Seminar “Inverse Rendering”

Written Report

Zeeshan Ahmad

Matrikel Number: 23072973

Friedrich-Alexander-Universität
Technische Fakultät

MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction
Zehao Yu, Songyou Peng and Andreas Geiger

1 Introduction

Digital Cameras project the 3D world onto a 2D image plane during rendering and lose 3D information including shape and texture. However, many real-world use cases require 3D information like robotics, animation, virtual reality, gaming, etc. Though various techniques have been proposed to solve this ill-posed inverse rendering problem but still it requires a lot more work. Current implicit surface reconstruction approaches work well for simple scenes and densely sampled scenes, however, they fail for large-scale scenes and few-shot reconstruction.

This paper studies the effect of off-the-shelf monocular cues for 3D reconstruction combined with neural implicit scene representations to overcome reconstruction ambiguities. Previously, the implicit-based methods used RGB reconstruction loss which doesn't provide enough constraints in large texture-less regions and results in an under-constrained approach. However, this method uses the depth and normal cues obtained from a pre-trained model to provide additional constraints to the implicit surface reconstruction methods during optimization and result in high-fidelity 3D reconstructions. The main contributions of this paper are listed below:

- It investigates the effect of monocular geometric cues to remove the reconstruction ambiguities and produce smooth 3D reconstructions in less time.
- The authors experimented with different design choices for neural implicit scene representation including MLPs and grid-based approaches.
- The paper also provides a detailed study on a variety of datasets ranging from simple object-level scenes to large-scale indoor scenes.

2 Scientific Context

Various methods have been proposed to represent the 3D geometry using coordinate-based neural networks because of their low memory footprint and inductive bias. Occupancy Networks [4] proposed a network that predicts the occupancy of any 3D point, and represents the surface as a decision boundary of a classifier. Though it works well for simple scenes and single objects but does not scale well to large and complex scenes. To solve this problem, Convolutional Occupancy Networks [7] proposes an encoder-decoder architecture comprising a CNN encoder and an occupancy decoder. This CNN encoder better encodes large scenes because it can make use of translation equivariance inductive bias to represent large scenes with better geometric details and a low memory footprint.

The implicit scene representation methods that we have discussed require 3D supervision during training which is not readily available. To overcome this challenge, several efforts have been made to train the implicit-based methods using only 2D multi-view images. One of the early works, Differentiable Volumetric Rendering [6] learns occupancy-based implicit 3D representation using only 2D posed images and object masks. It results in very detailed reconstructions but requires object masks. Following NeRF [5], various methods have been proposed to render the implicit scenes using volume rendering that does not require the object masks. One of them is VolSDF [8] which defines volume density as a function of geometry where geometry is represented by a Signed Distance Field (SDF). Nevertheless, such methods don't work well for large-scale scenes with texture-less areas.

Fortunately, the results for large scenes with texture-less areas can be improved by using the monocular geometric priors. A concurrent work Manhattan-SDF [3] explores the effect of using Manhattan world priors to overcome the reconstruction ambiguities in low-textured areas. However, this paper is based on the assumption that the monocular depth and normal maps obtained from pre-trained Omnidata [1] are accurate enough to provide additional constraints for

3D reconstruction. During the optimization of neural implicit surface models, these monocular geometric cues provide an additional supervision signal resulting in an improved reconstruction quality and also reducing the optimization time.

Inspired by the success of monocular priors in this paper, AG3D [10] has been proposed that generates novel 3D humans with accurate geometry and appearance only from 2D RGB images and their corresponding normal maps. However, it uses separate normal maps for the face and rest of the body to come up with better constrained optimization. Another recent work NICER-SLAM [9] learns a neural implicit representation along with optimizing the camera poses for a RGB-only SLAM. It represents the scene geometry with Signed Distance Field (SDF) and also uses monocular geometric cues for additional supervision.

3 Background and Foundations

3.1 Signed Distance Field

SDF can be seen as a function that returns the distance s to the closest surface given a 3D point $\mathbf{x}$ as input. The sign of the distance indicates whether it is inside or outside, for inside it is negative otherwise positive.

$$f : \mathbb{R}^3 \to \mathbb{R} \qquad \mathbf{x} \mapsto s = \text{SDF}(\mathbf{x}) \tag{1}$$

SDF has the following properties:

- Surface is represented by a zero-level set.
- For any point $\mathbf{x}$ on surface, normal is defined as $\nabla f(\mathbf{x}) = \mathbf{N}(\mathbf{x})$.
- Gradient at point $\mathbf{x}$ satisfies the eikonal equation. i.e., $\|\nabla f\| = 1$.

4 Method

The proposed method reconstructs the scene geometry from posed multi-view images, however, it also makes use of monocular geometric priors during optimization to improve the results in texture-less and less observed areas. The method is divided into four sections, 4.1 discussed the details of various design choices for neural implicit scene representation, 4.2 briefly elaborated the volume rendering of neural implicit scene representations, 4.3 discussed the effect of using monocular cues during the optimization, and 4.4 explained the optimization details of the method. The Fig. 1 provides an overview of MonoSDF pipeline.

4.1 Implicit Scene Representations

The geometry is represented by a continuous signed distance function f as we have discussed in 3.1. However, the authors of this paper have experimented with several different design choices to parameterize f as mentioned below:

Dense SDF Grid. It stores the SDF values directly on the vertices of the discrete SDF grid $\mathcal{G}_\theta$. To compute the SDF value $\hat{s}$ for a 3D point $\mathbf{x}$ it uses `interp` which is simply tri-linear interpolation.

$$\hat{s} = \texttt{interp}(\mathbf{x}, \mathcal{G}_\theta) \tag{2}$$

Single MLP. The signed distance function f is parameterized by a simple neural network as f_θ. The positional encoding γ is computed for 3D point $\mathbf{x}$ and then passed to f_θ to predict $\hat{s}$.

$$\hat{s} = f_\theta(\gamma(\mathbf{x})) \tag{3}$$

Single-Resolution Feature Grid with MLP Decoder. This simple hybrid method uses MLP f_θ and and a feature-grid Φ_θ combined to parameterize the signed distance function. The MLP f_θ is conditioned on interpolated features of 3D point $\mathbf{x}$ obtained from Φ_θ.

$$\hat{s} = f_\theta(\gamma(\mathbf{x}), \texttt{interp}(\mathbf{x}, \Phi_\theta)) \tag{4}$$

Multi-Resolution Feature Grids with MLP Decoder. This design approach uses L feature-grids $\{\Phi_\theta^l\}_{l=1}^L$ of different resolutions and then combines those interpolated features to provide condition to f_θ. It helps to combine features at different resolutions to result in a more robust representation.

$$\hat{s} = f_\theta(\gamma(\mathbf{x}), \{\texttt{interp}(\mathbf{x}, \Phi_\theta^l)\}_l) \tag{5}$$

Note that the number of parameters of feature-grid are independent of the size of the feature-grid because it uses hashing to store the feature vectors in a compact way.

Color Prediction. The authors used another network $\mathbf{c}_\theta$ to predict the RGB color for the input 3D point $\mathbf{x}$ to optimize the model using photometric reconstruction loss.

$$\hat{\mathbf{c}} = \mathbf{c}_\theta(\mathbf{x}, \mathbf{v}, \hat{\mathbf{n}}, \hat{\mathbf{z}}) \tag{6}$$

The color network $\mathbf{c}_\theta$ takes as input a 3D point $\mathbf{x}$, viewing direction $\mathbf{v}$, analytically computed normal vector $\hat{\mathbf{n}}$, and a feature-vector $\hat{\mathbf{z}}$ and predicts the color $\hat{\mathbf{c}}$. Here the feature-vector $\hat{\mathbf{z}}$ is predicted by the second head of the implicit representation network or computed using tri-linear interpolation `interp` for dense SDF grid.

4.2 Volume Rendering of Implicit Surfaces

SDF-to-Density Transformation. The RGB images are rendered from implicit representations in order to use photometric consistency loss for optimization. To render an image, the ray is cast from the camera center $\mathbf{o}$ in the viewing direction $\mathbf{v}$ and M point $\mathbf{x}_\mathbf{r}^i = \mathbf{o} + t_\mathbf{r}^i \mathbf{v}$ are sampled along the ray $\mathbf{r}$, and their SDF value $\hat{s}_\mathbf{r}^i$ and

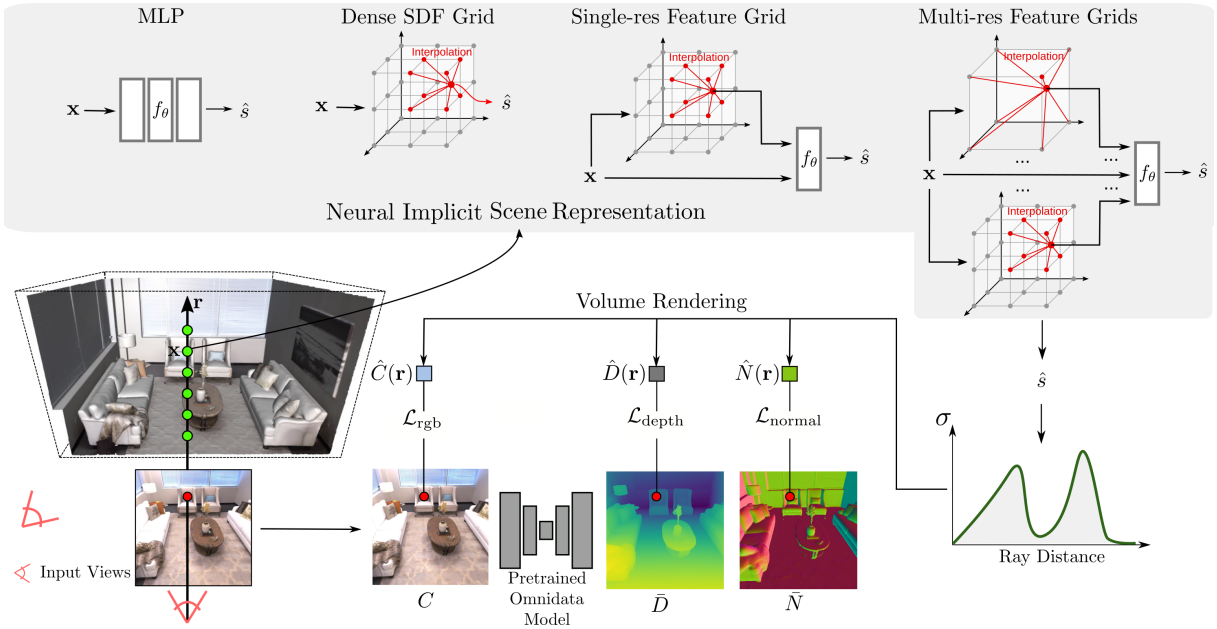

Fig. 1: **MonoSDF Pipeline.** A batch of rays is randomly sampled and their corresponding color, depth, and normal values are predicted and optimized w.r.t the RGB images, and monocular geometric priors. The method analyzes the different design choices for implicit scene representation by transforming the predicted SDF values to density and then using volume rendering to compute the photometric loss. For simplicity, the color prediction network is not shown here.

RGB value $\hat{\mathbf{c}}_\mathbf{r}^i$ is predicted. This predicted SDF value $\hat{\mathbf{s}}_\mathbf{r}^i$ is converted into density $\sigma_\mathbf{r}^i$ using Laplacian CDF following VolSDF [8] to perform volume rendering:

$$\sigma_\beta(s) = \begin{cases} \frac{1}{2\beta} \exp\left(\frac{s}{\beta}\right) & \text{if } s \leq 0 \\ \frac{1}{\beta}\left(1 - \frac{1}{2}\exp\left(-\frac{s}{\beta}\right)\right) & \text{if } s > 0 \end{cases} \tag{7}$$

where β is a learnable parameter.

Volume Rendering. The density $\sigma_\mathbf{r}^i$ is used to render color $\hat{C}(\mathbf{r})$ along the ray $\mathbf{r}$ similar to NeRF [5]:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{M} T_\mathbf{r}^i \, \alpha_\mathbf{r}^i \, \hat{\mathbf{c}}_\mathbf{r}^i \tag{8}$$

$$T_\mathbf{r}^i = \prod_{j=1}^{i-1} \left(1 - \alpha_\mathbf{r}^j\right) \qquad \alpha_\mathbf{r}^i = 1 - \exp\left(-\sigma_\mathbf{r}^i \delta_\mathbf{r}^i\right) \tag{9}$$

where $T_\mathbf{r}^i$ and $\alpha_\mathbf{r}^i$ are transmittance and alpha value of sample i along the ray $\mathbf{r}$.

Similarly, depth $\hat{D}(\mathbf{r})$ and normal $\hat{N}(\mathbf{r})$ can also be rendered for a particular ray $\mathbf{r}$:

$$\hat{D}(\mathbf{r}) = \sum_{i=1}^{M} T_\mathbf{r}^i \, \alpha_\mathbf{r}^i \, t_\mathbf{r}^i \qquad \hat{N}(\mathbf{r}) = \sum_{i=1}^{M} T_\mathbf{r}^i \, \alpha_\mathbf{r}^i \, \hat{\mathbf{n}}_\mathbf{r}^i \tag{10}$$

4.3 Exploiting Monocular Geometric Cues

Though implicit scene representation combined with volume rendering works really well, still they struggle to provide high-quality results for texture-less and less observed regions. The problem can be solved by providing the implicit representations with additional geometric priors e.g., depth and normal maps computed by pretrained Omnidata [1] model.

Monocular Depth Cues. Given an input RGB image, the depth map $\bar{D}$ can be obtained using the pre-trained model which provides the relative depth. These relative depth cues can provide semi-local relative information and improve the reconstruction quality where there are ambiguities regarding the depth.

Monocular Normal Cues. Similarly, monocular normal cues $\bar{N}$ can also be computed by using an off-the-shelf model that can provide local geometric details during reconstruction and remove noise.

4.4 Optimization

Reconstruction Loss. The predicted color $\hat{C}(\mathbf{r})$ from Eq. (8) is compared with the ground-truth 2D images $C(\mathbf{r})$ to compute RGB reconstruction loss:

$$\mathcal{L}_\text{rgb} = \sum_{\mathbf{r} \in \mathcal{R}} \|\hat{C}(\mathbf{r}) - C(\mathbf{r})\|_1 \tag{11}$$

where $\mathcal{R}$ is a set of rays in the minibatch.

Eikonal Loss. It regularizes the SDF values of 3D points by enforcing them to satisfy eikonal equation [2].

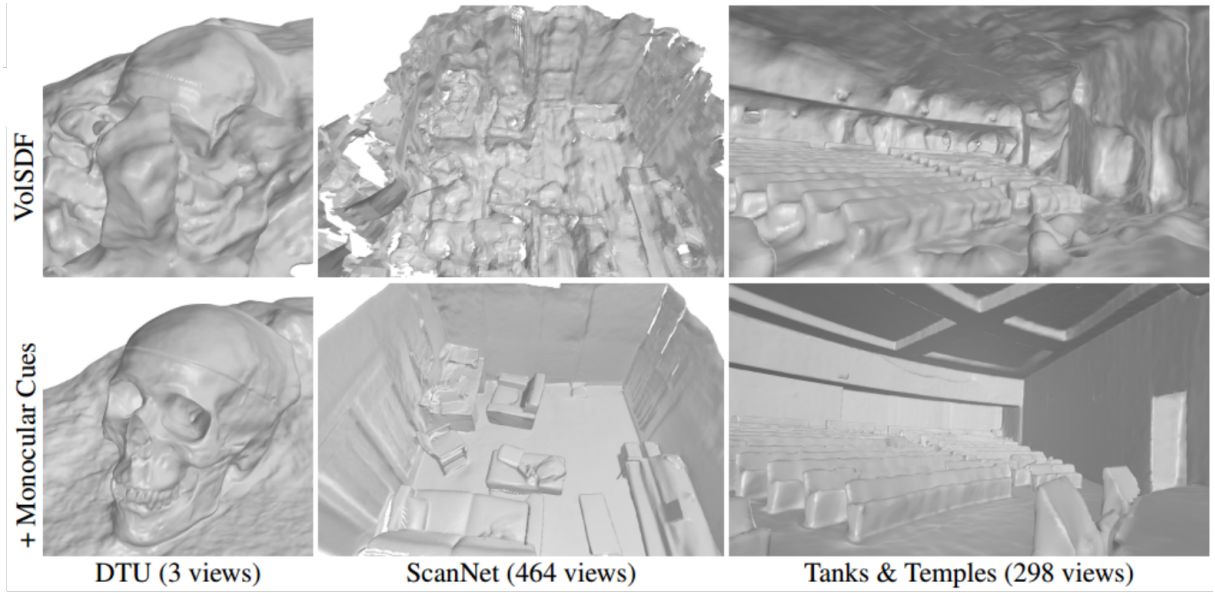

Fig. 2: **MonoSDF Results.** Top: VolSDF completely fails to reconstruct with limited views, and also in case of complex scenes containing multiple objects. Bottom: Implicit scene representations with monocular cues help fill the missing information for sparsely sampled scenes, and remove noise for large-scale scenes.

i.e., $\|\nabla f_\theta(\mathbf{x})\| = 1$.

$$\mathcal{L}_{\text{eikonal}} = \sum_{\mathbf{x}\in\mathcal{X}} (\|\nabla f_\theta(\mathbf{x})\|_2 - 1)^2 \tag{12}$$

Here $\mathcal{X}$ is joint set of uniformly sampled points and points near the surface.

Depth Consistency Loss. Rendered depth maps $\hat{D}(\mathbf{r})$ are compared with the predicted depth maps $\bar{D}$ to compute depth consistency loss:

$$\mathcal{L}_{\text{depth}} = \sum_{\mathbf{r}\in\mathcal{R}} \|(w\hat{D}(\mathbf{r}) + q) - \bar{D}(\mathbf{r})\|^2 \tag{13}$$

where w and q are scale and shift factors to align rendered and predicted depth maps $\hat{D}$ and $\bar{D}$ respectively. These w and q are computed analytically separately for each batch of rays.

Normal Consistency Loss. Similarly, the L1 loss and angular loss is computed for rendered normal $\hat{N}$ and predicted normal $\bar{N}$ to enforce normal consistency:

$$\mathcal{L}_{\text{normal}} = \sum_{\mathbf{r}\in\mathcal{R}} \|\hat{N}(\mathbf{r}) - \bar{N}(\mathbf{r})\|_1 + \|1 - \hat{N}(\mathbf{r})^\top \bar{N}(\mathbf{r})\|_1 \tag{14}$$

The overall loss used for optimization is a weighted sum of these individual losses:

$$\mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_1 \mathcal{L}_{\text{eikonal}} + \lambda_2 \mathcal{L}_{\text{depth}} + \lambda_3 \mathcal{L}_{\text{normal}} \tag{15}$$

where λ_1, λ_2, λ_3 are weight factors with values 0.1, 0.1, 0.05 respectively.

5 Results

The authors evaluated their model on four datasets: DTU (object-level scenes), Replica and ScanNet (real-world indoor scans), Tanks and Temples (large-scale indoor scenes). They reported the result for different choices of implicit neural scene representations and studied the effect of monocular geometric priors. For Replica, they highlighted that monocular cues improve results for both kinds of architectures grids as well as MLPs, however, feature-grids increase optimization speed while MLPs result in better accuracy.

For the ScanNet dataset, the images of the scenes are blurry and camera poses are also noisy. However, MLPs with monocular cues produce very good reconstructions shown in Fig. 2 because of their inductive smoothness bias. Similarly, neural implicit scene representations combined with monocular priors perform really well on Tanks and Temples dataset (see Fig. 2) which consists of huge indoor scenes. The authors claim that MonoSDF is the first method that can provide smooth reconstructions on this dataset.

They also evaluated their method in a sparse view (3 views) setting with DTU dataset. In Fig. 2, we can see few-shot reconstruction for DTU provided complete reconstructions for objects in the scene because of additional supervision signal provided by monocular cues. However, they also highlighted the failure case when the object is duplicated in front of each camera frustum. They hinted that this issue can be resolved by providing a sparse point cloud as an additional constraint during training.

6 Discussion & Conclusion

We have seen MonoSDF improves the 3D reconstruction and also reduces the optimization time by incorporating monocular geometric priors during the training of neural implicit scene representations. However, it is dependent on an off-the-shelf model that can predict depth and normal maps for the 2D scene images. Omnidata [1] generalizes really well across the scenes, however, it only accepts images with low-resolutions (384×384 pixels) but most of the real-world scenes have higher-resolutions and downsizing the images results in loss of performance. Though the authors have experimented with creating higher-resolution monocular cues in a sliding window manner over the image but they believe that it is not the best way to solve this issue.

In summary, this work investigated the different design choices for neural implicit scene representations, studied the effect of using monocular geometric priors, and experimented with a wide variety of datasets. It showed how monocular cues can improve the 3D reconstruction for large-scale scenes and less-observed regions. However, we have also discussed a failure case where the object is duplicated in front of each camera frustum. The authors hinted that the results can further be improved by using occlusion edges, planes, and curvature cues during training. They also suggested another interesting research direction to optimize scene representations and camera parameters jointly which will reduce the effect of motion blur and noisy camera poses.

Apart from this, recognition cues from classifiers can also be utilized during the training of implicit scene representations. The pretrained image classifiers are provided with a lot of augmented and multi-view image data that enables them to classify an object from almost every possible angle. If we extract that knowledge from CNNs or ViTs and provide it to the neural implicit scene representations we can further reduce the reconstruction errors, however, providing this knowledge to implicit scene representations is challenging.

References

[1] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. 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*, pages 10786–10796, 2021.

[2] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020.

[3] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *CVPR*, 2022.

[4] 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.

[5] 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.

[6] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[7] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision – ECCV 2020*, volume 3 of *Lecture Notes in Computer Science, 12348*, pages 523–540, Cham, August 2020. Springer.

[8] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[9] Zihan Zhu, Songyou Peng, Viktor Larsson, Zhaopeng Cui, Martin R Oswald, Andreas Geiger, and Marc Pollefeys. Nicer-slam: Neural implicit scene encoding for rgb slam. *arXiv preprint arXiv:2302.03594*, 2023.

[10] Jinlong Yang Michael J.Black Otmar Hilliges Andreas Geiger Zijian Dong, Xu Chen. AG3D: Learning to generate 3D avatars from 2D image collections. In *Arxiv*, 2023.