2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Neural Radiance Fields for Unbounded Lunar Surface Scene

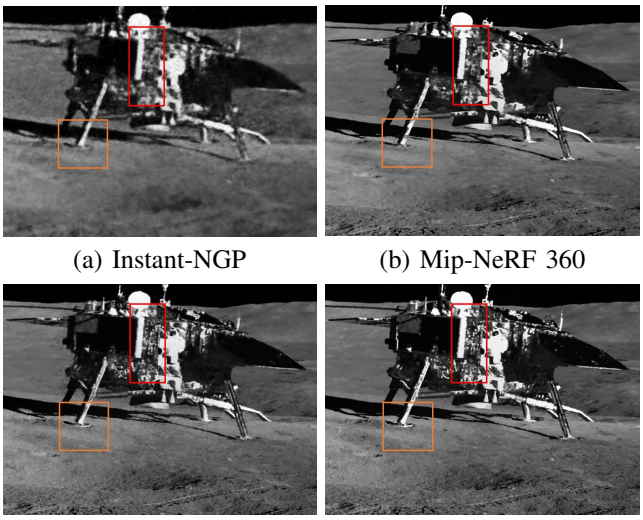
Xu Zhang, Linyan Cui, Jihao Yin

***Abstract*— Accurate understanding of lunar surface topography is vital for effective decision-making and remote control of lunar rovers during exploration missions. Conventional sensing methods often struggle to capture the intricate details of the lunar landscape. In response, we propose an innovative approach that leverages NeRF to synthesize new viewpoints within the expansive lunar environment. By blending 3D hash grids and 2D plane grids representations, our approach provides a comprehensive scene representation. We employ the technique of spiral sampling and feature rendering to enhance rendering quality while simultaneously reducing training time. Additionally, we leverage sparse point cloud to aid the model in better learning the geometric structure of the lunar environment. Through experimentation, we have demonstrated that our method is capable of synthesizing realistic images of lunar environments.**

I. INTRODUCTION

Lunar exploration through robotic rovers is a cornerstone of space research, enabling humanity to extend its reach beyond our planet. These missions are characterized by complex control strategies, where rover mobility hinges on comprehensive terrain understanding [1], visual localization [2], and decision-making. The conventional "move-wait" control pattern involves multiple steps: scientists, engineers, and operators collaborate to reconstruct the lunar terrain, devise optimal paths, and remotely guide the rover. However, this process often proves time-consuming, consequently diminishing task efficiency. One of the critical challenges faced by lunar rovers is their ability to accurately perceive and navigate through diverse and unfamiliar terrains. Effective decision-making and remote control of lunar rovers rely on a detailed understanding of the lunar surface. Traditional methods of lunar terrain perception have limitations in capturing complex 3D structures and handling extreme lighting conditions. As a result, there is a growing need for innovative solutions that can provide more accurate and informative representations of the lunar environment.

To overcome these challenges, the integration of virtual reality technology into lunar rover remote operations presents a transformative solution. By merging navigation camera imagery with advanced rendering techniques, virtual reality enhances the interaction between operators and rover. This approach provides operators with dynamic, versatile viewpoints of the lunar landscape, augmenting their perception and decision-making capabilities. Virtual reality-assisted remote operations pave the way for more informed and efficient decision-making, which is crucial for lunar rovers navigating complex lunar terrains. The key to effective virtual reality-assisted remote operations lies in constructing accurate and comprehensive virtual lunar environments. These environments necessitate dense 3D reconstruction or novel viewpoint synthesis based on existing images. Classical dense 3D reconstruction methods [3], [4], [5] rely on local image similarity, which often encounters issues in scenes characterized by repetitive textures and limited overlapping areas. In contrast, image-based novel viewpoint synthesis methods [6], [7], [8] generate high-quality images without explicit 3D model. Notably, NeRF [9] has garnered attention for its capabilities in this regard. NeRF's ability to model complex scenes and generate realistic images from sparse input data [10], [11], [12], aligns well with the requirements of lunar exploration missions.



(a) Instant-NGP (b) Mip-NeRF 360

(c) Zip-NeRF (d) ours

Fig. 1: Novel view synthesis comparison.

In this paper, we present an innovative approach that leverages NeRF to synthesize novel viewpoints within the expansive and unbounded lunar environment. Our aim is to enhance the perceptual capabilities of lunar rovers, allowing them to virtually observe the lunar surface from varying angles. Our approach combines 3D hierarchical feature grids and 2D planar feature grids to create a hybrid scene representation that captures the intricacies of lunar topography. Moreover, we introduce a multisampling and feature rendering technique that not only enhances the quality of synthesized images but also reduces the training time of the model. To enable accurate learning of the scene's geometric structure, we incorporate sparse point cloud supervision, refining the training process. Fig. 2 illustrates the overview of our approach.

*This work was supported by National Key R&D Program of China (2022ZD0117400) and The Key Laboratory of Spaceflight Dynamics Technology Foundation (ZBS-2023-004, 2022-JYAPAF-F1030).

All authors are with Image Processing Center, School of Astronautics, Beihang University, cuily@buaa.edu.cn

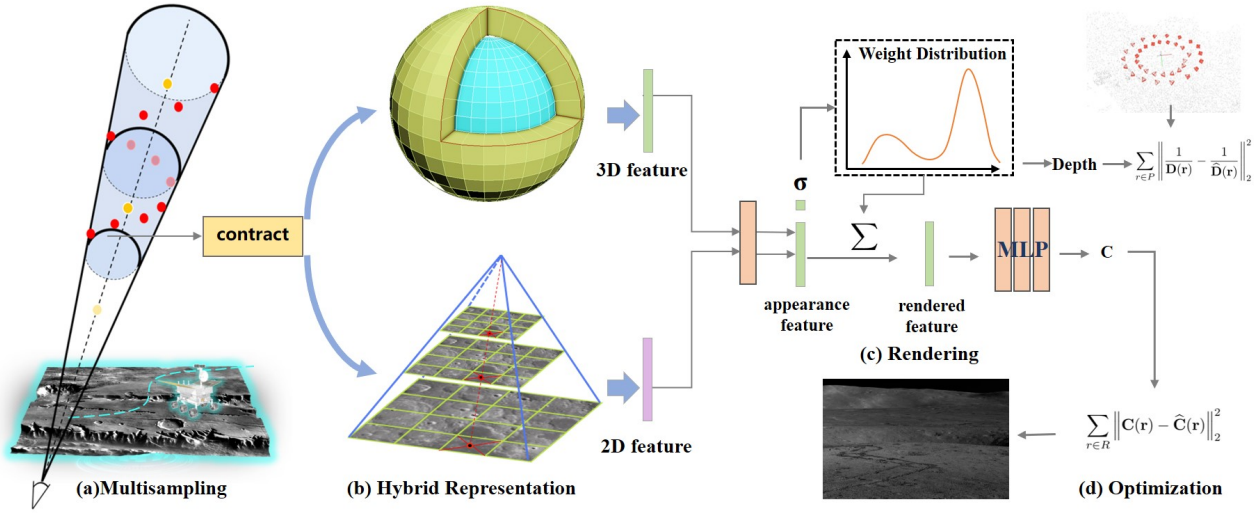


Fig. 2: Overview of our method. We utilize a spiral sampling approach to approximate the shape of a cone and employ hybrid feature grids to represent the scene. For each sampled point, we interpolate both 3D and 2D features, which are jointly fed into a shallow MLP to regress voxel density and appearance feature vector. Through feature rendering, we acquire pixel feature to regress color. Simultaneously, we employ the depth obtained from the projection of a sparse point cloud to supervise the rendered depth.

To summarize, our key contributions are:

- We propose a hybrid feature grid representation for unbounded lunar scene.
- We introduce a method involving spiral sampling and feature rendering, achieving anti-aliasing effects and simultaneously accelerating the rendering speed.
- Our approach achieves the best rendering results on both real and synthetic lunar datasets.

II. RELATED WORK

Neural Scene Representations. Neural scene representation has emerged as a transformative approach in computer vision. Pioneered by Neural Radiance Fields (NeRF), this methodology employs neural networks to model complex scenes, allowing for novel view synthesis and reconstruction from limited input data. NeRF and its follow-up works show impressive results on novel view synthesis [13], [14], [15], 3D reconstruction [16], [17], [18], pose estimation [19], [20], [21], training acceleration [22], [23] and dynamic scene representation [24], [25], [26].

Accelerating NeRF. NeRF [9] encodes the scene within the MLP in an implicit manner, resulting in efficient storage, yet the processes of reconstructing and rendering are notably time-consuming. A thread of studies demonstrate that the training of NeRF can be accelerated through the utilization of explicit representation. NSVF [27] utilizes the sparse voxel structure which enables the renderer to skip empty regions during ray tracing to achieve efficient rendering. DVGO [28] enhances the efficiency of radiance field reconstruction by optimizing voxel grids of neural features. The parameters of 3D dense grids grow by $O(N^3)$, resulting in significant memory wastage. Instant-NGP [29] effectively reduces the number of parameters that require optimization by employing hash tables. TensoRF [30] leverages tensor decomposition techniques to reconstruct radiance fields compactly as a joint representation of matrices and vectors. However, the rendering models utilized in the methods above treat pixels as individual points and disregard the relevant area around the sampling points. Our method addresses this issue by efficient multisampling, which enhances rendering quality and reduce computational time.

Unbounded Scene Reconstruction and Rendering. It's a longstanding research challenge in image processing and computer graphics. Early approaches primarily relies on structure-from-motion (SfM) [31], [32] and multi-view stereo (MVS) techniques [33], [34]. SfM algorithms, such as COLMAP [32], focus on recovering camera poses and sparse 3D points from a set of images, enabling the generation of initial scene representations. MVS methods extend these sparse representations to dense 3D reconstructions by leveraging photo-consistency and depth map fusion from multiple viewpoints. The domain of unbounded 3D reconstruction and rendering has witnessed substantial advancements with the emergence of novel techniques. NeRF++ [35] partitions unbounded scenes into foreground and background and models them separately using MLPs. Mip-NeRF 360 [36] uses inverse-sphere warping to map an infinitely large space to a bounded sphere. F^2-NeRF [37] introduces a novel space-warping technique called perspective warping, which is able to render high-quality images with arbitrary trajectories. Zip-NeRF [38] integrates iNGP's pyramid of grids into Mip-NeRF 360's framework, achieving improved rendering quality and faster training speed.

III. PRELIMINARIES

Given multi-view images with calibrated camera poses, NeRF [9] represents the 3D scene using the weights of a multilayer perceptron, that maps a spatial position $\mathbf{x} =$

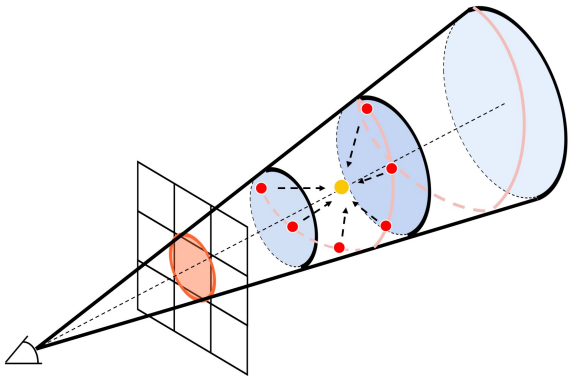


Fig. 3: Visualization of the multisampling. We employ spiral sampling to simulate the shape of a cone. By computing the weighted sum of the hybrid features of the sampled points, we obtain the feature of the conical frustum's center point.

(x, y, z) and viewing direction $\mathbf{d} = (d_x, d_y, d_z)$ to the corresponding density σ and view-dependent color $\mathbf{c}$:

$$(\sigma, \mathbf{f}) = \mathrm{MLP}_{\theta_1}(\mathbf{x}), \tag{1}$$

$$\mathbf{c} = \mathrm{MLP}_{\theta_2}(\mathbf{f}, \mathbf{d}) \tag{2}$$

where $\mathbf{f}$ is an intermediate embedding during the color learning process, and $\theta = [\theta_1, \theta_2]$ represent the parameters of the neural network.

For each pixel coordinate $\mathbf{p} \in \mathbb{R}^2$, NeRF samples N points $\mathbf{x}_i$ along ray $\mathbf{r}$ from given near and far bounds $\mathbf{t}_n$ and $\mathbf{t}_f$. It then calculates corresponding density σ and color $\mathbf{c}$. Via volume rendering, we can render the color at pixel $\mathbf{p}$:

$$\widehat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{N} w_i \mathbf{c_i} \tag{3}$$

$$w_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)(1 - \exp(-\sigma_i \delta_i)) \tag{4}$$

where w_i is the weight of each sampling point; δ_i is the distance between adjacent samples. To cope with unbounded scene, we employ the parameterization approach used in Mip-NeRF360 [36], mapping the entire space into a bounded region within the range of $[-2, 2]$:

$$\mathrm{contract}(\mathbf{x}) = \begin{cases} \mathbf{x}/r & \|\mathbf{x}\| \leqslant r \\ (2 - \frac{r}{\|\mathbf{x}\|})(\frac{\mathbf{x}}{\|\mathbf{x}\|}) & \|\mathbf{x}\| > r \end{cases} \tag{5}$$

where r represents the boundary of the region which is contracted into the range of $[-1, 1]$.

IV. METHOD

A. Hybrid Feature Grid Representation

For the sphere with a radius of r obtained from (5), we construct multi-resolution spherical hash feature grids $\mathbf{G}_{hash}$ to represent lunar surface scene in polar coordinates. Compared to Zip-NeRF [38] that directly uses square hash grids to cover the spherical region, which means some grid parameters located outside the contracted region are never utilized, spherical hash grids fully utilize parameters of the feature grids to represent the scene. For any 3D point $\mathbf{x} = [x, y, z]$ and level l, we perform feature encoding through trilinear interpolation:

$$\mathbf{f}_{3D}^{l} = \mathrm{interp3D}(\mathbf{G}_{hash}^{l}, \mathrm{contract}(\mathbf{x})) \tag{6}$$

Although 3D hash grids efficiently represent scenes, they are significantly affected by hash collisions at high resolutions, especially when dealing with images captured in an outward 360° view that contain large scale area. Considering that lunar surface environments can be approximated as a plane, we adopt multi-resolution axis-aligned plane grids $\mathbf{G}_{plane}$ to alleviate the impact of hash collision. Firstly, we project the point onto the xy plane to obtain the coordinates $\mathbf{x}_{xy}$ and apply the parameterization. Then we perform plane feature encoding through bilinear interpolation at each level:

$$\mathbf{f}_{2D}^{l} = \mathrm{interp2D}(\mathbf{G}_{plane}^{l}, \mathrm{contract}(\mathbf{x}_{xy})) \tag{7}$$

The two types of features from all levels are concatenated as the hybrid feature of point.

The parameters of plane grids grow quadratically with the highest resolution. Compared to simply increasing the length of the hash table, using additional plane grids is more memory-efficient.

B. Multisampling and Feature Rendering

We cast a cone with a radius of $\dot{r}t$ from the center of each pixel in line with the strategy in Mip-NeRF [13], where t is metric distance along the symmetry axis of cone. To account for the space occupied by the cone, the feature $\mathbf{f}$ of each conical frustum should be the integral of all points' features within it:

$$\mathbf{f} = \int W(\mathbf{p}) Q(\mathbf{p}) \mathrm{d}\mathbf{p} \tag{8}$$

where $\mathbf{p}$ is a point inside the conical frustum, Q denotes a function that queries the hybrid feature of $\mathbf{p}$, W is a function for weights based on the coordinates. However, due to the computational complexity, this equation is almost impractical to implement efficiently. Inspired by Zip-NeRF [38], to discretely approximate the conical volume, we sample M points along a spiral with k loops on the lateral side of the conical frustumas shown in Fig. 3:

$$\mathbf{p}_{i,j} = \left\{ \begin{bmatrix} \dot{r} l_j \cos(2\pi k j/M) \\ \dot{r} l_j \sin(2\pi k j/M) \\ l_j \end{bmatrix} | j = 0, 1, ...(M-1) \right\} \tag{9}$$

where $l_j = t_i + (t_{i+1} - t_i)(j + 1/2)/M$. We use normalized inverse distance between the sample points and the conical frustum center $\mathbf{x}_i$ to aggregate the local features $\mathbf{f}_{i,j}$ of these M sampling points, aiming to obtain a single feature $\mathbf{f}_i$ that describes the region inside the conical frustum:

$$\mathbf{f}_i = \sum_{j=1}^{M} \frac{\alpha_j}{\sum \alpha_j} \mathbf{f}_{i,j}, \alpha_j = \frac{1}{\|\mathbf{p}_{i,j} - \mathbf{x}_i\|} \tag{10}$$

Instead of directly employing the MLP to regress color and volumetric density for every sample point as NeRF and its

followers do, we feed the hybrid feature into a shallow MLP to obtain the volumetric density σ and appearance feature $\mathbf{f}_{app}$:

$$(\sigma_i, \mathbf{f}_{app\_i}) = \mathrm{MLP}_{shallow}(\mathbf{f}_i) \tag{11}$$

then calculate the weights for each conical frustum utilizing the obtained σ according to (4). Subsequently, we render the appearance features into an integrated feature $\mathbf{f}_{pixel}$ for the corresponding pixel, as shown in Fig. 4:

$$\mathbf{f}_{pixel} = \sum_{i=1}^{N} w_i \mathbf{f}_{app\_i} \tag{12}$$

A larger MLP is applied to decode colors from the rendered features:

$$\widehat{\mathbf{C}}(\mathbf{r}) = \mathrm{MLP}_{larger}(\mathbf{f}_{pixel}, \mathbf{d}) + w_{bg}\mathbf{c}_{bg}(\mathbf{d}) \tag{13}$$

where $\mathbf{c}_{bg}$ is a learnable variable related to the viewing direction, representing the space background color. It can be implemented by spherical harmonics functions or a small MLP:

$$\mathbf{c}_{bg} = f_{space}(\mathbf{d}) : \mathbb{S}^2 \rightarrow \mathbb{R}^3 \tag{14}$$

The weight of the background is calculated based on the sum of weights along the rays:

$$w_{bg} = 1 - \sum_{i=1}^{N} w_i \tag{15}$$

Using this approach, rendering a pixel color requires only one MLP evaluation, avoiding color regression for each sampling point. Despite sampling more points compared to other methods, our technique significantly accelerates the rendering process. The appearance loss L_{color} for ray $\mathbf{r}$ with true color $\mathbf{C}(\mathbf{r})$ is:

$$L_{color} = \sum_{r \in R} \left\| \mathbf{C}(\mathbf{r}) - \widehat{\mathbf{C}}(\mathbf{r}) \right\|_2^2 \tag{16}$$

C. Sparse Depth Supervision

The method of rendering feature is indeed reasonable and can effectively improve rendering speed in most cases. However, when majority of training images contain unbounded regions, this approach may fail to learn the geometric representation of the moon scene, as it does not find the surface distribution. In order to identify the correct optimization direction, similar to [12], we utilize the sparse point cloud obtained during the camera pose estimation using COLMAP [32]. By projecting this sparse point cloud into various viewpoints, we acquire sparse depth for supervision.

Similar to (12), the depth estimate is computed from the rendering weights and the upper bound distance $\inf$:

$$\widehat{\mathbf{D}}(\mathbf{r}) = \sum_{i=1}^{N} w_i t_i + w_{bg} \cdot \inf \tag{17}$$

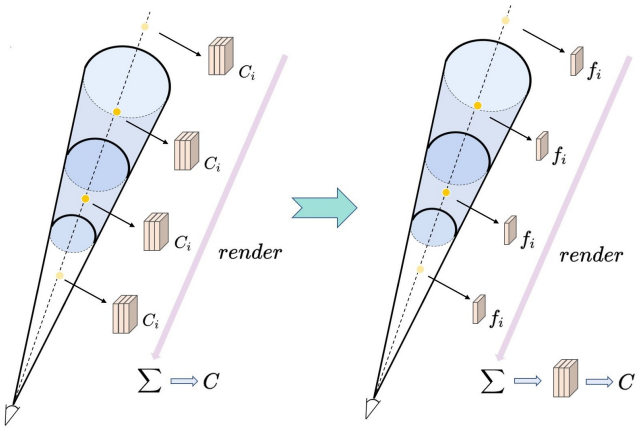


Fig. 4: In Mip-NeRF360 [36] and Zip-NeRF [38], color regression is performed for each conical frustum center point, followed by volume rendering to calculate pixel colors. In our model, the feature vector of the conical frustum center point is initially processed through a shallow MLP to derive volume density and appearance feature vector. These features along the ray are rendered and ultimately fed into an MLP to calculate pixel color.

The geometric loss is a L2 loss between the inverse projected and predicted depths, which results in a slightly better quality than MSE:

$$L_{geo} = \sum_{r \in P} \left\| \frac{1}{\mathbf{D}(\mathbf{r})} - \frac{1}{\widehat{\mathbf{D}}(\mathbf{r})} \right\|_2^2 \tag{18}$$

where P are the rays generated by projected pixels from the sparse point cloud. This step enables us to use geometric prior information to guide the optimization process, thereby enhancing the accuracy and stability of the algorithm.

D. Optimization

Considering that the images collected by the lunar rover contain the space background, L_{bg} regularizes the space background probability w_{N+1} to focus on either the foreground or the background:

$$L_{bg} = -w_{bg}\log(w_{bg}) - (1 - w_{bg})\log(1 - w_{bg}) \tag{19}$$

To optimize the neural scene representation, we minimize the following loss:

$$L = \lambda_{color} L_{color} + \lambda_{geo} L_{geo} + \lambda_{bg} L_{bg} \tag{20}$$

where $\lambda_{color}, \lambda_{geo}, \lambda_{bg}$ are weighting parameters.

V. EXPERIMENTS

A. Dataset

In the experiment, we assess the performance of our model on both synthetic and real lunar surface datasets.

Synthetic Dataset. We generate a series of lunar image sequences using UE4, employing the same capture approach as the real data, with the camera tilted downward, capturing an image every 20 degrees along the circumference. To emulate the real lunar surface environment as closely as

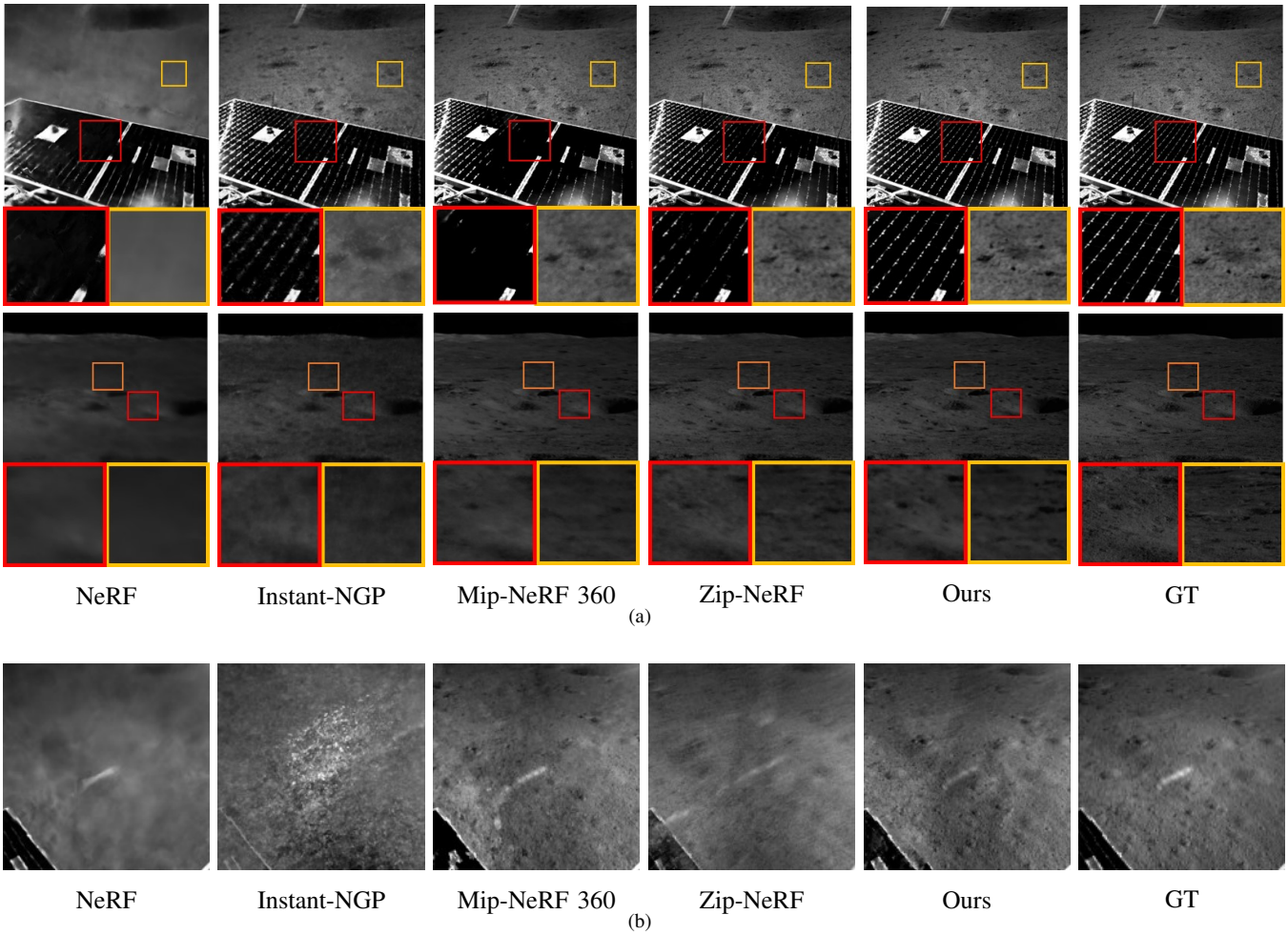


Fig. 5: Novel view synthesis comparison. The poses of the test viewpoints are derived from the left eye of binocular camera located at specific positions, as shown in (a). To compare the performance of various models under sparse viewpoints, we further exclude the right-eye images corresponding to the test viewpoints from the training dataset. The rendering results are shown in (b).

feasible, the synthetic images encompass not only regular scenes but also incorporate elements of strong reflections, shadows, and other intricate phenomena.

Real Dataset. The real lunar images were captured by the "Yutu-2" lunar rover in 2019. During the lunar rover's exploration, the onboard binocular camera captures an image sequence of a scene by following a circular path. The extrinsic parameters for both the real dataset and the synthetic dataset are estimated using COLMAP [32].

B. Baselines and Metrics

Our approach employs a hybrid representation, utilizing both hash grids and dense planar grids. We compare our model with various forms of NeRF methods, including MLP based method Mip-NeRF360 [36], hash grids based methods Instant-NGP [29] and Zip-NeRF [38]. We evaluate all models using PSNR, SSIM, and LPIPS$_{VGG}$ metrics.

C. Implementation Details

We construct 9 levels of spherical hash grids with resolutions from 2^4 to 2^{12}. The hash table length is 2^{21} and the stored feature vector dimension is 4. We design plane feature grids with four resolutions from 2^8 to 2^{11}. Each plane grid has a feature dimension of 4. The shallow MLP for σ has only 1 layer with 64 hidden units and ReLU activation. The larger MLP for color has 3 layers with 256 hidden units. In multisampling, we set the point number $M = 8$ and the loops $k = 2$. In the loss function, we set λ_{color}=1.0, λ_{geo}=0.1, λ_{bg}=0.005. Our model is implemented with Pytorch and is trained with a batch size of 2^{14} rays. The runtime for all experiments is measured on an NVIDIA Titan GPU.

D. Results on Real Datasets

We first report the results on real datasets in Fig. 5(a), where these perspectives originate from the left eye of a binocular camera, with the corresponding right eye participating in the training process. Our model achieves the best rendering quality among all the NeRFs. The images synthesized by the original NeRF exhibit significant blurriness, owing to the fact that the NDC coordinate system is effective for forward-facing unbounded scenes but is inadequate for addressing outward-looking unbounded scenes. The results

TABLE I: Performance on synthetic and real lunar surface datasets. Red and orange highlights indicate the 1st and 2nd best performing technique. Our method could achieve high rendering quality while keeping relatively low time consumption.

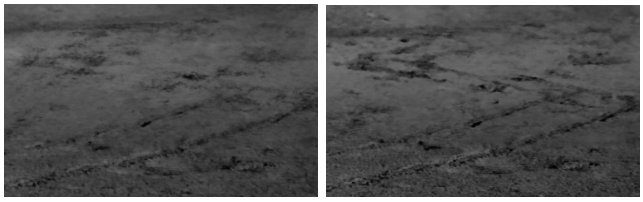
Dataset:	Syntheic I			Syntheic II			Real I			Real II			
Metric:	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	Avg. Time
NeRF	24.41	0.63	0.59	23.98	0.63	0.60	22.14	0.35	0.72	18.60	0.38	0.71	8 h
Instant-NGP	25.76	0.70	0.49	26.72	0.72	0.49	23.65	0.46	0.60	20.58	0.44	0.64	12 min
Mip-NeRF360	29.27	0.75	0.31	29.52	0.79	0.32	24.59	0.60	0.48	22.04	0.59	0.51	18 h
Zip-NeRF	29.58	0.77	0.26	29.17	0.80	0.28	26.54	0.71	0.35	23.03	0.65	0.44	1 h
Ours(w.o. plane grids)	29.71	0.82	0.31	30.26	0.82	0.30	26.96	0.74	0.35	23.58	0.69	0.42	45 min
Ours(w.o. f. rendering)	28.82	0.79	0.32	28.52	0.79	0.33	26.56	0.72	0.35	23.51	0.68	0.41	1.1 h
Ours(w.o. sparse depth)	30.77	0.82	0.27	31.10	0.82	0.29	27.03	0.72	0.36	23.56	0.67	0.44	50 min
Ours(all)	31.12	0.85	0.24	32.44	0.86	0.22	27.02	0.74	0.33	23.66	0.70	0.41	48 min

from Instant-NGP [29] are slightly sharper but distant elements still lack the desired clarity due to the absence of scene parameterization. The rendering outcomes of Mip-NeRF 360 [36] and Zip-NeRF [38] closely resemble real images; however, certain details are not well pronounced, and Mip-NeRF 360 suffers from prolonged training time. Thanks to the multisampling and feature rendering, our model captures the scene's details while requiring comparatively less training time. The comparison of the results in Fig. 1 is also the same. Detailed metrics for each model are listed in Table I.

When both the left and right eye images captured at a specific location are not included in the training data, we employ the pose of the left eye image for rendering, as shown in Fig. 5(b). Due to the limited overlap of adjacent training viewpoints in this region, Zip-NeRF and Instant NGP generate blurry results, while only our model reasonably reproduces the appearance of the scene. We leverage the sparse point cloud generated during the process of estimating extrinsics for training the radiance field. This enables our method to perform well even under sparse viewpoints. Removing the supervision of the sparse point cloud leads to a decrease in rendering quality, as shown in Fig. 6.

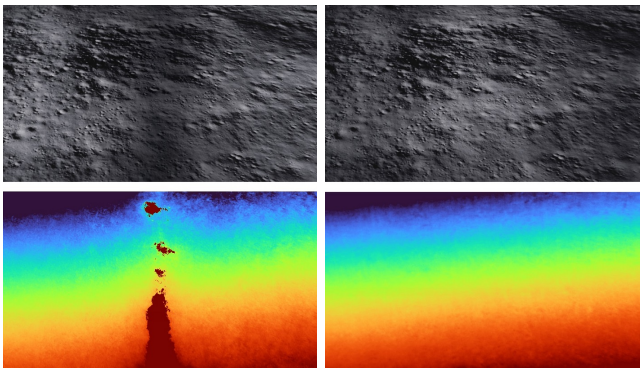
E. Results on Synthetic Datasets

The results in Table I demonstrates that our model continues to exhibit superior performance on the synthetic datasets. Similar to the performance on the real datasets, our approach more realistically reproduces the lunar simulated environment compared to other models. The ablation experiments reveal that the feature rendering technique effectively enhances the performance of our method on the synthetic datasets. This leads to a reduction of approximately 25% in time consumption, while concurrently improving rendering quality.



w.o. sparse depth ours

Fig. 6: Qualitative results of ablation study.



Zip-NeRF ours

Fig. 7: RGB images and depth maps rendered by Zip-NeRF [38] and our method.

Given UE4's scene editing capabilities, we conduct further experiments by modifying lighting intensity, angles, exposure, and more without altering the image capture approach. We observe that our model exhibits stronger robustness under complex lighting conditions, as shown in Fig. 7. After training with sparse viewpoint images, Zip-NeRF generates erroneous lighting and shadow effects when tested from different angles. By further scrutinizing the rendered depth maps, we find that the incorrect rendering results of Zip-NeRF stem from its failure to learn the scene's continuous geometric structure. Lighting and shadow effects mislead its optimization process. Conversely, our method accurately captures the variation of lighting with observation direction.

VI. CONCLUSIONS

This paper propose a hybrid representation of 3D spherical feature grids and 2D planar feature grids for the lunar environment. The introduction of the multisampling and feature rendering technique enhances the quality of synthesized images, while the incorporation of sparse point cloud supervision refines the radiance field training process, ensuring more accurate scene geometry capture. Through comprehensive experiments, we have demonstrated the effectiveness of our approach in generating novel viewpoints and enhancing the perception capabilities of lunar rovers. The results show the potential of NeRF-based synthesis in assisting lunar exploration missions, ultimately contributing to more efficient and successful space exploration endeavors.

REFERENCES

[1] R. E. Karlsen and G. Witus, "Terrain understanding for robot navigation," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 895–900.

[2] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[3] V. Vineet, O. Miksik, M. Lidegaard, M. Nießner, S. Golodetz, V. A. Prisacariu, O. Kähler, D. W. Murray, S. Izadi, P. Pérez *et al.*, "Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 75–82.

[4] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph." Robotics: Science and Systems, 2015.

[5] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 1, 2017.

[6] J. Flynn, M. Broxton, P. Debevec, M. DuVall, G. Fyffe, R. Overbeck, N. Snavely, and R. Tucker, "Deepview: View synthesis with learned gradient descent," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2367–2376.

[7] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–14, 2019.

[8] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik, "Multi-view supervision for single-view reconstruction via differentiable ray consistency," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2626–2634.

[9] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[10] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. Sajjadi, A. Geiger, and N. Radwan, "Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5480–5490.

[11] P. Truong, M.-J. Rakotosaona, F. Manhardt, and F. Tombari, "Sparf: Neural radiance fields from sparse and noisy poses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4190–4200.

[12] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 882–12 891.

[13] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.

[14] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7210–7219.

[15] S. Lewis, J. Pavlasek, and O. C. Jenkins, "Narf22: Neural articulated radiance fields for configuration-aware rendering," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 770–777.

[16] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural rgb-d surface reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6290–6301.

[17] X. Zhong, Y. Pan, J. Behley, and C. Stachniss, "Shine-mapping: Large-scale 3d mapping using sparse hierarchical implicit neural representations," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 8371–8377.

[18] J. Wang, T. Bleja, and L. Agapito, "Go-surf: Neural feature grid optimization for fast, high-fidelity rgb-d surface reconstruction," in *2022 International Conference on 3D Vision (3DV)*. IEEE, 2022, pp. 433–442.

[19] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "inerf: Inverting neural radiance fields for pose estimation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1323–1330.

[20] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5741–5751.

[21] Y. Lin, T. Müller, J. Tremblay, B. Wen, S. Tyree, A. Evans, P. A. Vela, and S. Birchfield, "Parallel inversion of neural radiance fields for robust pose estimation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9377–9384.

[22] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "Plenoctrees for real-time rendering of neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5752–5761.

[23] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5501–5510.

[24] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 318–10 327.

[25] C. Gao, A. Saraf, J. Kopf, and J.-B. Huang, "Dynamic view synthesis from dynamic monocular video," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5712–5721.

[26] T. Zhang, Y.-F. Lau, and Q. Chen, "A portable multiscopic camera for novel view and time synthesis in dynamic scenes," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2409–2416.

[27] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 651–15 663, 2020.

[28] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5459–5469.

[29] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[30] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European Conference on Computer Vision*. Springer, 2022, pp. 333–350.

[31] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri, "Global motion estimation from point matches," in *2012 Second international conference on 3D imaging, modeling, processing, visualization & transmission*. IEEE, 2012, pp. 81–88.

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

[33] S. Galliani, K. Lasinger, and K. Schindler, "Massively parallel multiview stereopsis by surface normal diffusion," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 873–881.

[34] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer, 2016, pp. 501–518.

[35] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "Nerf++: Analyzing and improving neural radiance fields," *arXiv preprint arXiv:2010.07492*, 2020.

[36] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5470–5479.

[37] P. Wang, Y. Liu, Z. Chen, L. Liu, Z. Liu, T. Komura, C. Theobalt, and W. Wang, "F2-nerf: Fast neural radiance field training with free camera trajectories," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4150–4159.

[38] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Zip-nerf: Anti-aliased grid-based neural radiance fields," *arXiv preprint arXiv:2304.06706*, 2023.