Transparent Object Reconstruction in 3D Gaussian Splatting with RGB-to-TIR Image Translation Diffusion Model

Eugene Baek, Hanjun Kim, Hyeongseok Suh
Seoul National University
{eugene0103, hanjun815, hyeongaa}@snu.ac.kr

Abstract

Recent advances in 3D scene reconstruction have addressed various challenges posed by transparent objects, which are commonly encountered in everyday life but present difficulties in image detection, segmentation, and 3D rendering due to their property of allowing light to pass through them. While innovative algorithms like 3D Gaussian Splatting (GS) have overcome several issues associated with Neural Radiance Fields (NeRF), accurately rendering transparent objects remained challenging. Some of our experiments confirmed that thermal infrared (TIR) images are more effective than RGB images in accurately recognizing the shapes of transparent objects in images. Based on this, we successfully developed an RGB-to-TIR module using a Stable Diffusion (SD) based model to convert RGB images into TIR images. Although we could not directly apply the diffusion module to 3D rendering due to pose-inconsistency issues in the dataset used for fine-tuning the diffusion module, we achieved higher accuracy in the 3D depth map when using a GAN-based sRGB-to-TIR model to convert RGB images into TIR images. This suggests that significant improvements in depth map accuracy can be achieved by improving the camera angle alignment in the RGB-TIR paired dataset used for diffusion fine-tuning.

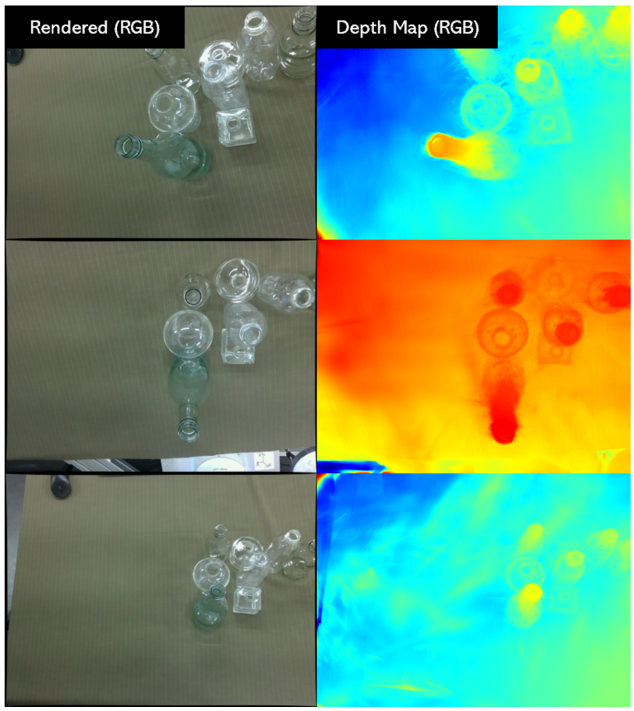


Figure 1. 3D GS with RGB inputs shows poor performance in 3D reconstruction for transparent objects.

1. Introduction

In recent years, the field of 3D scene reconstruction, typically performed with Neural Radiance Fields (NeRF) [28], has seen rapid advancements with the emergence of Gaussian Splatting (GS) [20]. Subsequent NeRF-based models have been capable of accurately generating complex scenes, but struggled with the drawback of being computationally costly in terms of generation speed [5, 9, 38]. GS distributes Gaussian kernels with color and transparency in 3D space, projecting them to create images. This method, both fast and flexible, addresses the shortcomings of traditional 3D rendering techniques [11, 20, 41].

However, 3D GS alone cannot accurately generate everything in the real world. Transparent objects, such as glass and plastics, are commonly encountered in everyday life, but 3D renderings fail to calculate the complex interactions between light and transparent surfaces due to their unique optical properties that allow light to pass through. Through experiments, we found that rendering transparent objects using multi-angle RGB images resulted in inaccurate volumes due to the scene beyond the transparent object being calculated as part of the object, or the edges of the transparent object not being captured accurately [43]. The experimental results are in Figure 1, and for the aforementioned reasons, the transparent objects appear as inaccurate

shapes in the depth map.

There have been various approaches to accurately recognize transparent objects in images [8, 16, 25, 26, 30, 42]. Among these, converting RGB images to TIR images has proven effective in robustly recognizing the volume and shape (edges) of objects. Although previous studies training Generative Adversarial Networks (GANs) [13] using contrastive learning for TIR image transformation, the training was unstable, requiring additional conditions to get reasonable results [23]. We defined this task as a style transfer problem and aimed to apply diffusion, which has recently surpassed GANs in performance in the image editing field, to this problem. We fine-tuned InstructPix2Pix [3] based on the Stable Diffusion (SD) [32], a diffusion model known for its stable training. While we successfully achieved RGB-to-TIR conversion in 2D, we failed to apply it to 3D rendering due to camera pose inconsistency issues in the training dataset. However, by confirming an improvement in the accuracy of the depth map when performing 3D rendering after TIR imaging using the GAN-based sRGB-to-TIR model [23], we verified the potential of improving the camera pose alignment in our diffusion module's fine-tuning dataset.

In this paper, we propose a novel approach using a diffusion-based module to enhance rendering of transparent objects by converting RGB images to TIR images. Through various experiments and evaluations, we demonstrate the success of our RGB-to-TIR diffusion module in 2D TIR imaging. We show that improvements to this module can overcome limitations of existing methods and enable more realistic rendering of transparent objects in 3D scenes.

The key contributions of our research can be summarized as follows:

1. We propose a method using an SD-based style transfer model to convert RGB images to TIR images.
2. We introduce a framework that augments the 3D GS pipeline, originally rendering sampled RGB images, with an RGB-to-TIR translation process. This framework enables accurate 3D rendering of transparent objects even from conventional RGB images captured with ordinary cameras.
3. Testing 3D rendering using TIR imaging via GAN-based models demonstrates the potential of our diffusion module's scalability.

2. Related Works

Detecting Transparent Objects. Numerous studies have been conducted on segmenting transparent objects such as mirrors or glass and reconstructing them in 3D from 2D images. Initially, various attempts were made to calculate depth or other parameters for glass segmentation in images [8, 16, 25, 26, 30, 42]. However, due to light reflection, even RGB-D data collected through depth cameras proved challenging to utilize for recognizing the surfaces of transparent objects. Transparent objects are amorphous in structure, lacking clear boundaries, making them difficult to identify using visible light. In contrast, when utilizing TIR (Thermal Infrared) imaging, these objects appear opaque in the long-wave infrared range, making it easier to distinguish them from the background. Consequently, the shapes of transparent objects can be accurately calculated [14, 16]. Based on this characteristic of TIR images, this paper aims to introduce methods for more accurately converting RGB images into TIR images.

In 3D reconstruction research, Dex-NeRF [17] leveraged additional light to improve NeRF learning by calculating whether the light is reflected, while SAID-NeRF [37] calculated the depth of hierarchically transparent objects using a mask generated by the Segment Anything Model (SAM) [22]. However, both of these methods are time-consuming, especially Dex-NeRF, which requires starting from data collection, leading to limited practical utility.

This paper aims to propose a general method for directly utilizing existing RGB datasets of transparent objects [6, 17, 21] for 3D reconstruction.

RGB-to-TIR Style Transfer. Previous GAN-based RGB-to-TIR image translation studies have effectively addressed the challenge of translating between two different domains. By leveraging the strengths of Bi-domain-based transformations, these methods have shown significant improvements in capturing the essential features of both RGB and TIR images. They demonstrate a strong ability to generalize across various scenarios and have successfully preserved the intricate details of the original RGB images, ensuring high-quality and accurate translations [23]. Building upon these works, employing a different base model could also lead to performance enhancements specifically in translating transparent objects RGB-to-TIR. Given that recent diffusion models have surpassed GANs in editing capabilities [10, 19, 40] and are being utilized in various tasks [3, 27, 33, 44], including style transfer, we anticipate that using a diffusion model could better address the RGB-to-TIR translation problem.

Diffusion [35] is a method for generating images using a diffusion process, which has garnered attention recently for surpassing the performance of GANs [13]. This model gradually alters pixel values to generate desired outputs. Specifically, SD [32] is known for executing this process reliably, minimizing issues like mode collapse during training and achieving high-quality image generation. InstructPix2Pix [3] is an image editing model based on SD that modifies or transforms images based on textual instructions. In this study, we chose this model to RGB-to-TIR style transfer, converting RGB images into TIR images. We also compared the results from GAN-based models, specifically CycleGAN [48] and sRGB-TIR [23], as baselines.

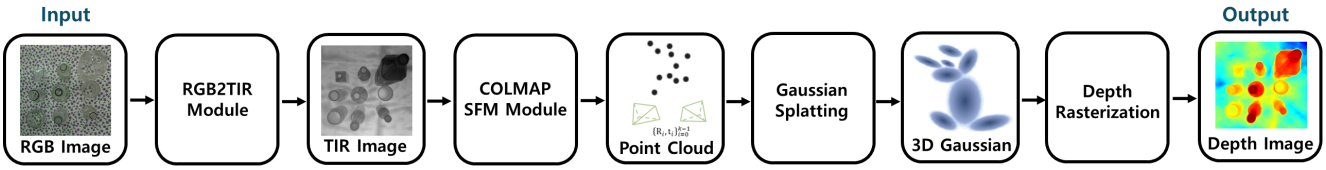


Figure 2. An overview of TIR-based 3D GS pipeline. Input RGB images of transparent objects are translated into TIR images through our RGB-to-TIR diffusion model (used sRGB-TIR model instead of diffusion). Then, 3D GS renders transparent objects with COLMAP SFM points from TIR images. We estimated depth maps by depth rasterization.

3D Representations. In 3D representations, methods utilizing neural radiance have been predominant. NeRF enables 3D rendering from 2D images using volumetric rendering based on position and direction. However, NeRF optimization is time-consuming, leading to practical limitations. Research efforts have focused on improving computational speed [5, 29, 38] or reducing input requirements [5, 9, 45]. The recent emergence of GS has significantly improved the speed and accuracy of traditional NeRF. GS has produced high-quality results in 3D reconstruction tasks within minutes, garnering significant interest in the generation field [11, 20].

Our research aims to extend the application of 3D GS to transparent object reconstruction, thus expanding existing 3D rendering techniques, which were previously limited to opaque objects, to encompass all real-world objects.

3. Method

3.1. Background

RGB-to-TIR Image Translation. Early approaches to RGB-to-TIR image translation heavily relied on image registration and fusion techniques. These methods typically involved aligning RGB and TIR images through geometric transformations and then merging the information from both modalities to create a composite image. Techniques such as wavelet transform [4], principal component analysis (PCA) [1], and gradient-based methods were used, but they often struggled with issues related to alignment accuracy and information loss. Additionally, handcrafted feature-based approaches like Scale-Invariant Feature Transform (SIFT) [24] and Speeded Up Robust Features (SURF) [2] were used to extract and match features between RGB and TIR images. However, these techniques were limited by the quality of the extracted features and the need for manual tuning.

The advent of CNNs brought innovation to the field of image translation. Early CNN-based models focused on learning the complex mappings between RGB and TIR images, demonstrating significant improvements in translation accuracy and robustness compared to traditional methods. GANs have played a particularly important role in this field, with the Pix2Pix [18] framework being widely adopted for RGB-to-TIR translation. CycleGAN addressed the challenge of scarce paired training data by using cycle consistency loss, allowing training on unpaired datasets. More recently, attention mechanisms have been integrated into CNN and GAN frameworks to further enhance translation quality. Attention modules help the network focus on salient regions of the image, improving the accuracy and detail of the translated images. These advancements have resulted in more accurate and visually appealing outcomes in RGB-to-TIR translation.

Text-guided Diffusion Models. Text-guided diffusion models [32] aim to map an arbitrary Gaussian noise vector $\mathbf{z}_T$ into an image $\mathbf{z}_0$ while aligning with a specific text condition $\mathbf{c}_T$, typically text embeddings derived from text encoders like CLIP [31]. This is achieved through a sequential denoising operation known as the reverse process. This process is driven by a noise prediction network ϵ_θ, which is optimized through loss functions:

$$L_{\text{simple}} = \mathbb{E}_{z_0,\epsilon\sim\mathcal{N}(0,I),t\sim U(1,T)}\|\epsilon - \epsilon_\theta(z_t, t, c_T)\|_2^2. \quad (1)$$

Here, ϵ_θ is abbreviated as $\epsilon_\theta(\mathbf{z}_t, c_T)$ by omitting the timestep condition for brevity in network output notation. Text-guided diffusion models typically incorporate text conditions during image generation using classifier-free guidance (CFG) [15]. CFG is represented as:

$$\tilde{\epsilon}_\theta(\mathbf{z}_t, c_T) = \epsilon_\theta(\mathbf{z}_t, \emptyset) + w \cdot (\epsilon_\theta(\mathbf{z}_t, c_T) - \epsilon_\theta(\mathbf{z}_t, \emptyset)). \quad (2)$$

Here, w denotes the text guidance scale controlling the influence of the text condition, and $\emptyset$ represents the null text embedding.

SFM Points by COLMAP. According to [20], to perform 3D Gaussian Splatting, SFM (Structure-from-Motion) points are required. SFM takes unordered images as input and outputs camera poses and 3D mapping points through correspondence search and incremental reconstruction [34]. In the correspondence search stage, the process begins with extracting keypoints from unordered images, typically using algorithms like SIFT to identify SURF in the images. Next, keypoints from multiple images are matched with those from other images to form pairs of related images. During this process, the RANSAC (Random Sample Consensus) [12] algorithm is employed for geometric verification, ensuring the selection of correct image pairs. These

matched image pairs then serve as crucial input data for the subsequent reconstruction stage. In the incremental reconstruction stage, the 3D structure is progressively reconstructed using a scene graph. In the initialization phase, two images are selected to form a basic 3D structure. Subsequently, the PnP (Perspective-n-Point) [12] algorithm estimates the pose of new images and integrates them into the existing structure. Through this process, the poses and 3D points for all images are gradually completed. Finally, the Bundle Adjustment (BA) [36] technique optimizes the estimated points and poses. BA operates by simultaneously adjusting all points and camera poses to minimize errors.

COLMAP is a prominent software that implements these SFM techniques. Given unordered images as input, COLMAP computes SFM points, camera intrinsic parameters, and camera poses through correspondence search and incremental reconstruction stages. Since our dataset only consists of images without poses, we integrated COLMAP into the pre-processing pipeline of 3D GS to directly estimate poses.

3D Gaussian Splatting. The algorithm described in [20] employs a 3D Gaussian-based model for real-time rendering. Utilizing 3D Gaussians to represent 3D scenes as point clouds was chosen due to its differentiability and ease of 2D projection, as highlighted in [20]. A 3D Gaussian is defined by its center $\mathbf{x}$ and covariance matrix Σ within the context of a point cloud:

$$G(x) = e^{-\frac{1}{2}(x)^\top \Sigma^{-1}(x)} \tag{3}$$

The covariance matrix Σ can be decomposed into a scale matrix S and a rotation matrix R:

$$\Sigma = RSS^\top R^\top \tag{4}$$

Here, S represents the scale of the 3D Gaussian, and R denotes the rotation. To render, it is necessary to project the 3D Gaussian into 2D. This process is achieved using the viewing transformation matrix W and the Jacobian matrix J approximating the affine projection. This yields the 2D covariance matrix (denoted as Σ').

$$\Sigma' = JW\Sigma W^\top J^\top \tag{5}$$

Here, J is the Jacobian matrix used for the 2D projection of the 3D Gaussian, defined by a combination of viewing transformation and projection transformation.

Utilizing this approach, Gaussian Splatting in [20] is broadly divided into four stages: Gaussian initialization for SFM points, 2D projection of 3D Gaussians, rendering using 2D Gaussians, and optimization. These processes can be executed in real-time, enabling fast and efficient rendering of high-resolution 3D scenes. Particularly, GS techniques yield smoother results compared to traditional point cloud rendering methods and operate efficiently even in complex scenes.

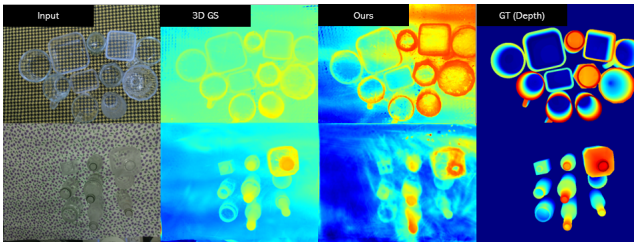


Figure 3. TIR images are more helpful than RGB images for 3D GS. These results show that our method is useful for accurate 3D rendering of transparent objects.

3.2. Overview

Our proposed method aims to significantly enhance the rendering of transparent objects within 3D environments. It comprises two key components: the RGB-to-TIR diffusion module and Depth Rasterization with 3D Gaussian Splatting module. In the RGB-to-TIR diffusion module, training utilizes two types of datasets: input RGB images and corresponding TIR images. To construct a dataset of transparent objects, we employ datasets as outlined in [21]. These datasets contain RGB and TIR image pairs, alongside pose and extrinsic data, which are leveraged in the 3D GS module.

The foundation of our diffusion model is rooted in the work presented in [32]. This work places a distinct emphasis on augmenting perceptual aspects, thereby yielding a stable diffusion model. This model can be refined in various ways through the use of the refine module. To adapt this model to our specific purpose, we employ an upgraded version, denoted as [3]. This advanced model, an evolution of the concepts introduced in [32], allows for training with paired images and facilitates reference text-guided image manipulation. Paired images, consisting of RGB and TIR image pairs, are utilized in conjunction with this model to seamlessly convert RGB images into TIR representations.

In the Depth Rasterization with 3D GS stage, TIR (Thermal Infrared) images generated by our diffusion model serve as inputs. This stage builds upon the principles outlined in [20], which introduces real-time, high-quality radiance field rendering. With an adequate set of TIR images generated across various poses, the SFM (Structure-from-Motion) points are calculated, and the 3D GS algorithm creates depth maps of transparent objects. Because TIR images are not in color, this paper focuses solely on the depth maps. The output of the entire process is depicted in Figure 2.

3.3. RGB-to-TIR

In our study, we used the InstructPix2Pix model based on Stable Diffusion to convert RGB images to TIR images. InstructPix2Pix originally generates training datasets using fine-tuned GPT-3 and Stable Diffusion. However, in our ex-

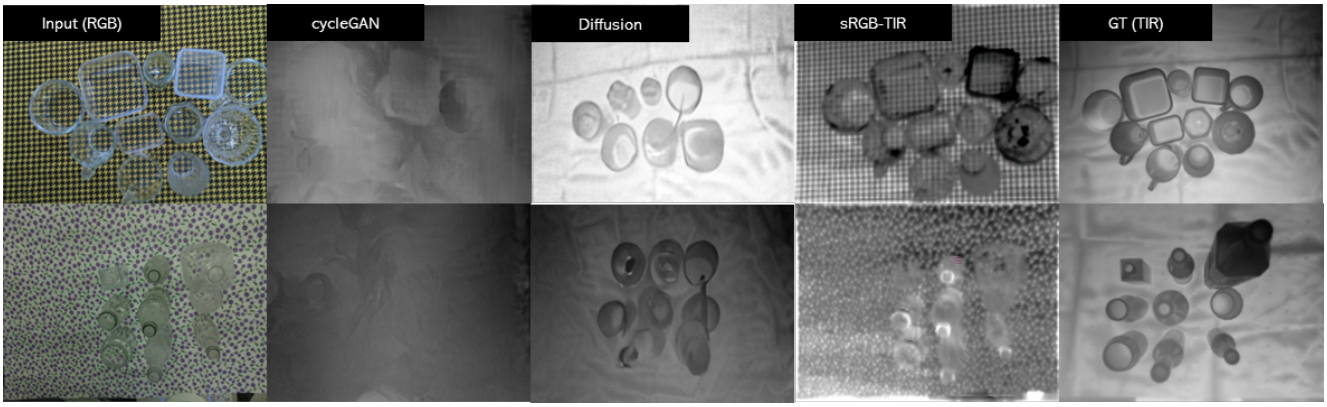


Figure 4. TIR images translated from RGB inputs with several image style transfer models. Although it seems that Diffusion-based InstructPix2Pix made a high quality TIR images, the shape of the objects are distorted. Among these, sRGB-TIR model performed well.

Method	RMSE (↓)		MAE (↓)	
Testset	Test_1	Test_11	Test_1	Test_11
3D GS	0.7038	0.4426	0.6129	0.3248
Ours	**0.5321**	**0.3761**	**0.4227**	**0.2738**

Table 1. Quantitative evaluation of depth map after 3D GS. Ours showed improved performance compared to original 3D GS.

Method	RMSE (↓)		MAE (↓)	
Testset	Test_1	Test_11	Test_1	Test_11
CycleGAN	0.5970	0.6675	0.4858	0.5733
InstructPix2Pix	0.5956	0.6420	0.4773	0.5438
sRGB-TIR	0.5321	0.3761	0.4227	0.2738

Table 2. Quantitative evaluation of translated TIR images. Fine-tuned InstructPix2Pix achieved higher accuracy in TIR image transformation compared to using the image editing model as is.

periments, we skipped the dataset generation phase because we already had prepared RGB-TIR image pairs, and fine-tuned the model using these prepared datasets.

InstructPix2Pix is a model that edits images based on specific textual instructions. It takes an input image and a text description to perform the desired transformation. For instance, if the text description is "change a specific part of this image to blue," the model identifies the specified part in the input image and changes it to blue. This model has a high degree of flexibility and can be applied to various image editing tasks.

Stable Diffusion is a model that plays a crucial role in image generation and transformation by progressively removing noise from a noisy image to produce a high-resolution image. This process is highly precise, allowing for the reproduction of fine details with high quality. InstructPix2Pix utilizes the principles of Stable Diffusion to transform specific parts of an image based on textual instructions.

In our experiment, the model was trained to take RGB images as input and generate corresponding TIR images as output. This process involves not just changing the style of the image but also maintaining the form and detailed information of objects while converting them into TIR images.

The dataset we used consisted of RGB-TIR image pairs captured in various environments. This dataset included various types of transparent objects, enabling the model to generate accurate TIR images in different situations. Through these image pairs, the model learned the characteristics of both RGB and TIR images, allowing it to generate corresponding TIR images when new RGB images were provided.

The training process of the InstructPix2Pix model was as follows: first, the model was trained using RGB-TIR image pairs as input. During this training, the following loss function was used to accurately learn the transformation between RGB and TIR images:

$$L = \mathbb{E}_{\mathcal{E}(x),\mathcal{E}(c_I),c_T,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon\theta(z_t, t, \mathcal{E}(c_I), c_T)\|_2^2\right] \quad (6)$$

where $\epsilon * \theta$ is the predicted noise, z_t is the noisy image at time t, $E(c_I)$ is the embedding of the input image c_I, c_T is the textual instruction, and ϵ is the noise drawn from a standard normal distribution. Through this loss function, the model was able to precisely learn the process of transforming RGB images to TIR images.

Once training was complete, the InstructPix2Pix model could take a new RGB image as input and perform the task of converting it into a TIR image. However, transformed TIR image failed to reproduce the shapes and details of various objects, including transparent ones. So, we used the same model with [23] for our RGB-to-TIR translation.

Method	SSIM (↑)		PSNR (↑)		LPIPS (↓)	
Testset	Test_1	Test_11	Test_1	Test_11	Test_1	Test_11
RGB	0.8371	0.9124	22.8956	26.1824	0.1209	0.1125
TIR	**0.9307**	**0.9691**	**34.5199**	**31.4292**	**0.2865**	**0.222**

Table 3. Quantitative evaluation of our method compared to previous works, computed over the TRansPose dataset.

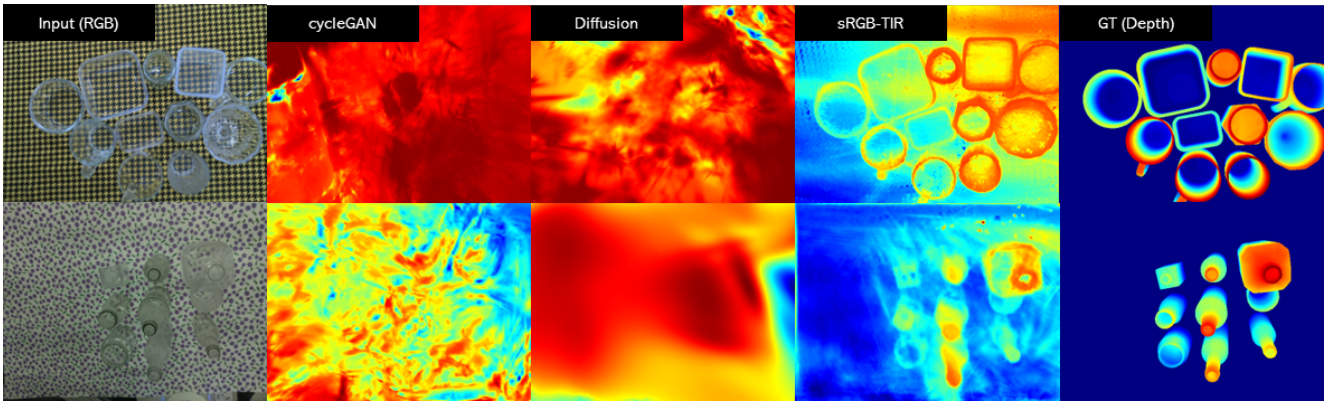


Figure 5. Depth map from depth rasterization with GS after translating TIR images with several image manipulation models.

3.4. Depth Rasterization with 3D GS

The overall process is illustrated in Figure 2. At this stage, we use the TIR images transformed by the previous diffusion module. First, we utilize COLMAP to compute SFM points, camera poses, and camera intrinsics. Using this information, we perform 3D Gaussian Splatting (GS), optimize the Gaussian splats, and render the depth map.

COLMAP SFM Module. Using a diverse set of images $\{I_i\}_{i=0}^{k-1}$, $I_i \in [0,1]^{H\times W\times 3}$, we run COLMAP. Based on [34], COLMAP provides the camera poses $R_i \in \mathbb{R}^{3\times 3}$ and $t_i \in \mathbb{R}^3$, and calculates the camera intrinsic parameters $K_i \in \mathbb{R}^{3\times 3}$ and the SFM points $P \in \mathbb{R}^{n\times 3}$. This information plays a crucial role in rendering the depth map in 3D GS.

Depth Map with 3D Gaussian. Using the SFM points of the TIR images calculated by the COLMAP SFM Module, we perform GS as proposed by [20]. Since TIR images lack color information, we perform depth rasterization in 3D GS as suggested by [7] instead of color rendering. First, initialize the SFM points calculated from the TIR images. According to [7], the information obtained through GS is used to render the depth map. Following the depth implementation method of NeRF, the equation is as follows:

$$D = \sum_{i\in N} d_i \alpha_i T_i, \tag{7}$$

Here, D denotes the rendered depth, and d represents the depth of each splat emitted from each camera. a represents the learned opacity multiplied by the covariance of the 2D Gaussian. Using the formulation from [7], depth maps of transparent objects can be rasterized using the proposed 3D Gaussian splatting method. Based on [7], the rendering process for transparent object depth maps can be outlined into four main stages:

1. Initialization: Initialize Gaussian splats based on SFM points with TIR images retrieved from 3.3. This involves setting up splats in the form similar to Equation 3, incorporating positions and covariances of points.
2. Projection: Project 3D Gaussians onto 2D, computing 2D covariance matrices as described in Equation 5.
3. Differentiable Tile Rasterizer: Splat the projected 2D Gaussian splats onto the image plane with the Equation 7.
4. Optimization: Optimize the rendered image. Adjust Gaussian positions and covariances to generate a more accurate depth map.

These four stages collectively enable the real-time rendering of depth maps for transparent objects using the 3D GS.

4. Experiment

In this section, we compared with the existing RGB image based 3D reconstruction method to verify the rendering performance of our model. Next, to verify the image translation performance of our RGB-to-TIR diffusion model, we compared it with the existing model that converts RGB images into TIR images.

Experiments were conducted on two datasets for verification. The first dataset is Transpose Datasets [21], and the

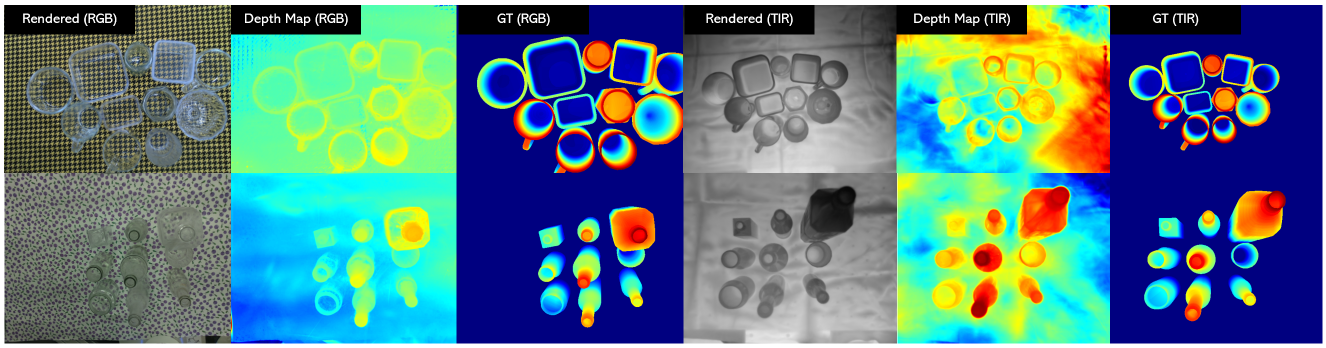


Figure 6. Comparisons in depth map after 3D GS from RGB images and TIR images.

second dataset is Datasets for Dex-NeRF [17].

4.1. Training Details

Dataset. The Transpose Datasets is the first large-scale multi-spectrum dataset that combines stereo RGB-D, TIR images, and object poses to facilitate research on transparent objects. The dataset includes 99 transparent objects, comprising 43 household items, 27 recyclable items, 29 chemical laboratory equivalents, and 12 non-transparent objects. The dataset is organized into seq_test, seq_all, seq_C, seq_H, and seq_T. For our experiments, we used the seq_test_01 and seq_test_11 sequences. In each sequence, RGB images were captured using the cam_L camera, while TIR images were captured using the 8-bit cam_T camera.

Setup.

4.2. Depth Estimation

We compared MAE (Equation 8) and RMSE (Equation 9) to evaluate the performance of 3DGS and 3D GS with our RGB-to-TIR translation module. Here, $i \in [0, \ldots, N]$ denotes the frame number, r represents the pixel position, and Ω_r is the set of all pixel positions across frames. $\hat{D}(r)$ denotes the inferred depth in meters, while $D(r)$ represents the ground truth depth in meters.

$$\text{MAE} = \frac{1}{n} \sum_{(i,r)\in\Omega_r} |\hat{D}_i(r) - D_i(r)|_1, \tag{8}$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{(i,r)\in\Omega_r} ||\hat{D}_i(r) - D_i(r)||^2} \tag{9}$$

In these equations, $|\omega_r|$ represents the cardinality of the set ω_r, i.e., the total number of pixel positions considered.

Table 1 presents the RMSE and MAE computed for the results obtained using the conventional 3D GS and our model for the test 1 and test 11 sequences of the Transpose Datasets. For RMSE, our model shows 0.1717 lower error compared to the conventional 3D GS in the test 1 sequence and 0.0665 lower error in the test 11 sequence. For MAE, our model exhibits 0.1902 lower error in the test 1 sequence and 0.051 lower error in the test 11 sequence compared to the conventional 3D GS.

Figure 3 compares the depth maps obtained using the conventional 3D GS and our model with the ground truth (GT) for the test1 and test11 sequences of the Transpose Datasets. The conventional 3D GS uses RGB input images directly, whereas our model utilizes TIR-transformed images. As evident from the figure, the depth map results from our model exhibit closer resemblance to the GT compared to those obtained from the conventional 3D GS.

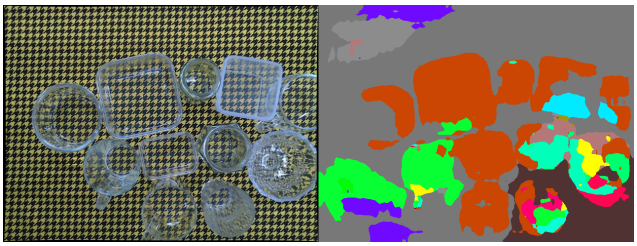
5. Ablations

5.1. RGB-to-TIR

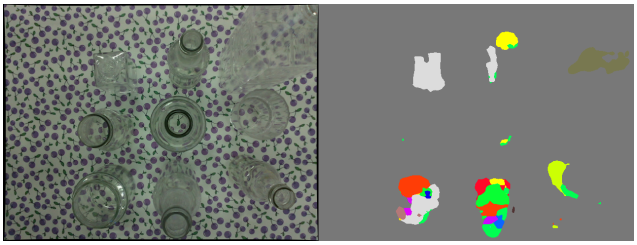
First, we compared the transformation results of various models that convert RGB images to TIR images to evaluate the RGB2TIR image transformation module of our 3D GS model. Table 2 presents the RMSE and MAE computed for the results obtained using CycleGAN, InstructPix2Pix, and sRGB-TIR to generate depth maps for the test 1 and test 11 sequences of the Transpose Datasets. We observed sRGB-TIR achieved highest performance.

Through the results of TIR transformation in Figure 4, we compared the results of converting RGB images to TIR images using CycleGAN, InstructPix2Pix, and sRGB-TIR for the test1 and test 11 sequences of the Transpose Datasets. For CycleGAN, we observed lower quality and significant disappearance of shapes of transparent objects. In the case of InstructPix2Pix, while style transformation to TIR images was effective, there were distortions in object shapes and instances of object creation or disappearance. Conversely, sRGB-TIR maintained object shapes intact compared to InstructPix2Pix, despite less accurate style transformation to TIR images. During this process, we identified issues in the camera poses of the RGB-TIR pair dataset used for finetuning InstructPix2Pix, which we acknowledge as a limitation to be addressed in future work.

Figure 5 compares the results of depth rasterization using



(a) Segmentation map of test 1 sequence



(b) Segmentation map of test 11 sequence

Figure 7. Segmenting transparent objects through existing segmentation model fails.

GS after converting RGB images to TIR images using CycleGAN, InstructPix2Pix, and sRGB-TIR for the test1 and test11 sequences of the Transpose Datasets. CycleGAN and InstructPix2Pix failed in 3D rendering due to shape distortions during the RGB-to-TIR image conversion, leading to inconsistency across frames. In contrast, sRGB-TIR successfully maintains object shapes while transforming the image style to TIR, enabling problem-free 3D rendering and depth image generation for transparent objects.

5.2. 3D reconstruction

Table 3 presents the evaluation results of SSIM [39], PSNR (Peak Signal-to-Noise Ratio), and LPIPS [46] for 3D reconstruction using RGB images and TIR images on the test 1 and test 11 sequences of the Transpose Datasets. Comparing the SSIM results, TIR images show 0.0936 higher SSIM for the test 1 sequence and 0.0567 higher SSIM for the test 11 sequence compared to RGB images. Comparing the PSNR results, TIR images exhibit 11.6243 higher PSNR for the test 1 sequence and 5.2468 higher PSNR for the test 11 sequence compared to RGB images. Lastly, LPIPS scores show that TIR images achieve 0.1656 higher for the test 1 sequence and 0.1095 higher for the test 11 sequence compared to RGB images.

Figure 6 compares the results of obtaining depth maps using RGB images and TIR images for the test 1 and test 11 sequences of the Transpose Datasets. The depth ground-truth of the Transpose Datasets is obtained by rendering object parts using CAD models, hence depth information for the background is not provided. We treated the depth for the background as the maximum depth value.

In the test 1 and test 11 sequences, using RGB images resulted in lower quality depth maps, whereas using TIR images showed improved results compared to RGB images. This indicates that RGB-to-TIR translation is effective for 3D rendering of transparent objects.

5.3. Segmentation

We also attempted to recognize transparent objects using segmentation instead of diffusion-based style transfer. Figure 7 shows the results of segmenting RGB images from test sequences 1 and 11 using a model pre-trained on the ADE20K dataset [47]. Despite attempting to use the segmentation maps for 3D GS, the objects were transparent, causing the model to fail in distinguishing between background and objects, resulting in unsuccessful semantic segmentation. Consequently, it demonstrated lower performance compared to RGB-to-TIR style transfer.

6. Conclusion

Transparency of objects represents a significant challenge in computer graphics. We address the issue that the 3D GS model struggles to render transparent objects accurately, aiming to resolve it by converting RGB images to TIR images for precise depth map generation.

When training RGB-to-TIR style transfer using diffusion-based InstructPix2Pix, it appeared effective in transforming images into TIR format, yet maintaining the shape of objects proved difficult, rendering it unsuitable for 3D GS. We concluded that using the sRGB-TIR model, a GAN-based approach to convert RGB images to TIR, improved depth map performance compared to rendering with RGB images alone and anticipate that improving camera pose alignment in our diffusion module's training dataset will enable accurate rendering of transparent objects. This points to future research directions and emphasizes the importance of RGB-to-TIR image transformation technology in achieving precise visual representation of transparent objects in various graphics applications.

In summary, our research aims for accurate 3D depth estimation of transparent objects, demonstrating the crucial role of RGB-to-TIR image translation techniques. Furthermore, future researches to develop better computer graphics techniques capable of handling complex inter-reflections and transparency of objects with greater accuracy.

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010. 3

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 404–417. Springer, 2006. 3

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 2, 4

[4] C Sidney Burrus, Ramesh A Gopinath, and Haitao Guo. Wavelets and wavelet transforms. *rice university, houston edition*, 98, 1998. 3

[5] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14124–14133, 2021. 1, 3

[6] Xiaotong Chen, Huijie Zhang, Zeren Yu, Anthony Opipari, and Odest Chadwicke Jenkins. Clearpose: Large-scale transparent object dataset and benchmark. In *European conference on computer vision*, pages 381–396. Springer, 2022. 2

[7] Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. Depth-regularized optimization for 3d gaussian splatting in few-shot images. *arXiv preprint arXiv:2311.13398*, 2023. 6

[8] Alex Costanzino, Pierluigi Zama Ramirez, Matteo Poggi, Fabio Tosi, Stefano Mattoccia, and Luigi Di Stefano. Learning depth estimation for transparent and mirror surfaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9244–9255, 2023. 2

[9] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 1, 3

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2

[11] Jiemin Fang, Junjie Wang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. *arXiv preprint arXiv:2311.16037*, 2023. 1, 3

[12] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3, 4

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[14] S Henke, Detlef Karstädt, Klaus-Peter Möllmann, Frank Pinno, and Michael Vollmer. Identification and suppression of thermal reflections in infrared thermal imaging. *Inframation Proc*, 5:287–98, 2004. 2

[15] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3

[16] Dong Huo, Jian Wang, Yiming Qian, and Yee-Hong Yang. Glass segmentation with rgb-thermal image pairs. *IEEE Transactions on Image Processing*, 32:1911–1926, 2023. 2

[17] Jeffrey Ichnowski, Yahav Avigal, Justin Kerr, and Ken Goldberg. Dex-nerf: Using a neural radiance field to grasp transparent objects. *arXiv preprint arXiv:2110.14217*, 2021. 2, 7

[18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 3

[19] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 2

[20] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. 1, 3, 4, 6

[21] Jeongyun Kim, Myung-Hwan Jeon, Sangwoo Jung, Wooseong Yang, Minwoo Jung, Jaeho Shin, and Ayoung Kim. Transpose: Large-scale multispectral dataset for transparent object. *The International Journal of Robotics Research*, page 02783649231213117, 2023. 2, 4, 6

[22] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2

[23] Dong-Guw Lee, Myung-Hwan Jeon, Younggun Cho, and Ayoung Kim. Edge-guided multi-domain rgb-to-tir image translation for training vision tasks with challenging labels. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8291–8298. IEEE, 2023. 2, 5

[24] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 3

[25] Haiyang Mei, Bo Dong, Wen Dong, Jiaxi Yang, Seung-Hwan Baek, Felix Heide, Pieter Peers, Xiaopeng Wei, and Xin Yang. Glass segmentation using intensity and spectral polarization cues. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 2

[26] Haiyang Mei, Xin Yang, Yang Wang, Yuanyuan Liu, Shengfeng He, Qiang Zhang, Xiaopeng Wei, and Rynson W.H. Lau. Don't hit me! glass detection in real-world scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[27] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 2

[28] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[29] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 3

[30] Gautham Narasimhan, Kai Zhang, Ben Eisner, Xingyu Lin, and David Held. Self-supervised transparent liquid segmen-

tation for robotic pouring. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4555–4561. IEEE, 2022. 2

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3, 4

[33] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 2

[34] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 3, 6

[35] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 2

[36] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*, pages 298–372. Springer, 2000. 4

[37] Avinash Ummadisingu, Jongkeum Choi, Koki Yamane, Shimpei Masuda, Naoki Fukaya, and Kuniyuki Takahashi. Said-nerf: Segmentation-aided nerf for depth completion of transparent objects. *arXiv preprint arXiv:2403.19607*, 2024. 2

[38] Huan Wang, Jian Ren, Zeng Huang, Kyle Olszewski, Menglei Chai, Yun Fu, and Sergey Tulyakov. R2l: Distilling neural radiance field to neural light field for efficient novel view synthesis. In *ECCV*, 2022. 1, 3

[39] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 8

[40] Chen Henry Wu and Fernando De la Torre. A latent space of stochastic diffusion models for zero-shot image editing and guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7378–7387, 2023. 2

[41] Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Adrian Prisacariu. Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing. *arXiv preprint arXiv:2403.08733*, 2024. 1

[42] Enze Xie, Wenjia Wang, Wenhai Wang, Mingyu Ding, Chunhua Shen, and Ping Luo. Segmenting transparent objects in the wild. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 696–711. Springer, 2020. 2

[43] Enze Xie, Wenjia Wang, Wenhai Wang, Peize Sun, Hang Xu, Ding Liang, and Ping Luo. Segmenting transparent object in the wild with transformer. *arXiv preprint arXiv:2101.08461*, 2021. 1

[44] Serin Yang, Hyunmin Hwang, and Jong Chul Ye. Zero-shot contrastive loss for text-guided diffusion image style transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22873–22882, 2023. 2

[45] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 3

[46] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 8

[47] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 8

[48] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2