GPGS: Consistent 3D Object Removal via Geometry-Aware 3D Inpainting and Projected Image Refinement in 3D Gaussian Splatting

Yongjoon Lee, Donghyeon Cho*

Department of Computer Science, Hanyang University, Seoul, South Korea
{dyd7168, doncho}@hanyang.ac.kr

Abstract

Object removal in 3D space is a key technology for immersive applications such as virtual reality (VR), augmented reality (AR), and the metaverse. While recent approaches have attempted to address this task using 2D inpainting models, they often suffer from two major limitations: (1) inaccurate geometric restoration in the removed regions, and (2) visual inconsistency across multiple viewpoints. To address these challenges, we propose GPGS, a novel pipeline built upon the 3D Gaussian Splatting (3DGS) framework. First, we perform geometry-aware 3D inpainting by leveraging a pre-trained point cloud completion model and a coarse-to-fine inference strategy, enabling accurate restoration of unseen 3D structures. Next, we introduce a projected image refinement method that improves the appearance of novel-view projections by addressing view-dependent artifacts such as brightness shifts and texture misalignments. GPGS further enhances overall scene consistency through fine-tuning of the original 3DGS scene using the refined multi-view images. Experimental results show that our GPGS makes geometrically accurate and visually coherent outputs, even in challenging 360° panoramic scenes, significantly outperforming existing methods.

Code — https://github.com/yongjoon99/GPGS

Introduction

3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) is a promising 3D reconstruction technique that generates novel views by reconstructing 3D scenes as Gaussians primitives from multiple input images. 3DGS has become a core technology in virtual content creation, including virtual reality (VR), augmented reality (AR), and the metaverse, and extensive research is ongoing to further improve its capabilities. In virtual reality applications, interaction between the user and the environment plays a crucial role. To support such interaction, it is necessary not only to reconstruct the 3D scene but also to enable editing of the reconstructed space. In particular, object removal in 3D space is an essential technology that allows users to freely manipulate or eliminate unnecessary or inconsistent objects, thus facilitating seamless interaction between real users and virtual environments.

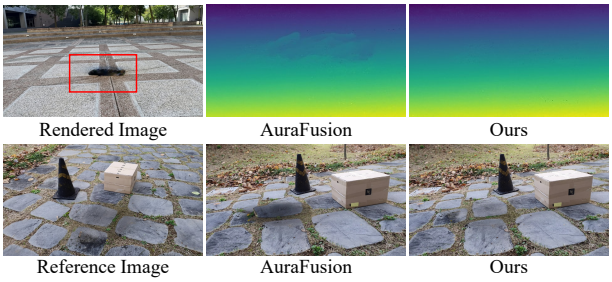

Figure 1: The first row shows the result of projecting the reference view onto another view according to the geometry completion method. Ours cleanly restores the geometry of the empty space in the rendered image. However, AuraFusion shows noisy restoration due to inaccurate depth. The second row shows the results of reference-guided inpainting of AuraFusion and our projected image refinement. Ours maintains the structure of the reference image well and shows natural texture. On the other hand, AuraFusion fails to maintain the structure and shows unnatural texture.

Due to its significance, research on object removal in 3D scenes has been actively explored. Most existing approaches (Mirzaei et al. 2023b; Wu et al. 2025; Lee et al. 2025) apply 2D image inpainting models to the input images to remove the target object, followed by re-training the 3DGS scene to reflect the changes. Early works focus on forward-facing scenes, inpainting the entire object region directly in the input images (Mirzaei et al. 2023b; Weder et al. 2023; Wang et al. 2024a; Liu et al. 2022; Mirzaei et al. 2023a). While this approach works well for narrow field-of-view (FoV) images, it struggles to produce consistent background inpainting for 360° panoramic data with wide FoVs.

To overcome this issue, recent studies have proposed methods that separate objects in 3D space, fill in visible regions using surrounding viewpoints, and inpaint only the unseen regions that are not visible from any view (Lee et al. 2025; Ye et al. 2024; Liu et al. 2024; Wu et al. 2025). These approaches reduce the inpainting area and improve performance, but as shown in Figure 1, two major challenges remain. The first challenge is to accurately restore the geometry of the empty regions left after the object has been re-

*Corresponding author.

moved. The second challenge is to reduce visual inconsistencies across multiple viewpoints, as inpainted regions may appear structurally or texturally inconsistent when viewed from different angles.

To address these challenges, we propose GPGS, a novel pipeline that leverages 3DGS scene representations and object masks. In particular, for accurate geometry restoration, we adopt Point-MAE (Pang et al. 2023) framework to complete the missing 3D structure in unseen regions through a coarse-to-fine point cloud inpainting process. This approach enables the recovery of accurate depth information, which is essential for projecting images into novel views without geometric distortion. Also, for addressing the limitations of inconsistency problem, such as missing regions due to pixel mismatch across views and unnatural appearance caused by view dependent lighting, we introduce a projected image refinement. Our refinement module corrects distortion in the projected images by combining brightness transfer and texture restoration using a CNN trained with Frequency Distribution Loss (FDL) (Ni et al. 2024), which is robust to spatial misalignment. Finally, we fine-tune the original 3DGS scene using the refined projected images to enhance the overall consistency and realism of the reconstructed scene.

To the best of our knowledge, this is the first work to achieve consistent object removal in 3DGS by combining inpainting in 3D space and refining the projected reference image. Our approach proves effective even in 360° panoramic settings, where existing methods often struggle with maintaining multi-view consistency.

Related Works

Object Removal in 3D Scenes

Most existing object removal studies in 3D space use 2D image inpainting models to remove objects from each image and retrain the 3D scene. Object removal methods that simply inpaint the entire object (Mirzaei et al. 2023b; Weder et al. 2023; Wang et al. 2024a) work well in forward-facing environments, but struggle in 360° environments with a wide field of view. To support object removal in 360° environments, several approaches (Lee et al. 2025; Ye et al. 2024; Huang, Chou, and Wang 2025) for removing objects in 3D space have emerged. These methods use 3D spatial information to restore beyond the object, reducing the area to which inpainting is applied, and thus work well in 360° environments. However, they cannot solve the consistency issues that arise because inpainting is applied to all views. Therefore, research has been conducted to solve the consistency issues between each inpainted image. There are efforts (Liu et al. 2024; Huang, Chou, and Wang 2025) that restore the scene using only a single reference image to ensure consistency. Infusion (Liu et al. 2024) performs more accurate depth inpainting to remove objects based on geometric alignment. While this approach guarantees consistency, it has the limitation of being unable to express view-dependent effects. In order to obtain consistent and view-dependent images, many studies (Wu et al. 2025; Shi et al. 2025; Mirzaei et al. 2024; Lin et al. 2024) propose methods that utilize both diffusion models (Rombach et al. 2022) and reference images. AuraFusion (Wu et al. 2025) performs reference-guided inpainting using the projection results of the reference image as a condition to enhance consistency between images. IMFine (Shi et al. 2025) uses the projection results of the reference image to adapt the diffusion model at test time to obtain consistent images. RefFusion (Mirzaei et al. 2024) uses Low-Rank Adaptation (LoRA) (Hu et al. 2022) with data obtained from a 3D scene to generate consistent images using a diffusion model. The method using the diffusion model can obtain natural images from any view, but there is a risk of inconsistency due to the diversity of the diffusion model output. GScream (Wang et al. 2024b) uses a single reference image and cross attention with the surrounding area to maintain texture consistency, but the GS model is limited to Scaffold-GS (Lu et al. 2024) and does not work well in 360° environments. To achieve stable and natural object removal, we refine the projected reference images with a CNN, effectively mitigating the instability of diffusion-based generation.

Point-MAE

Masked Auto Encoder (MAE) (He et al. 2022) is a self-supervised learning technique that masks part of an image and then restores it to learn the general features of the image. Point-MAE (Pang et al. 2023) is a study that applies MAE to point clouds and proposes self-supervised learning using point clouds. Point-MAE extracts center points that represent the shape of the input point cloud through the Farthest Point Sampling (FPS) algorithm (Qi et al. 2017). To train Point-MAE, it is first necessary to convert the point cloud into a patch format that can be used for training. Similar to how MAE creates image patches, Point-MAE uses the KNN algorithm at each center point to construct point patches. After constructing the point patches, the training process proceeds similarly to MAE. Point-MAE is pretrained using ShapeNet (Chang et al. 2015) and utilizes the general 3D features learned through ShapeNet for various downstream tasks. Using only the pretrained weights, it performs point completion to estimate masked point patches. Inspired by this, we propose a point cloud inpainting method that leverages the general 3D representations learned by Point-MAE.

Method

The objective of this paper is to achieve consistent object removal across multiple views, given the following inputs: a scene representation G^o learned via 3D Gaussian Splatting (3DGS), a set of training images $\{I_n\}$, their corresponding camera parameters $\{P_n\}$, and binary masks $\{M_n\}$ indicating the target object to be removed. To address this task, we propose a novel pipeline that projects a single inpainted reference image into different viewpoints and refines the projections to produce natural and visually consistent results. In particular, our proposed pipeline consists of three main stages: object separation, geometry-aware 3D inpainting, and projected image refinement, as illustrated in Figure 2. In the object separation stage, we make object-separated 3DGS G^s using the provided binary masks, and then the missing regions in the reference image are com-

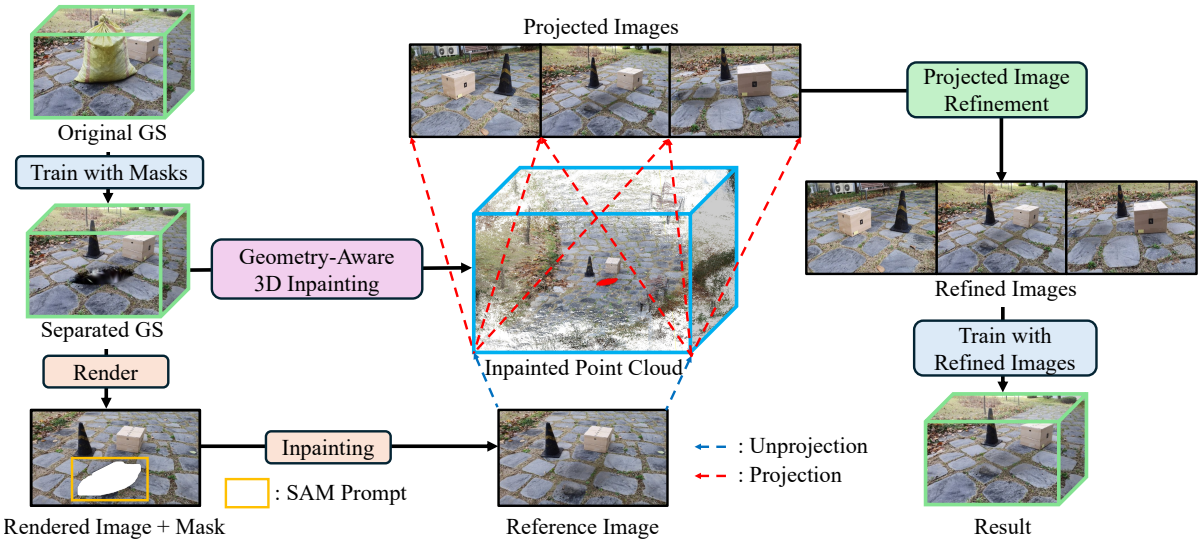

Figure 2: Overall pipeline of GPGS. We first separate the object from the original 3DGS using masks. We render the reference image from the separated 3DGS and inpaint the unseen region using a mask obtained manually through SAM. At the same time, we restore the geometry of the unseen region by inpainting the point cloud using geometry-aware 3D inpainting. We project the reference image onto other views and apply a projected image refinement to obtain refined images. We fine-tune the 3DGS model using the refined images to complete object removal.

pleted using image inpainting techniques. In the geometry-aware 3D inpainting stage, the 3D scene is represented as a point cloud, and missing geometric structures are reconstructed through point cloud inpainting, enabling accurate recovery of the underlying spatial layout. In the projected image refinement stage, the completed 3D geometry is used to project the inpainted reference image into different viewpoints. The projected images are further refined to ensure spatial consistency and high visual fidelity across all views. Finally, G^s is fine-tuned using the refined projected images from multiple views, thereby enhancing the overall consistency and visual realism of the reconstructed scene.

Object Separation

GPGS maintains consistency across viewpoints by completing the masked regions in multiple view images using only a single reference image. Therefore, we have to make a high quality inpainted reference image. To this end, we derive an optimal mask and apply an inpainting model to the manually selected reference image. According to (Lee et al. 2025; Wu et al. 2025; Shi et al. 2025; Huang, Chou, and Wang 2025), separating objects enables us to fill regions outside the object using 3D information, which reduces the area that needs to be inpainted and leads to higher-quality inpainting results. Reflecting this, we separate the object in 3D space before applying inpainting. In particular we perform object separation using a simple method that can be applied to any 3DGS model, as described below. First, we perform additional training using binary masks $\{M_n\}$ instead of the original input images $\{I_n\}$ used in standard 3DGS. Training with masks causes the color elements of the Gaussian points representing the object to approach one, while the remaining points approach zero. In other words, points corresponding to the object exhibit higher color intensities (*i.e.*, values approaching white), whereas the others remain near zero. When 3DGS scene has K points, by comparing the modified colors $c^m_{1:K}$ obtained after training with masks to the original colors $c^o_{1:K}$, we can identify the points corresponding to the object as follows.

$$M^{3d}_k = \begin{cases} 0, & \text{if } c^m_k < c^o_k, \\ 1, & \text{otherwise,} \end{cases} \tag{1}$$

where M^{3d}_k is a mask that determines whether the k-th point corresponds to an object, and we can get G^s by removing the masked point from G^o. After separating the objects, we construct an optimal binary mask of the reference image that indicates the unseen regions occluded across all viewpoint images for inpainting using SAM (Kirillov et al. 2023) manually. Then, we use LeftRefill (Cao et al. 2024) to inpaint the unseen regions, resulting in a high-quality reference image.

Geometry-Aware 3D Inpainting

To generate novel views that maintain consistency with the reference image, information from it should be propagated to the other viewpoints. The most straightforward way to transfer information from the reference image to another view is to project it using the corresponding depth map and camera parameters. We can render depth maps from a 3DGS with objects removed, but we get incomplete depth for unseen regions. To resolve this issue, existing methods use depth inpainting and depth alignment methods. Depth inpainting methods solve this problem by fine-tuning existing image inpainting models to inpaint depth well. Meanwhile,

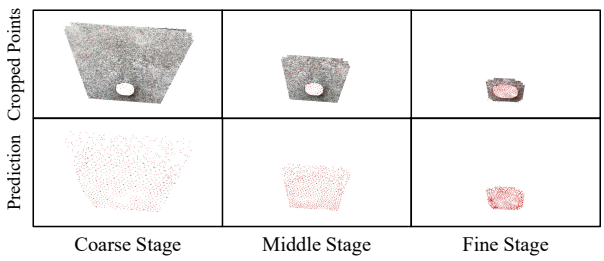

Figure 3: Point prediction results based on the coarse-to-fine strategy. Red points denote the Point-MAE prediction results, and grey points indicate the unprojected points from the reference view. As the process progresses from coarse to fine, the proportion of unseen regions is gradually reduced.

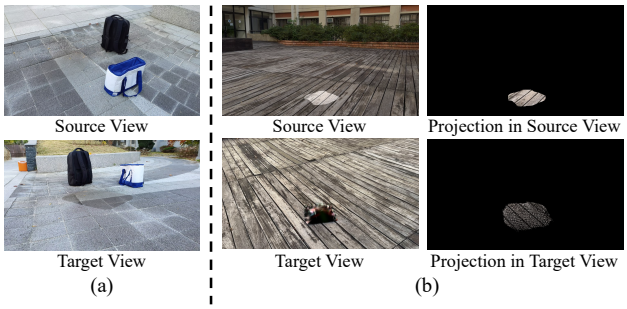

Figure 4: The source view provides pixels for projection, while the target view receives them. (a) Projection can produce brightness inconsistencies when view-dependent effects are ignored. (b) Projection may fail to cover unseen regions in other views due to insufficient projected points.

depth alignment methods estimate monocular depth from the reference image and then align it to the coordinate system of the 3D scene. However, both methods cannot guarantee accuracy during projection because even a small error in the estimated depth map is amplified during the projection.

To mitigate this problem, we propose a geometry-aware 3D inpainting that addresses the limitations of 2D inpainting by operating directly in 3D space. To perform inpainting in 3D space, we devise a method based on Point-MAE (Pang et al. 2023) that can fill in unseen regions. Point-MAE is a point cloud completion method that can generate point clouds that follow the geometric features of general point clouds by learning ShapeNet (Chang et al. 2015). In particular, it can accurately estimate point clouds in lost areas by general 3D feature. Inspired by this, we make point clouds of the 3D scene by unprojecting the rendered images from G^s and use Point-MAE to geometrically restore unseen regions. Estimating the point cloud of unseen regions allows us to obtain accurate depth, but there is a problem with directly applying Point-MAE. Point-MAE receives a point cloud as input and uses the FPS algorithm to extract 64 center points that represent shape of the whole point cloud. Then, it samples 32 points for each center point using the KNN algorithm to form 64 point patches. In the inference process, Point-MAE restores the corresponding point patch for the given center point. Therefore, it cannot directly estimate points for empty spaces without center points.

Fortunately, points estimated from adjacent center points can still populate regions that lack their own center points. To obtain dense points and fully fill the interior of these unseen regions, we adopt a coarse-to-fine prediction strategy. As shown in coarse stage in Figure 3, we initially crop a wide point cloud that includes the unseen region but has a small proportion of the unseen region area initially. Then, we crop the point cloud so that the proportion of the unseen region gradually increases and use the sparse points in unseen regions as center points to densely fill them. As shown in middle and fine stage of Figure 3, the larger the proportion of unseen regions, the more densely the unseen regions can be filled. Through the coarse-to-fine stage, points can be estimated evenly in unseen region. However, Point-MAE estimates a fixed number of 2,048 points, and most unseen regions require more points. Therefore, we repeatedly overlap the fine stage until the unseen region is sufficiently filled.

However, Point-MAE has general 3D information, but its ability may be limited when dealing with data it has never seen before. Thus, we fine-tune Point-MAE to be specific to the scene. To fine-tune Point-MAE, we make point clouds of 3D scene by unprojecting the rendered images and crop the point cloud for training data. Then, we fine-tune Point-MAE through training that masks and restores cropped point clouds using the Chamfer distance loss. Point-MAE trained on 3D scenes can restore unseen regions more accurately.

Projected Image Refinement

Using the depth obtained through geometry-aware 3D inpainting, reference images can be accurately projected into different views. However, two fundamental problems arise in the process of projecting a 2D image into another view. The first issue is that projection ignores view-dependent effects. Even if the same object is in the same location, its brightness changes depending on the light conditions when the view changes. However, projection simply transfers the intensity from a specific view to another view, which often results in unnatural brightness when viewed from another view, as shown in Figure 4(a). The second issue is that the number of points projecting the unseen region from the reference view may not be sufficient for the unseen region of the other view. In 3D space, the size of the unseen region is fixed, but in 2D images, the size varies depending on the view. As shown in Figure 4(b), if the size of the unseen region in the reference view is smaller than that in other views, the number of pixels projected is insufficient to fill the entire unseen region in other views. In such cases, blanks that cannot be filled by projection are created, and texture distortion occurs during the filling process. As a result, the projected output is accurate in terms of content but produces images with distorted brightness and textures.

To address the intensity inconsistency, we apply color transfer method (Reinhard et al. 2001) to match the brightness tone of the unseen region by leveraging the intensity

statistics of the surrounding area to the unseen region. To determine the surrounding area in all views, we apply morphological dilation to the unseen region mask of the reference image and project it onto other views to obtain the surrounding mask. Then, we convert RGB color to LAB color space, which can separates lightness from chromatic components, and then apply mean-std transfer only to the L channel that represents lightness as follows.

$$L_t^u = \frac{\sigma^s}{\sigma^u}(L^u - (\mu^u)) + \mu^s, \tag{2}$$

where L^u denotes the lightness elements of the unseen region, and L_t^u denotes the result of the intensity transfer applied to L^u. Note that σ^s and σ^u denote the standard deviations, and μ^s and μ^u denote the means of the lightness values in the surrounding and unseen regions, respectively.

To address the texture distortion, we propose a projection refinement network. The projection refinement network is a type of restoration network that makes projected results more natural. We construct a CNN consisting of five convolution layers for refinement learning. This network receives distorted images and predicts the residual between the Ground Truth (GT) image and the distorted image. To train the network, we need pairs of distorted images and GT images. However, there are no corresponding GT images for projection in unseen regions. To address this, as shown in Figure 5, we use a training strategy that learns restoration from the surrounding area and applies it to the unseen region. This strategy is based on the assumption that the surrounding area is similar to the unseen region. In this strategy, we slice the projected and GT images in the surrounding areas into 16×16 patches to construct the data for training. To train a model that restores the texture distortion of the projected image, a special loss function is required. This is because even with an accurate projection, slight misalignment occurs, making the use of a pixel-level loss function inappropriate. To effectively guide the texture restoration model, we use the FDL(Ni et al. 2024), which is robust to misalignment. The formula for FDL is as follows.

$$L_{FDL}(x, y) = WD(A_x, A_y) + WD(P_x, P_y), \tag{3}$$

where x is the output of model, y is the target signal, and WD denotes the Wasserstein Distance (Arjovsky, Chintala, and Bottou 2017). Also, A_x, A_y denotes the amplitude component of the discrete Fourier transform of the x, y, while P_x, P_y denotes the corresponding phase components. This loss is effective for restoring texture by transforming it into the frequency domain and is robust to misalignment by measuring the distance between distributions. Additionally, for learning robust to view and position changes, we concatenate coordinate and image index embeddings per image patch during training. After training, we apply the projection refinement network to the projected unseen region to refine the distorted results.

In the final training stage, we add initial Gaussian points to the unseen region using the reference image and depth obtained through geometry-aware 3D inpainting to quickly restore the unseen region. We complete natural and consistent object removal by fine-tuning G^s with images obtained through the projection refinement network.

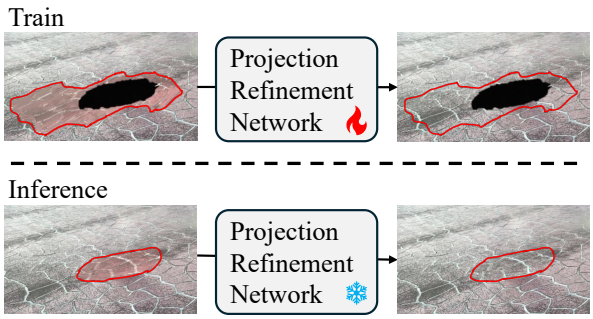

Figure 5: Training and inference of the projection refinement network. We train the projection refinement network from the surrounding areas of the unseen region. Then, we refine the unseen area by inference in the unseen area.

Experiments

Dataset. We conduct experiments on two 360°object removal datasets. First is COR-NeRF dataset (Lee et al. 2025), which consists of 11 unbounded scenes covering 360° views. Each scene has 150-200 train images with objects and a similar number of test images without objects. The second is the 360-USID dataset (Wu et al. 2025), which consists of 7 unbounded scenes covering 360° views. Each scene has 180–200 training images with objects and 30–40 test images without objects. It also provides reference images for object removal task. We choose these datasets for quantitative evaluation, as they provide GT data.

Metrics. To quantitatively evaluate the superiority of GPGS, we measure the following four metrics. PSNR and SSIM (Hore and Ziou 2010), which are traditional methods for viewing differences at the pixel level. LPIPS (Zhang et al. 2018) and FID (Heusel et al. 2017), which measure perceptual differences across the entire image. To focus on measuring the performance of object removal, we measure the metric in the bounding box area of the object mask.

Implementation Details

During the image generation stage, when separating objects, we conduct 300 iterations of training using mask-based learning. In the COR-NeRF data, we manually select one image to create a reference image. In the 360-USID data, we use the provided reference image and unseen mask. All comparison methods and GPGS use the same reference image. In the geometry-aware 3D inpainting stage, the unseen region mask is dilated morphologically in five steps, and the mask area is projected to perform a coarse-to-fine strategy. In the fine stage, inference is repeated and overlapped until the depth of the unseen region reaches 70%, and the remaining depth is filled by interpolation with the neighboring areas. We use the RaDe-GS (Zhang et al. 2024) as a 3DGS model to obtain better geometric reconstruction without floaters. All experiments use a single RTX 3090 GPU.

Figure 6: Qualitative comparisons on 360°object removal data. The top two rows are results from COR-NeRF data, and the bottom one row is results from 360-USID data. The results of each method are the red box areas in the input image. Compared to other methods, GPGS preserves content well and produces seamless results.

	COR-NeRF data			
Method	PSNR↑	SSIM↑	LPIPS↓	FID↓
Gaussian Grouping (Ye et al. 2024)	18.520	0.485	0.329	102.548
Infusion (Liu et al. 2024)	18.405	0.516	0.312	93.900
Gscream (Wang et al. 2024b)	18.105	0.466	0.415	198.178
AuraFusion (Wu et al. 2025)	**18.832**	0.510	0.294	76.003
GPGS (Ours)	18.762	**0.524**	**0.286**	**73.694**

Table 1: Quantitative evaluation of the COR-NeRF dataset. The bold score is the highest score and the underlined score is the second highest score.

	360-USID data			
Method	PSNR↑	SSIM↑	LPIPS↓	FID↓
Gaussian Grouping (Ye et al. 2024)	17.026	0.380	0.338	145.571
Infusion (Liu et al. 2024)	16.675	0.351	0.376	167.728
Gscream (Wang et al. 2024b)	14.593	0.236	0.660	450.412
AuraFusion (Wu et al. 2025)	**18.255**	0.433	0.329	146.000
GPGS (Ours)	17.972	**0.445**	**0.306**	**116.822**

Table 2: Quantitative evaluation of the 360-USID dataset. The bold score is the highest score and the underlined score is the second highest score.

Comparison with Other Methods

We evaluate GPGS by comparing it with the following four state-of-the-art methods. Gaussian Grouping (Ye et al. 2024), Infusion (Liu et al. 2024), Gscream (Wang et al. 2024b), AuraFusion (Wu et al. 2025). Since the 2D Gaussian splatting (Huang et al. 2024) used by AuraFusion does not work properly in the COR-NeRF data, we change the 3DGS model to RaDe-GS and conduct experiments. In addition, in COR-NeRF data, since AuraFusion cannot estimate the unseen region properly, we provide a correct unseen region mask created manually. In other words, favorable conditions are given to existing methods for fair comparisons.

Figure 6 shows the qualitative results of the baselines and GPGS. Gaussian grouping applies inpainting without any restrictions in all views, resulting in blurry images with consistency issues. In the case of Infusion, consistency can be maintained by using a single reference image and training only in that view, but produces blurred results in other views. In addition, artifacts occur due to inaccuracies in depth inpainting. AuraFusion performs inpainting guided by the reference image, achieving more consistent results compared to inpainting models that are not guided by the reference image. However, it cannot fully control the output diversity of the diffusion-based inpainting model, leading to consistency issues. In contrast, GPGS projects a single reference image to obtain consistent images from other views. Furthermore, by refining the projected images, we achieve seamless restoration results that match the surrounding areas in terms of brightness and texture. In particular, we observe that GPGS outperforms others in scenes with complex patterns, such as the first and third rows, whereas simpler scenes like the second row show smaller differences.

Table 1 and Table 2 show a comparison of quantitative evaluation results with other methods. GPGS outperforms other methods in all metrics except PSNR. PSNR is a pixel-level similarity metric that may, in some cases, assign higher scores to relatively blurry results, and inconsistent images result in blurry object removal results. Thus, although PSNR does not outperform, we believe this result demonstrates that GPGS achieves consistent and natural 3D object removal across all views.

Ablation Studies

To demonstrate the effectiveness of geometry-aware 3D inpainting and projected image refinement, we conduct the following ablation studies. The experiments use 360-USID data where the reference image is given as GT.

Ablation Study on Geometry Completion. To demonstrate the effectiveness of geometry-aware 3D inpainting, we conduct a comparison with other depth inpainting models. We select four baseline methods to compare existing depth inpainting methods with geometry-aware 3D inpainting. (1) LaMa (Suvorov et al. 2022), a transformer-based inpainting model. (2) Infusion (Liu et al. 2024), which fine-tunes the

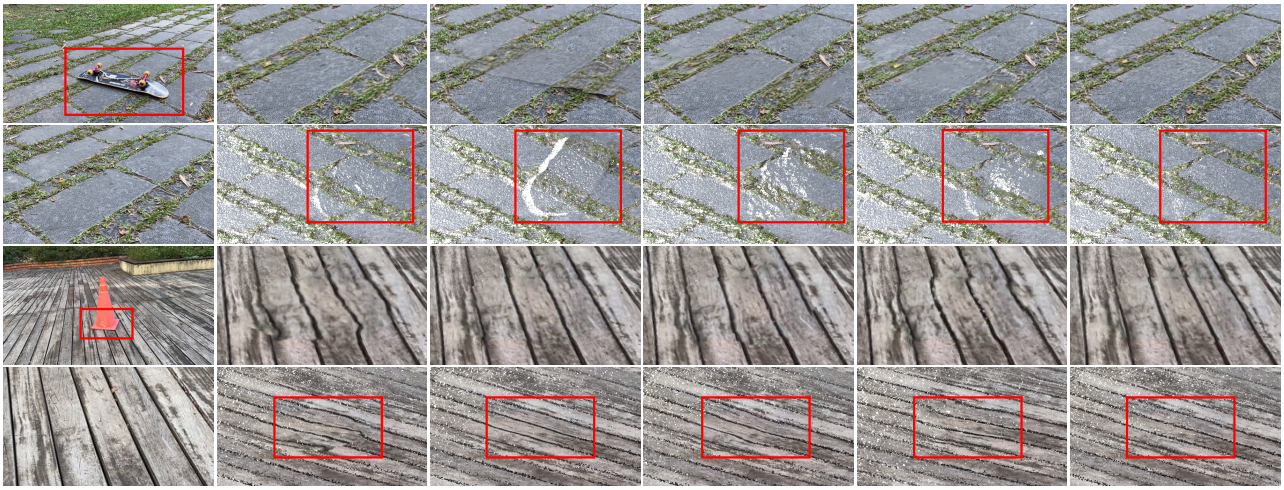

Figure 7: Qualitative comparison of geometry completion methods. The results of each method are the red box areas in the input image. For each sample, the first row shows the rendered images from the learned 3DGS, and the second row shows the result of projecting the images in the first row onto 3D points the estimated depth maps. Compared to other methods, our geometry-aware 3D inpainting maintains structure well.

	360-USID data			
Method	PSNR↑	SSIM↑	LPIPS↓	FID↓
LaMa (Suvorov et al. 2022)	17.839	0.433	0.323	130.045
Infusion (Liu et al. 2024)	17.828	<u>0.439</u>	<u>0.314</u>	<u>126.758</u>
AGDD (Wu et al. 2025)	<u>17.902</u>	0.437	0.328	136.474
Ours w/o Fine-tuning	17.820	0.430	0.346	151.462
Ours	**17.972**	**0.445**	**0.306**	**116.822**

Table 3: Quantitative comparison of geometry completion methods. The bold score is the highest score and the underlined score is the second highest score.

	360-USID data			
Method	PSNR↑	SSIM↑	LPIPS↓	FID↓
w/o Refinement	17.637	0.432	0.323	126.917
w/o Intensity Transfer	17.813	0.436	<u>0.322</u>	<u>126.274</u>
w/o FDL	<u>17.899</u>	**0.445**	0.335	147.244
Ours	**17.972**	**0.445**	**0.306**	**116.822**

Table 4: Quantitative evaluation of the refinement ablation experiments. The bold score is the highest score and the underlined score is the second highest score.

diffusion model. (3) AGDD proposed by AuraFusion (Wu et al. 2025). (4) Geometry-aware 3D inpainting without Point-MAE fine-tuning. We replace the geometry-aware 3D inpainting with baseline methods, while keeping the remaining processes identical to GPGS. Figure 7 shows a qualitative comparison of geometry completion methods. Other baselines may look acceptable in terms of rendering results, but it can be observed that their projection results fail to maintain the structure well. In contrast, our method maintains the structure of the reference image well with accurate depth estimation. Table 3 shows quantitative results of geometry completion methods. Our method outperforms other depth inpainting methods, which proves the effectiveness of geometry-aware 3D inpainting.

Ablation Study on Projected Image Refinement. To demonstrate the effects of each element in the our projected image refinement method, we compare the results by removing each element one by one. The experiments are as follows: (1) Training without applying the projection refinement network. (2) Training without applying intensity transfer. (3) Training using L1 loss instead of FDL in the projection refinement network. We do not change any other parts besides the projected image refinement method. Table 4 shows quantitative results of each experiment. The results show that using FDL in the projection refinement network significantly contributes to improving perceptual accuracy. Additionally, intensity transfer is found to have a notable impact on metrics such as PSNR and SSIM.

Conclusion

In this paper, we have proposed GPGS, a novel pipeline for consistent object removal in 3D scenes. First, we manually select a reference view and generate a completed reference image by performing accurate object separation followed by image inpainting to fill the missing regions. Then, we employ geometry-aware 3D inpainting via Point-MAE to accurately restore the missing 3D structures in unseen regions. In addition, a projected image refinement enhances the quality of projected images by correcting brightness and texture distortion. By fine-tuning the separated 3DGS with refined images, we enable seamless, coherent removal results. Extensive experiments on 360°datasets demonstrate that our approach outperforms state-of-the-art 3D inpainting methods in terms of perceptual quality and multi-view consistency.

Acknowledgements

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2020-II201373, Artificial Intelligence Graduate School Program(Hanyang University)) and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2025-00521432).

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Cao, C.; Cai, Y.; Dong, Q.; Wang, Y.; and Fu, Y. 2024. Leftrefill: Filling right canvas based on left reference through generalized text-to-image diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7705–7715.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16000–16009.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NIPS*.

Hore, A.; and Ziou, D. 2010. Image quality metrics: PSNR vs. SSIM. 2366–2369.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Huang, B.; Yu, Z.; Chen, A.; Geiger, A.; and Gao, S. 2024. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, 1–11.

Huang, S.-Y.; Chou, Z.-T.; and Wang, Y.-C. F. 2025. 3D Gaussian Inpainting with Depth-Guided Cross-View Consistency. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 26704–26713.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4): 139–1.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4015–4026.

Lee, Y.; Ryu, J.; Yoon, D.; and Cho, D. 2025. Consistent Object Removal from Masked Neural Radiance Fields by Estimating Never-Seen Regions in All-Views. In *International Conference on Pattern Recognition*, 416–431. Springer.

Lin, C. H.; Kim, C.; Huang, J.-B.; Li, Q.; Ma, C.-Y.; Kopf, J.; Yang, M.-H.; and Tseng, H.-Y. 2024. Taming latent diffusion model for neural radiance field inpainting. In *European Conference on Computer Vision*, 149–165. Springer.

Liu, H.-K.; Shen, I.; Chen, B.-Y.; et al. 2022. Nerf-in: Free-form nerf inpainting with rgb-d priors. *arXiv preprint arXiv:2206.04901*.

Liu, Z.; Ouyang, H.; Wang, Q.; Cheng, K. L.; Xiao, J.; Zhu, K.; Xue, N.; Liu, Y.; Shen, Y.; and Cao, Y. 2024. Infusion: Inpainting 3d gaussians via learning depth completion from diffusion prior. *arXiv preprint arXiv:2404.11613*.

Lu, T.; Yu, M.; Xu, L.; Xiangli, Y.; Wang, L.; Lin, D.; and Dai, B. 2024. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20654–20664.

Mirzaei, A.; Aumentado-Armstrong, T.; Brubaker, M. A.; Kelly, J.; Levinshtein, A.; Derpanis, K. G.; and Gilitschenski, I. 2023a. Reference-guided controllable inpainting of neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, 17815–17825.

Mirzaei, A.; Aumentado-Armstrong, T.; Derpanis, K. G.; Kelly, J.; Brubaker, M. A.; Gilitschenski, I.; and Levinshtein, A. 2023b. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20669–20679.

Mirzaei, A.; De Lutio, R.; Kim, S. W.; Acuna, D.; Kelly, J.; Fidler, S.; Gilitschenski, I.; and Gojcic, Z. 2024. Reffusion: Reference adapted diffusion models for 3d scene inpainting. *arXiv preprint arXiv:2404.10765*.

Ni, Z.; Wu, J.; Wang, Z.; Yang, W.; Wang, H.; and Ma, L. 2024. Misalignment-robust frequency distribution loss for image transformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2910–2919.

Pang, Y.; Tay, E. H. F.; Yuan, L.; and Chen, Z. 2023. Masked autoencoders for 3d point cloud self-supervised learning. *World Scientific Annual Review of Artificial Intelligence*, 1: 2440001.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Reinhard, E.; Adhikhmin, M.; Gooch, B.; and Shirley, P. 2001. Color transfer between images. *IEEE Computer graphics and applications*, 21(5): 34–41.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Shi, Z.; Huo, D.; Zhou, Y.; Min, Y.; Lu, J.; and Zuo, X. 2025. Imfine: 3d inpainting via geometry-guided multi-view refinement. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 26694–26703.

Suvorov, R.; Logacheva, E.; Mashikhin, A.; Remizova, A.; Ashukha, A.; Silvestrov, A.; Kong, N.; Goka, H.; Park, K.; and Lempitsky, V. 2022. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the*

IEEE/CVF winter conference on applications of computer vision, 2149–2159.

Wang, D.; Zhang, T.; Abboud, A.; and Süsstrunk, S. 2024a. Innerf360: Text-guided 3d-consistent object inpainting on 360-degree neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12677–12686.

Wang, Y.; Wu, Q.; Zhang, G.; and Xu, D. 2024b. Learning 3D Geometry and Feature Consistent Gaussian Splatting for Object Removal. In *European Conference on Computer Vision*, 1–17. Springer.

Weder, S.; Garcia-Hernando, G.; Monszpart, A.; Pollefeys, M.; Brostow, G. J.; Firman, M.; and Vicente, S. 2023. Removing objects from neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16528–16538.

Wu, C.-H.; Chen, Y.-J.; Chen, Y.-H.; Lee, J.-Y.; Ke, B.-H.; Mu, C.-W. T.; Huang, Y.-C.; Lin, C.-Y.; Chen, M.-H.; Lin, Y.-Y.; et al. 2025. AuraFusion360: Augmented Unseen Region Alignment for Reference-based 360deg Unbounded Scene Inpainting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 16366–16376.

Ye, M.; Danelljan, M.; Yu, F.; and Ke, L. 2024. Gaussian grouping: Segment and edit anything in 3d scenes. In *European Conference on Computer Vision*, 162–179. Springer.

Zhang, B.; Fang, C.; Shrestha, R.; Liang, Y.; Long, X.; and Tan, P. 2024. Rade-gs: Rasterizing depth in gaussian splatting. *arXiv preprint arXiv:2406.01467*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. 586–595.