NoPose-NeuS: Jointly Optimizing Camera Poses with Neural Implicit Surfaces for Multi-view Reconstruction

Mohamed Shawky Sabae[1], Hoda Anis Baraka[1], and Mayada Mansour Hadhoud[1,2]

[1]Faculty of Engineering, Cairo University
[2]University of Science and Technology, Zewail City
{mohamedshawky911, hoda.baraka, mayada.hadhoud}@eng.cu.edu.eg

Editors: Marco Fumero, Emanuele Rodolà, Clementine Domine, Francesco Locatello, Gintare Karolina Dziugaite, Mathilde Caron

Abstract

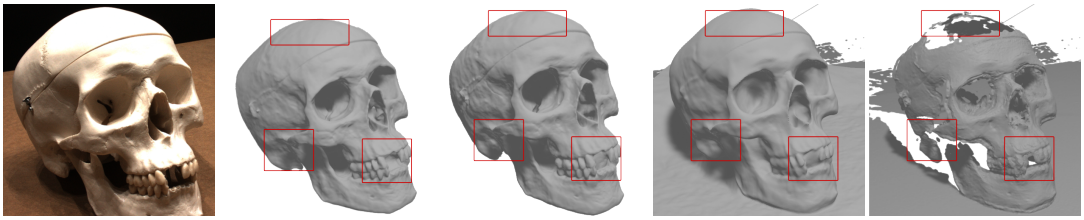
Learning neural implicit surfaces from volume rendering has become popular for multi-view reconstruction. Neural surface reconstruction approaches can recover complex 3D geometry that are difficult for classical Multi-view Stereo (MVS) approaches, such as non-Lambertian surfaces and thin structures. However, one key assumption for these methods is knowing accurate camera parameters for the input multi-view images, which are not always available. In this paper, we present NoPose-NeuS, a neural implicit surface reconstruction method that extends NeuS to jointly optimize camera poses with the geometry and color networks. We encode the camera poses as a multi-layer perceptron (MLP) and introduce two additional losses, which are multi-view feature consistency and rendered depth losses, to constrain the learned geometry for better estimated camera poses and scene surfaces. Extensive experiments on the DTU dataset show that the proposed method can estimate relatively accurate camera poses, while maintaining a high surface reconstruction quality with 0.89 mean Chamfer distance.

1 Introduction

3D reconstruction from multi-view images is a fundamental problem in computer vision and computer graphics. Traditionally, 3D reconstruction pipelines, such as COLMAP [1] [2], contain multiple steps to generate multi-view depth maps and fuse them into a dense point cloud representation, which is then used to recover scene surfaces. These methods rely on correspondence matching between RGB images, which causes artifacts and missing regions due to matching errors. Following the recent advances in NeRF [3], new approaches [4] [5] [6] have emerged to learn neural implicit surfaces using volume rendering. Generally, these works aim to jointly optimize implicit geometry and color networks guided by photometric loss from differentiable volume rendering. The geometry network is often encoded as a multi-layer perceptron (MLP) representing a signed distance function (SDF).

A common assumption in classical and neural rendering based techniques is the existence of accurate camera parameters for the input multi-view images. Actual camera parameters are not always easy to obtain in real situations, such as in casually captured images. Consequently, Structure from Motion (SfM) [1] methods are used to estimate camera parameters and sparse 3D points before performing multi-view geometry optimization. Recent NeRF-based methods [7] [8] [9] [10] propose joint optimization techniques of camera parameters with radiance fields.

Proceedings of the I edition of the Workshop on Unifying Representations in Neural Models (UniReps 2023).



Reference Image | **NoPose-NeuS** (ours) | NeuS [6] | MonoSDF [15] | COLMAP [2]

Figure 1: An example of reconstructed surfaces from the DTU dataset [14], showing the reconstruction quality of our proposed method compared to other methods.

In this work, we enable camera pose optimization in SDF-based surface reconstruction methods, particularly NeuS [6]. Following NeRFtrinsic Four [11], we use an MLP to predict camera poses from Gaussian Fourier features of camera indices. We impose two additional constraints, which are multi-view feature consistency, inspired by MVSDF [12] and D-NeuS [13], and rendered depth loss. These additional losses constrain the learned geometry to jointly optimize camera poses along with geometry and color networks. Results on the DTU dataset [14] show that the proposed method estimates camera poses with high relative accuracy, while outperforming classical MVS methods in terms of reconstruction quality, and offering comparable quality to other SDF-based surface reconstruction methods that depend on accurate input camera parameters, as shown in Figure 1.

The main contributions of this paper are summarized as follows:

- We propose a joint optimization method of camera poses with geometry and color networks of NeuS [6] constrained by additional multi-view feature consistency and rendered depth losses, in order to maintain high quality geometry compared to other SDF-based surface reconstruction methods relying on input camera poses.
- We evaluate our proposed method both qualitatively and quantitatively on the DTU dataset [14] and show high surface reconstruction quality with relatively accurate camera poses compared with other baselines.

2 Related Work

2.1 Classical Multi-view Stereo

Traditional Multi-view Stereo (MVS) methods [2] [16] [17] aim to recover a global 3D dense representation of the scene from a set of posed overlapping images. These methods estimate the pixel-wise depth map of each input image using pairwise matching of RGB image patches, and then fuse the depth maps into a dense point cloud. As a post-processing step, surfaces are recovered from the dense point cloud using methods like Screened Poisson surface reconstruction [18]. As these methods require camera parameters for the input RGB images, Structure from Motion (SfM) [1] is first applied to recover the camera parameters and the sparse point cloud representation of the scene. The reconstruction quality of the classical MVS methods is heavily affected by the quality of the correspondence matching, which is usually difficult for regions without rich textures.

Recently, deep learning-based MVS methods [19] [20] [21] showed better performance in estimating depth maps and dense representations. Furthermore, transformer-based approaches [22] [23] can capture long-range context information across images, further enhancing the quality of the reconstruction. Similar to classical MVS methods, learning-based methods rely on input camera parameters and struggle with low-texture regions. In this work, we relax the assumption of having input camera extrinsics (poses) guided by multi-view feature consistency, inspired by MVS approaches, to constrain volume rendering for surface reconstruction.

2.2 Neural Implicit Surface Representation and Reconstruction

Neural implicit representations have gained increasing attention recently, because of the ability to learn continuous and highly-complex functions using simple neural networks. Implicit 3D scene

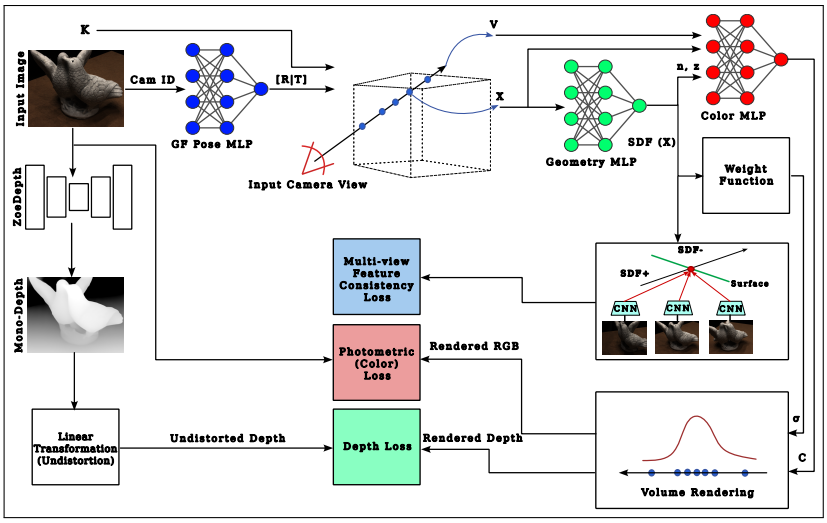


Figure 2: Overview of the proposed method. We aim to jointly learn camera poses along with scene geometry and color functions following the formulation of NeuS [6]. We use an MLP to predict the camera poses from Gaussian Fourier features of camera indices [11]. Furthermore, we impose two additional constraints: 1) multi-view feature consistency from MVSDF [12] and D-NeuS [13], 2) monocular depth supervision using predicted depth maps from ZoeDepth [27].

representation can encode 3D scenes with high spatial resolution into a multi-layer perceptron (MLP) with few layers. Consequently, this representation is successfully applied to multi-view 3D reconstruction. The related works in this area can be roughly categorized into surface rendering based methods and volume rendering based methods. Surface rendering based methods [24] [25] render the pixel color as the color of the intersection between the pixel ray and the scene geometry, which is vulnerable to self-occlusions and sudden changes in depth. On the other hand, volume rendering based methods [3] render the pixel color using alpha-composition of sampled point colors along the pixel ray. These methods are more robust to complex geometry changes, however the learned geometry is often noisy, due to the lack of constraints on the geometry level sets.

To mitigate such issues, methods, such as UNISURF [4], VolSDF [5] and NeuS [6], learn implicit geometry functions, represented as occupancy values or signed distance functions (SDF), using volume rendering. SDF-based methods [5] [6] are generally better, because the surface can be extracted as the zero-level set of the SDF using the Marching Cubes algorithm [26], which produces more accurate geometry. Similar to classical MVS, these methods assume knowing accurate camera parameters for the input images.

Following NeRF [3], methods, such as NeRFmm [7], BARF [8], SC-NeRF [9] and NoPe-NeRF [10], propose techniques to jointly optimize camera parameters with radiance fields to relax the assumption of knowing accurate camera parameters by imposing additional losses to the optimization process. In this work, we enable camera pose optimization in SDF-based surface reconstruction methods, particularly NeuS [6], by combining better camera parameterization and additional losses to constrain the learned geometry.

3 Proposed Method

Given a set of RGB images of a scene, the goal is to reconstruct its surface represented by the zero-level set of an implicit neural signed distance field (SDF) without knowing the camera poses or alternatively given rough initial camera poses. The overview of the proposed method is illustrated in Figure 2. We allow camera poses to be jointly optimized along with implicit neural networks. In this section, we first review the usage of volume rendering to learn SDF-based implicit neural surfaces. Then, we explain our parameterization of the camera poses and the two additional losses for improving the estimated camera poses and the overall surface quality.

3.1 SDF-based Surface Reconstruction using Volume Rendering

Using volume rendering to learn SDF-based implicit surfaces combines the advantages of surface rendering based and volume rendering based methods, where the scene space is constrained by a signed distance field. The surface S is then represented as the zero-set level of an implicit SDF field $S = \{x \in \mathbb{R}^3 | f(x) = 0\}$, where f is a function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ that maps a spatial position $x \in \mathbb{R}^3$ to its signed distance to the object surface. This mapping function is implicitly encoded using a multi-layer perceptron (MLP), which encodes the scene geometry. In addition to the MLP of scene geometry (SDF), another MLP is used to encode the color function $g : \mathbb{R}^3 \times \mathbb{S}^2 \times \mathbb{S}^2 \times \mathbb{R}^{N_f} \rightarrow \mathbb{R}^3$ that maps a spatial point $x \in \mathbb{R}^3$, its viewing direction $v \in \mathbb{S}^2$, its normal surface $n \in \mathbb{S}^2$ calculated by differentiating the SDF function $\nabla f(x)$, and a feature vector $v \in \mathbb{R}^{N_f}$ generated by the SDF network to the point color $c \in \mathbb{R}^3$. In order to train both networks, 3D points are sampled along a ray emitted through an image pixel as follows:

$$x(t) = o + tv | t \geq 0 \tag{1}$$

where o is the camera origin, v is the unit vector of the ray direction, and t is the distance between the 3D point x and the camera origin o. Then, sampled colors are weighted and accumulated along the ray using volume rendering in order to get the pixel color to compare with the ground-truth:

$$C(o, v) = \int_0^{+\infty} w(t) g(x(t), v, n, z) dt \tag{2}$$

where $C(o, v)$ is the output color of the corresponding pixel, $g(x(t), v, n, z)$ is the color of the 3D point $x(t)$ given the viewing direction v, and $w(t)$ is the weight of the 3D point $x(t)$. NeuS [6] introduced a weight function that is both unbiased and occlusion-aware:

$$w(t) = exp\left(- \int_0^t \rho(u) du \right) \rho(t) \tag{3}$$

where $\rho(t)$ is called an opaque density function, which is the counterpart of the density function $\sigma(t)$ in the standard volume rendering. Equation 2 can be approximated using discretization; refer to NeuS [6] for more details on this. The rendered pixel colors are then compared with the ground-truth input image pixel colors for supervising the networks training.

3.2 Camera Parameterization

The camera parameters $\Pi = K[R|t]$ include camera intrinsics K_i, which transform the points from the camera i coordinates into the image i coordinates, and extrinsics (poses) $T_i = [R_i|t_i]$, which transform the world coordinates into camera i coordinates. We assume that the camera intrinsics K are known, as they are usually included in the image metadata. We only consider optimizing the camera poses $[R|t]$ in this work, where $R \in SO(3)$, $t \in \mathbb{R}^3$ and $[R|t] \in SE(3)$.

Following NeRFtrinsic Four [11], the index of each camera is mapped to a higher-dimensional space using Gaussian Fourier feature mapping from Tancik et al. [28].

$$\gamma(v) = [cos(2\pi Bv), sin(2\pi Bv)]^T, B \in \mathbb{R}^{m \times d} \tag{4}$$

where v is the low-dimensional input and B is a matrix for the Gaussian mapping, whose values are sampled from $N(0, \sigma^2)$ and frequency parameter m.

These features are then passed through an MLP with GELU activation functions to predict the pose for each camera, which contains translation vector $t \in \mathbb{R}^3$ and rotation vector in axis-angle representation $r \in so(3)$ that is used to construct the rotation matrix $SO(3)$. This formulation gives the camera poses more degrees of freedom to learn than directly optimizing rotation and translation vectors, which enables joint optimization of camera, geometry, and color networks.

3.3 Multi-view Consistency

Generally, using multi-view consistency constraints is common in 3D reconstruction methods. This becomes more important when optimizing camera poses, as the multi-view consistency constraints ensure correct relative poses between cameras. Photo-consistency approaches use photometric distance across RGB images, which is typically used in classical MVS methods [2]. Meanwhile, feature consistency approaches compare pixels in feature maps of different views.

We use feature consistency on surface points to constrain camera and surface optimization. Similar to D-NeuS [13], we utilize the SDF values of the sampled 3D points to extract the surface points. This is done using linear interpolation to find the zero-crossing of the SDF values between the last positive and the first consecutive negative SDF values.

Once the surface points are obtained, features of this point are compared across multiple views, similar to MVSDF [12]. Features are extracted by applying a convolutional neural network (CNN), pretrained for supervised MVS [21], on the RGB images. The final multi-view feature consistency loss is formulated as follows:

$$L_{feature} = \frac{1}{N_c N_s} \sum_{i=1}^{N_s} |F_0(p_0) - F_i(K_i(R_i x^{'} + t_i))| \tag{5}$$

where N_c is the number of channels in the feature maps, N_s is the number of neighboring source views, F is the extracted feature map for a specific view, p_0 is the pixel through which the ray is cast in the reference view, x' is the interpolated surface point, and $K_i(R_i x^{'} + t_i)$ is the surface point projected on the source view i using its camera parameters K_i, R_i, t_i.

It is clear that the multi-view feature consistency loss imposes direct constraint on the predicted camera poses R_i, t_i to enforce correct relative pose between the reference and source cameras.

3.4 Depth Supervision

In order to improve the quality of the reconstructed surface and keep the estimated camera translation within reasonable limits, we apply monocular depth loss against ground-truth depth maps. Ground-truth depth maps $\bar{D}$ are predicted by applying a pretrained monocular depth predictor on the input RGB images. We choose ZoeDepth [27] for monocular depth prediction, which offers state-of-the-art monocular depth quality. However, the predicted depth maps from the RGB images are usually not multi-view consistent.

Consequently, we use monocular depth undistortion technique from Nope-NeRF [10], which considers learning scale and shift parameters $\{(\alpha_i, \beta_i)|i = 0...N-1\}$ for each view. These parameters are used to linearly transform monocular depth maps to recover multi-view consistent depth maps $\bar{D}^*$ to be used for depth supervision:

$$\bar{D}_i^* = \alpha_i \bar{D} + \beta_i \tag{6}$$

The scale α_i and shift β_i parameters are jointly optimized along with camera poses, geometry and color MLPs. Furthermore, the predicted depth maps $\hat{D}$ are obtained using volume rendering as follows:

$$\hat{D} = \int_0^{+\infty} w(t) t dt \tag{7}$$

Similar to Equation 2, $w(t)$ is the weight of the 3D point $x(t)$ and t is the distance between the distance between the 3D point $x(t)$ and the camera origin o from Equation 1.

The rendered depth maps $\hat{D}$ are then compared with the undistorted ground-truth depth maps $\bar{D}_i^*$ using L1 loss on N_r rays (pixels) in the minibatch:

$$L_{depth} = \frac{1}{N_r} \sum_{j}^{N_r} ||\hat{D}_j - \bar{D}_j^*||_1 \tag{8}$$

3.5 Overall Training Loss

The overall loss to jointly optimize camera poses, depth undistortion parameters, geometry network and color network is formulated as follows:

$$L = L_{rgb} + \lambda_1 L_{eikonal} + \lambda_2 L_{mask} + \lambda_3 L_{feature} + \lambda_4 L_{depth} \tag{9}$$

The color loss L_{rgb} is defined as L1 photometric loss between the rendered RGB $\hat{C}$ and the input RGB images C on N_r rays (pixels) in the minibatch:

$$L_{rgb} = \frac{1}{N_r} \sum_{j}^{N_r} ||\hat{C}_j - C_j||_1 \tag{10}$$

Moreover, Eikonal loss $L_{eikonal}$ is applied on the sampled points to regularize the gradients of the SDF field predicted by the geometry network f:

$$L_{eikonal} = \frac{1}{N_r N_p} \sum_{j,k}^{N_r, N_p} (||\nabla f(x_{j,k})||_2 - 1)^2 \tag{11}$$

where N_r is the number of rays (pixels) in the minibatch, and N_p is the number of sampled 3D points per ray. Also, we use an optional mask loss L_{mask}, which is defined as binary cross entropy loss against ground-truth mask, as described in the original NeuS paper [6].

Finally, we apply coarse-to-fine optimization from BARF [8], which adds increasingly higher frequencies to the positional encoding of both 3D position and viewing direction during training. This is proven to reduce the likelihood of converging to a local minimum for camera pose optimization.

4 Experimental Results

4.1 Dataset

To evaluate our method against other baselines, we use the DTU dataset [14], which is widely-used for evaluating 3D reconstruction methods with a challenging variety of geometry, materials and appearance. We choose the same 15 scenes as those used in IDR [25] and NeuS [6]. Each scene contains 48 or 64 images of a resolution of 1200×1600. Ground truth camera parameters are also provided to evaluate our estimated camera poses using relative pose error (RPE) between pairs of image views. Furthermore, reference point clouds for all scenes are provided in the dataset for quantitative evaluation using the Chamfer distance provided by the official dataset evaluation protocol.

4.2 Experimental Setup

Baselines. We compare the quality of our reconstructed geometry to the widely-used classical MVS pipeline COLMAP [1] [2], as well as other SDF-based surface reconstruction methods: NeuS [6] and MonoSDF [15], which offer state-of-the-art results on the DTU dataset. For MonoSDF, we use the MLP representation trained on all input views for fair comparison. Moreover, we quantitatively compare our estimated camera poses with those estimated by COLMAP using the relative pose error (RPE) between pairs of image views.

Implementation Details. Similar to NeuS [6], the geometry network contains 8 hidden layers, each of size 256 and a skip connection from the input to the output of the 4th layer. The color (radiance) network contains 4 hidden layers, each of size 256. We follow the same hierarchical sampling strategy for volume rendering from NeuS. Positional encoding is applied to position x and viewing direction

v, with frequencies of 6 and 4, respectively. However, we use a coarse-to-fine scheduling strategy, similar to BARF [8], to smoothly mask the frequencies of positional encoding on an interval $[0.1, 0.5]$. Moreover, the camera pose network is modeled as an MLP with 3 layers with a hidden size of 64 and GELU activation functions, similar to NeRFtrinsic Four [11]. The input frequency parameter m is set to 128, which results in an embedding size of 256. We allow the estimation of actual camera poses or relative transformation from an initial pose. By default, all cameras are zero initialized at the center of a unit sphere, following NeuS initialization of the SDF network. For multi-view feature consistency loss, we use $N_s = 2$ where each reference view is compared to two source views using $N_c = 32$ feature channels. We train our method for 300k iterations and 512 sampled rays per batch for 21 hours on a single Nvidia RTX 2080ti GPU. We use the ADAM optimizer [29] with a learning rate of $5e - 4$. Also, we set λ_1, λ_2, λ_3 and λ_4 to 0.1, 0.1, 0.5 and 0.01, respectively. After optimization, we use the Marching Cubes algorithm [26] to extract mesh from the learned SDF field using bounding boxes defined by estimated camera poses with volume size of 512^3 voxels.

4.3 Results

We conducted our comparisons with baselines both quantitatively and qualitatively. In Table 1, we report the Chamfer distances on the selected scenes from the DTU dataset [14]. The results show that our method offers comparable results to NeuS [6] and MonoSDF [15], which are SDF-based surface reconstruction methods that rely on input camera parameters. Meanwhile, our method outperforms the classical MVS pipeline COLMAP [2] in most cases. Furthermore, we show the relative pose errors in Table 2 against COLMAP, which is the only method in our baselines that optimizes camera poses. The results show that our method maintains accurate relative poses between different views as good as the classical MVS pipelines. This is mainly due to the imposed constraints by the multi-view feature consistency and depth losses. Note that the estimated camera poses are sensitive to initialization, as discussed in 4.4.

Scan	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	**Mean**
COLMAP [2]	**0.81**	2.05	0.73	1.22	1.79	1.58	1.02	3.05	1.40	2.05	1.00	1.32	0.49	0.78	1.17	1.36
NeuS [6]	1.00	**1.37**	0.93	**0.43**	1.10	0.65	**0.57**	1.48	**1.09**	0.83	0.52	1.20	**0.35**	**0.49**	**0.54**	**0.84**
MonoSDF [15]	0.83	1.61	**0.65**	0.47	**0.92**	0.87	0.87	**1.30**	1.25	**0.68**	0.65	**0.96**	0.41	0.62	0.58	**0.84**
NoPose-NeuS	0.91	1.51	0.95	0.44	1.01	**0.63**	0.79	1.53	1.22	0.88	**0.51**	1.35	0.39	0.55	0.67	0.89

Table 1: Quantitative evaluation on the DTU dataset using the Chamfer distance (lower values are better). Results of the baselines are reported from the original papers, except for COLMAP results, which are taken from MonoSDF paper [15]. The best score for each scan is marked in **bold**.

	Scan	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	**Mean**
RPE_r	COLMAP [2]	**0.52**	0.93	**0.62**	0.71	0.72	0.65	**0.61**	0.89	0.66	**0.71**	0.55	**0.64**	**0.57**	0.57	0.63	0.67
	NoPose-NeuS	0.55	**0.89**	0.69	**0.54**	**0.66**	**0.54**	0.62	**0.79**	**0.57**	0.75	**0.48**	0.66	0.60	**0.51**	**0.58**	**0.63**
RPE_t	COLMAP [2]	**0.95**	**1.01**	1.12	0.99	0.81	0.93	**0.67**	1.05	**1.15**	1.05	0.91	0.88	**0.91**	0.86	0.99	0.95
	NoPose-NeuS	1.01	1.08	**0.91**	**0.88**	**0.76**	**0.80**	0.87	**1.02**	1.19	**1.04**	**0.79**	**0.82**	0.96	**0.85**	**0.97**	**0.93**

Table 2: Quantitative evaluation of the estimated camera poses using relative pose error (lower values are better). We compare our estimated poses to those estimated by COLMAP in terms of relative rotation and translation errors. The rotation error (RPE_r) is reported in degrees, and the translation error (RPE_t) is scaled by 100. The best score for each scan is marked in **bold**.

Moreover, Figure 3 shows our qualitative results against the baselines. Our method is able to recover highly-accurate geometry, while jointly optimizing camera poses. COLMAP results suffer from noisy and discontinuous surfaces (scans 37, 65 and 106). Meanwhile, MonoSDF (MLP) can reconstruct smooth surfaces with high accuracy (scan 63), however it struggles with thin structures (scan 37). Our

method reconstructs continuous surfaces with a high level of quality, which outperforms COLMAP in all cases. Moreover, our method can handle complex geometry (scans 24, 37, 65 and 106), offering comparable geometry quality to NeuS and MonoSDF (MLP), which rely on accurate input camera parameters. However, flat surfaces are better in NeuS in some cases (scan 24 and 37). Also, MonoSDF (MLP) is better for smooth surfaces in other cases (scan 63).

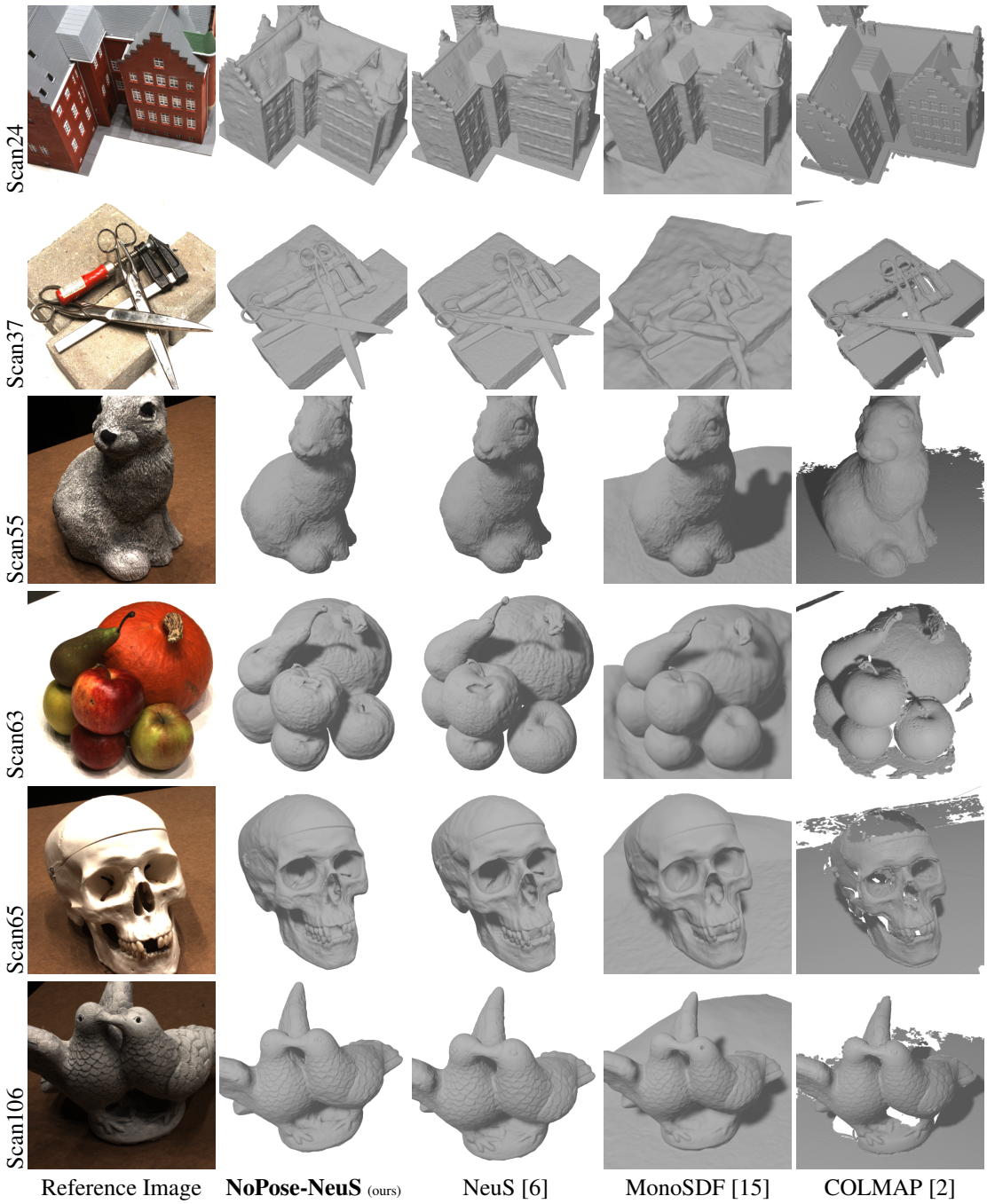


Figure 3: Qualitative evaluation of the surface reconstruction of our method against COLMAP [2], NeuS [6] and MonoSDF [15].

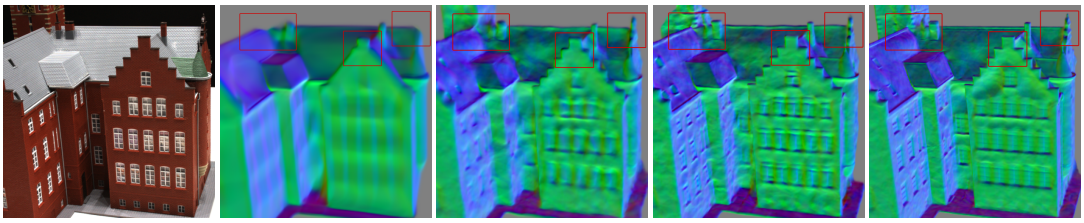
4.4 Discussion

Camera Pose Initialization. Good camera pose initialization is essential for our method to achieve high-quality reconstruction. We performed an analysis to show the effect of camera initialization. As illustrated in Figure 4, initializing camera poses (translation vector $t \in \mathbb{R}^3$ and rotation vector $r \in so(3)$) with random values that do not follow any structure can result in losing fine details, as the optimization process cannot recover the correct relative poses between different views. However, it is better that the camera poses are zero initialized (center of a unit sphere), as the geometry network is initialized to produce an approximate SDF of a unit sphere. Note that near-optimal initialization of camera poses helps the model to converge faster to good results.

Reference Image | a) Random Initial Poses | b) Centered Initial Poses

Figure 4: Qualitative results showing the effect of camera pose initialization on the reconstruction quality. We show rendered surface normal maps and RGB images of two models: a) initialized with random noisy camera poses, b) initialized with zero (centered) camera poses.

Ablation Study. We conducted an ablation study of the DTU dataset to evaluate the different components of our loss function. We started with the original losses from NeuS (L_{rgb}, $L_{eikonal}$ and L_{mask}), then progressively combined our additional losses. We show the rendered surface normal maps of DTU scan 24 for different settings in Figure 5. It is clear that using only the original NeuS losses results in over-smoothed geometry, while adding only feature consistency or depth loss results in noisy geometry. This is mainly due to high relative camera pose error. The final loss (described in Equation 9) offers the best reconstruction quality.



Reference Image | W/o feature/depth | W/o feature | W/o depth | **Ours**

Figure 5: Qualitative results of the ablation study conducted on the DTU dataset. To further illustrate the importance of the two additional losses, we show the rendered normal maps in different cases.

5 Conclusion

We introduced NoPose-NeuS, a neural implicit surface reconstruction method that enables camera pose optimization in NeuS [6]. In our work, we encode the camera poses as an MLP, which is jointly optimized with the geometry and color networks. Furthermore, we impose two additional losses, which are multi-view feature consistency and rendered depth loss, to constrain the learned camera poses and 3D geometry. Our method can recover relatively accurate camera poses, while maintaining the quality of the surface reconstruction. The main limitation of our approach is the sensitivity to the camera initialization, as it assumes a bounded scene following NeuS formulation. Therefore, an interesting future work is to relax this assumption from the camera parameterization and the SDF network initialization. Moreover, we can further optimize the camera intrinsics for full camera calibration.

References

[1] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] 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.

[3] 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, 2020.

[4] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction, 2021.

[5] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces, 2021.

[6] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction, 2023.

[7] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters, 2022.

[8] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields, 2021.

[9] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Animashree Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields, 2021.

[10] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior, 2023.

[11] Hannah Schieber, Fabian Deuser, Bernhard Egger, Norbert Oswald, and Daniel Roth. Nerftrinsic four: An end-to-end trainable nerf jointly optimizing diverse intrinsic and extrinsic camera parameters, 2023.

[12] Jingyang Zhang, Yao Yao, and Long Quan. Learning signed distance field for multi-view surface reconstruction, 2021.

[13] Decai Chen, Peng Zhang, Ingo Feldmann, Oliver Schreer, and Peter Eisert. Recovering fine details for neural implicit surface reconstruction, 2022.

[14] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, pages 1–16, 2016.

[15] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction, 2022.

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

[17] S. Galliani, Katrin Lasinger, Konrad Schindler, and Konrad Schindler. Gipuma: Massively parallel multi-view stereo reconstruction. 2016.

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

[19] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo, 2018.

[20] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo, 2020.

[21] Jingyang Zhang, Yao Yao, Shiwei Li, Zixin Luo, and Tian Fang. Visibility-aware multi-view stereo network, 2020.

[22] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. Transmvsnet: Global context-aware multi-view stereo network with transformers, 2021.

[23] Chenjie Cao, Xinlin Ren, and Yanwei Fu. Mvsformer: Multi-view stereo by learning robust image features and temperature-based depth, 2022.

[24] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision, 2020.

[25] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance, 2020.

[26] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, aug 1987.

[27] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth, 2023.

[28] 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, 2020.

[29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.