VOLUMETRIC DISENTANGLEMENT FOR 3D SCENE MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, advances in differential volumetric rendering enabled significant breakthroughs in the photo-realistic and fine-detailed reconstruction of complex 3D scenes, which is key for many virtual reality applications. However, in the context of augmented reality, one may also wish to effect semantic manipulations or augmentations of objects within a scene. To this end, we propose a volumetric framework for (i) disentangling or separating, the volumetric representation of a given foreground object from the background, and (ii) semantically manipulating the foreground object, as well as the background. Our framework takes as input a set of 2D masks specifying the desired foreground object for training views, together with the associated 2D views and poses, and produces a foreground-background disentanglement that respects the surrounding illumination, reflections, and partial occlusions, which can be applied to both training and novel views. Unlike previous work, our method does not rely on 3D information in the form of 3D object bounding boxes or a scene voxel grid. It correctly captures reflective foreground objects, objects occluded by the background, and objects with noisy and inaccurate masks. Our method enables the separate control of pixel color and depth as well as 3D similarity transformations of both the foreground and background objects. We subsequently demonstrate our framework's applicability on several downstream manipulation tasks, going beyond the placement and movement of foreground objects. These tasks include object camouflage, non-negative 3D object inpainting, 3D object translation, 3D object inpainting, and 3D text-based object manipulation.

1 INTRODUCTION

The ability to interact with a 3D environment is of fundamental importance for many augmented reality (AR) application domains such as interactive visualization, entertainment, games, and robotics Mekni & Lemieux (2014). Such interactions are often semantic in nature, capturing specified entities in a 3D scene and manipulating them accordingly. To this end, we propose a novel framework for the disentanglement and manipulation of objects in a 3D scene. Given a small set of 2D masks delineating the desired foreground object together with the associated 2D views and poses, and no other 3D information, our method produces a volumetric representation of both the foreground object and the background. Our volumetric representation enables separate control of pixel color and depth, as well as scale, rotation, and translation of the foreground object and the background. Using this disentangled representation, we demonstrate a suite of downstream manipulation tasks involving both the foreground and background volumes, going beyond previous work, and including 3D camouflage and 3D semantic text-based manipulation. Fig. 1 illustrates our proposed volumetric disentanglement and a sampling of the downstream volumetric manipulations that this disentanglement enables. We note that while the *foreground/background* terminology is useful for painting a mental picture, we wish to emphasize that the disentanglement is not limited to foreground objects, and works equally well for objects positioned further back (and partially occluded).

Neural Radiance Fields (NeRF) Mildenhall et al. (2020) delivered a significant breakthrough in the ability to reconstruct complex 3D scenes with high fidelity and a high level of detail. However, NeRF has no control over individual semantic objects within a scene. To this end, ObjectNeRF Yang et al. (2021) proposed to represent foreground objects by rendering rays with masked regions. While ObjectNeRF learns foreground object representation independently from the background, our method

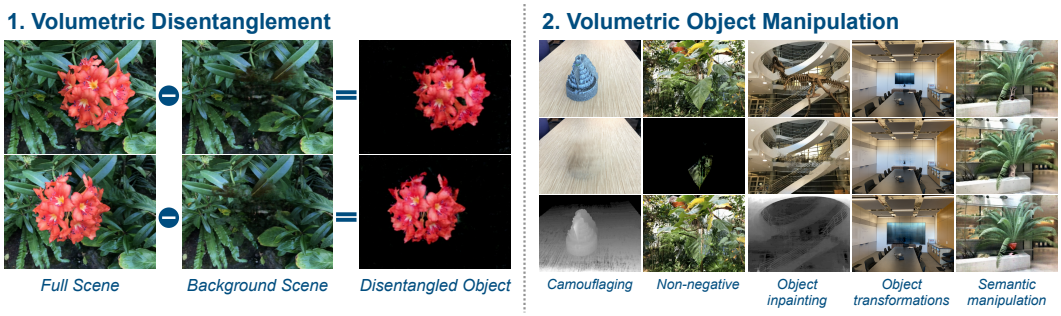


Figure 1: **Volumetric disentanglement framework.** We introduce a framework for the volumetric disentanglement of foreground objects as well as the background from a full scene (1). Our volumetric disentanglement can then be used for many downstream tasks of interest to designers and artists in AR applications (2), including 3D object camouflage, non-negative 3D inpainting, 3D object inpainting, 3D object transformation, and 3D text-based semantic manipulation.

instead disentangles the foreground from the background using a volumetric composition. In particular, the foreground object is extracted using a volumetric "subtraction" of the background from the full scene. In doing so, our method correctly captures reflective objects and those occluded by the background, as well as objects with noisy and inaccurate masks. Further, unlike our method, ObjectNeRF requires additional 3D information in the form of 3D bounding boxes to render the background and edit objects at test time and relies on an accurate estimation of depth for training.

Given a set of 2D training views and poses of a scene, as well as masks, specifying the foreground object, our method first trains a neural radiance field to reconstruct the background and its associated effects, following a similar procedure to NeRF Mildenhall et al. (2020). Due to the prior induced through volumetric rendering, the resulting neural field captures the background volume that also includes objects appearing behind or occluding the foreground object, and captures associated effects such as illumination and reflections. By training a neural radiance field to reconstruct the volume of the entire 3D scene and the volume of the background separately, the representation of the foreground can be computed in a compositional manner from the two volumes Drebin et al. (1988) as illustrated in Fig. 1, without using any other 3D information. We note that the background and foreground can be rendered from both training and novel views.

Having disentangled the foreground object from the rest of the 3D scene, we can now perform a range of downstream tasks, going beyond the placement and movement of objects, as shown in Yang et al. (2021). For example, optical-see-through devices can only add light to the scene, meaning that the generation must be non-negative with respect to the input scene Luo et al. (2021). In other cases, one may wish to keep the depth of the original scene intact Owens et al. (2014); Guo et al. (2022), and only modify the textures or colors of objects. Our framework enables properties such as color, depth, and affine transformations of both the foreground object and background to be manipulated separately, and therefore can handle such manipulation tasks.

Lastly, we consider the ability to affect semantic manipulations to the foreground. To this end, we consider the recently proposed multi-modal embedding of CLIP Radford et al. (2021). Using CLIP, we are able to manipulate the foreground object semantically using text. Recent work such as Michel et al. (2021); Wang et al. (2021a); Sanghi et al. (2021b) considered the ability to manipulate 3D scenes semantically using text. We demonstrate a similar capability, but one which transcends to individual objects in our 3D scene, while adhering to the semantics of the background. We also note that while 2D counterparts may exist for each of the proposed manipulations, our disentangled volumetric manipulation offers 3D-consistent semantic manipulation of foreground objects.

2 Related Work

3D Disentanglement We focus on the disentanglement of semantic and geometric properties in 3D scenes. For a more comprehensive overview, see Ahmed et al. (2018). CLIP-NeRF Wang et al. (2021a) disentangle the shape and appearance of NeRF Mildenhall et al. (2020) and, subsequently, uses CLIP Radford et al. (2021) to manipulate these properties. Other works disentangle pose Wang

et al. (2021b); Yen-Chen et al. (2021), illumination Srinivasan et al. (2021); Boss et al. (2021), texture and shape Liu et al. (2021); Jang & de Agapito (2021); Deng et al. (2020); Noguchi et al. (2021). These works are limited to an entire volumetric scene or object but not to objects within a scene. Further, they are limited to specific categories on constrained domains (e.g human parts).

Another line of work considers the disentanglement of objects in a full 3D scene. Niemeyer & Geiger (2021); Nguyen-Phuoc et al. (2020) consider the generation of scenes in a compositional manner. In contrast, we disentangle an *existing* scene into the foreground and background volumes, while they generate such volumes from scratch. A subsequent line of works considers the disentanglement of objects in an existing scene. Several representations can be used to learn 3D scenes such as point clouds Shu et al. (2019); Yang et al. (2019); Hui et al. (2020); Achlioptas et al. (2017), meshes Hanocka et al. (2019); Groueix et al. (2018); Wang et al. (2018); Pan et al. (2019), or voxels Riegler et al. (2017); Xie et al. (2019); Wu et al. (2016); Brock et al. (2016). However, work using these representations for disentanglement Broadhurst et al. (2001); Kutulakos & Seitz (2000) are typically restricted in topology or resolution or make strong assumptions about scenes.

Recently, a number of methods proposed to use neural fields (NeRFs) to represent individual objects in the scene. Guo et al. (2020) use an object library and learn a per object scattering field which can then be composed together to represent a scene where the object's movement, lighting, and reflection can be controlled. Our method instead decomposes an existing scene into the foreground and background objects, capturing their relations, and subsequently allowing for object-specific edits. Ost et al. (2021) use a scene graph representation to decompose dynamic objects, but rely on a dynamic scene as input, and are restricted to only one class of objects with similar shapes. Fu et al. (2022); Kundu et al. (2022) consider specific types of semantic categories, for instance by considering a specialized domain s.a traffic scenes. Unlike these works, our work is not limited to the type of editable objects in the scene and enables a wider variety of manipulations including 3D object camouflage and 3D semantic manipulation of individual objects in a scene. Recently, and concurrently to our work, Kobayashi et al. (2022) proposed a disentanglement framework for neural fields using text or image patches. While it enables the disentanglement of coarse concepts based on text or image patch, it does not allow for the fine-grained control which a mask can provide in selecting the object to be disentangled.

Perhaps most similar to our work is ObjectNeRF Yang et al. (2021). ObjectNeRF uses an object branch to render rays with masked regions for foreground objects. At test time, it uses 3D bounding boxes of individual objects to edit their movement and placement. Similarly to ObjectNeRF, our method inherits NeRF's ability to produce novel views for both foreground and background objects. However, our method differs from ObjectNeRF in multiple ways: (i). Our method requires input 2D segmentation masks for input training views and does not require 3D bounding boxes for editing foreground objects. Similarly, no 3D structure in the form of a voxel grid is required during training. (ii). Unlike ObjectNeRF, our method correctly captures objects with noisy and inaccurate masks as well as reflective objects and those occluded by the background. (iii). Our method relies on ground truth RGB images for existing views for our loss objectives, and does not require an occlusion loss which requires an accurate estimate of the scene's depth of existing and novel views. (iv). Lastly, our method goes beyond the editing of objects' movement and placement and enables zero-shot manipulations (does not require any 3D or 2D training data) such as 3D object camouflage, and 3D text-based semantic manipulation of individual objects.

3D Manipulation Our framework enables the manipulation of localized regions in a scene. While 2D counterparts, such as 2D inpainting approaches exist Guillemot & Le Meur (2013); Yu et al. (2019); Efros & Leung (1999); Efros & Freeman (2001), they cannot generate 3D consistent manipulations. One set of approaches consider editing the entire scene. Canfes et al. (2022) considers texture and shape manipulation of 3D meshes. CLIP-Forge Sanghi et al. (2021a) generates objects matching a text prompt using CLIP embeddings. Text2Mesh Michel et al. (2021) manipulate the texture or style of an object. DreamFields Jain et al. (2021a) learn a neural radiance field representing 3D objects from scratch. Unlike these works, our work is concerned with manipulating a local region in an existing scene. Jang & de Agapito (2021) and Liu et al. (2021) modify the shape and color code of objects using coarse 2D user scribbles, but require a curated dataset of objects under different colors and views, and are limited to synthetic objects. In contrast, our method enables the manipulation of objects in complex scenes, semantically, according to a target text prompt.

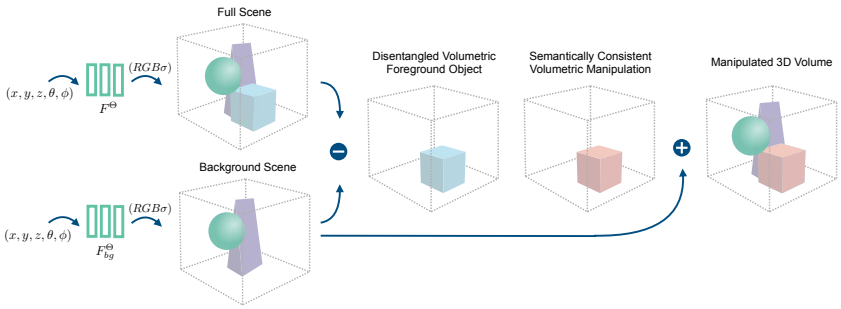


Figure 2: **Overview of our disentanglement framework.** First, we learn a volumetric representation of the background and full scene (Sec. 3.1). Second, by subtracting the full and the background volumes, we obtain a disentangled foreground volume. Third, we perform a wide range of manipulations on this volume, which adhere to the background volume. This is illustrated here by changing the color of the cube from blue to red. Finally, we can place the foreground object back into the original scene by adding it volumetrically to the background scene, obtaining a manipulated scene.

3 METHOD

Given a 3D scene, we wish to disentangle semantic objects from the rest of the scene. First, we describe the 3D volumetric representation used to disentangle objects and control objects separately (Sec. 3.1). The disentanglement of foreground and background volumes opens a wide range of downstream applications. We provide a framework that explores some of these applications by manipulating objects in a semantic manner (Sec. 3.2). An illustration of our framework is provided in Fig. 2. Additional training and implementation details are provided in Appendix A.

3.1 DISENTANGLED OBJECT REPRESENTATION

The ability to disentangle the foreground object volumetrically from the background requires a volumetric representation that correctly handles multiple challenges: (i). Foreground occluding objects, which may be covered by a foreground mask, should not be included in the foreground volume, (ii). Regions occluded by the foreground object should be visible in the background volume, (iii). Illumination and reflectance effects, affecting the foreground object in the full scene volume, should affect the now unoccluded regions of the background in a natural way. To this end, we build upon the representation of neural radiance fields Mildenhall et al. (2020).

Neural Radiance Fields. A neural radiance field Mildenhall et al. (2020) is a continuous function f whose input is a 3D position $p = (x, y, z) \in \mathbb{R}^3$ along with a viewing direction $d = (\theta, \phi) \in \mathbb{S}^2$, indicating a position along a camera ray. The output of f is an RGB color $c \in \mathbb{R}^3$ and volume density $\alpha \in \mathbb{R}^+$. We first apply a frequency-based encoding γ to correctly capture high-frequency details using $\gamma(p) = [\cos(2\pi \mathbf{B}p), \sin(2\pi \mathbf{B}p)]^{\mathrm{T}}$, where $\mathbf{B} \in \mathbb{R}^{n \times 3}$ is a randomly drawn Gaussian matrix whose entries are drawn from $\mathcal{N}(0, \sigma^2)$, where σ is a hyperparameter. f is then parameterized as an MLP f_θ whose input is $(\gamma(p), \gamma(d))$ and output is c and σ.

Object Representation. Given camera extrinsics ξ, we assume a set $\{(c_r^i, \sigma_r^i)\}_{i=1}^N$ of color and volume density values predicted by f_θ for N randomly chosen points along camera ray r. A rendering operator then maps these values to an RGB color c_r as follows:

$$c_r = \sum_{i=1}^{N} w_r^i \cdot c_r^i \qquad w_r^i = \prod_{j=1}^{i} (1 - \sigma_j^r) \cdot T_r^i \qquad T_r^i = 1 - \exp(\sigma_r^i \cdot \delta_i^r) \tag{1}$$

where σ_r^i and T_r^i are the alpha and transmittance values for point i along ray r and $\delta_i^r = t^{i+1} - t^i$ is the distance between adjacent samples. For training, we assume a set of posed views $\{x_i\}_{i=1}^M$ together with their associated foreground object masks $\{m_i\}_{i=1}^M$. We set $\{\hat{x_i}\}_{i=1}^M$ to be the corresponding colors to $\{x_i\}_{i=1}^M$ as predicted by Eq. (1). Fig. 3 gives an overview of the training. To train the background (resp. full) volume, we minimize the masked (resp. unmasked) reconstruction loss

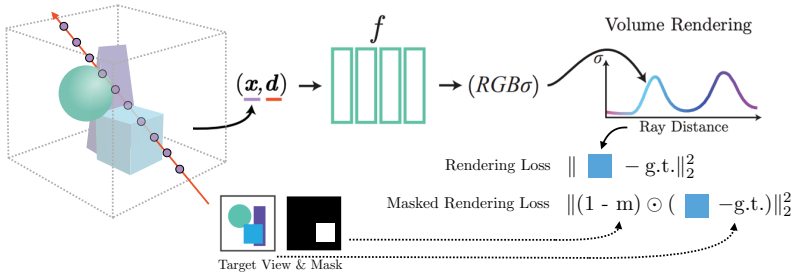


Figure 3: **Training losses for the background and full scene.** We train a neural radiance field for the *full scene* with rendering loss on all ground truth pixels. To train the *background scene*, we apply a masked rendering loss, where regions that are projected inside a 2D mask $(1 - m)$, are not penalized in the loss. The network learns to reconstruct this region based on correlated effects, such as how light from the surrounding affects the masked regions and multi-view geometry.

between real and estimated views:

$$\mathcal{L}_{bg} = \sum_{i=1}^{M} ||(1 - m_i) \odot (x_i - \hat{x}_i)||_2^2 \qquad \mathcal{L}_{full} = \sum_{i=1}^{M} ||x_i - \hat{x}_i||_2^2 \tag{2}$$

Let $w_{bg}^{i_r}$ and $c_{bg}^{i_r}$ be the value of w_r^i and c_r^i in Eq. (1) predicted for the background volume and similarly let $w_{full}^{i_r}$ and $c_{full}^{i_r}$ be the value of w_r^i and c_r^i in Eq. (1) predicted for the full volume. A natural representation of the foreground object can then be found using the principle of volume mixing Drebin et al. (1988):

$$c_r^{fg} = \sum_{i=1}^{N} w_{fg}^{i_r} \cdot c_{fg}^{i_r} \qquad w_{fg}^{i_r} = w_{full}^{i_r} - w_{bg}^{i_r} \qquad c_{fg}^{i_r} = c_{full}^{i_r} - c_{bg}^{i_r} \tag{3}$$

c_r^{fg} is the foreground volume color at the pixel corresponding to ray r. Eq. (3) renders the color of the foreground object for all pixels across different views.

Object Controls. We note that camera parameters, as well as chosen poses, rays, and sampled points along the rays, are chosen to be identical for both the full volume and the background volume, and hence also identical to the foreground volume. Given this canonical setting, the corresponding points along the rays for both the foreground and background can be easily found.

Due to the above-mentioned correspondence, one can independently modify ${w_{fg}}^{i_r}$ and ${c_{fg}}^{i_r}$ to get ${w'_{fg}}^{i_r}$ and ${c'_{fg}}^{i_r}$ for the foreground volume as well as ${w_{bg}}^{i_r}$ and ${c_{bg}}^{i_r}$ to get ${w'_{bg}}^{i_r}$ and ${c'_{bg}}^{i_r}$ for the background volume. In order to recombine the modified background with the modified foreground, we note that every 3D point along the ray should only be colored, either according to the background volume or according to the foreground volumes, but not by both, as they are disentangled. We can then recombine the modified foreground and background:

$$c_r^c = \sum_{i=1}^{N} {w'_{bg}}^{i_r} \cdot {c'_{bg}}^{i_r} + {w'_{fg}}^{i_r} \cdot {c'_{fg}}^{i_r} \tag{4}$$

c_r^c is the recombined color of the pixel corresponding to ray r. In our experiments, we only modify the foreground and so ${w'_{bg}}^{i_r} = {w_{bg}}^{i_r}$, ${c'_{bg}}^{i_r} = {c_{bg}}^{i_r}$.

3.2 Object Manipulation

Given the ability to control the foreground and background volumes separately, we now propose a set of downstream manipulation tasks that emerge from our disentangled representation. As noted in Sec. 3.1, we can now control the weights, colors as well as translation parameters separately for the foreground and background volumes and so introduce a set of manipulation tasks that use the controls. We note that the task of *Object Removal* is equivalent to displaying the background.

Object Transformation. Due to the alignment of camera parameters, as well as chosen poses, rays, and sampled points along the rays, one can apply a transformation on the background and foreground volumes separately, before recombining the volumes together. For either the foreground or the background, and for a given transformation T, we simply evaluate the color and weight of point p using f_θ at position $T^{-1}(p)$ and then recombine the volumes together using Eq. (4).

Object Camouflage. Here we wish to change the texture of the foreground 3D object such that it is difficult to detect from its background Owens et al. (2014); Guo et al. (2022). Such an ability can be useful in the context of diminished reality Mori et al. (2017). To do so, we fix the depth of the foreground object while manipulating its texture. As the depth of the foreground is derived from ${w_{fg}}^{i_r}$, we fix ${w'_{fg}}^{i_r} = {w_{fg}}^{i_r}$ and only optimize ${c'_{fg}}^{i_r}$. We follow Eq. (4), in compositing the foreground and background volumes. Let the resulting output for each view i be $\hat{x}_i^c$, and let $\hat{x}_i^{bg}$ be the corresponding output for the background volume. We optimize a neural radiance field for foreground colors ${c'_{fg}}^{i_r}$ to minimize $\mathcal{L}_{camouflage} = \sum_{i=1}^{M} ||\hat{x}_i^c - \hat{x}_i^{bg}||_2^2$. As depth is fixed, only the foreground object colors are changed to match the background volume as closely.

Non-negative 3D Inpainting. Next, we consider the setting of non-negative image generation Luo et al. (2021). We are interested in performing non-negative changes to views of the full scene so as to most closely resemble the background. This constraint is imposed in optical-see-through devices that can only add light onto an image. In this case, we learn a residual volume to render views $\hat{x}_i^{residual}$ as in Eq. (1) to minimize $\mathcal{L}_{non-negative} = \sum_{i=1}^{M} ||\hat{x}_i^{full} + \hat{x}_i^{residual} - \hat{x}_i^{bg}||_2^2$. where $\hat{x}_i^{full}$ are rendered views of the full scene as in Eq. (2). That is, we learn a residual volume whose views are $\hat{x}_i^{residual}$, such that when added to the full volume views, most closely resemble the background.

Semantic Manipulation. Next, we consider a mechanism for the semantic manipulation of the foreground. We consider the recently proposed model of CLIP Radford et al. (2021), which can be used to embed an image I and text prompt t (or image I_2), and to subsequently compare the cosine similarity of the embeddings. Let this operation be $sim(I, t)$ (resp. $sim(I, I_2)$), where a value of 1 indicates perceptually similar text (resp. image) and image. Let $\hat{x}_i^c$ be the result of applying Eq. (4), while fixing the background colors and weights as well as the foreground weights. That is, we only optimize the foreground colors ${c'_{fg}}^{i_r}$. For a user-specified target text t, we consider the objective:

$$\mathcal{L}_{semantic} = \sum_{i=1}^{M} 1 - sim\left(\hat{x}_i^c \odot m_i + \hat{x}_i^{bg} \odot (1 - m_i), t\right) \tag{5}$$

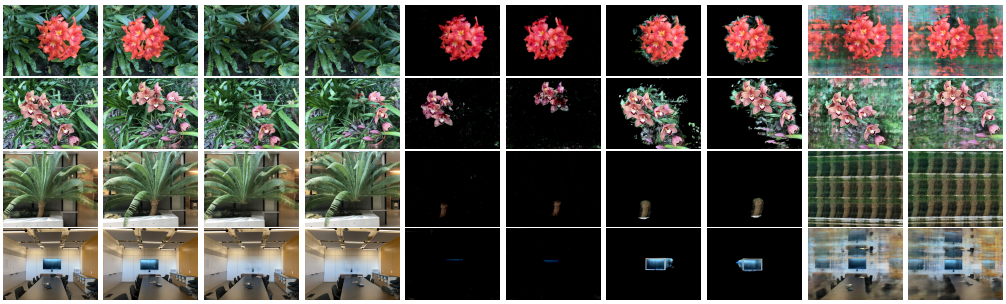
$$+ 1 - sim\left(\hat{x}_i^c \odot m_i + \hat{x}_i^{bg} \odot (1 - m_i), \hat{x}_i^{bg} \odot (1 - m_i)\right) \tag{6}$$

$$+ ||\hat{x}_i^c \odot (1 - m_i), \hat{x}_i^{bg} \odot (1 - m_i)||_2^2 \tag{7}$$

We note that while only the colors of the foreground volume can be manipulated, we enforce that such changes only occur within the localized masked region of the foreground, and so take the background from the fixed background volume. To do so, instead of applying clip similarity directly with $\hat{x}_i^c$, we apply it with $\hat{x}_i^c \odot m_i + \hat{x}_i^{bg} \odot (1 - m_i)$. Therefore, CLIP's similarity can only be improved by making local changes that occur within the masked region of the foreground object, but can 'see' the background as well as the foreground for context. We enforce the generated volume views are similar to both the target text (Eq. (5)) and the background (Eq. (6)). To further enforce that no changes are made to the background, we constrain the background of the combined volume views to match those of the background using Eq. (7).

4 EXPERIMENTS

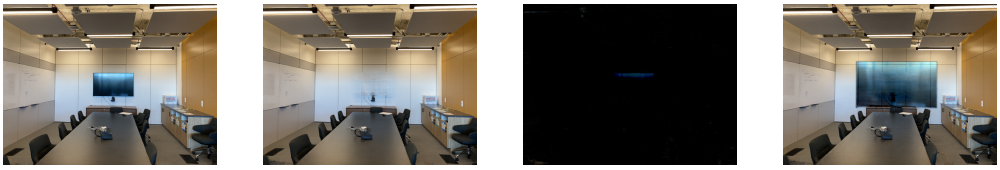
We divide the experimental section into two parts. First, we consider the ability to successfully disentangle the foreground and background volumes from the rest of the scene. Second, we demonstrate some of the many manipulation tasks this disentanglement enables, as described in Sec. 3.2. Corresponding 3D scenes from multiple existing and novel views are provided as a supplementary.



Full Background Foreground Yang et al. (2021) Foreground Training

Figure 4: Two rendered *novel* views of the full scene, background, and foreground. Note that as the TV screen is black, the foreground object appears black as well. For ObjectNeRF Yang et al. (2021), however, the object representation also captures the background reflections. We also show the result of directly training a neural field on the masked foreground object itself (foreground training). As masks are noisy and may include much of the background, this results in a very noisy result which also include much of the background. While no 2D mask annotation is given for novel views, corresponding masks for training views are provided in the appendix.

Figure 5: Two uniformly sampled renderings of the full and the background volumes for three different scenes. The removed object is visually enhanced by a 2D mask.



(a) Original Scene (b) Background (c) Foreground Obj. (d) New Scene

Figure 6: **Foreground object transformation**. Our method makes plausible predictions in occluded regions (behind the TV) by understanding the correlated effects from the rest of the scene, such as the light reflections in the TV screen, which are not visible in the foreground. After scaling the foreground object and placing it back into the scene, the correlated effects are introduced again, resulting in photo-realistic and view consistent light reflections on the TV screen.

(a) (b) (c) (d) (e)

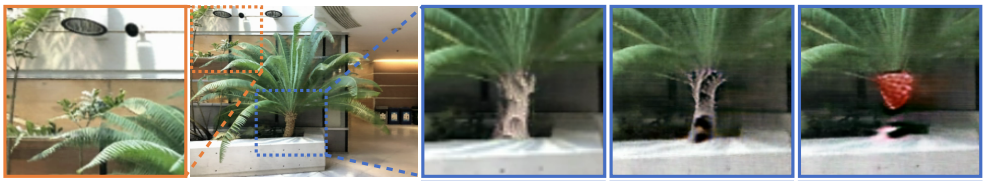
Figure 7: **Object camouflage for two different random views of a fortress scene.** (a) original scene, (b) background scene, (c) disparity map of the background scene, (d) camouflaged scene, (e) disparity map of the camouflaged scene.

(a) (b) (c) (d)

Figure 8: **Non-negative object inpainting for two views for a scene of leaves.** Given the full scene (a), a residual scene is added (b) resulting in scene (c), with the aim of being close to the background without the leaf (d).

	Object Removal			Object Extraction		Semantic Object Manipulation		
	Ours	DeepFill-v2	EdgeConnect	Ours	ObjectNeRF	Ours	GLIDE	Blended
Q1	**3.86**	2.44	2.37	**3.87**	2.85	**3.85**	1.10	1.26
Q2	**3.84**	1.52	1.86	**3.91**	2.62	**3.78**	1.20	1.26

Table 1: A user study performed for the tasks of Object Removal, Object Extraction, and 3D Semantic Object Manipulation. A mean opinion score (1-5) is shown.



(a) (b) (c) (d) (e)

Figure 9: **3D Object Manipulation.** Insets of the disentangled (a) window mullion and manipulated (c)-(e) tree trunk in the original scene (b). Note how the window mullion is removed without removing the leaf of the fern that occludes it from the first view. The query text to manipulate the trunks is (c) *Old tree*, (d) *Aspen tree*, and (e) *Strawberry*.

4.1 Object Disentanglement

Fig. 4 shows views from different scenes of the LLFF dataset Mildenhall et al. (2019), where we separate the full scene, background, and foreground in a volumetrically and semantically consistent manner. We compare our method to ObjectNeRF Yang et al. (2021). We note that ObjectNeRF requires 3D bounding boxes to extract the background volume which we do not use. Hence, we consider only the extracted foreground by ObjectNeRF. As can be seen, ObjectNeRF's extracted foreground object captures much of the background as well. This is most visible for the orchids and tree trunk examples (second and third rows). Further, for the TV extraction (fourth row), our object representation isolates the reflections on the TV from the background (hence the TV appears black), while ObjectNeRF considers those reflections as part of the object representation. Our reflection-independent object representation allows us to resize the TV in a manner that correctly adheres to those reflections as those resulting from background light sources. This can be seen in Fig. 6. As a further comparison we consider a neural field trained to reconstruct only the masked region. Due to some noisy masks, shown in the appendix, this results in a noisy result which captures much of the background.

Fig. 5 depicts the consistency of the removal of a leaf, a T-rex, and a whiteboard for two different views. The background neural radiance field makes plausible predictions of the background scene via multi-view geometry and based on the correlated effects from the scene. *E.g.* the background behind the leaf or the legs of the T-rex might be occluded by the 2D mask from one view, but visible from another. However, the background behind the whiteboard is occluded from every angle. Nevertheless, the background neural radiance field makes a plausible prediction of the background based on the correlated effects from the surrounding scene. Further, our model can handle the disentanglement of non-planar objects, such as the T-rex, well.

In the 2D domain, as far as we can ascertain, the closest 2D task to object disentanglement is that of object inpainting. We consider two prominent baselines of DeepFill-v2 Yu et al. (2019) and EdgeConnect Nazeri et al. (2019) for this task and compare our method on the scenes of leaves and whiteboard removal as in Fig. 5. We train the baseline on the same training images and their associated masks. In order to compare our method on the same novel views, we train a NeRF Mildenhall et al. (2020) on the resulting outputs, resulting in a scene with the same novel views as ours. Unlike our method, the results have 3D inconsistencies, artifacts and flickering between views. The visual comparison is provided in the supplementary.

To assess our method numerically, we conduct a user study and ask users to rate from a scale of $1-5$: (Q1) "How well was the object removed/extracted?" and (Q2) "How realistic is the resulting object/scene?" We consider 25 users and mean opinion scores are shown in Tab. 1. For object

extraction we consider ObjectNeRF Yang et al. (2021) and consider the scenes in Fig. 4, For object removal, we consider 2D baselines of DeepFill-v2 Yu et al. (2019), EdgeConnect Nazeri et al. (2019), as detailed above, and consider the leaves and whiteboard scenes as in Fig. 5. As no 3D bounding box is provided, we did not consider ObjectNeRF Yang et al. (2021) for object removal.

4.2 Object Manipulation

Foreground Transformation. We consider the ability to scale the foreground object and place the rescaled object back into the scene by changing the focal length used to generate the rays of the foreground object, and then volumetrically adding it back into our background volume. Fig. 6 shows an example where the disentangled TV is twice as large. We note that other transformations such as translation and rotation are possible in a similar manner. Fig. 6 highlights several properties of our volumetric disentanglement volume. First, the network is able to "hallucinate" how a plausible background looks in regions occluded across all views (*e.g.* behind the TV). It does this based on correlated effects from the rest of the scene. A second property is that it can disentanglement correlated effects such as the reflections on the TV screen, which is evident from the almost completely black TV in the foreground scene. Lastly, these correlated effects result in consistent and photo-realistic reflections, when we place the rescaled TV back into the scene.

Object Camouflage. Another manipulation is of camouflaging an object Owens et al. (2014); Guo et al. (2022), *i.e.* only changing the texture of the object and not its shape. Fig. 7 illustrates examples of camouflaging with fixed depth, but free texture changes. While the depth of the camouflaged object and that of the foreground object match, the appearance is that of the background.

Non-Negative Inpainting. In optical see-through AR, one might also wish to camouflage objects Luo et al. (2021) or inpaint them. However, in see-through AR one can only add light. Fig. 8 shows how adding light can make the appearance of camouflage in a 3D consistent manner.

3D Object Manipulation. We now consider 3D object manipulation. Fig. 9 shows two views of a fern. We have disentangled both the window mullion in the upper left corner and the tree trunk from the rest of the scene. Even though the window mullion is occluded in the first view, and thus our 2D mask is masking the occluding leaf in front of the window mullion, this occluding object is not part of the disentangled window mullion object. The 3D manipulations are shown in (c)-(e) in Fig. 9. For the strawberry manipulation in (e), note how part of the tree trunk was camouflaged to more closely resemble the shape of a strawberry. We compare to 2D text-based inpainting methods of GLIDE Nichol et al. (2021) and Blended Diffusion Avrahami et al. (2021), where we follow the same procedure as in Sec. 4.1. We consider a similar user study as detailed in Sec. 4.1, where Q1 is modified to: "How well was the object semantically manipulated according to the target text prompt?" and consider the fern scene of Fig. 9, for the text prompts of "strawberry" and "old tree".

4.3 Discussion and Limitations

Our work has some limitations. When light from the background affects the foreground object, we correctly disentangle the illuminations on the object. However, when the object is a light source, we cannot completely disentangle the object as seen in Appendix Fig. 10 (a) and in the supplementary. Another limitation is with respect to the semantic manipulation of foreground objects. We found that manipulating too large objects results in an under-constrained optimization because the signal provided by CLIP is not sufficient. We also note that, while our work can handle noisy masks, we require masks for all training views. We leave the task of reducing the number of masks for future work. Further, artifacts may arise when the training views do not provide sufficient information to generate the foreground using correlated background effects of multi-view geometry. Our work is orthogonal to recent speed and generalization extensions of NeRF that could be combined with our method. The number of 2D masks required by our method is also upper bounded by the number of training views and so methods such as DietNeRF Jain et al. (2021b) and SinNeRF Xu et al. (2022) could be combined with our method to reduce the number of 2D masks required to even a single mask. Alternatively one can use zero-shot segmentation approaches such as Shin et al. (2022) to obtain masks in a zero-shot manner. In Appendix B, we consider, for the task of foreground object translation (Fig. 6), alternatives to the recombining method of Eq. (4). Lastly, we note that our method can handle noisy annotations of the foreground. In Appendix Fig. 10(c), we demonstrate the masks used for the leaves scene, which were extracted using an off-the-shelf segmentation algorithm.

5 CONCLUSION

In this work, we presented a framework for the volumetric disentanglement of foreground objects from a background scene. The disentangled foreground object is obtained by volumetrically subtracting a learned volume representation of the background with one from the entire scene. The foreground-background disentanglement adheres to object occlusions and background effects such as illumination and reflections. We established that our disentanglement facilitates separate control of color, depth, and transformations for both the foreground and background objects. This enables a wide range of applications going beyond object movement and placement, of which we have demonstrated those of object camouflage, non-negative generation, and 3D object manipulation.

REFERENCES

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J Guibas. Learning representations and generative models for 3d point clouds. *arXiv preprint arXiv:1707.02392*, 2017.

Eman Ahmed, Alexandre Saint, Abd El Rahman Shabayek, Kseniya Cherenkova, Rig Das, Gleb Gusev, Djamila Aouada, and Björn E. Ottersten. A survey on deep learning advances on different 3d data representations. *arXiv: Computer Vision and Pattern Recognition*, 2018.

Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. *arXiv preprint arXiv:2111.14818*, 2021.

Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

Adrian Broadhurst, Tom W Drummond, and Roberto Cipolla. A probabilistic framework for space carving. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, volume 1, pp. 388–393. IEEE, 2001.

Andrew Brock, Theodore Lim, James Millar Ritchie, and Nicholas J. Weston. Generative and discriminative voxel modeling with convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2016.

Zehranaz Canfes, M Furkan Atasoy, Alara Dirik, and Pinar Yanardag. Text and image guided 3d avatar generation and manipulation. *arXiv preprint arXiv:2202.06079*, 2022.

Boyang Deng, John P Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Nasa neural articulated shape approximation. In *European Conference on Computer Vision*, pp. 612–628. Springer, 2020.

Robert A Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *ACM Siggraph Computer Graphics*, 22(4):65–74, 1988.

Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 341–346, 2001.

Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pp. 1033–1038. IEEE, 1999.

Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. *arXiv preprint arXiv:2203.15224*, 2022.

Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Christine Guillemot and Olivier Le Meur. Image inpainting: Overview and recent advances. *IEEE signal processing magazine*, 31(1):127–144, 2013.

Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. Object-centric neural scene rendering. *arXiv preprint arXiv:2012.08503*, 2020.

Rui Guo, Jasmine Collins, Oscar de Lima, and Andrew Owens. Ganmouflage: 3d object nondetection with texture fields. *arXiv preprint arXiv:2201.07202*, 2022.

Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: A network with an edge. *ACM Trans. Graph.*, 38(4), jul 2019. ISSN 0730-0301. doi: 10.1145/3306346.3322959. URL https://doi.org/10.1145/3306346.3322959.

Le Hui, Rui Xu, Jin Xie, Jianjun Qian, and Jian Yang. Progressive point cloud deconvolution generation network. In *ECCV*, 2020.

Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. *arXiv preprint arXiv:2112.01455*, 2021a.

Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5885–5894, 2021b.

Won Jun Jang and Lourdes de Agapito. Codenerf: Disentangled neural radiance fields for object categories. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12929–12938, 2021.

Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *arXiv preprint arXiv:2205.15585*, 2022.

Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12871–12881, 2022.

Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38(3):199–218, 2000.

Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

Katie Luo, Guandao Yang, Wenqi Xian, Harald Haraldsson, Bharath Hariharan, and Serge Belongie. Stay positive: Non-negative image synthesis for augmented reality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10050–10060, 2021.

Mehdi Mekni and Andre Lemieux. Augmented reality: Applications, challenges and future trends. *Applied computational science*, 20:205–214, 2014.

Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. *arXiv preprint arXiv:2112.03221*, 2021.

B Mildenhall, P Srinivasan, M Tancik, JT Barron, and RRR Ng. Representing scenes as neural radiance fields for view synthesis. In *Proc. of European Conference on Computer Vision, Virtual*, 2020.

Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019.

Shohei Mori, Sei Ikeda, and Hideo Saito. A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects. *IPSJ Transactions on Computer Vision and Applications*, 9(1):1–14, 2017.

Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019.

Thu H Nguyen-Phuoc, Christian Richardt, Long Mai, Yongliang Yang, and Niloy Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. *Advances in Neural Information Processing Systems*, 33:6767–6778, 2020.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11453–11464, 2021.

Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5762–5772, 2021.

Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2856–2865, 2021.

Andrew Owens, Connelly Barnes, Alex Flint, Hanumant Singh, and William Freeman. Camouflaging an object from many viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2782–2789, 2014.

Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9964–9973, 2019.

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*, pp. 8748–8763. PMLR, 2021.

Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Aditya Sanghi, Hang Chu, J. Lambourne, Ye Wang, Chin-Yi Cheng, and Marco Fumero. Clip-forge: Towards zero-shot text-to-shape generation. *ArXiv*, abs/2110.02624, 2021a.

Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, and Marco Fumero. Clip-forge: Towards zero-shot text-to-shape generation. *arXiv preprint arXiv:2110.02624*, 2021b.

Gyungin Shin, Weidi Xie, and Samuel Albanie. Reco: Retrieve and co-segment for zero-shot transfer. *arXiv:2206.07045*, 2022.

Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3858–3867, 2019.

Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *CVPR*, 2021.

Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. *arXiv preprint arXiv:2112.05139*, 2021a.

Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018.

Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf-: Neural radiance fields without known camera parameters. *ArXiv*, abs/2102.07064, 2021b.

Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016.

Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *ICCV*, 2019.

Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance fields on complex scenes from a single image. *arXiv preprint arXiv:2204.00928*, 2022.

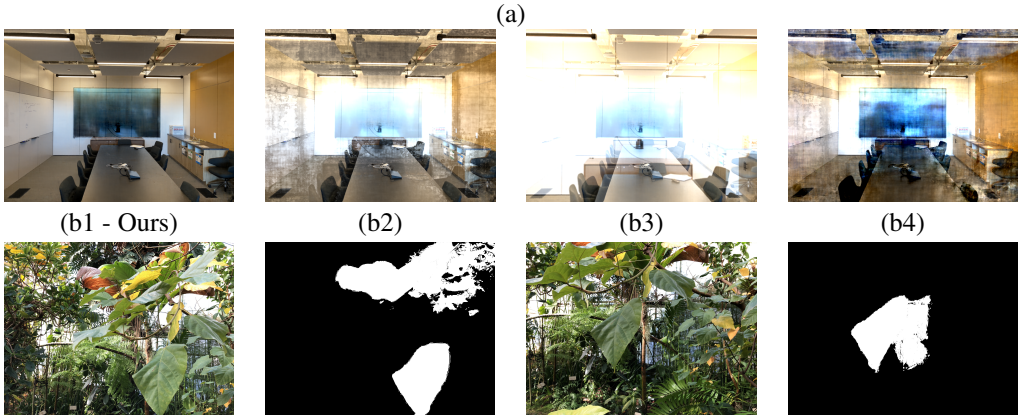
Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13779–13788, 2021.

Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4541–4550, 2019.

Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. iNeRF: Inverting neural radiance fields for pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4471–4480, 2019.

(a)



(b1 - Ours) (b2) (b3) (b4)

(c)

Figure 10: (a) *Failure to completely remove a light source.* The original light source is shown in blue in the middle image and for the background, using our method, on the right. In orange and green are regions affected by the light source, resulting in the failure to completely remove it. (b1-b4) *Ablation for composition*. Alternatives to the composition shown in Eq. (4) for foreground object translation (Fig. 6). (c) *Robustness to noisy 2D masks.* Our method can handle noisy 2D masked obtained automatically.

A Implementation Details

For training, we consider the natural non-synthetic scenes given in Mildenhall et al. (2020), together with their associated pose information. An off-the-shelf segmentation or manual annotation is used to extract masks. We note that masks need not be exact, and may capture more than the desired object (see main paper for details). Our rendering resolution for training the background and full scenes is 504×378. For the manipulation tasks, the same resolution is used for *3D inpainting*, *object camouflage*, *transformation* and *non-negative inpainting* tasks. For the *semantic manipulation* task, our rendering resolution is 252×189. For the CLIP Radford et al. (2021) input, for a given view, we sample a 128×128 grid of points from the 252×189 output and then upsample it to 224×224, which is the required input resolution of CLIP. We normalize the images and apply text and image embedding as in CLIP Radford et al. (2021). We follow NeRF Mildenhall et al. (2020), in optimizing both "coarse" and "fine" networks for a neural radiance field, and follow the same sampling strategy of points along the ray. All neural fields are parametrized using an MLP with ReLU activation of the same architecture as Mildenhall et al. (2020). We use an Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with a learning rate that begins with 5×10^{-4} and decays exponentially to 5×10^{-5}.

B Additional Visualizations

As noted in the main text, Fig. 10 (a) shows the failure to remove a light source. In Fig. 10 (b1 to b4), we show, for the task of foreground object translation (Fig. 6), alternatives to the recombining method of Eq. (4), with (a2) ${c'_{full}}^{i_r}$ instead of ${c'_{fg}}^{i_r}$, (a3) ${w'_{full}}^{i_r}$ instead of ${w'_{fg}}^{i_r}$, (a4) $c^c_r =$

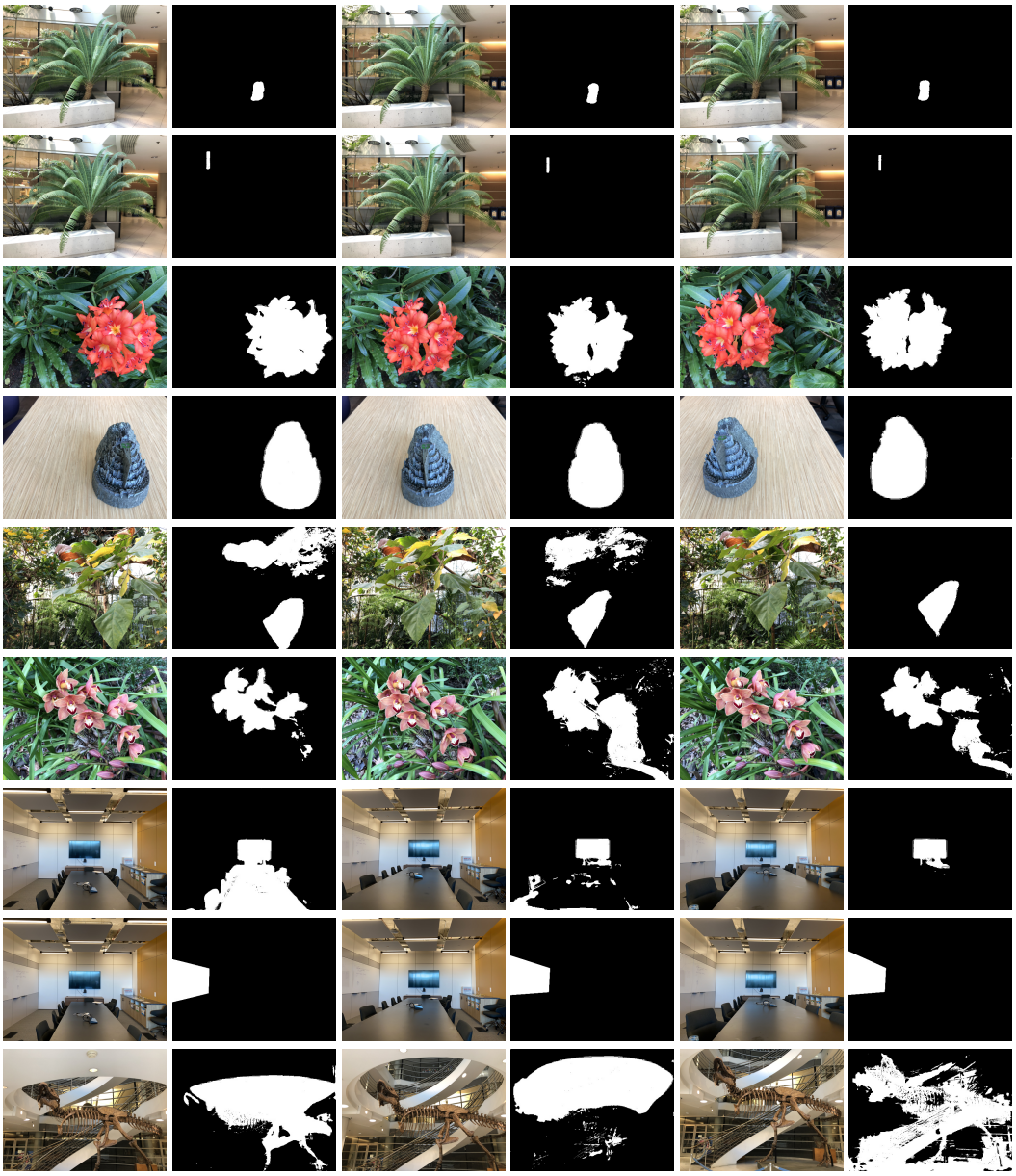


Figure 11: Sample of the masks used for our method for training views.

$\sum_{i=1}^{N}(w'_{bg}{}^{i_r} + w'_{fg}{}^{i_r}) \cdot (c'_{bg}{}^{i_r} + c'_{fg}{}^{i_r})$. Fig. 10 (c) shows examples of the noisy masks used for the leaf scene disentanglement which our method handles correctly.

C TRAINING MASKS

We provide a sample of the training masks used for training views in Fig. 11.