ParaSurRe: Parallel Surface Reconstruction with No Pose Prior

Wenyu Li, Zongxin Ye, Sidun Liu, Ziteng Zhang, Xi Wang, Peng Qiao†, Yong Dou
School of Computer
National University of Defense Technology
{wenyu18, yezongxin21, liusidun, zzt, xiwang, pengqiao, yongdou} @nudt.edu.cn

***Abstract*—Surface reconstruction from multi-view images without pose prior is challenging. Recent advances integrate incremental Structure from Motion (SfM) pipeline into surface optimization, enabling simultaneous surface reconstruction and pose estimation. However, due to the inherent incremental registration scheme of SfM, the efficiency of these methods is far from satisfactory, e.g., reconstruction of an object captured by 49 images costs over 9 hours using a high-end GPU. Inspired by divide-and-conquer strategy, we present a Parallel Surface Reconstruction method, coined as ParaSurRe, where image collections are divided into non-overlapped clusters and the incremental reconstructions are performed individually in each cluster. Owing to image partitioning, each cluster only accurately reconstructs a part of the surface. The major challenge is to merge multiple partial neural implicit surfaces into a complete one. We propose a confidence-aware surface fusion strategy and a geometry-guided refinement to tackle this issue. Experiments on real-world datasets demonstrate that ParaSurRe reconstructs delicate surfaces from unposed images, and achieves competitive pose estimation performance compared with state-of-the-art methods, with up to 6.5× speedup on a scene captured by 81 images.**

***Index Terms*—Surface Reconstruction, Neural Radiance Fields, Structure from Motion, Pose Estimation**

I. Introduction

Surface reconstruction from unordered image collections is a critical problem in computer vision. Recently, with the advances in differentiable rendering, neural implicit surface reconstruction methods represent the surface as zero level set of signed distance field (SDF), and achieve high-quality reconstructions [1]–[8].

A prerequisite for these methods is the camera poses recovered from Structure from Motion (SfM). However, SfM can not recover accurate poses in environments with reflections or untextured surfaces, and also lacks differentiability. Recent works [1] integrate incremental SfM pipeline into the learning process of neural implicit surface, enabling simultaneous surface reconstruction and pose estimation, referred as **neural incremental reconstruction methods**. With the top-down regularization from implicit representations, they manage to achieve accurate pose estimation results.

A major limitation of these methods is their long training time (over 9 hours for a scene captured by 49 images on a NVIDIA V100 GPU). The low efficiency is mainly caused by two reasons. Firstly, the incremental SfM pipeline requires to register the next-best-view frame one by one. Such iterative registration schemes remarkably decrease efficiency. Secondly, Neural Bundle Adjustment (NBA) that jointly optimizes implicit representation and camera poses is frequently leveraged in optimization. When provided with large collections of images, the NBA takes more time to converge.

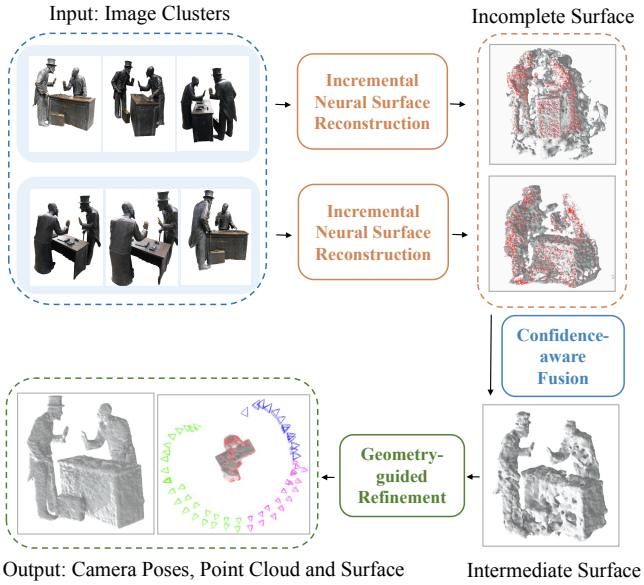


Fig. 1. ParaSurRe has three components: an incremental neural surface reconstruction network for each cluster, a confidence-aware SDF fusion module that merge partial surfaces into a complete surface, and a geometry-guided refinement network that optimizes both surfaces and poses. The blue, magenta, and green colors in the bottom left corner represent cameras belonging to different clusters.

Inspired by divide-and-conquer strategy from traditional parallel SfM approaches [9], [10], we divide input image collections into non-overlapped clusters, reconstruct these clusters independently, and merge all local results into a consistent model. We observe that the neural incremental reconstruction methods benefit from this strategy: First, with inter-independent clusters, the registration process within individual cluster can be conducted in parallel. Second, due to the limited number of local images, the neural bundle adjustment process in each cluster can converge in a short period. Third, reconstructing scenes in parallel can mitigate the endemic drift issue of incremental SfM.

† Corresponding authors.

Based on these insights, we present ParaSurRe, a neural implicit surface reconstruction method capable to train in parallel, as illustrated in Fig. 1. Due to image partitioning, the reconstructed surface in each local cluster is accurate only within the visible area. The primary challenge is to merge multiple partial neural implicit surfaces into a complete one. To tackle this challenge, we concentrate on three key aspects. Firstly, we introduce an SDF confidence prediction module to filter out erroneous predictions in invisible area. Secondly, we highlight the critical role of smooth maximum in preserving smoothness at the joint sections of the boundary. Lastly, we incorporate a geometry-guided refinement network to further improve surface quality and pose accuracy. Evaluations on DTU [11] and BlendedMVS [12] show that ParaSurRe can achieve competitive pose estimation accuracy to the state-of-the-art (SOTA) neural incremental method Level-S^{2}fM [1] and the defacto standard SfM software Colmap [13], while also providing comparable or superior surface quality to Level-S^{2}fM [1]. Notably, all ParaSurRe results are obtained through parallel training. For example, we divide the Horse scene from BlendedMVS dataset into 9 clusters, enabling simultaneous training on multiple GPUs and achieving 6.5$\times$ speedup.

In summary, the contributions of this work are three-fold:

- First, we propose a confidence-aware surface fusion strategy and following a geometry-guided refinement to merge partial neural implicit surfaces.
- Second, building upon our fusion method, we present the integrated pipeline ParaSurRe, facilitating parallel surface reconstruction from unposed images while simultaneously recover camera poses.
- Third, our proposed ParaSurRe can attain compelling surface quality and competitive pose estimation compared to state-of-the-art methods, while achieving good training efficiency.

II. Related Works

A. Neural Implicit Surface Reconstruction

Neural implicit representations have recently become popular for surface reconstruction. VolSDF [7] defines the volume density as Laplace's cumulative distribution function of the SDF representation and produces smooth geometry reconstruction. As a contemporary work, NeuS [5] proposes a new volume rendering method to train a neural SDF representation and leads to accurate surface reconstruction. MonoSDF [8] explores the geometric cues from monocular prediction network to improve reconstruction results of textureless areas. However, these approaches require accurate camera poses as input. Surface reconstruction from unposed images is still a challenging task.

B. Unposed Neural Implicit Network Optimization

In the absence of pose prior, many works have been proposed to simultaneously optimize pose and radiance field. NeRFmm [14] firstly takes pose as learnable parameters when optimizing NeRF in forward-facing scenes. BARF [15] proposes a coarse-to-fine training strategy to improve the optimization of pose parameters. GNeRF [16] utilizes generative adversarial networks for pose estimation. Nope-NeRF [17] incorporates monocular depth as prior to constrain the relative poses between consecutive images. The above methods optimize all poses globally and are sensitive to initial pose. Level-S^{2}fM [1] provides a new perspective to revisit incremental camera estimation and presents a neural incremental SfM pipeline. However, Level-S^{2}fM [1] takes an extremely long training time, which makes it unsuited for reconstructing densely captured objects.

III. Approach

Given a collection of images with unknown camrera poses, our method aims to reconstruct surface while simultaneously recover camera poses. Building upon the Level-S^{2}fM [1] framework, we address efficiency concerns through parallel learning. The overview of our framework is shown in Fig. 1. We will explain full pipeline in three parts: First, we present the data partitioning and local reconstruction method in Section III-A. Then, we describe the SDF fusion process in Section III-B. Finally, we discuss the refinement process in Section III-C.

A. Local Reconstruction

In this stage, we partition the scene graph into non-overlapping clusters, and each cluster undergoes individual neural incremental surface reconstruction.

Data Partitioning. We first establish exhaustive 2D-2D keypoint correspondences between image pairs and filter out non-overlapped pairs through geometric verification. Subsequently, we represent all valid correspondences as a scene graph, with images as nodes and keypoint correspondences as edges. The number of correspondences reflects the similarity between images. For dividing image collections into clusters, we iteratively apply the NCuts [18] algorithm to cut the scene graph until the number of images within each cluster is below a specified threshold. Following division, each cluster possesses a local scene graph comprising images and correspondences.

Incremental Neural Surface Reconstruction. Within each cluster C_i, we run neural incremental surface reconstruction approaches, Level-S^{2}fM [1], which estimates camera poses $\{P_j\}_i$, reconstructs the surface model F_i, and generates sparse point cloud X_i. Due to limited number of images in each cluster, a network with small model capacity is sufficient for training in a short time.

Determining transformations between clusters. Implicit surface models, reconstructed in their respective coordinate systems, need to be transformed into a consistent space before fusion. Specifically, $G_{ij} \in Sim(3)$ is a 3D similarity transformation which represents the relative transformation from C_i to C_j. To recover transformation G_{ij}, we register images $\{I_j\}$ from cluster C_j into cluster C_i by identifying 3D-2D correspondences and solving Perspective-N-Point(PNP) problems to obtain transformation candidates. Subsequently, we filter out poor candidate with few inliers and consider the median transformation as estimation.

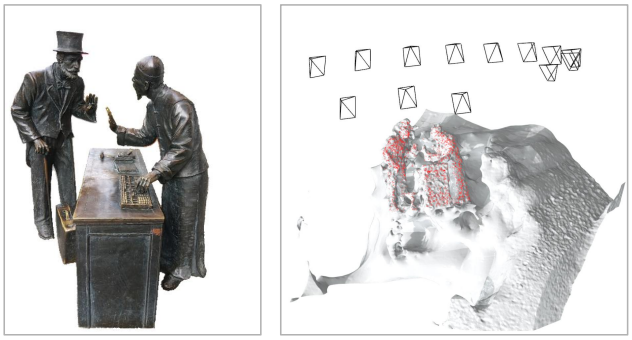


Fig. 2. The reconstructed surface, point cloud and cameras of one cluster. In this cluster, all cameras are positioned on one side of the statues, which causes the reconstructed surface being accurate only within the visible range.

B. Confidence-aware SDF Fusion

Without loss of generality, in this section, we will explain our SDF fusion algorithm with concentration on two clusters. Suppose that we have two neural implicit surface models F_1 and F_2, reconstructed by image clusters C_1 and C_2 respectively, and let G represents relative transformation from C_1 to C_2. For a given input 3D location x, we define the fusion process as the calculation of an appropriate SDF value from the outputs of the two models:

$$F(x) = Fusion(F_1(x), F_2(Gx)) \tag{1}$$

SDF confidence prediction. Given the limited field of view of each cluster, F_i only predicts accurate SDF value in the visible region, while predicts erroneous values behind visible surface, such as backgrounds, as illustrated in Fig. 2. We employ an additional network f_i to regress the confidence level for prediction $F_i(x)$ to filter out improper prediction. A key observation is that transmittance $T(t)$ has a close relation to confidence level: since transmittance $T(t)$ represents the probability that the ray travels from near bound t_n to t without hitting any other particle, points with high transmittance indicate their location in free space or on the surface, and should have high confidence levels; conversely, points with transmittance near 0 signify their position behind the first visible boundary, leading to low confidence levels. The confidence $f_i(r(t))$ of location $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ within the near and far bounds t_n and t_f is supervised by the transmittance $T(t)$ of that point, where o is the camera center, v is the unit direction vector of emitted ray:

$$\begin{aligned} \mathcal{L}_c &= \sum ||T(t) - f_i(\mathbf{r}(t))|| \\ \text{where } T(t) &= exp(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\, ds) \end{aligned} \tag{2}$$

To eliminate unreliable SDF values, we incorporate the confidence predictions from network f_i into the SDF values. We indentify point locations with confidence levels below a threshold and set their SDF values to infinitely small. This approach effectively mitigates the influence of inaccurate SDF estimation during the fusion stage.

Smooth maximum. In general, the visible regions of each cluster are complementary to each other. An example is shown in Fig. 3: the green cameras mainly capture the left side of the object, leaving the right side non-visible; the blue cameras can only capture the right side. The ideal complete surface should be the boundary of area where the red and blue curves intersect, which is mathematically equivalent to the maximum of two SDF models [19]. However, applying maximum on two individual neural implicit surface result in shapes with discontinuity, as shown in Fig. 4. To get rid of the discontinuity of maximum function, we propose to use smooth maximum [19] that smoothly blends shapes while keeps the smoothness at the joint sections of the boundary, as shown in Algorithm 1.

C. Geometry-guided refinement

Although the aforementioned SDF fusion method yields a reasonable surface of the target object, the surface still contains holes and lacks thin structures due to information loss caused by data partitioning (e.g., the walking stick in the statues in the bottom right corner of Fig. 1). We employ a geometry-guided refinement network to enhance the surface smoothness and recover lost information. Additionally, we jointly optimize camera poses to improve pose accuracy.

A straightforward approach to this idea would involve optimizing all camera poses concurrently with the SDF network. However, despite using the coarse pose approximation as an initialization, we observe that directly optimizing hundreds of camera poses remains challenging and easily results in

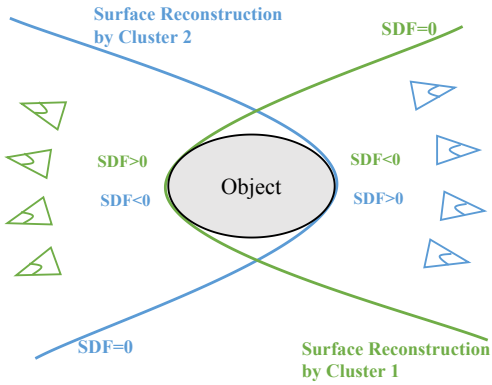
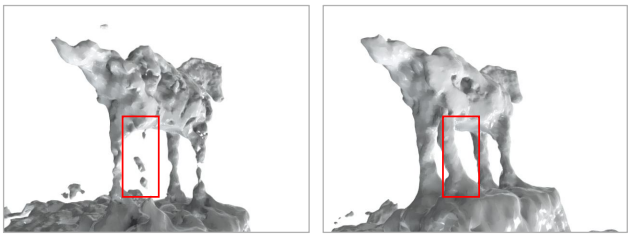


Fig. 3. Illustration on each cluster's visible surface. The ideal complete surface should be the boundary of area where the green and blue curves intersect.

Algorithm 1: Smooth maximum for SDF fusion

Input: Two SDF value s_1 and s_2, Smooth factor k
Output: Fused SDF value t

$r \leftarrow (s_1 + k, s_2 + k))$;
$r \leftarrow max(r, (0, 0))$;
$r_{norm} = ||r||$;
$t \leftarrow min(-r, max(a, b))$;
$t \leftarrow t + r_{norm}$;



Maximum Smooth Maximum

Fig. 4. Visualization of SDF fusion result on Horse scene from BlendedMVS dataset. Smooth maximum can effectively decrease visual artifacts.

failure. To enhance the stability of the optimization process, we propose refining the relative 3D similarity transformation $G \in Sim(3)$ between clusters, by utilizing the fact that estimated poses in each cluster is of high quality. We employ photometric loss between rendered images I^* and captured images I to optimize SDF network ϕ, and transformation G.

$$\mathcal{L}_{rgb} = \sum |I_i - I_i^*| \tag{3}$$

To further encourage the smoothness of the reconstructed surfaces, we impose a prior by regularizing the normalization of SDF network gradient, denoted as eikonal loss:

$$\mathcal{L}_{eikon} = \sum (||\nabla F(x_i)|| - 1)^2 \tag{4}$$

In the absence of explicit geometry regularization, the joint optimization of poses and surface can admit degenerated solution, which is referred as shape-radiance ambiguity [20]. To avoid such ambiguity, we leverage the 2D-2D keypoint correspondences as geometry guidance, as they offer explicit constraints derived from two-view geometry. Specifically, we project local point clouds into camera sets in other clusters and calculate the reprojection loss at the pixel level:

$$\mathcal{L}_{reproj} = \sum_{i=0}^{n} \sum_{j=0, j\neq i}^{n} (|\Pi(X_i, G_{ij}) - x_j|) \tag{5}$$

where X_i is the 3D points in cluster C_i, G_{ij} is the relative transformation from C_i to C_j, and x_j is the 2D key points in cluster C_j that have correspondences with X_i.

The overall refinement loss $\mathcal{L}$ can be formulated as:

$$\mathcal{L} = \mathcal{L}_{rgb} + \alpha \mathcal{L}_{eikon} + \beta \mathcal{L}_{reproj} \tag{6}$$

Progressive Training. Following Instant NGP [21], our network is parameterized by multi-resolution hash grids and a small MLP to achieve fast convergence, where features are linearly-interpolated from grids. However, due to the discontinuity of the linearly-interpolated feature, the fluctuation of gradient with respect to camera poses is frequent at high resolution [22], which is harmful to stable training. Inspired by BARF [15], we start refinement from only a set of low-frequency bands to reduce negative fluctuation in early stage, and progressively activate finer feature grids to capture more details, until the highest level L. We weight the k-th level feature vector by weight w_k, where w_k is defined as:

$$w_k(\alpha) = \begin{cases} 0 & \text{if } \alpha < k \\ 1 - (\alpha - k) & \text{if } 0 \leq \alpha - k < 1 \\ 1 & \text{if } 1 \leq \alpha - k \end{cases} \tag{7}$$

and $\alpha \in [0, L]$ is a parameter proportional to the refinement progress.

TABLE I
EFFICIENCY EVALUATION ON DTU AND BLENDEDMVS DATASET.

Scene	N_I	N_c	Training Time(hours)		Speedup
			Level-S^{2}fM [1]	ParaSurRe	
scan65	49	6	9.56	1.88	5.09
scan106	64	7	13.71	2.01	6.82
scan114	64	2	13.43	4.32	3.10
scan122	64	2	13.62	4.50	3.02
Fountain	91	3	18.40	4.31	4.26
Stone	20	2	2.85	1.75	1.62
Horse	81	9	15.34	2.36	6.5
Statues	59	7	14.18	3.50	4.05

IV. EXPERIMENTS

A. Experimental Setup

We conduct experiments on four scenes from DTU [11], each containing 49-65 images. We also use four representative and challenging scenes from BlendedMVS [12], including Statues, Horse, Fountain, and Stone, each containing 20-91 images. We consider three types of evaluation metrics. Firstly, we show training time on NVIDIA V100 GPUs and evaluate the efficiency of ParaSurRe against Level-S^{2}fM [1] on all test scenes. Secondly, since there are no ground truth meshes for our dataset, we provide qualitative assessments of extracted meshes produced by different methods for surface quality comparison. Thirdly, to fairly evaluate the accuracy of poses, we align test poses with ground truth camera trajectory by Procrustes analysis, and then follow the protocol of visual odometry to measure the difference. Specifically, we compute mean angle difference and ATE (Absolute Trajectory Error) to measure rotation error and translation error respectively.

B. Efficiency

Table I report the quantitative efficiency results for Level-S^{2}fM [1] and ParaSurRe, where N_I and N_c are the numbers of images and clusters respectively. Note that Level-S^{2}fM is not explicitly designed for parallel training, while our ParaSurRe is capable of training across multiple devices, and the expected speedup should be equal to the count of clusters N_c. We also investigate the scalability of our approach to different numbers of clusters. For this experiment, we find that our approach ParaSurRe can successfully handle up to 9 clusters on Horse dataset with total 81 images, which corresponds to about 9 images in each cluster. As the number of images in each cluster is below than 10, due to the extremely sparse point cloud, it becomes challenging to determine transformations between clusters through the PNP algorithm.

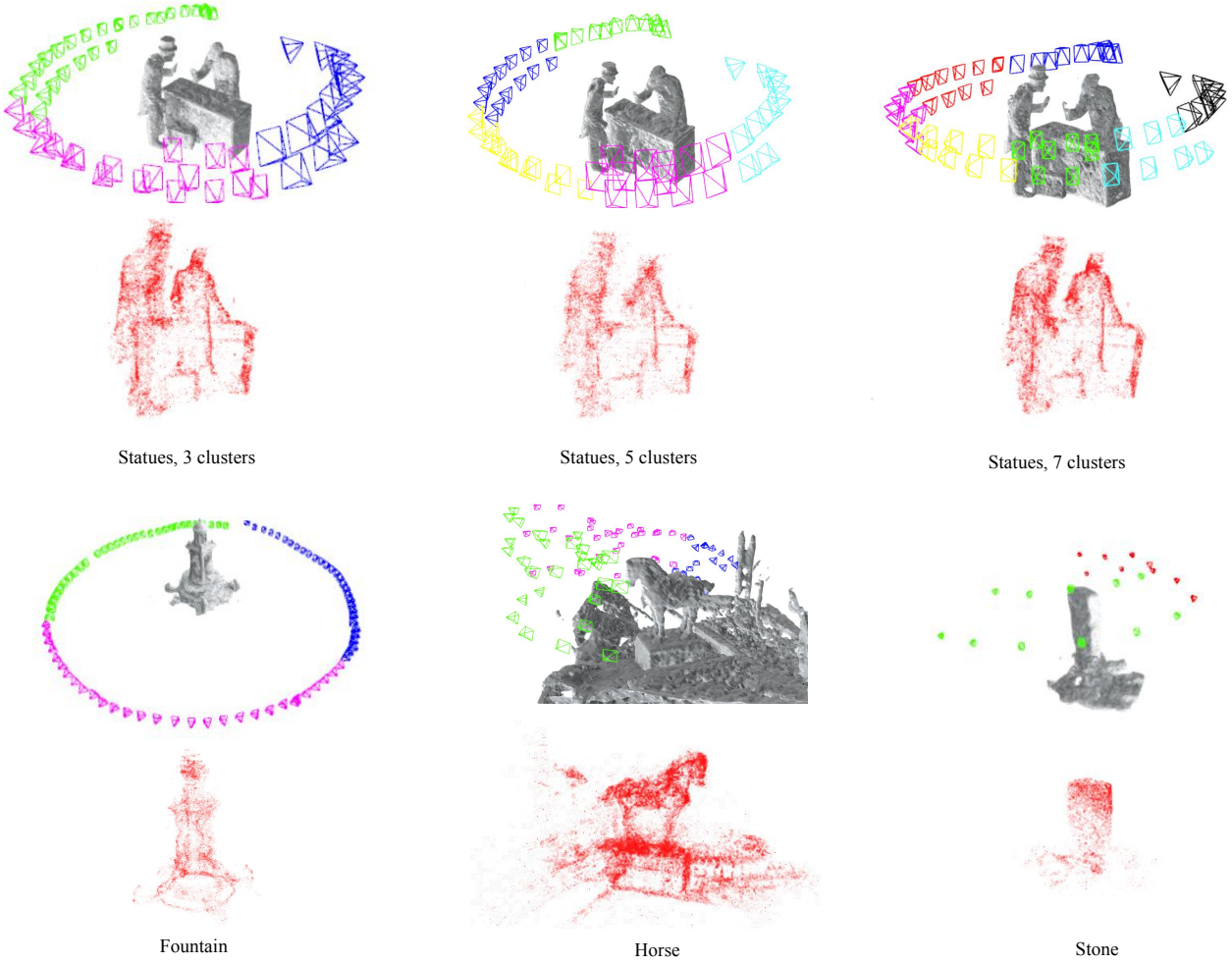


Fig. 5. Qualitative results for surface reconstruction and pose estimation of ParaSurRe. For each scene, we show the corresponding fused surface with cameras, and point clouds.

C. Surface Reconstruction

We use Marching Cube [23] to extract mesh models from zero level set of SDF field, then we compare ParaSurRe against Level-S^{2}fM [1] for surface reconstruction. In Fig. 6, we present visualizations of surface reconstructed by different methods. The extracted meshes of Level-S^{2}fM [1] are noisy and fail to reconstruct complex geometry accurately (especially on Statues and Fountain scenes), while our surfaces preserve more detail with high fidelity. We also present the surfaces extracted from SDF field without refinement in Fig. 6, from which we know that SDF refinement plays an important role in surface reconstruction. We report more qualitative results for ParaSurRe in Fig. 5, where point clouds, cameras, and meshes are visualized. Different-colored cameras are located in separate clusters.

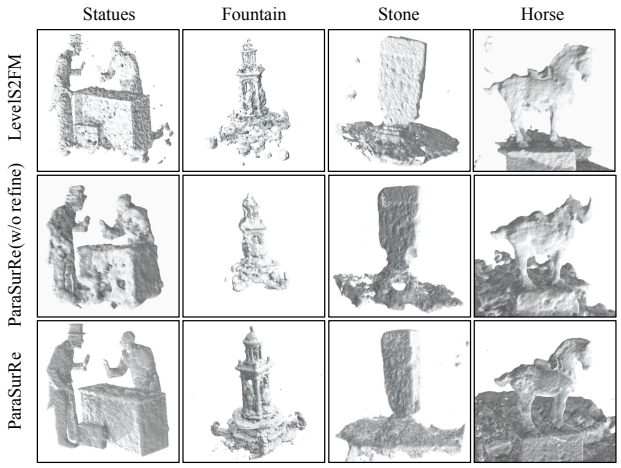


Fig. 6. Visualization of reconstructed meshes by Level-S^{2}fM and ParaSurRe on scenes from BlendedMVS and DTU.

D. Pose Estimation

We report pose estimation results of Colmap, Level-S^{2}fM [1] and ParaSurRe in Table II. On the DTU dataset, our ParaSurRe achieves similar pose estimation accuracy with Level-S^{2}fM [1]. Both ParaSurRe and Level-S^{2}fM obtain a relative improvement of **36%** in rotation error compared

TABLE II
POSE ESTIMATION RESULTS ON DTU [11] AND BLENDEDMVS [12]. THE BEST AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN PURPLE AND CYAN RESPECTIVELY.

Method	DTU		BlendedMVS	
	Rot.↓	Trans.↓	Rot.↓	Trans.↓
Colmap [13]	0.589	0.431	1.435	0.019
Level-S^{2}fM [1]	0.384	1.017	0.790	0.044
Ours w/o $\mathcal{L}_{reproj}$	0.624	2.234	1.721	0.088
Ours	0.376	1.112	1.398	0.043

with Colmap. On the BlendedMVS dataset, our ParaSurRe gets slightly inferior results compared with Level-S^{2}fM [1]. ParaSurRe and Level-S^{2}fM [1] are both better than Colmap in terms of rotation error, while are worse than Colmap in terms of translation error. We also analyse the effectiveness of the geometry gudiance $\mathcal{L}_{reproj}$ that have been added in our refinement process. We observe that the reprojection loss is the main contributor to improve pose accuracy. When removing reprojection loss, there is less constraint for scene geometry, and the camera pose quality drops due to shape-radiance ambiguity problem [20].

V. CONCLUSIONS

In this work, we present ParaSurRe, a method that solves two tasks in a unified and parallel manner: pose estimation and surface reconstruction, which are fundamental requirements for 3D understanding and reconstruction. We demonstrate that ParaSurRe can speed up training and achieve competitive performance on popular datasets. We hope our model design can be useful for future exploration of a unified vision model with fast training speed.

REFERENCES

[1] Yuxi Xiao, Nan Xue, Tianfu Wu, and Gui-Song Xia, "Level-s^{2}fm: Structure from motion on neural level set of implicit surfaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17205–17214.

[2] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.

[3] Michael Oechsle, Songyou Peng, and Andreas Geiger, "Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5589–5599.

[4] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[5] 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," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27171–27183, 2021.

[6] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu, "Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction," in *ICCV*, 2023, pp. 3295–3306.

[7] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman, "Volume rendering of neural implicit surfaces," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4805–4815, 2021.

[8] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger, "Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction," *Advances in neural information processing systems*, vol. 35, pp. 25018–25032, 2022.

[9] Yu Chen, Shuhan Shen, Yisong Chen, and Guoping Wang, "Graph-based parallel large scale structure from motion," *Pattern Recognition*, vol. 107, pp. 107537, 2020.

[10] Siyu Zhu, Tianwei Shen, Lei Zhou, Runze Zhang, Jinglu Wang, Tian Fang, and Long Quan, "Parallel structure from motion from local increment to global averaging," *arXiv preprint arXiv:1702.08601*, 2017.

[11] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs, "Large scale multi-view stereopsis evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 406–413.

[12] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan, "Blendedmvs: A large-scale dataset for generalized multi-view stereo networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1790–1799.

[13] Johannes L Schonberger and Jan-Michael Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[14] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu, "Nerf–: Neural radiance fields without known camera parameters," *arXiv preprint arXiv:2102.07064*, 2021.

[15] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey, "Barf: Bundle-adjusting neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5741–5751.

[16] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, and Jingyi He, "Gnerf: Gan-based neural radiance field without posed camera," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6351–6361.

[17] Wenjing Bian, Zirui Wang, Kejie Li, and Victor Adrian Prisacariu, "Nope-nerf: Optimising neural radiance field with no pose prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4160–4169.

[18] Jianbo Shi and Jitendra Malik, "Normalized cuts and image segmentation," in *Proceedings of IEEE computer society conference on computer vision and pattern recognition*. IEEE, 1997, pp. 731–737.

[19] Johann Korndörfer, Benjamin Keinert, Urs Ganse, Michael Sänger, Simon Ley, Konstanze Burkhardt, Mario Spuler, and Jörn Heusipp, "Hgsdf: A glsl library for building signed distance functions," https://mercury.sexy/hg_sdf, 2015.

[20] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun, "Nerf++: Analyzing and improving neural radiance fields," *arXiv preprint arXiv:2010.07492*, 2020.

[21] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *TOG*, vol. 41, no. 4, pp. 1–15, 2022.

[22] Hwan Heo, Taekyung Kim, Jiyoung Lee, Jaewon Lee, Soohyun Kim, Hyunwoo J Kim, and Jin-Hwa Kim, "Robust camera pose refinement for multi-resolution hash encoding," *arXiv preprint arXiv:2302.01571*, 2023.

[23] William E Lorensen and Harvey E Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353. 1998.