Expanding the Unknown: 3D Gaussian Splatting with Diffusion Models for Unseen Region Reconstruction

Anonymous
Dept. xxx
Sogang University
Seoul, Republic of Korea

***Abstract*—Recent advancements in learnable spatial representation structures, such as Neural Radiance Fields (NeRF) and 3D Gaussian Splatting, have improved 3D scene reconstruction from 2D images, enhancing computational efficiency, scalability, and memory usage. However, in multi-view environments, reconstruction performance degrades in regions with limited field of view (FOV) overlap, especially in Forward-Facing Scene datasets. To address this, we assume extended camera poses and use Depth Image-Based Rendering (DIBR) for data augmentation, generating new views beyond the original FOV. Additionally, we employ diffusion models to generate new viewpoints in data-scarce areas and fine-tune them with Low-Rank Adaptation (LoRA) to maintain spatial consistency with existing views. Our approach significantly improves reconstruction quality in outer regions by combining extended camera poses, DIBR, and diffusion models. It works effectively in both single-image and multi-view setups, enhancing 3D reconstruction from sparse camera coverage and limited training data.**

***Index Terms*—novel view synthesis, neural rendering**

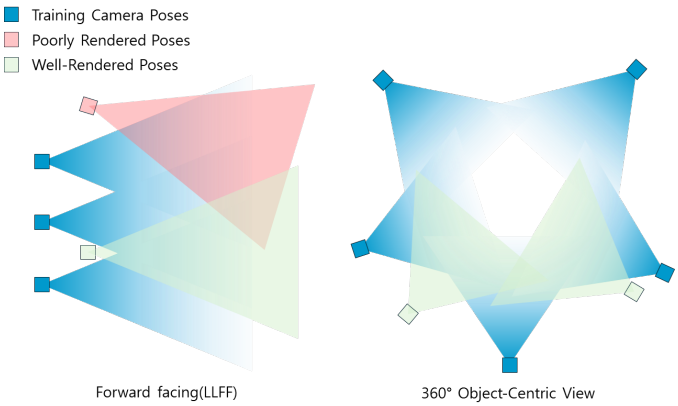


Fig. 1. In LLFF, Poorly Rendered Poses occur when the field of view is limited, leading to degraded reconstruction quality in areas with insufficient camera coverage.

I. Introduction

Research utilizing learnable spatial representation structures such as Neural Radiance Field (NeRF) and 3D Gaussian Splatting to reconstruct 3D scenes from 2D images has been actively pursued. These studies propose various approaches with the goals of more efficient computation, higher scalability, and lower memory usage. Most research is conducted in a multi-view environment, where data is captured from multiple cameras at different angles of the same scene. Multi-view environments are essential for high-quality reconstruction because they help maintain spatial consistency within the scene, reducing noise and enabling accurate 3D reconstruction. Therefore, multi-view datasets form an important foundation for evaluating the performance of these techniques.

However, even in multi-view environments, consistently high reconstruction quality cannot be guaranteed across all areas. High-quality reconstruction is achieved in regions with a large overlap of fields of view (FOV), but in outer regions with less overlap, reconstruction performance deteriorates. This issue is especially prominent in areas captured by only a few cameras, which becomes a major factor in reducing overall reconstruction accuracy. This problem arises because many existing 3D Gaussian Splatting studies utilize datasets constructed with forward-facing camera arrays (Forward-Facing Scenes). For instance, in environments like the LLFF dataset, multiple cameras are typically arranged in grids, such as 7×3 or 8×2, to create a multi-view environment.

These Forward-Facing Scene datasets are often substituted by synthetic datasets like 360° Object-Centric Views, as capturing real scenes in the required configuration is challenging. A 360° Object-Centric View, which captures images from various angles centered on an object, has the advantage of evenly distributing FOV overlaps. However, 360° capturing is often difficult in real environments, and most real-scene datasets still rely on the Forward-Facing Scene format. As a result, in Forward-Facing Scenes, reconstruction performs well only in areas with FOV overlap, while performance in outer regions suffers.

To address this issue, we assume extended camera poses to relax the constraints of the existing camera array and improve reconstruction quality in outer regions. Based on this, we employ Depth Image-Based Rendering (DIBR) to perform data augmentation by generating new views beyond the original FOV range. DIBR utilizes depth information to

generate images from new viewpoints that were not originally captured, effectively increasing the visual diversity in multi-view environments without additional data collection. This approach complements the limitations of previous research and significantly improves the quality of reconstruction in outer regions, contributing to more uniform spatial consistency.

In addition, we use diffusion models to enhance scene reconstruction quality and overcome the limitations of the training views. By using generative models to extend areas with insufficient data, our objective is to improve the overall quality of the scene. To achieve this, it is necessary to maintain the reliability of the existing training views while generating additional areas through the diffusion model. When applying the diffusion model, we must ensure that the new viewpoints generated by the model maintain consistency with the existing scene. Therefore, we fine-tune the model using Low-Rank Adaptation (LoRA) [1] to generate plausible spaces. This allows the diffusion model to generate realistic new spaces while preserving spatial consistency with the original views. This approach plays a crucial role in improving the accuracy of scene reconstruction and ensuring visual diversity from various viewpoints.

In this paper, we propose several novel approaches to address the limitations of traditional 3D scene reconstruction methods:

- Extended Camera Pose Assumption: We alleviate the constraints imposed by traditional camera arrays, improving reconstruction quality in outer regions by using Depth Image-Based Rendering (DIBR) for data augmentation.
- Use of Diffusion Models: We leverage diffusion models to extend data-scarce areas, overcoming the limitations of the training views and enhancing the overall scene quality by generating new perspectives while maintaining spatial consistency. Additionally, this approach can be applied to expand both single images and multi-view setups, generating new viewpoints to improve reconstruction from limited data or sparse camera coverage.
- Improved Spatial Consistency: Through the integration of these techniques, we significantly enhance the spatial consistency of the reconstructed scene, ensuring a more reliable and diverse reconstruction from various viewpoints.

II. Related Works

Monocular Depth Estimation-based 3D Reconstruction Early 3D reconstruction methods, such as Pixel2Mesh [2], represented objects as triangular meshes, while PIFu [3], [4] utilized memory-efficient implicit functions to recover fine details, including occluded regions. However, these approaches heavily relied on object-specific priors and 3D supervision, which limited their generalizability to diverse or complex scenes. Other methods, such as Make3D [5], segmented scenes into planar regions and combined their orientations and positions to construct a coherent 3D structure. Earlier approaches often depended on hand-crafted priors, such as shading [6] or edge contours [7], to infer 3D geometry. [8] also addresses the challenge of estimating accurate 3D scene shapes from single monocular images by proposing a two-stage framework that predicts depth up to an unknown scale and shift, then refines it using 3D point cloud data.

Recent advancements leverage Monocular Depth Estimation (MDE) to generate depth maps for scene reconstruction. MDE, which estimates the distance of every pixel in a 2D image, has seen significant progress with diffusion-based models like Marigold [9] and foundation-model-based approaches such as MiDaS [10] and Depth Anything [11], which produce finely detailed depth maps. These models adopt affine-invariant depth learning strategies that enhance geometric consistency in depth estimation. However, they are limited to relative depth predictions and lack the metric depth information required for complete 3D reconstruction. Addressing this limitation requires estimating the missing depth shift and focal length. ZoeDepth [12] is able to accomplish this and generate metric depth, but has to fine-tune on a similar dataset for it.

Also, there have been several attempts to leverage different type of domain data as a multi-modal in monocular depth estimation task. VPD [13] proposed a multi-modal architecture which took an image and a text prompt as inputs. It extracted feature maps from both data respectively and trained the correlation between them. The concatenated feature is then delivered to a decoder which outputs the estimated depth map. EVP [14] carried on the methods of [13] by enhancing the denoising UNet. It proposed multi-attention methods to refine multi-scale feature maps which were the output of the denoising UNet, so the correlation map could be delivered efficiently to a decoder. These methods had the similarity of taking both image and text as inputs and generate the correlational feature maps, which were then processed by the decoder to produce the depth map. However, they require additional text prompt data and performance of image-text alignment fails to meet expectations.

LeReS [15] incorporates a 3D point cloud encoder to predict these parameters, enabling the recovery of realistic 3D scene shapes. For this reason, we adopt LeReS as the monocular depth estimation model used in this study.

Sparse-view 3D Reconstruction Enhancement Learning-based spatial representation methods, such as Neural Radiance Fields (NeRF) [16] and 3D Gaussian Splatting (3DGS) [17], have emerged as essential tools for reconstructing 3D scenes from 2D images. NeRF employs neural networks to model the radiance field, enabling high-quality scene rendering [18]–[20], while 3DGS offers a memory-efficient approach by representing scenes as Gaussian splats. However, both methods are highly dependent on multi-view settings with overlapping fields of view (FOV) to maintain spatial consistency.

While most research focuses on reconstructing scenes using a sufficient number of views, real-world capturing scenarios often involve sparse views, with a focus on key regions. This has sparked research focused on overcoming the challenges of sparse-view environments. Such studies aim to develop methods that can achieve high-quality 3D reconstruction even with limited views, with a particular emphasis on techniques

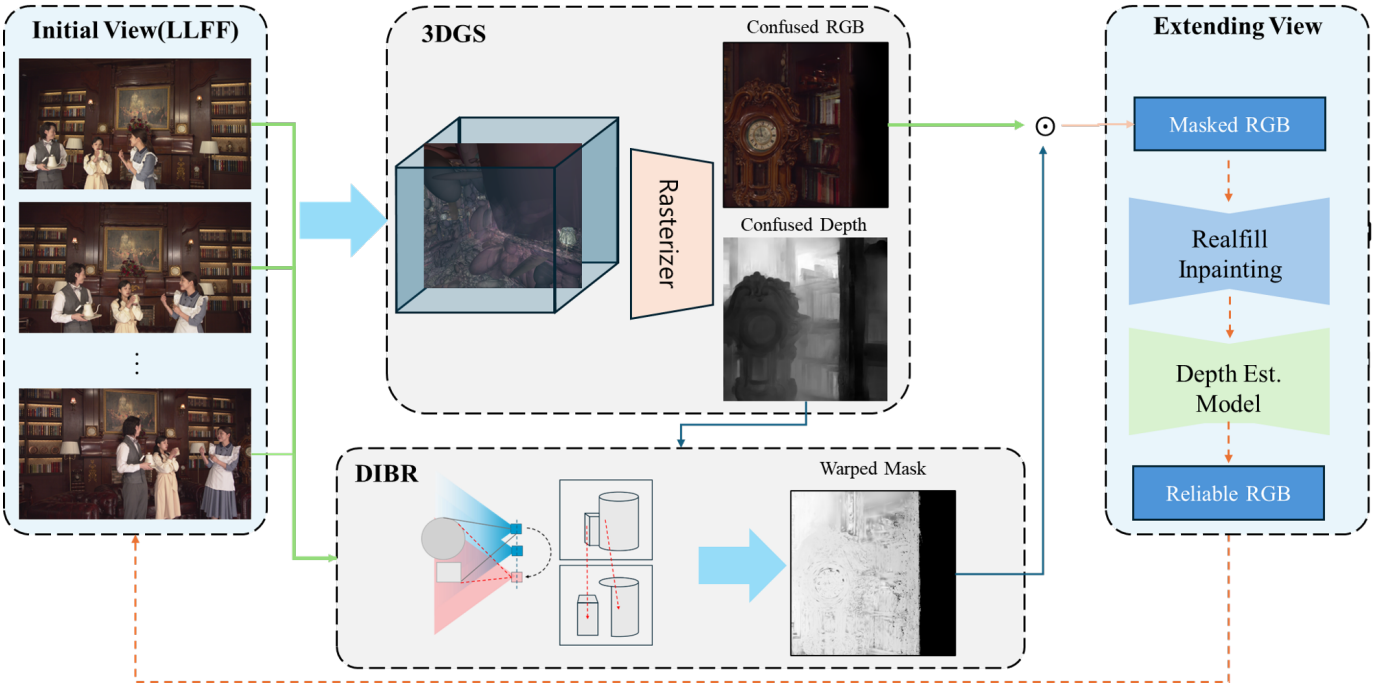


Fig. 2. **Our model framework.** The initial image can be either a multi-view or a single image. From this, a scene is constructed using initial 3D Gaussians. Efficient rasterization enables rendering of both RGB and depth maps. When rendering outside the camera group poses, rendering RGB and depth from an expanded pose can lead to confused results. Since a reliable mask cannot be obtained directly, a trustworthy mask is generated through Depth Image-Based Rendering (DIBR) by warping the mask for the expanded pose. This mask is then multiplied with the RGB image to construct a scene matching the expanded pose.

that perform effectively in sparse-view settings. [21] unifies neural rendering and probabilistic image generation to handle uncertainty and generate unseen regions in sparse-view 3D reconstruction.

Sparse-view Novel View Synthesis (NVS) introduces further challenges for methods such as 3DGS. To address this, MVPGS [22] integrates learning-based Multi-view Stereo (MVS) and forward-warping techniques, enhancing geometric initialization quality and mitigating overfitting through appearance constraints and monocular depth regularization. By introducing view-consistent geometry constraints for Gaussian parameters, it enables high-quality scene reconstruction in sparse-view settings with real-time rendering. Building upon this, MVSGaussian [23] introduces a generalizable 3D Gaussian representation based on MVS, improving novel view synthesis with a hybrid Gaussian rendering method and facilitating fast fine-tuning through a multi-view geometric aggregation strategy for per-scene optimization. Compared to NeRF-based methods and vanilla 3DGS, MVSGaussian offers superior synthesis quality, real-time rendering, and reduced computational costs, achieving state-of-the-art generalization and efficiency across various datasets.

Sparse-view scenarios are relatively uncommon in practical applications. In forward-facing or 360-degree photography, cameras typically focus on capturing key areas of the scene, prioritizing relevant information. This means that sparse-view configurations are seldom encountered. Methods addressing sparse-view synthesis may not always reflect the typical constraints encountered in practical 3D reconstruction tasks. Thus, we focus on scenarios such as LLFF, where we aim to address the sparse-view challenges commonly encountered in forward-facing or 360-degree capture methods.

Diffusion based 3D Generating Diffusion models, initially developed for generative image synthesis, have been increasingly applied to 3D scene reconstruction and data augmentation. These models utilize a probabilistic framework to iteratively generate realistic images by reversing the diffusion process. DreamGaussian [24] refines 3D representations by leveraging the diffusion process. In the initial stage, random points are used to generate preliminary Gaussians via SDS loss, which often results in blurry and low-detail outputs. To improve this, a Mesh Extraction process is introduced to generate textured meshes from the Gaussian representations. These meshes are then refined using a diffusion model to enhance texture quality and resolve spatial inconsistencies. Finally, the UV map is refined, producing high-quality 3D data that strikes a balance between computational efficiency and visual fidelity. 3D Gaussian Enhancer [25] presents a VAE (Variational Autoencoder)-based framework that incorporates a video diffusion prior for novel view enhancement. This method effectively improves both image quality and view consistency in diffusion-based 3D generation. Sparse images are initially used to build the Gaussian space, followed by novel view rendering with noise. These rendered images are treated as a video, with a video diffusion model, including a denoising U-Net, applied to remove noise. This process

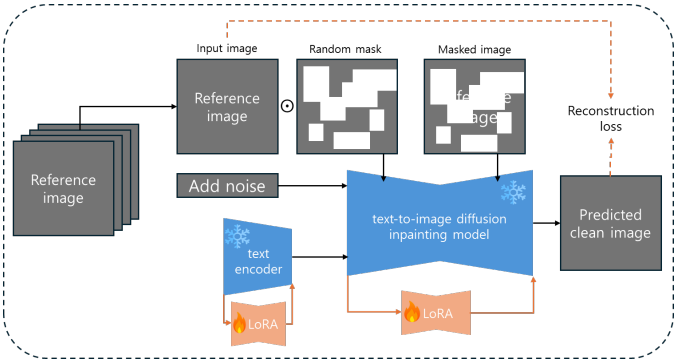


Fig. 3. **RealFill framework.** A random mask is applied to the reference image, and image diffusion inpainting is performed. The diffusion model is modified using LoRA to learn how to generate the predicted clean image. The model is trained using the input image and a reconstruction loss.

ultimately yields clean video RGB frames and camera poses. [26] leverages large-scale vision model priors to enhance 3D Gaussian Splatting for sparse-view 3D reconstruction, addressing challenges in data scarcity and reconstruction quality.

Together, these two research directions underscore the importance of both learning-based spatial representations and generative diffusion techniques in addressing the limitations of 3D scene reconstruction. By combining efficient 3D representations with generative models, our work aims to bridge the gap between high-quality reconstruction and real-world applicability in multi-view environments.

III. Main Method

A. Preliminary

Introduction to 3D Gaussian Splatting (3DGS) Here, we provide a concise overview of the formulation and rendering process of 3D Gaussian Splatting (3DGS) [27]. 3DGS models a scene as a collection of anisotropic 3D Gaussian spheres, enabling high-quality novel view synthesis (NVS) with exceptionally low rendering latency. Each Gaussian sphere is defined by its center position $\mu \in \mathbb{R}^3$, a scaling factor $\mathbf{s} \in \mathbb{R}^3$, and a rotation quaternion $\mathbf{q} \in \mathbb{R}^4$. The corresponding Gaussian distribution is expressed as:

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}, \tag{1}$$

where the covariance matrix $\Sigma = RSS^T R^T$ is determined by the scaling matrix S (derived from $\mathbf{s}$) and the rotation matrix R (computed from $\mathbf{q}$). To capture view-dependent appearance, each Gaussian sphere also includes spherical harmonics (SH) coefficients $\mathcal{C} \in \mathbb{R}^k$ (where k is the number of SH functions) and an opacity parameter $\alpha \in \mathbb{R}$. The Gaussian distribution defined in Eq. 1 is utilized to compute both color and opacity.

In the rendering process, all 3D Gaussian spheres are projected onto the 2D camera plane using a differentiable Gaussian splatting pipeline. The covariance matrix Σ' in camera coordinates is computed using the viewing transformation matrix W and the Jacobian matrix J of the affine approximation of the projective transformation:

$$\Sigma' = JW\Sigma W^T J^T. \tag{2}$$

This differentiable splatting efficiently maps 3D Gaussian spheres to 2D Gaussian distributions, facilitating fast α-blending for rendering and enabling color supervision. For each pixel, the final color is computed by aggregating contributions from M overlapping Gaussian spheres, which are sorted by depth along the viewing direction. The color at a pixel is given by:

$$C = \sum_{i \in M} \mathcal{C}_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_i). \tag{3}$$

This process ensures accurate blending of colors and opacities for photorealistic rendering.

Leveraging Diffusion Models for Scene Reconstruction RealFill [28] is a generative approach for image completion that fills missing areas in an image using content derived from up to n reference images (n $\leq$ 5), denoted as $\mathcal{X}_{ref} := \{I^k_{\text{ref}}\}_{k=1}^n$. The method also uses a target image $I_{tgt} \in \mathbb{R}^{H \times W \times 3}$ and an associated binary mask $M_{tgt} \in \{0,1\}^{H \times W}$, where 0 represents the existing regions in I_{tgt}, and 1 indicates the areas to be filled. The model generates a realistic output image $I_{out} \in \mathbb{R}^{H \times W \times 3}$ by conditioning on the target image I_{tgt} and mask M_{tgt}.

To achieve this, a pretrained generative model is fine-tuned on the reference images to incorporate scene-specific knowledge. This fine-tuned model then generates I_{out} conditioned on I_{tgt} and M_{tgt}, ensuring that the generated content aligns with the scene's context. The process begins with the Stable Diffusion v2 inpainting model [29], into which LoRA weights are injected into its text encoder and U-Net. The model is subsequently fine-tuned on both $\mathcal{X}_{ref}$ and I_{tgt} using randomly generated binary masks m. The loss function is defined as:

$$\mathcal{L} = \mathbb{E}_{x,t,\epsilon,m} \|\epsilon_\theta(x, t, p, m, (1-m) \odot x) - \epsilon\|_2^2, \tag{4}$$

where $x \in \mathcal{X}_{\text{ref}} \cup \{I_{\text{tgt}}\}$, p is a fixed language prompt containing a rare token, $\odot$ denotes the element-wise product, and $(1-m) \odot x$ represents the masked clean image. For the target image I_{tgt}, the loss is calculated only in the regions that exist in the original image. To create the random mask m for each training example, random rectangles are generated, and either their union or the complement of their union is used as the mask, as illustrated in Fig. 3.

During inference, a Denoising Diffusion Probablistic Model (DDPM) [30], [31] sampler generates the image I_{gen}, conditioned on p, I_{tgt}, and M_{tgt}. To ensure a seamless transition between the generated region and the original content of I_{tgt}, the binary mask M_{tgt} is feathered, producing $M_{feathered}$, which blends the boundaries. The final output image I_{out} is then obtained by alpha compositing I_{gen} and I_{tgt} based on $M_{feathered}$, as expressed in the following equation:

$$I_{\text{out}} = M_{\text{feathered}} \odot I_{\text{gen}} + (1 - M_{\text{feathered}}) \odot I_{\text{tgt}}. \tag{5}$$

Since the diffusion process depends on random seeds, generated images vary across runs. Therefore, multiple samples are generated in a batch $\{I_{out}\}$. To select the best output, the

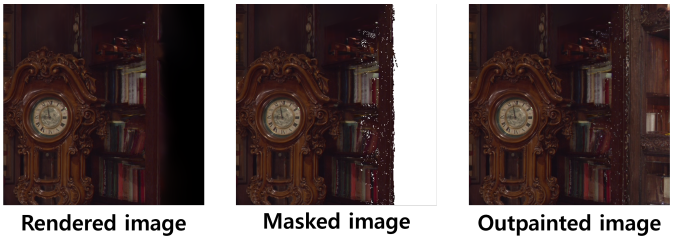
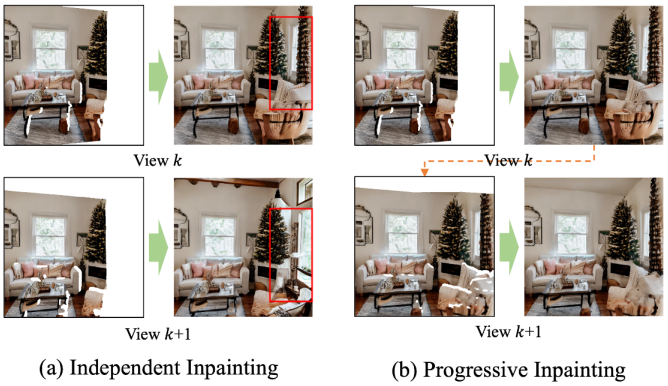


Fig. 4. The leftmost image is the one rendered using 3D Gaussian. The middle image is the masked version of the rendered image, and the rightmost image is the final out-painted result.



(a) Independent Inpainting (b) Progressive Inpainting

Fig. 5. **progressive outpainting.** The intuitive independent inpainting strategy simultaneously performs rendering and inpainting for each view. Since there are no 3D constraints during 2D inpainting, the overlapping regions inpainted in different views can be view-inconsistent (as shown in the red box). In contrast, the progressive inpainting strategy achieves view-consistent inpainting results by reflecting the previously inpainted content into the next view.

filled region of each I_{out} is compared to the reference images $\mathcal{X}_{ref}$ based on the number of feature correspondences. The outputs are then ranked, and the highest-quality result is used.

B. Scene Initialization

Single Image When scene expansion is performed with a single image, only one image is used as input. We use a pre-trained monocular depth model to generate a depth map for the input image. Using the generated depth map and the input image, we reconstruct 3D points as follows:

$$\mathbf{p}_s = [R|t] \cdot \left(D(u,v) \cdot K^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \right), \tag{6}$$

where $\mathbf{p}_s$ represents the reconstructed 3D point, and K denotes the camera's intrinsic parameters. For a single input image, the camera pose is set as the identity matrix, and the intrinsic focal length is determined based on the image resolution.

The 3D points reconstructed from monocular depth are used for Gaussian initialization and data augmentation. For Gaussian initialization, the reconstructed points serve as the initial mean values, while the covariance parameters are set in the same manner as in 3D Gaussian Splatting [27]. Training a 3D Gaussian with a single image can result in significant overfitting to that specific view. To mitigate this issue, we employ depth image-based rendering (DIBR) for data augmentation, generating a support set of eight views around the input image. This support set is used for supervised training of the initial 3D Gaussian alongside the input image.

Multi View Image When multi-view images are used as input, we initialize the 3D Gaussian using camera parameters and point clouds obtained from COLMAP. The 3D Gaussian is trained using these parameters, and the trained model is subsequently used to render depth for the input multi-view images. Both the trained 3D Gaussian and the rendered depth maps are used as initialization values. Unlike the single image case, additional support set generation using depth image-based rendering (DIBR) is not performed for multi-view inputs, as sufficient diversity is inherently provided by the multi-view setup.

C. Progressive Scene Expansion

We expand the scene progressively by training the initialized 3D Gaussian and applying out-painting, enabling the continuous and natural generation of the scene. First, a new camera pose is generated in the direction of expansion by applying rotation and translation to the previous camera pose. The initialized 3D Gaussian is rendered for this new camera pose, but it cannot fully cover the entire image. To address this limitation, out-painting is applied to fill the uncovered regions. To generate the mask required for out-painting, depth image-based rendering (DIBR) is performed for all previously utilized views in the direction of the new camera pose. During DIBR, textures from existing views are projected, and regions where textures cannot reach are identified as unseen areas, which are then masked. These masked regions represent areas to be expanded. The generated images and masks are input into an out-painting model to produce extended images, which are added to the training dataset of the 3D Gaussian Splatting model. The 3D Gaussian is further trained using the updated dataset, continuing from its previously learned state. This process is iteratively repeated, progressively expanding the scene.

IV. Experiments

A. Datasets

We utilized a video dataset captured using a multi-camera setup mounted on a fixed rig. The videos were recorded in static scenes without dynamic objects, and the 0th frame of each video was extracted for training. Since the diffusion-based out-painting model is limited to a resolution of 512, we cropped the scenes to a size of 512 for expansion. When expanding the scene using a single image, we used the cropped images. For scene expansion using multi-view data, we utilized the original FHD-resolution videos, proceeding with a size of 512 in the direction of expansion.

B. Results of Rendering

Qualitative Results. We conducted scene expansion experiments in two cases: single-view and multi-view inputs. The results for single-view input can be observed in Fig 6 and Fig

Fig. 6. **Single View Qualitative Results.** Single view scene expansion results using RealFill. Scene expanded using only the central image as input.

Fig. 7. **Single View Qualitative Results.** Results of scene expansion for a single view through parameter adjustment.

7. It was confirmed that continuously expanded images can be rendered even with a single input. Fig 8 shows the results of expanded images obtained with multi-view input. A total of 21 images were used as input, and the expanded images in the outer regions can be observed.

Fig. 8. **Single View Qualitative Results.**

V. Conclusion

We proposed a novel methodology for expanding scenes using a 3D rendering model. By utilizing 3D Gaussian Splatting, we obtained a representation of the 3D space while simultaneously expanding the scene using an out-painting model. The expanded scene was progressively updated into the 3D GS model. This approach can be applied in future tasks to render continuous scenes with only a few views and can also be used to expand 3D spaces. However, a limitation remains due to the resolution constraints of diffusion-based models.

References

[1] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[2] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2mesh: Generating 3d mesh models from single rgb images," 2018. [Online]. Available: https://arxiv.org/abs/1804.01654

[3] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization," 2019. [Online]. Available: https://arxiv.org/abs/1905.05172

[4] S. Saito, T. Simon, J. Saragih, and H. Joo, "Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization," 2020. [Online]. Available: https://arxiv.org/abs/2004.00452

[5] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 824–840, 2009.

[6] E. Prados and O. Faugeras, "Shape from shading: a well-posed problem?" in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, 2005, pp. 870–877 vol. 2.

[7] O. A. Karpenko and J. F. Hughes, "Smoothsketch: 3d free-form shapes from complex sketches," *ACM Trans. Graph.*, vol. 25, no. 3, p. 589–598, Jul. 2006. [Online]. Available: https://doi.org/10.1145/1141911.1141928

[8] W. Yin, J. Zhang, O. Wang, S. Niklaus, S. Chen, Y. Liu, and C. Shen, "Towards accurate reconstruction of 3d scene shape from a single monocular image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 6480–6494, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251903426

[9] B. Ke, A. Obukhov, S. Huang, N. Metzger, R. C. Daudt, and K. Schindler, "Repurposing diffusion-based image generators for monocular depth estimation," 2024. [Online]. Available: https://arxiv.org/abs/2312.02145

[10] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," 2020. [Online]. Available: https://arxiv.org/abs/1907.01341

[11] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," 2024. [Online]. Available: https://arxiv.org/abs/2401.10891

[12] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "Zoedepth: Zero-shot transfer by combining relative and metric depth," 2023. [Online]. Available: https://arxiv.org/abs/2302.12288

[13] W. Zhao, Y. Rao, Z. Liu, B. Liu, J. Zhou, and J. Lu, "Unleashing text-to-image diffusion models for visual perception," *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5706–5716, 2023.

[14] M. Lavrenyuk, S. F. Bhat, M. Müller, and P. Wonka, "EVP: enhanced visual perception using inverse multi-attentive feature refinement and regularized image-text alignment," *ArXiv*, vol. abs/2312.08548, 2023.

[15] W. Yin, J. Zhang, O. Wang, S. Niklaus, L. Mai, S. Chen, and C. Shen, "Learning to recover 3d scene shape from a single image," 2020. [Online]. Available: https://arxiv.org/abs/2012.09365

[16] 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.

[17] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics (SIGGRAPH Conference Proceedings)*, vol. 42, no. 4, July 2023. [Online]. Available: http://www-sop.inria.fr/reves/Basilic/2023/KKLD23

[18] C. Wang, X. Wu, Y.-C. Guo, S.-H. Zhang, Y.-W. Tai, and S.-M. Hu, "Nerf-sr: High quality neural radiance fields using supersampling," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 6445–6454.

[19] 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.

[20] 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.

[21] Z.-X. Zou, W. Cheng, Y.-P. Cao, S.-S. Huang, Y. Shan, and S. Zhang, "Sparse3d: Distilling multiview-consistent diffusion for object reconstruction from sparse views," *ArXiv*, vol. abs/2308.14078, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:261244080

[22] W. Xu, H. Gao, S. Shen, R. Peng, J. Jiao, and R. Wang, "Mvpgs: Excavating multi-view priors for gaussian splatting from sparse input views," *arXiv preprint arXiv:2409.14316*, 2024.

[23] T. Liu, G. Wang, S. Hu, L. Shen, X. Ye, Y. Zang, Z. Cao, W. Li, and Z. Liu, "Mvsgaussian: Fast generalizable gaussian splatting reconstruction from multi-view stereo," in *European Conference on Computer Vision*. Springer, 2025, pp. 37–53.

[24] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng, "Dreamgaussian: Generative gaussian splatting for efficient 3d content creation," *arXiv preprint arXiv:2309.16653*, 2023.

[25] X. Liu, C. Zhou, and S. Huang, "3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors," *arXiv preprint arXiv:2410.16266*, 2024.

[26] H. Yu, X. Long, and P. Tan, "Lm-gaussian: Boost sparse-view 3d gaussian splatting with large model priors," *ArXiv*, vol. abs/2409.03456, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272423748

[27] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[28] L. Tang, N. Ruiz, Q. Chu, Y. Li, A. Holynski, D. E. Jacobs, B. Hariharan, Y. Pritch, N. Wadhwa, K. Aberman, and M. Rubinstein, "Realfill: Reference-driven generation for authentic image completion," *ACM Transactions on Graphics*, vol. 43, no. 4, p. 1–12, Jul. 2024. [Online]. Available: http://dx.doi.org/10.1145/3658237

[29] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10 684–10 695.

[30] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020. [Online]. Available: https://arxiv.org/abs/2006.11239

[31] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," 2022. [Online]. Available: https://arxiv.org/abs/2010.02502