Computational Visual Media
https://doi.org/10.1007/s41095-023-0337-5 Vol. 9, No. 4, December 2023, 687–697

Research Article

Neural 3D reconstruction from sparse views using geometric priors

Tai-Jiang Mu[1], Hao-Xiang Chen[1], Jun-Xiong Cai[1], and Ning Guo[2] (✉)

Abstract Sparse view 3D reconstruction has attracted increasing attention with the development of neural implicit 3D representation. Existing methods usually only make use of 2D views, requiring a dense set of input views for accurate 3D reconstruction. In this paper, we show that accurate 3D reconstruction can be achieved by incorporating geometric priors into neural implicit 3D reconstruction. Our method adopts the signed distance function as the 3D representation, and learns a generalizable 3D surface reconstruction model from sparse views. Specifically, we build a more effective and sparse feature volume from the input views by using corresponding depth maps, which can be provided by depth sensors or directly predicted from the input views. We recover better geometric details by imposing both depth and surface normal constraints in addition to the color loss when training the neural implicit 3D representation. Experiments demonstrate that our method both outperforms state-of-the-art approaches, and achieves good generalizability.

Keywords sparse views; 3D reconstruction; volume rendering; geometric priors; neural implicit 3D representation

1 Introduction

Reconstructing 3D surfaces from sparse 2D views is a classic computer vision problem with various applications in virtual reality, augmented reality, animation, etc. Recently, much progress has been made with the development of neural implicit 3D representation. Unlike traditional methods which directly triangulate explicit 3D surface points by feature matching, neural implicit methods use multi-layer perceptrons (MLPs) to parameterize the underlying scene to be reconstructed using occupancy or signed distance fields. Such an implicit representation is usually learned by imposing photometric consistency via RGB image reconstruction loss through a volume rendering technique.

Solely applying photometric consistency obviously leads to an underconstrained surface reconstruction problem, since many geometries may exist that can reproduce the same colors upon volume rendering. Furthermore, the photometric consistency constraint is less effective in noisy or weakly textured regions. Thus, geometry directly recovered from a photometric consistency based neural implicit representation, such as NeRF [1], usually suffers from over-smoothing and noise. Reconstructing the underlying geometry with better, finer detail requires a dense set of 2D views.

In this paper, we show that a more accurate and detailed geometry can be achieved by incorporating monocular geometric cues when training the neural 3D implicit reconstruction method. Depths and normals are the two most common kinds of monocular geometric cues provided by the local geometry of the underlying surface. These cues can be estimated from monocular images with reasonable quality using deep learning approaches, such as Omnidata [2] and MVSNet [3], or the data can be provided directly by depth sensors. Photometric cues and geometric cues are complementary: depths and normals can help to infer the geometry in textureless regions, while photometric cues can help to enrich details. Thus, in addition to RGB image reconstruction loss, we also

1 BNRist, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: T.-J. Mu, taijiang@tsinghua.edu.cn; H.-X. Chen, chx20@mails.tsinghua.edu.cn; J.-X. Cai, caijunxiong000@163.com.

2 Academy of Military Sciences, Beijing 100091, China. E-mail: guoning10@nudt.edu.cn (✉).

Manuscript received: 2022-11-25; accepted: 2023-01-31

impose depth and normal reconstruction loss terms to the volume rendering results.

The geometric cues can also serve as an initial guess for the underlying geometry, which helps to build a reliable and sparse feature volume for the neural implicit 3D representation, making it easier to train. To obtain accurate reconstruction, previous methods [4] usually adopted a hierarchical implicit representation, which first reconstructs coarse geometry and then recovers details with a finer resolution. Though such a hierarchical architecture can be efficient, the final results are still be affected by uncertainty and noise in the coarse geometry. If depths are available, we can avoid the use of a hierarchical architecture by directly determine an initial feature volume from the point cloud reprojected from the depth maps. This initial geometry is sparse and has comparable accuracy to the learned geometry [4], and can also be further refined. Including geometry priors during training of the neural implicit representation and determining the initial geometry directly from the geometric cues (i) improves the quality of the reconstructed geometry, (ii) improves the generalizability of the method, and (iii) leads to faster convergence, giving an overall better approach for sparse view 3D reconstruction.

In summary, our method makes the following contributions:

- a general approach exploiting geometric priors to improve 3D reconstruction quality and generalization for neural sparse view implicit 3D reconstruction,
- initial geometric reasoning for neural implicit surface models, which is more effective and easier to train, and
- extensive experiments which demonstrate that our method achieves state-of-the-art sparse view 3D reconstruction.

2 Related work

2.1 Multi-view 3D reconstruction using an explicit representation

Typical multi-view 3D reconstruction is conducted by first extracting features from 2D views, then reasoning about the underlying geometry producing each view, and finally fusing the view geometry to obtain the final 3D geometry for the whole object or scene. The fusion process differs depending on the 3D representation used, such as voxels [5–8], a point cloud [9, 10], or depths [3, 11–14]. Compared to traditional methods [15–17] based on hand-crafted geometric features, current CNN-based methods are more capable of extracting robust features and have produced promising results. In particular, using readily available depth estimation neural networks [2, 3, 11, 18], depth-based methods [19], together with dedicated fusion [20–22], can provide high-quality reconstruction from densely captured images. However, these methods have shortcomings when facing image noise or weak textures, and can fail to recovery a complete surface given insufficient images or sparse views.

2.2 Neural implicit 3D reconstruction

With the success of novel view synthesis using neural radiance fields (NeRF) [1, 23–27], neural 3D implicit representations were quickly applied to multi-view 3D reconstruction [4, 28–36]. Such methods usually extract the geometry from the predicted voxel density, occupancy field, or signed distance function (SDF). However, the geometry can suffer from images noise, and can be unreliable given weak textures or incomplete data, since they usually only impose photometric consistency when learning the implicit representation. To learn a better surface for incompletely visible objects, SNeS [37] explored the use of an object symmetry prior to help recover the invisible parts. Some recent reconstruction methods consider depths [38] or normals [39, 40] as geometric priors to help reconstruction; however, to obtain finely detailed geometry, these methods usually require a large number of views to be input to perform per-scene optimization, leading to difficulties to generalize to new scenes.

To increase the generalizability of the networks, efforts have been made to enable the network to memorize the input views or geometric cues. MVSNeRF [41] encodes the input views into a feature volume for better view synthesis. StereoNeRF [42] learns stereo correspondences from the input sparse views. PixelNeRF [43] and IBRNet [44] encode pixel colors into the network in addition to 3D coordinates and viewing directions. Depth [45, 46] and normal [47] priors have also been explored in neural rendering to better constrain the underlying geometry. Although these methods can synthesize plausible images, the extracted geometries can still

suffer from noise, since all 3D spaces are considered in their representations. Our method, in contract, can determine a more accurate, sparse initial geometry with the help of geometric cues, especially using depth estimation from the input views.

3 Neural sparse view 3D reconstruction with geometric priors

3.1 Overview

The pipeline of our method is illustrated in Fig. 1. It uses a volume rendering scheme for surface reconstruction [4]. Given sparse input views (typically 3–7 views), our method first obtains a depth map for each view. Then the *sparse geometry reasoning module* builds a sparse feature volume from these depth maps by projecting them back to the world space to assemble a sparse point cloud of the underlying geometry. The resulting sparse feature volume is then fed into the *geometry-guided surface reconstruction module* to reconstruct an accurate surface with fine details, doing so by imposing geometric constraints based on losses from rendered depths and normals, in addition the photometric loss.

3.2 Sparse geometry reasoning with depths

We parameterize the underlying geometry to be reconstructed using the signed distance function (SDF), following previous methods [1, 4, 47]. To efficiently reconstruct the underlying geometry, previous methods usually exploit a coarse-to-fine process, first estimating the coarse geometry for the whole scene, and then only refining details in occupied (non-empty) regions as determined by the coarse geometry. While this approach is efficient, it is not robust to noise and unreliable photometric consistency, since it usually only uses RGB images to build the underlying geometry.

Given that depth maps of monocular images can be reliably and efficiently estimated by deep learning techniques [2, 3], or easily acquired by depth sensors, we make use of these depth maps for monocular RGB images to build a sparse and reliable coarse geometry for our neural implicit model. Specifically, given N sparse images I_i, $i = 0, \ldots, N-1$, with poses (R_i, t_i), we first determine or obtain corresponding depth maps D_i. These depth maps are then reprojected to the world coordinate system to produce a composite point cloud $P = \bigcup_{i=0}^{N-1} \pi_i^{-1}(D_i)$ using the camera intrinsic parameters and poses of all images I_i, where $\pi_i^{-1}(D_i)$ denotes the reprojected 3D locations of pixels in depth map D_i.

To construct a feature volume G for geometry reasoning, we follow a prior method [3] for multiview depth estimation to build a cost volume C. We first extract a 2D feature map F_i from each input image using a 2D feature extraction network, and voxelize the fused point cloud P into regular voxels with voxel size d. We then project all points $P(v)$ contained in each voxel v to feature map F_i and determine the voxel's feature $F_i(v)$ as the average feature:

$$F_i(v) = \mathrm{Avg}_{p \in P(v)} F_i(p) \tag{1}$$

The cost feature volume of a voxel v is thus obtained by computing the variance of all the projected features of the voxel to all input views:

$$C(v) = \mathrm{Var}(\{F_i(v)\}_{i=0}^{N-1}) \tag{2}$$

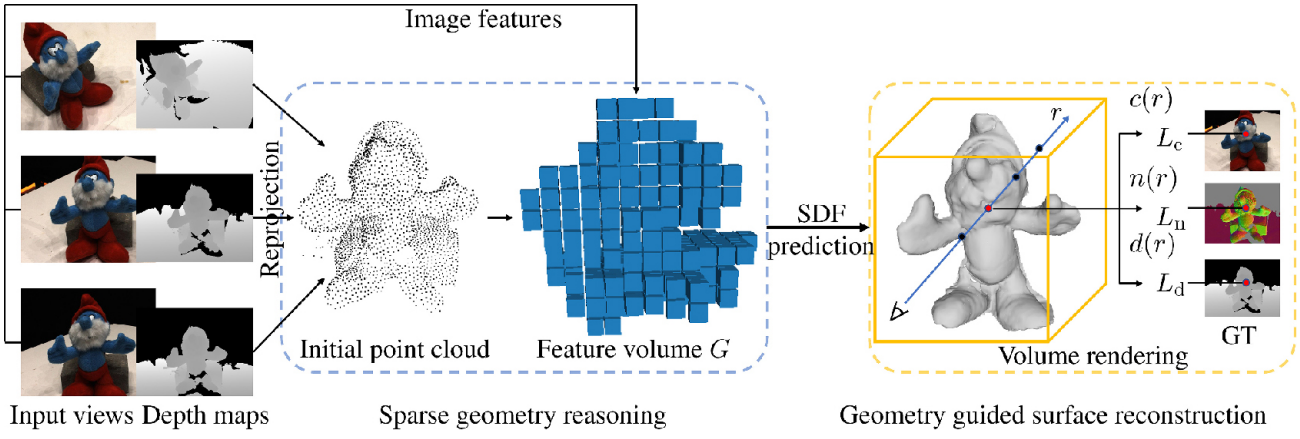

Fig. 1 Pipeline. Our method takes sparse views as input and reconstructs the underlying 3D surface by (i) reliably determining a coarse geometry from the depth maps, and (ii) in addition to photometric consistency L_c, imposing both depth consistency L_d and normal consistency L_n, leading to a more general and accurate framework for the neural implicit model.

Finally, the geometric feature volume G is obtained by applying a sparse 3D CNN to C:

$$G = \mathrm{CNN}_{\mathrm{sparse3d}}(C) \tag{3}$$

Note that this feature volume is inherently sparse because of the sparsity of the point cloud. To account for possible errors in depth maps, we also dilate each voxel by a distance δ_d.

In this way, our method directly reconstructs the underlying geometry using only one level of implicit field, i.e., the finest level of previous methods, while still being as efficient as possible, and achieving more accurate results.

3.3 Geometry guided surface reconstruction

Following Ref. [4], given a query 3D position q, our implicit 3D representation directly predicts its signed distance $\mathrm{SDF}(q)$. Specifically, an MLP network is applied to predict the surface from the interpolated geometric features of G, concatenated with q's positional encoding PE:

$$\mathrm{SDF}(q) = \mathrm{MLP}_{\mathrm{sdf}}(\mathrm{PE}(q), G(q)) \tag{4}$$

NeuS [28] used dedicated volume rendering for multiview 3D reconstruction using a neural implicit surface and later was extended to sparse view 3D reconstruction [4]. However, the neural implicit surface is optimized only using photometric supervision, which may suffer from noise and unreliable photometric consistency in textureless regions.

In addition to photometric consistency, we try to exploit complementary geometric priors to reconstruct more accurate and detailed geometries from the sparse input views. Specifically, to render the depth $d(r)$ and normal $n(r)$ as well as the color $c(r)$ of a ray r going through the underlying scene, we first query the depths d_i, normals n_i, and colors c_i, and SDF values s_i for all M sampled points $\{p(t_i)\}$ along the ray; then we convert s_i to densities σ_i using NeuS [28]:

$$\sigma_i = \max(-\Phi_s^{'}(s_i)/\Phi_s(s_i), 0) \tag{5}$$

where $\Phi_s(x) = (1 + \mathrm{e}^{-sx})^{-1}$ and s is a learnable parameter. Finally, we combine depths, normals, and colors with the densities to obtain the rendered values:

$$d(r) = \sum_{i=0}^{M-1} T_i(1 - \exp(-\delta_i\sigma_i))d_i \tag{6}$$

$$n(r) = \sum_{i=0}^{M-1} T_i(1 - \exp(-\delta_i\sigma_i))n_i \tag{7}$$

$$c(r) = \sum_{i=0}^{M-1} T_i(1 - \exp(-\delta_i\sigma_i))c_i \tag{8}$$

where $\delta_i = ||p(t_{i+1}) - p(t_i)||_2$ is the distance between two consecutive sample points and $T_i = \exp(-\Sigma_{j=0}^{i-1}\delta_i\sigma_j)$ is the accumulated transmittance. Note that we follow Ref. [4] to calculate the color at each sampled point which blends the colors of pixels or patches from the input views. d_i is the ray distance from the sampled point to the ray original and n_i is the spatial gradient of the sample point at the predicted SDF.

We now can train a more accurate neural implicit surface by imposing consistency on depth and normal as well as color with the total loss in Eq. (9):

$$\begin{cases} L_{\mathrm{total}} = L_{\mathrm{c}} + w_{\mathrm{n}}L_{\mathrm{n}} + w_{\mathrm{d}}L_{\mathrm{d}} \\ L_{\mathrm{c}} = \Sigma_{r\in\mathcal{R}}\|c(r) - \widetilde{c}(r)\|_2^2 \\ L_{\mathrm{n}} = \Sigma_{r\in\mathcal{R}}\|n(r) - \widetilde{n}(r)\|_2^2 \\ L_{\mathrm{d}} = \Sigma_{r\in\mathcal{R}}\|d(r) - \widetilde{d}(r)\|_2^2 \end{cases} \tag{9}$$

where L_{c}, L_{n}, and L_{d} are the photometric loss, normal loss, and depth loss, respectively, $\mathcal{R}$ is the set of all sample rays, $\widetilde{c}(r)$, $\widetilde{n}(r)$, and $\widetilde{d}(r)$ are the ground-truth color, normal, and depth of the sample ray r, respectively, and w_{n} and w_{d} weight these losses.

4 Experiments and results

4.1 Dataset

We trained our generalizable sparse view 3D reconstruction model on the DTU multiview stereo dataset [48], which contains 75 scenes for training and a further 15 non-overlapping scenes for testing. We centrally cropped the images to a resolution of 512×640 for both training and testing. To train our network, ground-truth normals were estimated from the ground-truth point cloud for the underlying scene provided with the dataset. While depth maps could be estimated using current learning-based methods [2, 3] for each view, to ensure depth consistency between views, we used the ground-truth depth to determine the initial geometry required by our network for both training and testing. To account for sparsity and errors in the depth map, we set the dilation range δ_d to 7 voxels. Our network was trained with 6 views, including 1 reference view and 5 source views.

4.2 Implementation details

We adopted a feature pyramid network [49] as the multi-scale 2D image feature extraction network and used a U-net like sparse 3D convolution network [50]. The 3D voxel resolution was set to $192 \times 192 \times 192$ and the weights in the loss function were set as $w_{\mathrm{d}} = 0.1$

and $w_d = 0.9$. We trained our model for 20k iterations with an initial learning rate of 2×10^{-4}, adjusted using a cosine decay schedule, with a factor of 0.5 at 10k and 15k steps. Our model was trained using the Jittor deep learning framework [51] on a Titan RTX GPU using a batch size of 512 rays.

4.3 Metrics

To evaluate the accuracy of 3D reconstruction, we adopt the commonly used chamfer distance, which measures point distances between the predicted and ground-truth geometries. Following Ref. [4], we make use of the foreground object masks provided in IDR [34] to remove the background from the reconstructed results when computing metrics.

4.4 Quantitative and qualitative results

We first compared our method to a baseline neural surface reconstruction method SparseNeuS [4], which ignores both depth and normal cues when learning the 3D implicit field. For a fair comparison, we followed the setting of SparseNeuS by performing evaluation on two sets of three views for each test scene. The final metrics average these pairs of results. We also compared our method to other leading general sparse view 3D reconstruction methods, including (i) generic neural rendering methods such as PixelNeRF [43], IBRNet [44], and MVSNeRF [41], where the reconstructed mesh is extracted from the learned implicit field, and (ii) the widely used classic MVS method COLMAP [19], where the reconstructed mesh is extracted from the reconstructed point cloud. Note that, to test the generalizability of all methods, they were not fine-tuned to suit each scene. Per-scene chamfer distances and mean chamfer distance on the DTU test set are reported in Table 1. All values except those for our method are directly drawn from Ref. [4]. We also present a visual comparison of sample output from SparseNeuS and our method in Fig. 2; we also show cosine similarity of the predicted geometry's normals to the ground-truth values.

Table 1 Chamfer distance assessment of reconstruction errors using the DTU test dataset, for various methods

Method \ Scan	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
PixelNeRF	5.13	8.07	5.85	4.40	7.11	4.64	5.68	6.76	9.05	6.11	3.95	5.92	6.26	6.89	6.93	6.28
IBRNet	2.29	3.70	2.66	1.83	3.02	2.83	1.77	2.28	2.73	1.96	1.87	2.13	1.58	2.05	2.09	2.32
MVSNeRF	1.96	3.27	2.54	1.93	2.57	2.71	1.82	1.72	2.29	1.75	1.72	1.47	1.29	2.09	2.26	2.09
SparseNeuS	1.68	3.06	2.25	1.10	2.37	2.18	1.28	1.47	1.80	1.23	1.19	1.17	0.75	1.56	1.55	1.64
Colmap	**0.90**	2.89	**1.63**	1.08	2.18	1.94	1.61	**1.30**	2.34	1.28	1.10	1.42	0.76	1.17	**1.14**	1.52
Ours	1.17	**2.35**	1.94	**1.06**	**1.11**	**1.42**	**0.95**	1.64	**1.18**	**1.02**	**0.96**	**0.81**	**0.67**	**1.15**	1.31	**1.25**

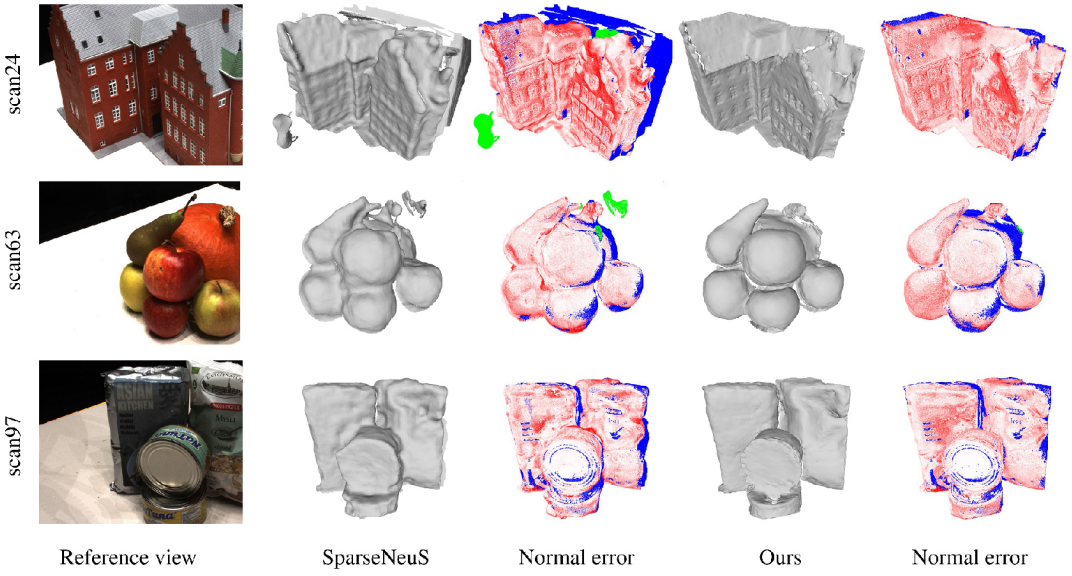

Fig. 2 Comparison of results from our method and SparseNeuS [4] for scans 24, 63, and 97 of the DTU [48] test set. Left to right: input reference view, reconstructed SparseNeuS mesh, normal error map for SparseNeuS, reconstructed mesh from our method, and normal error map for our method. Brighter red indicates larger normal error. Green and blue indicate regions omitted when calculating the normal error.

Our method achieves the most accurate reconstruction as assessed in terms of chamfer distance. Generic neural rendering methods, such as PixelNeRF, IBRNet, and MVSNeRF struggle to recover fine geometric details using only the input images. Compared to SparseNeuS, with the help of depth guided sparse geometry reasoning and the constraints from both depth and normal priors, our method can reconstruct more accurate and detailed geometry. Furthermore, unlike SparseNeuS, our method does not need to train two (coarse and fine) networks, making training easier. Note that our method can outperform the COLMAP classical MVS method without the need of per-scene fine-tuning, which is required by SparseNeuS. Indeed, our results are even a little better than the fine-tuned results of SparseNeuS, having a mean chamfer distance of 1.27 on the DTU test dataset.

The reconstructed results for other scenes from the DTU test set are shown in Fig. 3.

4.5 Ablation study

To demonstrate the effectiveness of our approach, we conducted experiments by ablating the core modules of our method providing sparse geometry reasoning, depth constraints, and normal constraints. In the following evaluation, results were reconstructed using 6 views for consistency with the training process. These views were selected according to view pairs provided by the DTU dataset [48], where the first three views are one set of three views used in SparseNeuS.

The study was performed as follows:

- Without sparse geometry reasoning (SGR). We replaced our sparse geometry reasoning module with the two-stage geometry reasoning module from SparseNeuS [4] and used the same settings of resolutions for coarse and fine voxels. Depth and normal constraints were imposed using the same weights as in our unaltered method.
- Without normal prior. We set the weight for the normal loss to zero in our full method.
- Without depth prior. We set the weight for the depth loss to zero in our full method.

A quantitative analysis of 3D reconstruction results is given in Table 2 and a qualitative assessment is presented in Fig. 4. As we can see, eliminating the SGR module leads to over smoothed surfaces and less accurate geometry; both depth and normal priors

Table 2 Ablation study. Mean chamfer distance with and without the sparse geometry reasoning (SGR) module, the normal constraint and the depth constraint

SGR	Depth	Normal	CD
×	✓	✓	1.737
✓	×	✓	1.304
✓	✓	×	1.834
✓	✓	✓	**1.281**

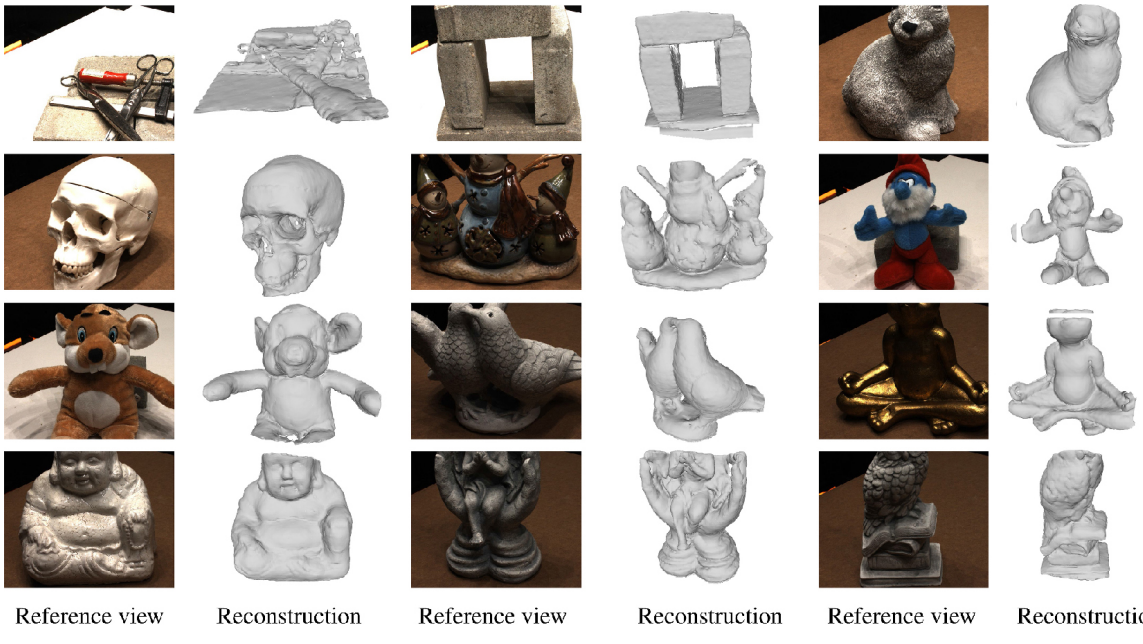

Fig. 3 Further reconstructed results for DTU test scenes using 6 views. Left: reference view. Right: mesh reconstructed by our method.

Fig. 4 Ablation study. Left to right: (a) reference image, (b) our method without sparse geometry reasoning (SGR), (c) our method without depth priors, (d) our method without normal priors, and (e) our full model. The SGR module helps to recover more accurate and detailed geometry; both depth and normal cues improve local details of the underlying geometry.

contribute to the geometric details. We also present novel view synthesis results in Fig. 5 by rendering an unseen view for several DTU test scenes. The results show that, though originally designed for 3D reconstruction, our geometric priors, especially the sparse geometry reasoning, are also beneficial when synthesizing novel views.

4.6 Parameter study

Our method is affected by four main parameters, including the weight for depth loss w_{d}, the weight for normal loss w_{n}, the voxel size d (inversely proportional to the number of voxels in each direction), and the depth dilation range δ_d. We varied both w_{d} and w_{n} from 0.0 to 0.9 with a step size of 0.1 and tested two settings of voxel resolution, i.e., 96 or 192 voxels in each dimension. The mean chamfer distances for different parameter configurations, using the DTU dataset test scenes are listed in Table 3. We can observe that

(1) d is responsible for geometric reconstruction accuracy: the smaller d is, the more accurate the geometry, but at a cost of increased computation.
(2) As the depth weight starts to increase (from 0.01 to 0.1), the geometry becomes more and more accurate; however, when it continues to increase, the accuracy drops. This may be somewhat affected by the sparse geometry reasoning module, which uses the depth information to construct an initial geometry for the underlying scene.

Table 3 Parameter study, considering mean chamfer distance w.r.t. normal loss weight, depth loss weight, number of voxels in each direction, and depth dilation range

w_{d}	w_{n}	Voxels	δ_d	CD
0.01	0.2	192	7	1.639
0.01	0.2	192	10	1.535
0.1	0.2	96	7	1.993
0.1	0.2	192	7	1.473
0.1	0.3	192	7	1.381
0.1	0.4	192	7	1.372
0.1	0.5	192	7	1.369
0.1	0.6	192	7	1.293
0.1	0.7	192	7	1.357
0.1	0.8	192	7	1.359
0.1	0.9	192	7	**1.281**
0.2	0.2	192	7	1.510
0.3	0.2	192	7	1.707
0.4	0.2	192	7	1.675
0.5	0.2	192	7	1.882
0.6	0.2	192	7	1.927
0.7	0.2	192	7	2.853
0.8	0.2	192	7	3.028
0.9	0.2	192	7	3.230

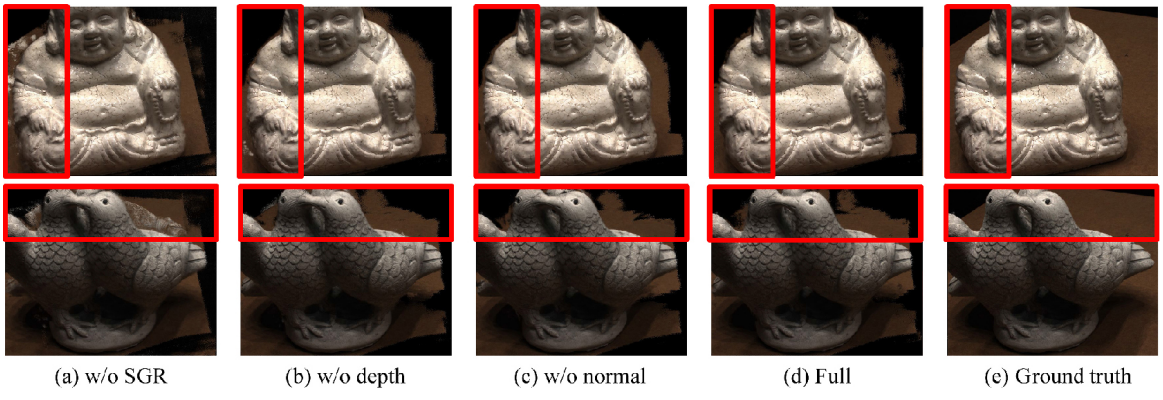

Fig. 5 Novel view synthesis. Left to right: (a) our method without sparse geometry reasoning (SGR), (b) our method without depth prior, (c) our method without normal prior, (d) our full model, and (e) the ground truth. The differences are highlighted in red box.

(3) A larger normal loss weight reconstructs more geometric details.

(4) A larger depth dilation range can cover more true surface regions and thus be more robust to noise in the depth maps, achieving more accurate reconstruction results, but at a cost of a greater computational burden.

To balance the accuracy of the recovered geometry and computational effort, we suggest setting $w_{\mathrm{d}} = 0.1$, $w_{\mathrm{n}} = 0.9$, and using higher voxel resolution and a smaller depth dilation range.

5 Conclusions and future work

This paper has presented a general framework for neural implicit 3D reconstruction from sparse views. By leveraging geometric priors, our method can determine a sparse and reliable coarse implicit geometry for optimization. This is done by imposing both depth consistency and normal consistency, as well as photometric consistency, on the training loss function. This makes the framework more general and accurate.

Currently, we set a fixed dilation range for the depth when constructing the initial geometry. This could be further improved in practice if the uncertainty of the depth map is known. Our model can also be per-scene fine-tuned given more views of a specific scene, using only the color loss.

In future, we would also like to apply our method to outdoor large scene 3D reconstruction from remote sensing images or aerial images, for which accurate depths and normals are even hard to obtain, by leveraging accurate mapping data.

Acknowledgements

We thank the anonymous reviewers for their valuable comments on this paper. This work was supported by the National Natural Science Foundation of China (Grant No. 61902210).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; Ng, R. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* Vol. 65, No. 1, 99–106, 2022.

[2] Eftekhar, A.; Sax, A.; Malik, J.; Zamir, A. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3D scans. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 10766–10776, 2021.

[3] Yao, Y.; Luo, Z.; Li, S.; Fang, T.; Quan, L. MVSNet: Depth inference for unstructured multi-view stereo. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11212*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 785–801, 2018.

[4] Long, X.; Lin, C.; Wang, P.; Komura, T.; Wang, W. SparseNeuS: Fast generalizable neural surface reconstruction from sparse views. In: *Computer Vision – ECCV 2022. Lecture Notes in Computer Science, Vol. 13692*. Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; Hassner, T. Eds. Springer Cham, 210–227, 2022.

[5] Seitz, S. M.; Dyer, C. R. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision* Vol. 35, No. 2, 151–173, 1999.

[6] Kar, A.; Häne, C.; Malik, J. Learning a multi-view stereo machine. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 364–375, 2017.

[7] Sun, J. M.; Xie, Y. M.; Chen, L. H.; Zhou, X. W.; Bao, H. J. NeuralRecon: Real-time coherent 3D reconstruction from monocular video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 15593–15602, 2021.

[8] Ji, M. Q.; Zhang, J. Z.; Dai, Q. H.; Fang, L. SurfaceNet: An end-to-end 3D neural network for very sparse multi-view stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 43, No. 11, 4078–4093, 2021.

[9] Lhuillier, M.; Quan, L. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 27, No. 3, 418–433, 2005.

[10] Furukawa, Y.; Ponce, J. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 32, No. 8, 1362–1376, 2010.

[11] Gu, X. D.; Fan, Z. W.; Zhu, S. Y.; Dai, Z. Z.; Tan, F. T.; Tan, P. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2492–2501, 2020.

[12] Long, X.; Liu, L.; Theobalt, C.; Wang, W. Occlusion-aware depth estimation with adaptive normal constraints. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12354*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 640–657, 2020.

[13] Long, X. X.; Lin, C.; Liu, L. J.; Li, W.; Theobalt, C.; Yang, R. G.; Wang, W. Adaptive surface normal constraint for depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 12829–12838, 2021.

[14] Long, X. X.; Liu, L. J.; Li, W.; Theobalt, C.; Wang, W. P. Multi-view depth estimation using epipolar spatio-temporal networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8254–8263, 2021.

[15] Fuentes-Pacheco, J.; Ruiz-Ascencio, J.; Rendón-Mancha, J. M. Visual simultaneous localization and mapping: A survey. *Artificial Intelligence Review* Vol. 43, No. 1, 55–81, 2015.

[16] Xu, Z. W.; Rong, Z.; Wu, Y. H. A survey: Which features are required for dynamic visual simultaneous localization and mapping? *Visual Computing for Industry, Biomedicine, and Art* Vol. 4, No. 1, Article No. 20, 2021.

[17] Özyeşil, O.; Voroninski, V.; Basri, R.; Singer, A. A survey of structure from motion. *Acta Numerica* Vol. 26, 305–364, 2017.

[18] Li, Y. Z.; Luo, F.; Xiao, C. X. Self-supervised coarse-to-fine monocular depth estimation using a lightweight attention module. *Computational Visual Media* Vol. 8, No. 4, 631–647, 2022.

[19] Schönberger, J. L.; Frahm, J. M. Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4104–4113, 2016.

[20] Choy, C. B.; Xu, D.; Gwak, J.; Chen, K.; Savarese, S. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9912*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 628–644, 2016.

[21] Huang, P. H.; Matzen, K.; Kopf, J.; Ahuja, N.; Huang, J. B. DeepMVS: Learning multi-view stereopsis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2821–2830, 2018.

[22] Wang, D.; Cui, X. R.; Chen, X.; Zou, Z. X.; Shi, T. Y.; Salcudean, S.; Wang, Z. J.; Ward, R. Multi-view 3D reconstruction with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 5702–5711, 2021.

[23] Liu, L.; Gu, J.; Lin, K. Z.; Chua, T.; Theobalt, C. Neural sparse voxel fields. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, Article No. 1313, 15651–15663, 2020.

[24] Trevithick, A.; Yang, B. GRF: Learning a general radiance field for 3D representation and rendering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 15162–15172, 2021.

[25] Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; Srinivasan, P. P. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 5835–5844, 2021.

[26] Verbin, D.; Hedman, P.; Mildenhall, B.; Zickler, T.; Barron, J. T.; Srinivasan, P. P. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5481–5490, 2022.

[27] Guo, Y. C.; Kang, D.; Bao, L. C.; He, Y.; Zhang, S. H. NeRFReN: Neural radiance fields with reflections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 18388–18397, 2022.

[28] Wang, P.; Liu, L.; Liu, Y.; Theobalt, C.; Komura, T.; Wang, W. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In: Proceedings of the 35th Conference on Neural Information Processing Systems, 27171–27183, 2021.

[29] Zhang, J. Y.; Yao, Y.; Quan, L. Learning signed distance field for multi-view surface reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 6505–6514, 2021.

[30] Niemeyer, M.; Mescheder, L.; Oechsle, M.; Geiger, A. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3501–3512, 2020.

[31] Oechsle, M.; Peng, S. Y.; Geiger, A. UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 5569–5579, 2021.

[32] Darmon, F.; Bascle, B.; Devaux, J. C.; Monasse, P.; Aubry, M. Improving neural implicit surfaces geometry with patch warping. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6250–6259, 2022.

[33] Yariv, L.; Gu, J.; Kasten, Y.; Lipman, Y. Volume rendering of neural implicit surfaces. In: Proceedings of the 35th Conference on Neural Information Processing Systems, 4805–4815, 2021.

[34] Yariv, L.; Kasten, Y.; Moran, D.; Galun, M.; Atzmon, M.; Basri, R.; Lipman, Y. Multiview neural surface reconstruction by disentangling geometry and appearance. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, Article No. 210, 2492–2502, 2020.

[35] Liu, S. H.; Zhang, Y. D.; Peng, S. Y.; Shi, B. X.; Pollefeys, M.; Cui, Z. P. DIST: Rendering deep implicit signed distance function with differentiable sphere tracing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016–2025, 2020.

[36] Kellnhofer, P.; Jebe, L. C.; Jones, A.; Spicer, R.; Pulli, K.; Wetzstein, G. Neural lumigraph rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4285–4295, 2021.

[37] Insafutdinov, E.; Campbell, D.; Henriques, J. F.; Vedaldi, A. SNeS: Learning probably symmetric neural surfaces from incomplete data. In: *Computer Vision – ECCV 2022. Lecture Notes in Computer Science, Vol. 13692*. Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; Hassner, T. Eds. Springer Cham, 367–383, 2022.

[38] Azinović, D.; Martin-Brualla, R.; Goldman, D. B.; Nießner, M.; Thies, J. Neural RGB-D surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6280–6291, 2022.

[39] Guo, H. Y.; Peng, S. D.; Lin, H. T.; Wang, Q. Q.; Zhang, G. F.; Bao, H. J.; Zhou, X. Neural 3D scene reconstruction with the Manhattan-world assumption. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5501–5510, 2022.

[40] Wang, J. P.; Wang, P.; Long, X. X.; Theobalt, C.; Komura, T.; Liu, L. J.; Wang, W. NeuRIS: Neural reconstruction of indoor scenes using normal priors. In: *Computer Vision – ECCV 2022. Lecture Notes in Computer Science, Vol. 13692*. Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; Hassner, T. Eds. Springer Cham, 139–155, 2022.

[41] Chen, A. P.; Xu, Z. X.; Zhao, F. Q.; Zhang, X. S.; Xiang, F. B.; Yu, J. Y.; Su, H. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 14104–14113, 2021.

[42] Chibane, J.; Bansal, A.; Lazova, V.; Pons-Moll, G. Stereo radiance fields (SRF): Learning view synthesis for sparse views of novel scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7907–7916, 2021.

[43] Yu, A.; Ye, V.; Tancik, M.; Kanazawa, A. pixelNeRF: Neural radiance fields from one or few images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4576–4585, 2021.

[44] Wang, Q. Q.; Wang, Z. C.; Genova, K.; Srinivasan, P.; Zhou, H.; Barron, J. T.; Martin-Brualla, R.; Snavely, N.; Funkhouser T. A. IBRNet: Learning multi-view image-based rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4688–4697, 2021.

[45] Roessle, B.; Barron, J. T.; Mildenhall, B.; Srinivasan, P. P.; Nießner, M. Dense depth priors for neural radiance fields from sparse input views. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12882–12891, 2022.

[46] Johari, M. M.; Lepoittevin, Y.; Fleuret, F. GeoNeRF: Generalizing NeRF with geometry priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 18344–18347, 2022.

[47] Yu, Z.; Peng, S.; Niemeyer, M.; Sattler, T.; Geiger, A. MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction. In: Proceedings of the 36th Conference on Neural Information Processing Systems, 2022.

[48] 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, 406–413, 2014.

[49] Lin, T. Y.; Dollár, P.; Girshick, R.; He, K. M.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 936–944, 2017.

[50] Tang, H. T.; Liu, Z. J.; Zhao, S. Y.; Lin, Y. J.; Lin, J.; Wang, H. R.; Han, S. Searching efficient 3D architectures with sparse point-voxel convolution. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12373*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 685–702, 2020.

[51] Hu, S. M.; Liang, D.; Yang, G. Y.; Yang, G. W.; Zhou, W. Y. Jittor: A novel deep learning framework with meta-operators and unified graph execution. *Science China Information Sciences* Vol. 63, No. 12, Article No. 222103, 2020.

Tai-Jiang Mu is an assistant researcher in the Department of Computer Science and Technology at Tsinghua University. He received his bachelor degree and Ph.D. degree in computer science and technology from Tsinghua University in 2011 and 2016, respectively. His research interests include computer graphics, visual media learning, 3D reconstruction, and 3D understanding.

Hao-Xiang Chen received his bachelor degree in computer science from Jilin University in 2020. He is currently a Ph.D. candidate in the Department of Computer Science and Technology, Tsinghua University. His research interests include 3D reconstruction and 3D computer vision.

Jun-Xiong Cai is currently a postdoctoral researcher at Tsinghua University, where he received his Ph.D. degree in computer science and technology in 2020. His research interests include computer graphics, computer vision, and 3D geometry processing.

Ning Guo is an assistant researcher at the Academy of Military Sciences. He received his bachelor degree, master degree, and Ph.D. degree in information and communication engineering from the National University of Defense Technology in 2014, 2016, and 2020, respectively. His research interests include digital earth, 3D GIS, 3D reconstruction, and spatial databases.