Supplementary Materials for SparseNeuS: Fast Generalizable Neural Surface Reconstruction from Sparse Views

Xiaoxiao Long[1] Cheng Lin[2] Peng Wang[1]
Taku Komura[1] Wenping Wang[3]

[1] The University of Hong Kong
[2] Tencent Games
[3] Texas A&M University

1 Details of Patch-based Color Blending

Besides the pixel-based color blending, we introduce patch-based color blending to jointly evaluate local and contextual radiance consistency, thus yielding more reliable color predictions. To render the colors of a patch with size $k \times k$, we leverage local surface assumption and homography transformation for an efficient implementation.

The key idea is to estimate a local plane of a sampled point to efficiently derive the local patch. Given a sampled point q in the query ray, we leverage the property of the SDF network $s(q)$ to estimate the normal direction n_q by computing the spatial gradient, i.e., $n_q = \nabla s(q)$. Then, we select a set of points on the local plane (q, n_q), project the selected points to each view, and obtain the colors by interpolation on each input image. This projection operation is implemented by homography transformation. Let H be the homography between the view to be rendered I_r and the i_{th} input view I_i induced by the local plane (q, n_q):

$$H = K \left(R_i + \frac{t_i n_q^T R_r^T}{n_q^T \left(q + R_r^T t_r \right)} \right) K^{-1}, \tag{1}$$

where K is the intrinsic matrix, R_i is the 3×3 rotation matrix of I_i relative to I_r, t_i is the 3D translation vector of I_i relative to I_r, and (R_i, t_i) is the pose of the view I_r in the world coordinate system. Given the homography, we can obtain the projected pixel location Hq on the view I_i, that is, the matrix product of H and q, and then obtain q's corresponding color by interpolation.

This homography is also applied to the set of points selected on the local plane (q, n_q), so we can obtain their colors in the view I_i by interpolation. All the points on the local plane share the same blending weights with q, and thus only one query of the blending weights is needed. By blending the patch colors interpolated from each view $\{I_i\}_{i=0}^{N-1}$ with the blending weights, we obtain the final patch colors of point q. Same as pixel-based blending, we use SDF-based volume rendering [7] to aggregate the interpolated patch colors of all the points

sampled in the query ray r to generate the final predicted patch colors of the ray.

Using local plane assumption, we consider the neighboring geometric information of a query 3D position, which encodes contextual information of local patches and enforces better geometric consistency. By adopting patch-based volume rendering, synthesized regions contain more global information than single pixels, thus producing more informative and consistent shape context, especially in the regions with weak texture and changing intensity.

2 More Implementation Details

Network details. Feature Pyramid Network [3] is used as the image feature extraction network to extract multi-scale features from input images. We implement the sparse 3D CNN networks using a U-Net like architecture, and use torchsparse [6] as the implementation of 3D sparse convolution. The signed distance function (SDF) f_θ is modeled by an MLP consisting of 4 hidden layers with a hidden size of 256. The blending network f_c used in fine-tuning is modeled by an MLP consisting of 3 hidden layers with a hidden size of 256. Positional encoding [4] is applied to 3D locations with 6 frequencies and to view directions with 4 frequencies. Same as NeuS [7], we adopt a hierarchical sampling strategy to sample points in the query ray for volume rendering, where the numbers of the coarse and fine sampling are both 64.

Training parameters. The loss weights of total loss Eq.7 are set to $\alpha = 0.1, \beta = 0.02$. The sdf scaling parameter τ of sparseness loss term Eq.10 is set to 100. For the consistency-aware color loss term Eq.6 used in fine-tuning, by default, λ_0 is set to 0.01 and λ_1 is set to 0.015. The ratio λ_0/λ_1 sometimes needs to be tuned for better reconstruction results for each scene: decreasing the ratio λ_0/λ_1 will lead to more regions being kept; otherwise, more regions are excluded and the surfaces are cleaner.

Data preparation. We observe that the images of the DTU dataset contain large black backgrounds and the regions have considerable image noises. Hence, we utilize a simple threshold-based denoising strategy to clean the images of training scenes. We first detect the pixels where intensities are smaller than a threshold $\tau = 10$ as the invalid black regions, and thus yielding a mask for each image. The mask is then processed by image dilation and erosion operations to reduce isolated outliers. Finally, we evaluate the areas of the connected components in the masks, and only keep the connected components whose areas are larger than s, where s is set to the 10% of the whole image. Given the masks, the detected black invalid regions are set to 0. By the simple denoising operation, the noises of the black background regions in the DTU training images are mostly removed.

3 More experiments

Different number of views as input. Despite the good performance given the input of sparse images, our method can deal with an arbitrary number of input views. We investigate how the reconstruction quality is improved with more views as input. We conduct experiments on Scan105 of DTU dataset, with $2 \sim 8$ views as input, of which results are shown in Figure 1. Our method is still able to produce plausible geometries using only two views of an unseen object. With more views included, the reconstruction quality can also be progressively improved, and finally converges to a fairly low reconstruction error.

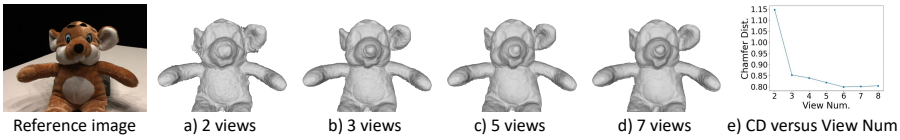

Fig. 1: The results with different number of views as input.

More qualitative results. We present more qualitative comparisons with MVSNerf [1], Colmap [5] and NeuS [7] on DTU [2] and BlendedMVS [8] datasets. As shown in Figure 2, the extracted meshes of MVSNerf always suffer from noisy surfaces, while our results via fast network inference are much smoother and less noisy. This is because MVSNerf adopts density representation which lacks local surface constraint.

After a short-time per-scene fine-tuning, our results are noticeably improved with fine-grained details and become more accurate and cleaner. Compared with the results of NeuS, our reconstructed surfaces are more complete and accurate. NeuS suffers from radiance ambiguity problem, and its geometries are incomplete and distorted.

More comparisons on BlendedMVS dataset are presented in Figure 3. Although our method is not trained on BlendedMVS, our generic model shows strong generalizability and produces cleaner and more complete results than those of MVSNerf. For example, for the Buddha head in Figure 3, Colmap fails to recover complete geometry and can only produce sparse points, while ours produces much more complete results.

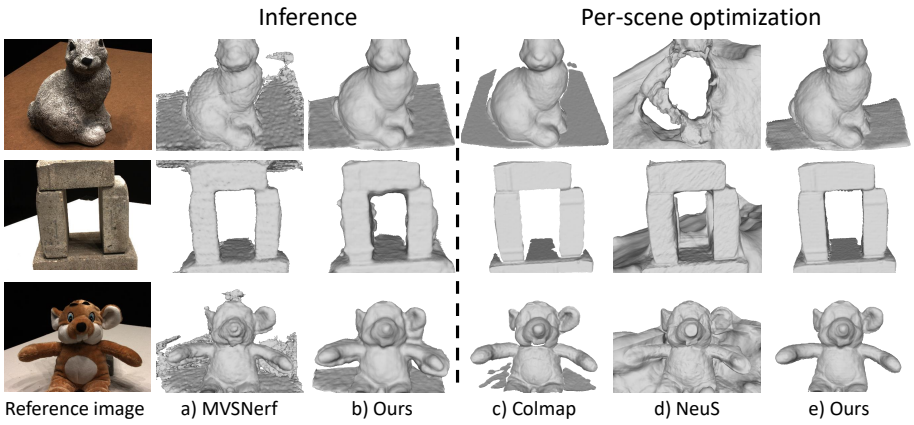

Fig. 2: Visual comparisons on DTU [2] dataset.

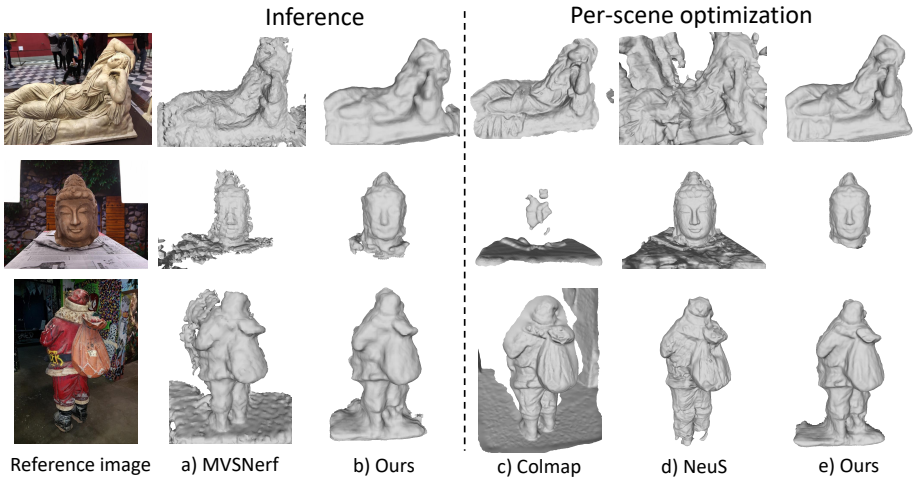

Fig. 3: Visual comparisons on BlendedMVS [8] dataset.

References

1. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14124–14133 (2021)
2. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 406–413 (2014)
3. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
4. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European conference on computer vision. pp. 405–421. Springer (2020)
5. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
6. Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., Han, S.: Searching efficient 3d architectures with sparse point-voxel convolution. In: European conference on computer vision. pp. 685–702. Springer (2020)
7. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. Advances in Neural Information Processing Systems **34** (2021)
8. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1790–1799 (2020)