Neural RGB-D Surface Reconstruction

Dejan Azinović[1] Ricardo Martin-Brualla[2] Dan B Goldman[2] Matthias Nießner[1] Justus Thies[1,3]

[1]Technical University of Munich [2]Google Research [3]Max Planck Institute for Intelligent Systems

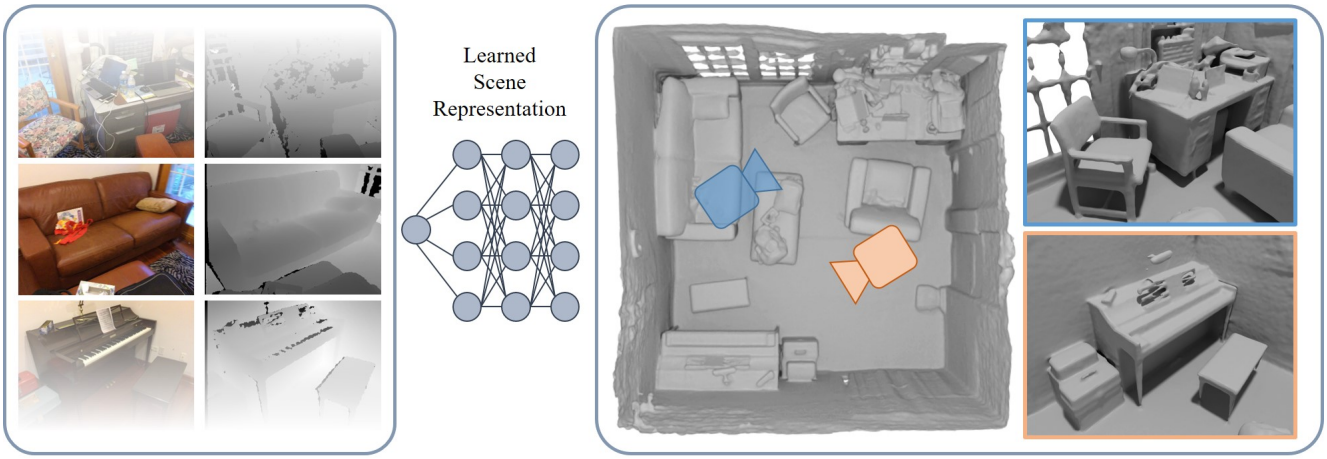


Figure 1: Our method obtains a high-quality 3D reconstruction from an RGB-D input sequence by training a multi-layer perceptron. The core idea is to reformulate the neural radiance field definition in NeRF [37], and replace it with a differentiable rendering formulation based on signed distance fields which is specifically tailored to geometry reconstruction.

Abstract

In this work, we explore how to leverage the success of implicit novel view synthesis methods for surface reconstruction. Methods which learn a neural radiance field have shown amazing image synthesis results, but the underlying geometry representation is only a coarse approximation of the real geometry. We demonstrate how depth measurements can be incorporated into the radiance field formulation to produce more detailed and complete reconstruction results than using methods based on either color or depth data alone. In contrast to a density field as the underlying geometry representation, we propose to learn a deep neural network which stores a truncated signed distance field. Using this representation, we show that one can still leverage differentiable volume rendering to estimate color values of the observed images during training to compute a reconstruction loss. This is beneficial for learning the signed distance field in regions with missing depth measurements. Furthermore, we correct misalignment errors of the camera, improving the overall reconstruction quality. In several experiments, we showcase our method and compare to existing works on classical RGB-D fusion and learned representations.

1. Introduction

Research on neural networks for scene representations and image synthesis has made impressive progress in recent years [54]. Methods that learn volumetric representations [37, 31] from color images captured by a smartphone camera can be employed to synthesize near photo-realistic images from novel viewpoints. While the focus of these methods lies on the reproduction of color images, they are not able to reconstruct metric and clean (noise-free) meshes. To overcome these limitations, we show that there is a significant advantage in taking additional range measurements from consumer-level depth cameras into account. Inexpen-

sive depth cameras are broadly accessible and are also built into modern smartphones. While classical reconstruction methods [6, 22, 39] that purely rely on depth measurements struggle with the limitations of physical sensors (noise, limited range, transparent objects, etc.), a NeRF-based reconstruction formulation allows to also leverage the dense color information. Methods like BundleFusion [8] take advantage of color observations to compute sparse SIFT [33] features for re-localization and refinement of camera poses (loop closure). For the actual geometry reconstruction (volumetric fusion), only the depth maps are taken into account. Missing depth measurements in these maps, lead to holes and incomplete geometry in the reconstruction. This limitation is also shared by learned surface reconstruction methods that only rely on the range data [36, 49]. In contrast, our method is able to reconstruct geometry in regions where only color information is available. Specifically, we adapt the neural radiance field formulation of Mildenhall et al. [37] to learn a truncated signed distance field (TSDF), while still being able to leverage differentiable volumetric integration for color reproduction. To compensate wrong initial camera alignments which we compute based on the depth measurements, we jointly optimize our scene representation network with the camera poses. The implicit function represented by the scene representation network allows us to predict signed distance values at arbitrary points in space which is used to extract a mesh using marching cubes.

In summary, our work proposes an RGB-D based scene reconstruction method that leverages both dense color and depth observations. It is based on an effective incorporation of depth measurements into the training of a neural radiance field using a signed distance representation to store the scene geometry. Comparisons to state-of-the-art scene reconstruction methods show that our approach improves the quality of geometry reconstructions both qualitatively and quantitatively.

2. Related Work

Our approach reconstructs geometry from a sequence of RGB-D frames, leveraging both dense color and depth information. In the following, we will outline the literature that is related to our approach.

3D reconstruction methods. A common strategy for 3D reconstruction is stereo matching from two or multiple color views [44, 18]. Such techniques assume knowledge of the camera alignment which can be estimated using Structure-from-Motion [45] or SLAM [11, 10, 13] methods. Multi-view stereo techniques may use disjoint representations, like oriented patches [15], volumes [28], or meshes [20] to reconstruct the scene or object. Other approaches rely on fusing multiple depth measurements from range scanners [6] using signed distance functions. KinectFusion [38] combines such representation with real-time tracking to reconstruct scenes in real-time. It has been extended by VoxelHashing [39] to handle large scenes. BundleFusion [8] builds on top of VoxelHashing and leverages offline computation to further refine the camera poses and the recovered model to solve the loop closure problem.

Deep learning for 3D reconstruction. The advent of convolutional neural networks has enabled incorporating higher-level priors and reasoning to 3D reconstruction pipelines. Monocular depth estimation [29, 17, 14] learns how to estimate the depth of a single view by learning from a large dataset of RGB-D images or stereo pairs. Multi-view stereo can be learned using 3D CNNs on voxel grids [23, 59], or multi-plane images [12]. Large synthetic shape datasets [4] enable modeling object categories, where generative models can be used for single view reconstruction [53, 58]. Neural Volumes [31] uses a Variational Autoencoder [26] to decompose a dynamic scene into a template represented in a voxel grid, and a deformation field. Other methods use 2D CNNs to extract useful features for 3D reconstruction, like normals, or shading [62]. 3D reconstruction models can also be combined with conditional GANs [21], to perform deferred rendering on an estimated mesh [55] or reconstruction using a voxel grid [50].

Coordinate-based models. Most recently a new family of models has become popular. These models represent scenes in the weights of a network, containing only fully connected layers, i.e. a multi-layer perceptron (MLP) [54]. Such models take as input the coordinates of a 3D location in the world, and output various fields of scalar and vector values. Coordinate-based models can represent shapes by modeling signed distance fields [40] and occupancy values [36]. Scene Representation Networks [51] combine such representations with a learned renderer implemented as a recurrent neural network, that marches rays through the scene. PIFu [42] and PIFuHD [43] propose to estimate a pixel-aligned implicit function to reconstruct human bodies from single images.

Most recently, Mildenhall et al. [37] proposed representing a scene as a neural radiance field (NeRF) using a coordinate-based model, that outputs a density and color for each 3D point. These values can then be rendered using traditional volume rendering integrals [35]. A key component of their technique is a positional encoding layer, that uses sinusoidal functions to improve the learning properties of the MLP. Alternatives to the positional encodings such as Fourier features [52] or sinusoidal activation layers [49] have been proposed. NeRFs have been extended to handle in-the-wild data with different lighting and occluders [34], dynamic scenes [30, 41], avatars [16], and adapted

for generative modeling [3, 48] and image-based rendering [61, 56]. Others have focused on resectioning a camera given a learned NeRF [60], and optimizing for the camera poses while learning a NeRF [57]. Our work is the first one to incorporate depth measurements in NeRF, and also proposes a camera refinement scheme to further improve the quality of alignment.

3. Method

We propose an optimization-based approach for geometry reconstruction that uses a deep neural network as the scene representation. Specifically, we assume an RGB-D sequence of a consumer-level camera as input (e.g., from a Microsoft Kinect). We leverage both the N color frames $\mathcal{I}_i$ as well as the corresponding depth frames $\mathcal{D}_i$. As an initialization, we obtain camera poses $\mathcal{T}_i$ using BundleFusion [8] which uses SIFT features detected in the color images to refine the sequential depth-based frame-by-frame tracking of VoxelHashing [39]. Using this input data, we optimize a continuous volumetric representation of the scene that stores the radiance as well as a truncated signed distance per point. Differentiable volumetric integration of the radiance values [35] is applied to compute color images from this representation, as illustrated in Fig. 2. We globally optimize the scene representation across all frames, based on color and depth reconstruction energies. We show that the combination of a TSDF-based geometry representation and volumetric rendering is a key aspect for reconstructing parts of the geometry for which no depth measurements were captured by the sensor. Since camera calibration and pose estimation are prone to errors, we allow for corrections of both intrinsic and extrinsic camera parameters. At evaluation time, we use Marching Cubes [32] to extract a triangle mesh from the optimized implicit scene representation.

3.1. Scene Representation

We build upon the work of Mildenhall et al. [37] and represent scenes with a multi-layer perceptron (MLP). The MLP can be queried at arbitrary positions $\vec{p}_i$ in space to compute a truncated signed distance value D_i and a radiance value c_i. Similar to [37], we observe an improvement in the quality of results after applying sinusoidal positional encodings $\gamma(\vec{p}_i)$ and $\gamma(\vec{d})$ to the queried 3D points and view directions.

Inspired by the recent success of volumetric integration in neural rendering, we render color as a weighted sum of radiance values sampled near the object surface. Instead of computing the weights as probabilities of light reflecting at a given sample point based on the density of the medium, we compute weights directly from signed distance values as the product of two sigmoid functions:

$$w_i = \sigma(s \cdot D_i) \cdot \sigma(-s \cdot D_i). \tag{1}$$

This bell-shaped function has its peak at the surface, i.e. at the zero-crossing of the signed distance values. The factor s controls how quickly the values fall to zero as the distance from the surface increases. We set $s = 20$ in our experiments. The color along a specific ray is approximated as a weighted sum of the K sampled colors:

$$C = \frac{1}{K} \sum_{i=0}^{K-1} w_i \cdot c_i. \tag{2}$$

This scheme gives the highest integration weight to the point on the surface, while points farther away from the surface have lower weights. Although such an approach is not derived from a physically-based rendering model, as is the case with volumetric integration over density values, it represents an elegant way to render color in a signed distance field in a differentiable manner, and we show that it helps deduce depth values through a photometric loss (see Sec. 4). We observe better reconstruction results than with a density-based volumetric approach such as [37]. In particular, this approach allows to always predict hard boundaries between occupied and free space, instead of using the density to model partially transparent blobs that explain view-dependent effects, which lead to noisy reconstructions.

Network Architecture Our network is composed of two MLPs as depicted in Fig. 2. The first MLP takes only the encoding of a queried 3D point $\vec{p}$ as input and outputs the truncated signed distance D_i to the nearest surface. The task of the second MLP is to produce surface color values for a given view direction $\vec{d}$. To this end, we concatenate the positional encoding of the view direction $\gamma(\vec{d})$ for which to produce color, and a 2-dimensional appearance latent code to the output of the final layer of the first MLP and pass the concatenated data to the second MLP. The view vector allows our method to deal with view-dependent effects like specular highlights, which would otherwise have to be modeled by deforming the geometry. Since color data is often subject to varying exposure or white-balance, we learn the latent appearance code vector for each frame in the input data to mitigate this issue [34].

Camera Correction The camera poses $\mathcal{T}_i$ consist of a rotation in Euler angle representation and a translation vector, resulting in 6 parameters per frame which are initialized with BundleFusion poses and refined during the joint optimization. Inspired by [63], an additional 2D deformation field of the camera pixel space in form of a 6-layer ReLU MLP is added to account for possible distortions in the input images or inaccuracies of the intrinsic camera parameters. Note that this correction field is the same for every frame. During optimization, camera rays are first shifted with the

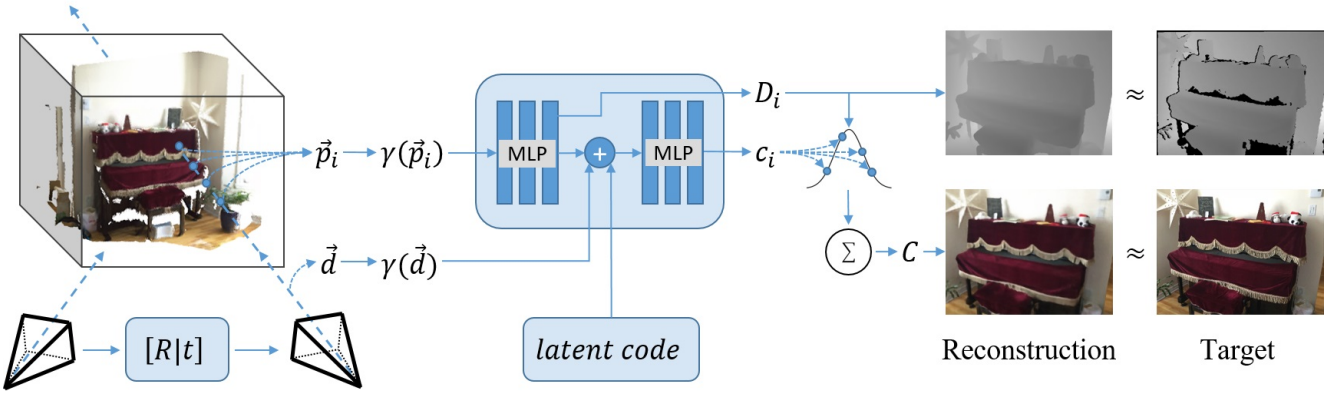


Figure 2: Differentiable volumetric rendering is used to reconstruct the scene that has been captured using an RGB-D camera. Multi-layer perceptrons (MLPs) are used to store a continuous volumetric representation of the scene, giving access to radiance c_i and signed distance values D_i of point $\vec{p}_i$ ($\gamma(\vec{p}_i)$ applies a positional encoding to the point coordinate). To optimize the parameters of these MLPs, the latent code (used to correct for effects like auto-white balancing) and the camera poses, we shoot rays $\vec{d}$ through the input image pixels and integrate the radiance along the ray. In each optimization step, the parameters are updated such that the error between the integrated color C and the observed image, as well as the distance between the observed signed-distance values and the prediction D_i is reduced. To model view-dependent effects, we condition the color MLP that emits the radiance c_i with the positionally encoded viewing direction $\gamma(\vec{d})$.

2D vector retrieved from the deformation field, before being transformed to world space using the camera pose $\mathcal{T}_i$. Finally, sample points are drawn on the corrected rays.

3.2. Optimization

We optimize our scene representation network by randomly sampling a batch of P_b pixels from the input dataset of color and depth images. For each pixel p in the batch, a ray is generated using its corresponding camera pose. S_p sample points are generated on the ray. Our global objective function $\mathcal{L}(\mathcal{P})$ is minimized w.r.t. the unknown parameters $\mathcal{P}$ (the network parameters Θ and the camera poses $\mathcal{T}_i$) over all B input batches and is defined as:

$$\mathcal{L}(\mathcal{P}) = \sum_{b=0}^{B-1} \lambda_1 \mathcal{L}^b_{rgb}(\mathcal{P}) + \lambda_2 \mathcal{L}^b_{fs}(\mathcal{P}) + \lambda_3 \mathcal{L}^b_{tr}(\mathcal{P}). \quad (3)$$

$\mathcal{L}^b_{rgb}(\mathcal{P})$ measures the squared difference between the observed pixel colors $\hat{C}_p$ and predicted pixel colors C_p of the b-th batch of rays:

$$\mathcal{L}^b_{rgb}(\mathcal{P}) = \frac{1}{|P_b|} \sum_{p \in P_b} (C_p - \hat{C}_p)^2. \quad (4)$$

$\mathcal{L}^b_{fs}$ is a 'free-space' objective, which forces the MLP to predict a value of 1 for samples $s \in S^{fs}_p$ which lie between the camera origin and the truncation region of a surface:

$$\mathcal{L}^b_{fs}(\mathcal{P}) = \frac{1}{|P_b|} \sum_{p \in P_b} \frac{1}{|S^{fs}_p|} \sum_{s \in S^{fs}_p} (D_s - 1)^2. \quad (5)$$

For samples within the truncation region ($s \in S^{tr}_p$), we apply $\mathcal{L}^b_{tr}(\mathcal{P})$, the signed distance objective of samples close to the surface.

$$\mathcal{L}^b_{tr}(\mathcal{P}) = \frac{1}{P_b} \sum_{p \in P_b} \frac{1}{|S^{tr}_p|} \sum_{s \in S^{tr}_p} (D_s - \hat{D}_s)^2. \quad (6)$$

In our experiments, we use a truncation region of 5 cm which we internally map to a range of $[-1, 1]$ (1 for samples in front of the surface, to -1 for sample points behind the surface).

The S_p sample points on the ray are generated in two steps. In the first step S'_c sample points are generated on the ray using stratified sampling. Evaluating the MLP on these S'_c sample points allows us to get a coarse estimate for the ray depth by explicitly searching for the zero-crossing in the predicted signed distance values. In the second step, another S'_f sample points are generated around the zero-crossing and a second forward pass of the MLP is performed with these additional samples. The output of the MLP is concatenated to the output from the first step and color is integrated using all $S'_c + S'_f$ samples, before computing the objective loss. It is important that the sampling rate in the first step is high enough to produce samples within the truncation region of the signed distance field, otherwise the zero-crossing may be missed.

We implement our method in Tensorflow using the ADAM optimizer with a learning rate of 5×10^{-4}. We set the loss weights to $\lambda_1 = 0.1$, $\lambda_2 = 10$ and $\lambda_3 = 6 \times 10^3$. We train all of our experiments for 2×10^5 iterations, where

in each iteration we compute the gradient w.r.t. $|P_b| = 1024$ randomly chosen rays. We set number of S'_f samples to 16. S'_c is chosen so that there is on average one sample for every 1.5 cm of the ray length. The ray length itself needs to be greater than the largest distance in the scene that is to be reconstructed and ranges from 4 to 8 meters in our scenes.

4. Results

In the following, we evaluate our method on real, as well as on synthetic data. For the shown results, we use Marching Cubes [32] with a spatial resolution of 1 cm to extract a mesh from the reconstructed signed distance function.

Results on real data. We test our method on the ScanNet dataset [7] which provides RGB-D sequences of room-scale scenes. The data has been captured with a StructureIO camera which provides quality similar to that of a Kinect v1. The depth measurements are noisy and often miss structures like chair legs or other thin geometry. To this end our method proposes the additional usage of a dense color reconstruction loss, since regions that are missed by the range sensor are often captured by the color camera. To compensate for the exposure and white balancing of the used camera, our approach learns a per-frame latent code as proposed in [34]. In Fig. 4, we compare our method to the original ScanNet BundleFusion reconstructions which often suffer from severe camera pose misalignment. Our approach jointly optimizes for the scene representation network as well as the camera poses, leading to substantially reduced misalignment artifacts in the reconstructed geometry. In Fig. 3, we show that with the help of the photometric loss, the MLP can predict valid signed distance values in areas with no depth measurements (*e.g.*, stool legs).

Quantitative evaluation. We perform a quantitative evaluation of our method on a synthetic scene[1] for which the ground truth geometry and a camera trajectory of 1676 frames are known. For each frame of the trajectory we render a photo-realistic image using Blender [5, 9]. We apply noise and artifacts, similar to those of a depth sensor, to the ground truth depth maps [1, 2, 19]. These depth maps are then given as input to BundleFusion [8] to compute a geometry reconstruction and estimate the trajectory poses. Using these poses as initialization, we train our method to learn the TSDF based on the noisy input depth and the rendered color images.

In addition to a comparison with BundleFusion, we compare our results to two other methods that use both color and depth data to reconstruct geometry. The first one is a NeRF network trained with an additional depth loss. NeRF proposes using the expected ray termination distance as a way to visualize the depth of the scene. In our baseline, we add an additional loss to NeRF where this value is compared against the input depth using an L2 loss. Note that this baseline still uses NeRF's density field to represent geometry. The other method is a combination of COLMAP [45, 46, 47] and screened Poisson surface reconstruction [24]. Camera pose estimates from COLMAP are used to project all depth maps back to world space. Screened Poisson surface reconstruction is performed on the resulting point cloud to produce a mesh. Tab. 1 presents a numerical comparison between our method and the baseline approaches. The scene used for this evaluation is shown in Fig. 6.

[1] Synthetic scene downloaded from: https://blendswap.com/blend/5014

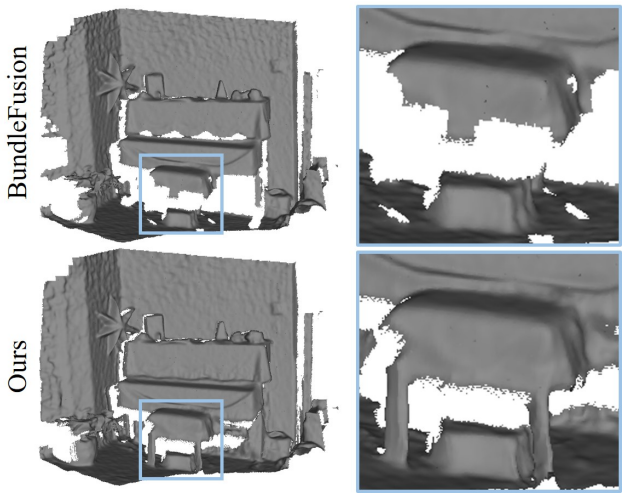


Figure 3: The photometric loss encourages correct depth prediction in areas where the depth sensor did not capture any measurements.

Method	C-ℓ_1 ↓	IoU ↑	NC ↑	F-score ↑
BundleFusion	0.062	0.528	0.869	0.701
COLMAP + Poisson	0.083	0.512	0.840	0.688
NeRF + Depth	0.073	0.385	0.716	0.619
Ours	**0.027**	**0.744**	**0.910**	**0.909**

Table 1: We compare the quality of our reconstruction on a synthetic scene for which ground truth data is available. The Chamfer ℓ_1 distance, normal consistency and the F-score [27] are computed between point clouds sampled with a density of 1 point per cm^2. We use a threshold of 5 cm for the F-score. We further voxelize each mesh to compute the intersection-over-union (IoU) between the predictions and ground truth.

Ablation studies. We conduct several ablation studies to justify our choice of network architecture and training parameters. In Fig. 7, we show how different components of

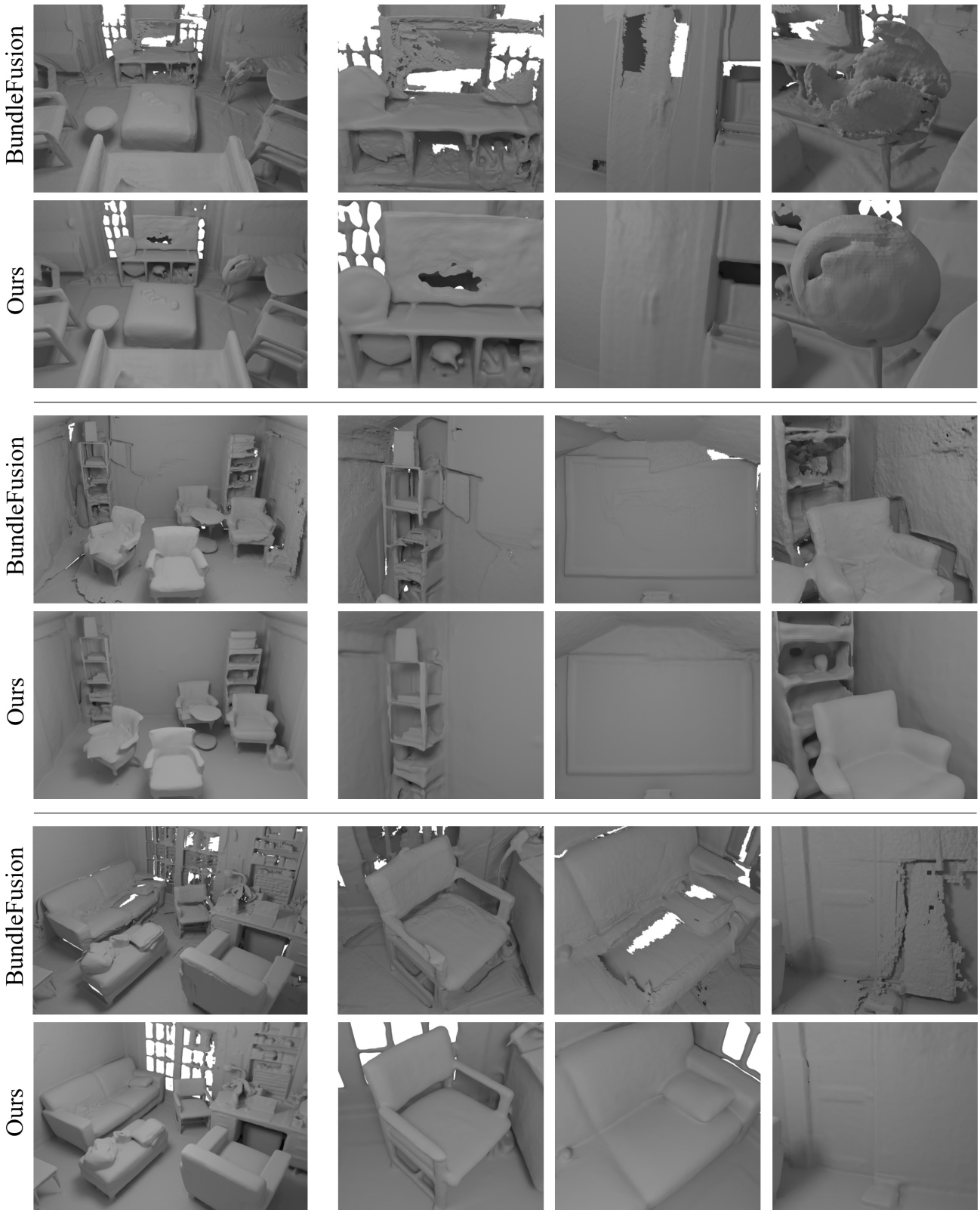
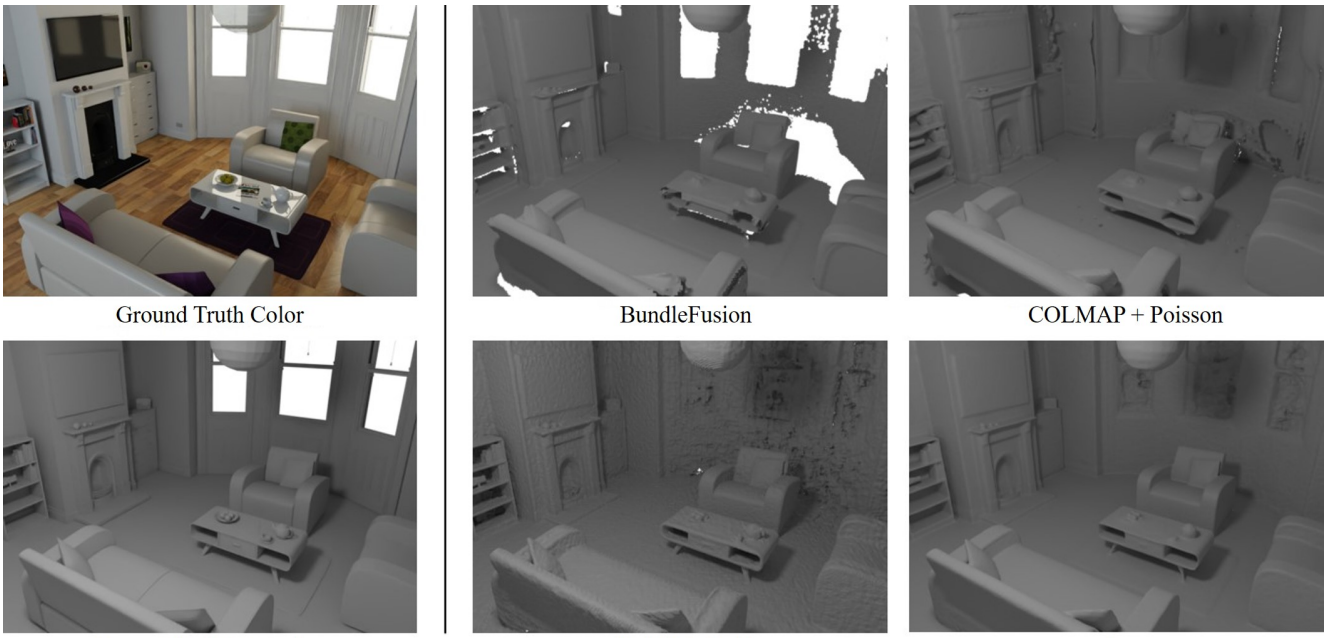


Figure 4: We compare our reconstructions with the original Scannet reconstructions of scenes 2, 12 and 50. Our method significantly reduces geometry misalignment and reconstructs geometry in regions where depth measurements were missing, such as the TV in the first row.

BundleFusion | Ours with Pose Correction | Ours with Pose and Distortion Correction

Figure 5: Our method improves the camera alignment over the baseline, as visible in the tiles of the floor. The additional distortion correction in image space, results in straight and aligned edges in the reconstruction.



Ground Truth Color | BundleFusion | COLMAP + Poisson

Ground Truth Geometry | NeRF + Depth | Ours

Figure 6: The first column shows a rendered frame from the recorded trajectory, along with the ground truth geometry. In the second and third column we show geometry that was reconstructed with baseline methods and our method. The BundleFusion reconstruction is incomplete in some regions, screened Poisson attempts to fit noise in the depth data, while the NeRF reconstruction suffers from noise in the density field. Our method manages to fill in gaps in geometry, while maintaining the smoothness of classic fusion approaches.

our method affect the geometry reconstruction. In particular, we justify our choice of moving from a density-based representation to a signed distance field. While representing scenes with a density field works great for color integration, extracting the geometry itself is a challenging problem. Although small variations in density may not affect the integrated color much, they cause visible noise in the extracted geometry and produce floating artifacts in free space. These artifacts can be reduced by choosing a different isolevel for geometry extraction with marching cubes, but this leads to a less complete reconstruction. In contrast, a signed distance field can model a smooth boundary between occupied and free space and we show that it can be faithfully represented by an MLP. However, the reconstruction quality is still limited by the provided camera poses. Optimizing for pose corrections further improves the quality of our reconstructions.

Figure 5 showcases the effects of our camera pose and distortion correction compared to BundleFusion. Blurry frames and sparse features lead to systematic camera pose errors in BundleFusion. Our method attempts to improve both the camera poses and the camera distortion model, and is able to bring the scene content into better alignment, thus achieving higher reconstruction quality.

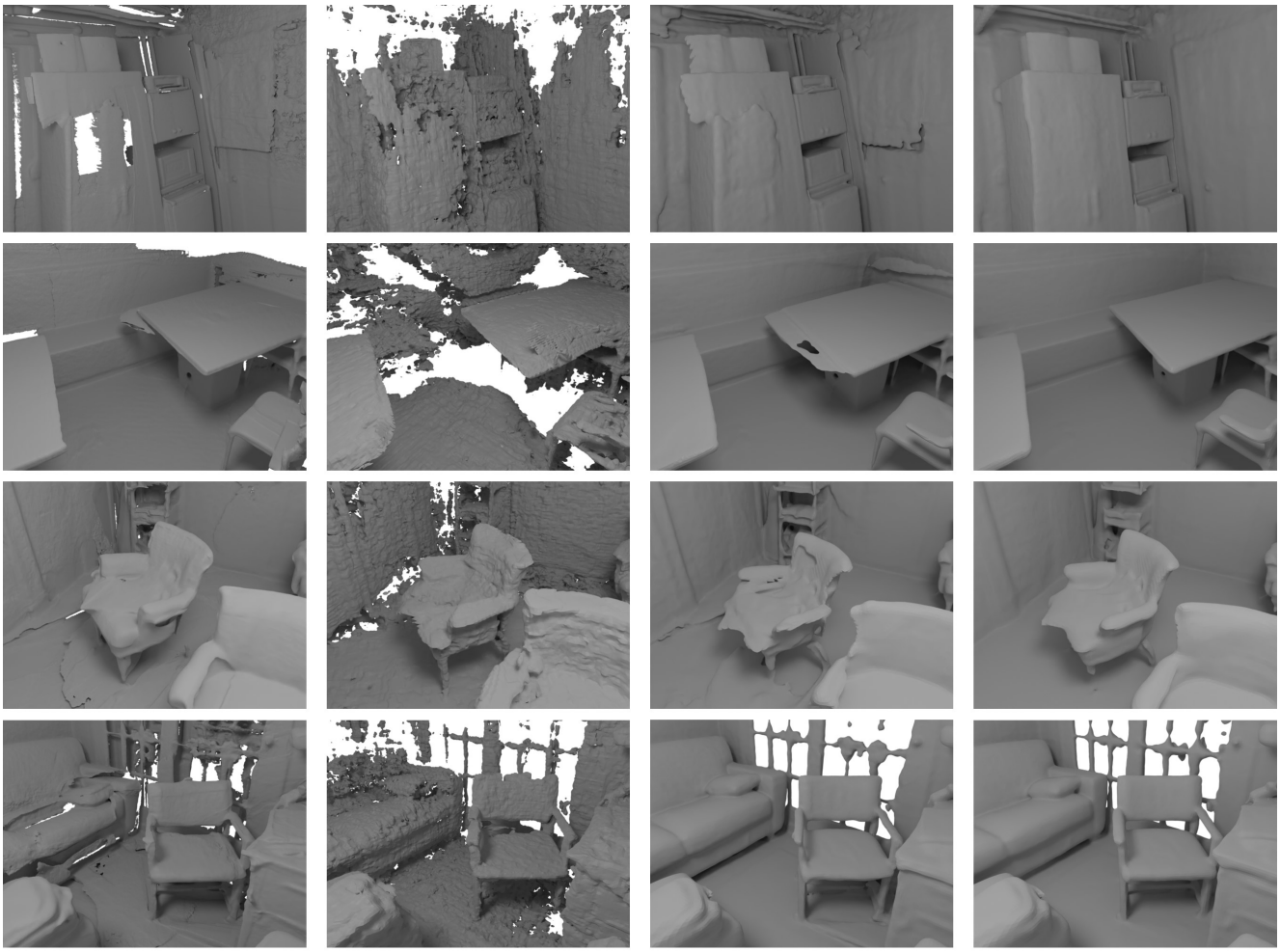


Figure 7: We compare our model without pose optimization and our full model with both the pose estimation and distortion correction to BundleFusion and a NeRF trained with depth supervision in scenes 2, 5, 12 and 50 of the ScanNet dataset. Our model without pose optimization recovers smoother meshes, but still suffers from misalignment artifacts, which are solved by our full model to recover a clean reconstruction.

Limitations and future work. The primary limitation of our method is its speed. Compared to classical fusion approaches, training a neural network to learn a signed distance field is several orders of magnitudes slower. A neural network requires only a fraction of the memory that an explicit voxel grid requires, but this comes at the cost of missing high-frequency local detail in very large scenes. Nevertheless, we hope to inspire future work in this area. We believe that there is significant potential in using neural rendering approaches to improve geometry reconstruction. One avenue worth exploring is the use of local models to improve sharpness and local detail. We also believe that novel view synthesis itself can benefit from better underlying geometry models and hope to see further research in this direction.

5. Conclusion

We have presented a new method for 3D surface reconstruction from RGB-D sequences by leveraging the recent success of implicit novel synthesis techniques. Instead of learning neural radiance fields with densities as geometric representation, we re-formulate the approach to use a truncated signed distance representation. This allows us to efficiently incorporate depth observations while still benefiting from the differentiable volumetric rendering of the original radiance field formulation. As a result, we obtain high-quality surface reconstructions, outperforming existing works on traditional RGB-D fusion as well as learned representations. Overall, we believe our work is a stepping stone towards leveraging the success of implicit, differentiable representations for 3D surface reconstruction.

Acknowledgements

This work was supported by a Google Gift Grant, a TUM-IAS Rudolf Mößbauer Fellowship, an NVidia Professorship Award, the ERC Starting Grant Scan2CAD (804724), and the German Research Foundation (DFG) Grant Making Machine Learning on Static and Dynamic 3D Data Practical. We would also like to thank Angela Dai for the video voice-over.

References

[1] Jonathan T. Barron and Jitendra Malik. Intrinsic scene properties from a single rgb-d image. *CVPR*, 2013. 5

[2] Jeannette Bohg, Javier Romero, Alexander Herzog, and Stefan Schaal. Robot arm pose estimation through pixel-wise part classification. *ICRA*, 2014. 5

[3] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *arXiv*, 2020. 3

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[5] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 5

[6] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 303–312, New York, NY, USA, 1996. Association for Computing Machinery. 2

[7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 5

[8] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG)*, 36(4):76a, 2017. 2, 3, 5, 12

[9] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. *arXiv preprint arXiv:1911.01911*, 2019. 5

[10] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision (ECCV)*, September 2014. 2

[11] J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *2013 IEEE International Conference on Computer Vision*, pages 1449–1456, 2013. 2

[12] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5515–5524, 2016. 2

[13] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, 2014. 2

[14] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018. 2

[15] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010. 2

[16] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[17] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017. 2

[18] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007. 2

[19] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. *ICRA*, 2014. 5

[20] Vu Hoang Hiep, Renaud Keriven, Patrick Labatut, and Jean-Philippe Pons. Towards high-resolution large-scale multi-view stereo. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1430–1437. IEEE, 2009. 2

[21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2

[22] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011. 2

[23] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. *arXiv preprint arXiv:1708.05375*, 2017. 2

[24] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3), July 2013. 5

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 12

[26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[27] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Trans. Graph.*, 36(4), July 2017. 5

[28] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38(3):199–218, 2000. 2

[29] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016. 2

[30] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. *arXiv preprint arXiv:2011.13084*, 2020. 2

[31] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes. *ACM Transactions on Graphics*, 38(4):1–14, Jul 2019. 1, 2

[32] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, page 163–169, New York, NY, USA, 1987. Association for Computing Machinery. 3, 5

[33] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999. 2

[34] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021. 2, 3, 5

[35] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. 2, 3

[36] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[37] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2, 3, 12

[38] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011. 2

[39] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 2013. 2, 3

[40] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[41] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo-Martin Brualla. Deformable neural radiance fields. *arXiv preprint arXiv:2011.12948*, 2020. 2

[42] Shunsuke Saito, , Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. *arXiv preprint arXiv:1905.05172*, 2019. 2

[43] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2020. 2

[44] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, pages 131–140, 2001. 2

[45] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 5

[46] Johannes Lutz Schönberger, True Price, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Asian Conference on Computer Vision (ACCV)*, 2016. 5

[47] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 5

[48] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

[49] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *arXiv*, 2020. 2

[50] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019. 2

[51] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, 2019. 2

[52] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 2

[53] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision*, 2016. 2

[54] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. In *Computer Graphics Forum*, volume 39, pages 701–727. Wiley Online Library, 2020. 1, 2

[55] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 2

[56] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. *CVPR*, 2021. 3

[57] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters, 2021. 3

[58] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NIPS*, 2016. 2

[59] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018. 2

[60] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. 2020. 3

[61] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 3

[62] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 175–185, 2018. 2

[63] Qian-Yi Zhou and Vladlen Koltun. Color map optimization for 3d reconstruction with consumer depth cameras. 33(4), July 2014. 3

BundleFusion | NeRF with Depth | Ours w/o Pose Optimization | Ours-Full

Figure 8: Our method obtains a high-quality 3D reconstruction from an RGB-D input sequence by training a multi-layer perceptron. In comparison to state-of-the-art methods like BundleFusion [8] or the theoretical NeRF [37] with additional depth constraints, our approach results in cleaner and more complete reconstructions. As can be seen, the pose optimization of our approach is key to resolving misalignment artifacts.

APPENDIX

In this appendix we show an ablation study on additional scenes from the ScanNet dataset (see Fig. 8). For the purpose of reproducibility, we also provide further details on the parameters that were used for optimization in each of the scenes.

A. Implementation Details

We implement our method in TensorFlow v2.4.1 using the ADAM [25] optimizer with a learning rate of 5×10^{-4} and an exponential learning rate decay of 10^{-1} over 2.5×10^5 iterations. In each iteration, we compute a gradient w.r.t. $|P_b| = 1024$ randomly chosen rays. We set number of S'_f samples to 16. S'_c is chosen so that there is on average one sample for every 1.5 cm of the ray length. Tab. 2 gives an overview of ray length and number of samples for each of the experiments. Internally, we translate and scale each scene so that it lies within a $[-1, 1]^3$ cube. Depending on scene size, our method takes between 15 and 25 hours to converge on a single Nvidia RTX 2080 Ti.

We set the loss weights to $\lambda_1 = 0.1$, $\lambda_2 = 10$ and $\lambda_3 = 6 \times 10^3$. We use 8 bands for the positional encoding of the point coordinates and 4 bands to encode the view direction vector. To account for possible distortions or inaccuracies of the intrinsic parameters, a 2D deformation field of the camera pixel space in form of a 6-layer MLP, with a width of 128, is added.

Scene	S'_c	$ray\ length$ (m)
Scene 0	512	8
Scene 2	256	4
Scene 5	256	4
Scene 12	320	5
Scene 24	512	8
Scene 50	256	4
Scene 54	256	4
Synthetic	512	8

Table 2: We list the number of samples S'_c and the ray length in meters that were used to reconstruct each of the ScanNet scenes and the synthetic scene. Note that these settings are dependent on the scene size.