Removing Objects From Neural Radiance Fields

Silvan Weder[1,2] Guillermo Garcia-Hernando[1] Áron Monszpart[1] Marc Pollefeys[2]
Gabriel Brostow[1,3] Michael Firman[1] Sara Vicente[1]
[1]Niantic [2]ETH Zurich [3]University College London
http://nianticlabs.github.io/nerf-object-removal

Abstract

Neural Radiance Fields (NeRFs) are emerging as a ubiquitous scene representation that allows for novel view synthesis. Increasingly, NeRFs will be shareable with other people. Before sharing a NeRF, though, it might be desirable to remove personal information or unsightly objects. Such removal is not easily achieved with the current NeRF editing frameworks. We propose a framework to remove objects from a NeRF representation created from an RGB-D sequence. Our NeRF inpainting method leverages recent work in 2D image inpainting and is guided by a user-provided mask. Our algorithm is underpinned by a confidence based view selection procedure. It chooses which of the individual 2D inpainted images to use in the creation of the NeRF, so that the resulting inpainted NeRF is 3D consistent. We show that our method for NeRF editing is effective for synthesizing plausible inpaintings in a multi-view coherent manner, outperforming competing methods. We validate our approach by proposing a new and still-challenging dataset for the task of NeRF inpainting.

1. Introduction

Since the initial publication of Neural Radiance Fields (NeRFs) [42], there has been an explosion of extensions to the original framework, *e.g.,* [3, 4, 8, 12, 25, 35, 39, 42]. NeRFs are being used beyond the initial task of novel view synthesis. It is already appealing to get them into the hands of non-expert users for novel applications, *e.g.,* for NeRF editing [80] or live capture and training [47], and these more casual use cases are driving interesting new technical issues.

One of those issues is how to seamlessly remove parts of the rendered scene. Removing parts of the scene can be desirable for a variety of reasons. For example, a house scan being shared on a property selling website may need unappealing or personally identifiable objects to be removed [68]. Similarly, objects could be removed so they can be replaced in an augmented reality application, *e.g.,* removing a chair from a scan to see how a new chair fits in the environment [51]. Removing objects might also be desirable when a NeRF is part of a traditional computer vision pipeline, *e.g.,* removing parked cars from scans that are going to be used for relocalization [44].

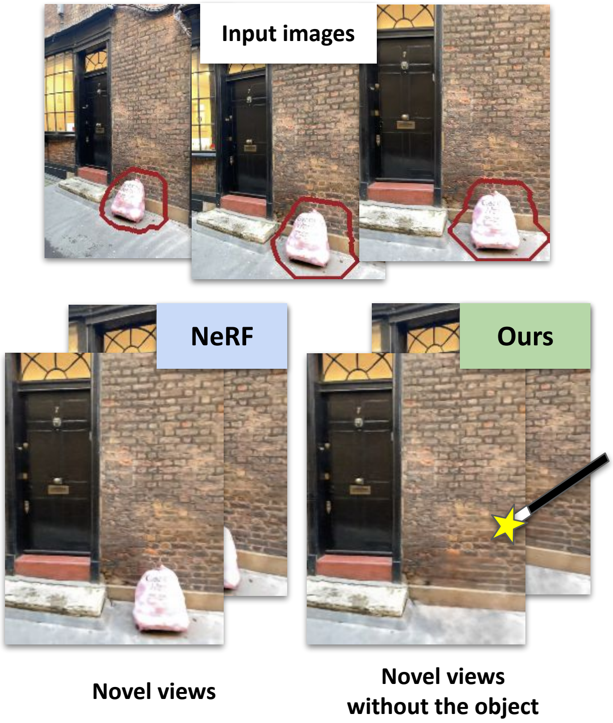

Figure 1. **Removal of unsightly objects.** Our method allows for objects to be plausibly removed from NeRF reconstructions, inpainting missing regions whilst preserving multi-view coherence.

Some editing of NeRFs has already been explored. For example, object-centric representations disentangle labeled objects from the background, which allows editing of the trained scene with user-guided transformations [74, 77], while semantic decomposition allows selective editing and transparency for certain semantic parts of the scene [26]. However, these previous approaches only augment information from the input scan, limiting their generative capabilities, *i.e.,* the hallucination of elements that have not been observed from any view.

With this work, we tackle the problem of removing objects from scenes, while realistically filling the resulting holes, as shown in Fig. 1. Solving this problem requires: a) exploiting multi-view information when parts of the scene are observed in some frames but occluded in others and, b) leveraging a generative process to fill areas that are never observed. To this end, we pair the multi-view consistency of NeRFs with the generative power of 2D inpainting models [69] that are trained on large scale 2D image datasets. Such 2D inpaintings are not multi-view consistent by construction, and may contain severe artefacts. Using these inpaintings directly causes corrupted reconstructions, so we design a new confidence-based view-selection scheme that iteratively removes inconsistent inpaintings from the optimization. We validate our approach on a new dataset and show that we outperform existing approaches for novel view synthesis on standard metrics of image quality, as well as producing multi-view consistent results.

In summary, we make the following contributions: 1) We propose the first approach focusing on inpainting NeRFs by leveraging the power of single image inpainting. 2) We introduce a novel view-selection mechanism that automatically removes inconsistent views from the optimization. 3) We present a new dataset for evaluating object removal and inpainting in indoor and outdoor scenes.

2. Related work

Image inpainting. Image inpainting tackles the problem of plausibly filling in missing regions in a single image. A typical approach is to use an image-to-image network with an adversarial loss, *e.g.,* [22, 53, 79, 82], or with a diffusion model [40]. Different ways have been proposed to encode the input image, *e.g.,* using masked [32] or Fourier convolutions [69]. Image inpainting was extended to also inpaint depth images by [14]. However, these methods do not give temporal consistency between video frames, nor the ability to synthesize novel views.

Removing moving objects from videos. While video inpainting is a well studied problem in computer vision [75, 81], most works focus on removing moving objects. This is typically achieved with guidance from nearby frames, *e.g.,* via estimating flow [15, 21, 75], sometimes using depth [6, 31]. Perhaps counter-intuitively, moving objects make the task easier, since their movement disoccludes the background, making most parts of the scene visible in at least one of the frames.

Removing static objects from videos. Where occluded pixels are visible in other frames in the sequence, these can be used to inpaint regions [28, 45, 46]. For example, [37, 38] remove static foreground distractors from videos, *e.g.,* fences and raindrops. However, there are typically still pixels which cannot be seen in other views, for which some other method is required to fill them in. For example, [18] propagates patches from visible pixels into the region to inpaint, and [45, 70] inpaint missing pixels via PatchMatch. Kim et al. [24] rely on a pre-computed mesh of each scene for object removal. Our key difference to these methods is that our inpaintings can be extrapolated to novel viewpoints.

2.1. Novel view synthesis and NeRFs

NeRF [42] is a highly popular image-based rendering method which uses a differentiable volume-rendering formulation to represent a scene; a multi-layer perceptron (MLP) is trained to regress the color and opacity given a 3D coordinate and ray viewing direction. This combined works on implicit 3D scene representations [9, 34, 41, 52, 58–60], with light-field rendering [11] and novel view synthesis [17, 36, 76]. Extensions include work that reduces aliasing artefacts [3], can cope with unbounded scenes [4], reconstructs a scene from only sparse views [8, 25, 39, 49] or makes NeRFs more efficient, *e.g.,* by using octrees [71, 78] or other sparse structures [61].

Depth-aware Neural Radiance fields. To overcome NeRFs requirement for dense views and the limits in the quality of the reconstructed geometry, depths can be used in training [12, 57]. These can be sparse depths from structure-from-motion [63, 64], or depth from sensors [55, 67].

Object-centric and semantic NeRFs for editing. One direction of progress in NeRFs is the decomposition of the scene into its constituent objects [50, 74, 77]. This is done based on motion for dynamic scenes [50], or instance segmentation for static scenes [77]. Both lines of work also model the background of the scene as a separate model. However, similar to video inpainting, dynamic scenes allow a better modelling of the background since more of it is visible. In contrast, visual artefacts can be seen in the background representation of [74, 77], which model static scenes. Methods that decompose the scene based on semantics [26, 84] can also be used to remove objects. However, they do not try to complete the scene when a semantic part is removed and, for example, [26] discusses how "the background behind the deleted objects can be noisy or have a hole because it lacks observation".

Generative models for novel view synthesis. 3D aware generative models can be used to synthesize views of an object or scene from different viewpoints, in a 3D consistent manner [13, 48, 56, 65]. In contrast with NeRF models, which only have a test time component and "overfit" to a specific scene, generative models can be used to hallucinate views of novel objects by sampling in the latent variable space. There has also been some interest in 3D generative models that work for full indoor scenes [13, 29, 56]. However, their capacity to fit the source views (or memorization) can be limited, as shown in the qualitative results of [13]. To train the generative model, [13, 29, 56] require a large

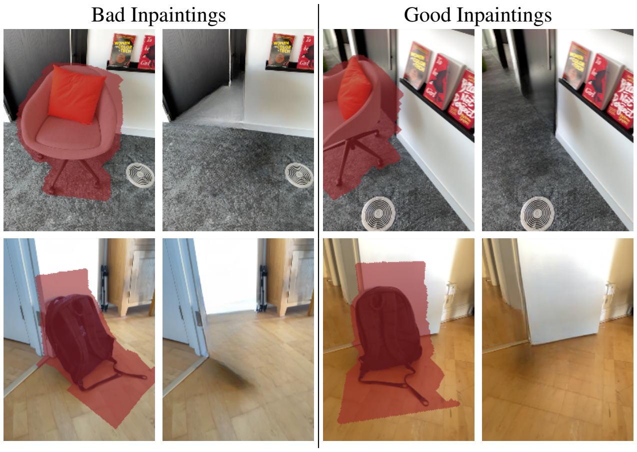

Figure 2. **Per-frame inpainting** can give plausible results for each frame, but they are not consistent between viewpoints and sometimes contain severe artefacts corrupting the optimization.

dataset of indoor scenes with RGB and camera poses and in some cases depth [13, 29]. In contrast, our use of a 2D pretrained inpainting network, which can be trained on any image, is less dependent on the existence of training data and less constrained to indoor scenarios.

Inpainting in novel view synthesis. Inpainting is often used as a *component* of novel view synthesis, to estimate textures for regions unobserved in the inputs [66, 72], *e.g.,* for panorama generation [20, 27]. Philip et al. [54] enable object removal from image-based-rendering, but with an assumption that background regions are locally planar. Similarly to our approach, two concurrent works [33, 43] leverage single image inpaintings to remove objects from NeRFs. To deal with multi-view inconsistencies, [33] manually selects a single view to use in the NeRF optimization inside the mask, while [43] uses a perceptual loss.

3. Method

We assume we are given an RGB-D sequence with camera poses and intrinsics. Depths and poses could be acquired, for example, using a dense structure-from-motion pipeline [63, 64]. For most of our experiments we capture posed RGB-D sequences directly using Apple's ARKit framework [2], but we also show that we can relax this requirement through use of a multi-view stereo method to estimate depth for an RGB sequence. Along the way, we also assume access to a per-frame mask of the object to be removed. The goal is to learn a NeRF model from this input, which can be used to synthesize consistent novel views, where the per-frame masked region should be plausibly inpainted. An overview of our method is shown in Figure 3.

3.1. RGB and depth inpainting network

Our method relies on a 2D single image inpainting method to inpaint each RGB image individually. Furthermore, we also require a depth inpainting network. We use both networks as black-boxes and our approach is agnostic to the method chosen. Future improvements in single image inpainting can be directly translated to improvements to our method. Given an image I_n and corresponding mask M_n, the per-image inpainting algorithm produces a new image $\tilde{I}_n$. Similarly, the depth inpainting algorithm produces an inpainted depth map $\tilde{D}_n$. We show some results for the 2D inpainting network in Figure 2.

3.2. Background on NeRFs

Following the original NeRF paper [42], we represent the scene as an MLP F_Θ that predicts color $\mathbf{c} = [r, g, b]$ and density σ, for a 5 dimensional input containing x, y, z position and two viewing directions. The predicted color for pixel $\mathbf{r}$, $\hat{I}_n(\mathbf{r})$, is obtained by volume rendering along its associated ray, so

$$\hat{I}_n(\mathbf{r}) = \sum_{i=1}^{K} \underbrace{T_i(1 - \exp\left(-\sigma_i\delta_i\right))}_{w_i} \mathbf{c}_i,\ T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j\delta_j\right), \tag{1}$$

where K is the number of samples along the ray, t_i is a sample location, $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples, and w_i is the alpha compositing weight which, by construction, sum to less than or equal to 1.

The NeRF loss operates on training images as

$$\mathcal{L}_{\text{RGB}} = \sum_{n=1}^{N} \sum_{\mathbf{r}\in\Omega_n} \left\| I_n(\mathbf{r}) - \hat{I}_n(\mathbf{r}) \right\|^2, \tag{2}$$

where $I_n(\mathbf{r})$ is the input RGB value for pixel $\mathbf{r}$, and $\hat{I}_n(\mathbf{r})$ is its predicted color. Ω_n indicates the 2D domain of image n. The parameters of the MLP, Θ, are optimized to minimize this loss. Similarly to [57], if input depth is available, then an additional loss can be added,

$$\mathcal{L}_{\text{depth}} = \sum_{n=1}^{N} \sum_{\mathbf{r}\in\Omega_n} \left| D_n(\mathbf{r}) - \hat{D}_n(\mathbf{r}) \right|,\ \text{with}\ \hat{D}_n(\mathbf{r}) = \sum_{i=1}^{K} w_i t_i, \tag{3}$$

where $D_n(\mathbf{r})$ is the input depth for pixel $\mathbf{r}$, and $\hat{D}_n(\mathbf{r})$ is the corresponding predicted depth.

Finally, a distortion regularizer loss was introduced in [4] to better constrain the NeRF optimization and remove "floaters". It encourages the non-zero compositing weights w_i to be concentrated in a small region along the ray, so for each pixel $\mathbf{r}$,

$$l_{\text{dist}}(\mathbf{r}) = \sum_{i,j} w_i w_j \left| \frac{t_i + t_{i+1}}{2} - \frac{t_j + t_{j+1}}{2} \right| + \frac{1}{3} \sum_i w_i^2 \delta_i. \tag{4}$$

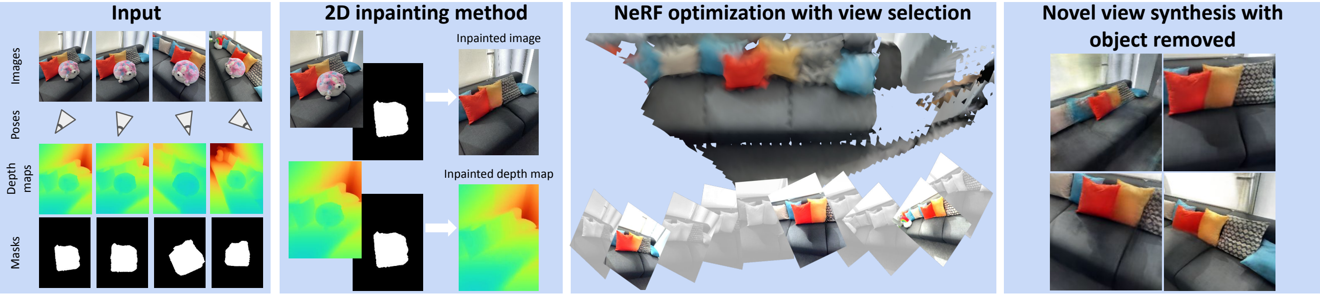

Figure 3. **An overview of our method** We take a sequence of posed RGB-D images together with corresponding 2D masks as input. The 2D frames are inpainted using [69] and then used to optimize a neural radiance field. During optimization, our confidence-based view selection automatically removes inconsistent views from the optimization preventing unwanted artefacts in the final result. Finally, novel views can be rendered from the scene, where the object has been removed.

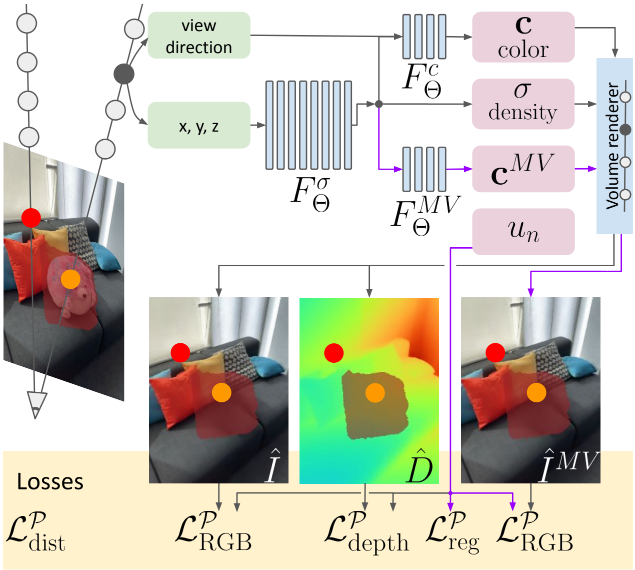
view direction
x, y, z
F^{σ}_{Θ}
F^{c}_{Θ}
F^{MV}_{Θ}
$\mathbf{c}$ color
σ density
$\mathbf{c}^{MV}$
u_n
Volume renderer
$\hat{I}$
$\hat{D}$
$\hat{I}^{MV}$
Losses
$\mathcal{L}^{\mathcal{P}}_{\text{dist}}$
$\mathcal{L}^{\mathcal{P}}_{\text{RGB}}$
$\mathcal{L}^{\mathcal{P}}_{\text{depth}}$
$\mathcal{L}^{\mathcal{P}}_{\text{reg}}$
$\mathcal{L}^{\mathcal{P}}_{\text{RGB}}$

Figure 4. **Our architecture.** Our NeRF formulation contains two color heads F^{c}_{Θ} and F^{MV}_{Θ}, where F^{MV}_{Θ} does not take the view direction as input. This is important to encourage multi-view consistency (see text for details). It also includes uncertainty variables u_n that are used to jointly model the scene and the uncertainty of the 2D inpaintings for our automated view selection.

3.3. Confidence-based view selection

Despite most of the individual inpainted RGB images $\tilde{I}_n$ looking realistic, they still suffer from two issues: 1) some of the inpaintings are incorrect, and 2) despite individual plausibility, they are not multi-view consistent, *i.e.,* the same area observed in multiple views is not necessarily completed in a consistent way (Figure 2). For this reason, we propose a confidence-based view selection scheme, that automatically chooses which views are used in the NeRF optimization. We associate to each image I_n a non-negative uncertainty measure u_n. The corresponding per-image confidence, e^{-u_n}, is used to re-weight the NeRF losses. This confidence value can be seen as a loss attenuation term, similar to the aleatoric uncertainty prediction term in [23].

The RGB loss for our model is then set out as

$$\mathcal{L}^{\mathcal{P}}_{\text{RGB}}(\hat{I}) = \sum_{n=1}^{N} \sum_{\mathbf{r}\in\Omega_n\setminus M_n} \left\| I_n(\mathbf{r}) - \hat{I}_n(\mathbf{r}) \right\|^2 + \sum_{n\in\mathcal{P}} e^{-u_n} \sum_{\mathbf{r}\in M_n} \left\| \tilde{I}_n(\mathbf{r}) - \hat{I}_n(\mathbf{r}) \right\|^2 \qquad (5)$$

where the color for pixels $\mathbf{r}$ is supervised by the inpainted image for pixels inside the mask, and by the input RGB image for pixels outside the mask. Note that the second term of this loss is only computed over a restricted set of images $\mathcal{P}$, where $\mathcal{P} \subseteq \{1, .., N\}$. This is indicated by the superscript $\mathcal{P}$ in the loss term $\mathcal{L}^{\mathcal{P}}_{\text{RGB}}$. In practice, that means that only some inpainted regions are used in the NeRF optimization. We discuss below how we choose the set $\mathcal{P}$.

We use a similar split into pixels inside and outside the mask for the depth loss, so

$$\mathcal{L}^{\mathcal{P}}_{\text{depth}} = \sum_{n=1}^{N} \sum_{\mathbf{r}\in\Omega_n\setminus M_n} \left| D_n(\mathbf{r}) - \hat{D}_n(\mathbf{r}) \right| + \sum_{n\in\mathcal{P}} e^{-u_n} \sum_{\mathbf{r}\in M_n} \left| \tilde{D}_n(\mathbf{r}) - \hat{D}_n(\mathbf{r}) \right|. \qquad (6)$$

Finally, we include two regularizers. One is on the uncertainty weights $\mathcal{L}^{\mathcal{P}}_{\text{reg}} = \sum_{n\in\mathcal{P}} u_n$, to prevent a trivial solution where e^{-u_n} is 0. The other is a distortion regularizer, based on [4], using the loss detailed in Equation (4), so

$$\mathcal{L}^{\mathcal{P}}_{\text{dist}} = \sum_{n=1}^{N} \sum_{\mathbf{r}\in\Omega_n\setminus M_n} l_{\text{dist}}(\mathbf{r}) + \sum_{n\in\mathcal{P}} \sum_{\mathbf{r}\in M_n} l_{\text{dist}}(\mathbf{r}). \qquad (7)$$

View direction and multi-view consistency. When optimizing the NeRF, we made three observations: a) the multi-view inconsistencies in the inpaintings are modelled by the network using the viewing direction; b) we can enforce multi-view consistency by removing the viewing direction from the input; and c) the inconsistencies introduce cloud-like artefacts in the density when not using the viewing direction as input. To prevent a) and c) and correctly optimize the variables u_n that capture the uncertainty about the inpaintings $\tilde{I}_n$, we propose: 1) adding an auxiliary network head, F^{MV}_{Θ}, to the NeRF that does not take the viewing direction as input and, 2) stopping the gradient from the color

Algorithm 1: Iterative refinement using confidence based view selection.

```
Data: Input images $I_n$, Inpainted images $\tilde{I}_n$, Depth maps
      $D_n$, Inpainted depth maps $\tilde{D}_n$, Masks $M_n$
Result: Trained NeRF model $F_\Theta$ with object removed
/* Set of images used for training NeRF is
   initialized with all images.                       */
$\mathcal{P} \leftarrow \{1, ..., N\}$ and $u_n \leftarrow 0, \quad n \in \mathcal{P}$
for $i \leftarrow 0$ to $K_{outer}$ do
    $\Theta \leftarrow$ randomly initialized
    /* Gradient iterations of NeRF training.          */
    for $j \leftarrow 0$ to $K_{grad}$ do
        $\Theta \leftarrow \Theta - \nabla_\Theta \mathcal{L}^\mathcal{P}$
        $\mathbf{U}^\mathcal{P} \leftarrow \mathbf{U}^\mathcal{P} - \nabla_{\mathbf{U}^\mathcal{P}} \mathcal{L}^\mathcal{P}$
    /* Calculate median of confidence values.         */
    $m = \text{median}(\{e^{-u_n}, n \in \mathcal{P}\})$
    /* Update $\mathcal{P}$ by removing images with small
       confidence.                                    */
    for $n \in \mathcal{P}$ do
        if $e^{-u_n} < m$ then
            $\mathcal{P} \leftarrow \mathcal{P} \setminus n$
```

inpainting and F_Θ^{MV} to the density, leaving the uncertainty variable u_n as the only view-dependent input. This design forces the model to encode the inconsistencies between inpaintings into the uncertainty prediction while keeping the model consistent across views. F_Θ^{MV} has a loss term based on Equation 5: $\mathcal{L}_{\text{RGB}}^\mathcal{P}(\hat{I}^{MV})$. The outputs of the auxiliary head F_Θ^{MV} are not used in the final rendering. Instead, the loss associated to this extra head is a regularisation term. See Figure 4 for an illustration of our architecture.

Our final loss is then $\mathcal{L}^\mathcal{P} = \lambda_{\text{RGB}} \mathcal{L}_{\text{RGB}}^\mathcal{P}(\hat{I}) + \lambda_{\text{RGB}} \mathcal{L}_{\text{RGB}}^\mathcal{P}(\hat{I}^{MV}) + \lambda_{\text{depth}} \mathcal{L}_{\text{depth}}^\mathcal{P} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}^\mathcal{P} + \lambda_{\text{dist}} \mathcal{L}_{\text{dist}}^\mathcal{P}$, which is optimized over the MLP parameters $\Theta = \left\{\Theta^\sigma, \Theta^c, \Theta^{MV}\right\}$ and the uncertainty weights $\mathbf{U}^\mathcal{P} = \{u_n, n \in \mathcal{P}\}$. The confidence of all images is initialized to 1, *i.e.,* $u_n := 0$.

Iterative refinement. We use the predicted per-image uncertainty, u_n, in an iterative scheme that progressively removes non-confident images from the NeRF optimization, *i.e.,* we iteratively update the set $\mathcal{P}$ of images that contribute to the loss in masked regions. After K_{grad} steps of optimizing $\mathcal{L}^\mathcal{P}$, we find the median estimated confidence value m. We then remove from the training set all 2D *inpainted regions* which have associated confidence scores less than m. We then retrain the NeRF with the updated training set, and repeat these steps K_{outer} times. Note that images excluded from $\mathcal{P}$ still participate in the optimization, but only for rays in the unmasked regions as they contain valuable information about the scene. This is summarized in Algorithm 1.

3.4. Implementation details

Masking the object to be removed. Similarly to other inpainting methods, our method requires per-frame masks as input. Manually annotating each frame with a 2D mask, as done in other inpainting methods [30, 69, 75], is time consuming. Instead, we manually annotate a 3D box that contains the object using MeshLab [10] to visualise and annotate a 3D point cloud. This only has to be done once per scene. Alternatively, we could have relied on 2D object segmentation methods, *e.g.,* [16], or 3D object bounding box detectors, *e.g.,* one of the baselines in [1].

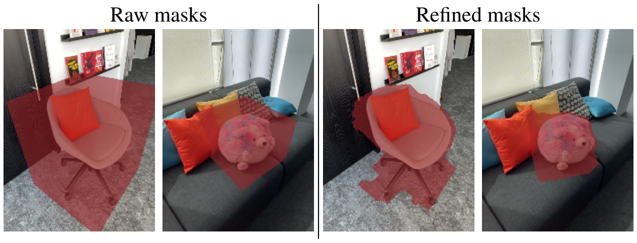

Figure 5. **Mask refinement** Our mask refinement leads to smaller masks and therefore higher quality inpainting.

Mask refinement. In practice, we observe that masks obtained from the annotated 3D bounding boxes can be quite coarse and include large parts of the background. Since large masks have a negative effect on the inpainting quality, we propose a mask refinement step to obtain masks which are tighter around the object. This step is not required if input masks are already tight. Intuitively, this mask refinement step removes parts of the 3D bounding box that are empty space. We start by taking all points in the reconstructed 3D point cloud that are inside the 3D bounding box. The refined mask is then obtained by rendering these points into each image and performing a simple comparison with the depth map to check occlusions in the current image. The resulting mask is cleaned up by dilating any pixel leaks caused by sensor noise using binary dilation and erosion. The effect of our mask refinement step can be seen in Figure 5.

Inpainting network. We used [69] for inpainting both RGB and depth. The inpainting of RGB images and depth maps is done independently and we use the reference network provided by the authors of [69] for both. Our depth maps are preprocessed by clipping to $5\,\text{m}$ and linearly mapping depths in $[0\,\text{m}, 5\,\text{m}]$ to pixel values of $[0, 255]$. We observed empirically that this approach provided good results, but an inpainting method specific for depth maps could improve over this baseline.

NeRF estimation. The implementation of our method is built upon [49] and [3]. We weight the terms in the loss function with $\lambda_{\text{RGB}} = \lambda_{\text{depth}} = \lambda_{\text{dist}} = 1$, and $\lambda_{\text{reg}} = 0.005$. We do a filtering step, where we remove low confidence images every $K_{\text{grad}} = 50,000$ steps, resulting in $K_{\text{outer}} = 4$ filtering steps. Timings for our method are comparable to those of a standard NeRF, in our case [49] and [3]. More implementation details are in the supplementary material.

	Synthetic objects - Masked					Real objects - Masked				
	PSNR↑	SSIM↑	LPIPS↓	Depth L_1 ↓	Depth L_2 ↓	PSNR↑	SSIM↑	LPIPS↓	Depth L_1 ↓	Depth L_2 ↓
Image and video inpainting baselines										
LaMa [69][†]	27.999	0.898	**0.060**	$\underline{0.070}$	**0.075**	-	-	-	-	-
E^{2}FGVI [30][†‡]	24.568	0.874	0.102	-	-	-	-	-	-	-
3D scene completion baselines										
PixelSynth [56][‡]	25.481	0.887	0.116	-	-	**25.438**	0.851	0.152	-	-
CompNVS [29]	17.389	0.823	0.171	1.697	3.641	-	-	-	-	-
Object compositional NeRF [77]*	-	-	-	-	-	21.757	0.836	0.134	0.312	0.341
Ablations										
Masked NeRF	26.126	0.882	0.093	0.084	0.105	21.644	0.815	0.142	$\underline{0.096}$	$\underline{0.054}$
Inpainted NeRF	$\underline{28.760}$	0.905	0.086	0.278	0.400	23.705	0.848	0.134	0.145	0.121
Inpainted NeRF + inpainted Depth	27.568	0.898	0.094	0.231	0.318	23.652	0.844	0.136	0.313	0.387
Ours – no depth	28.290	$\underline{0.906}$	0.079	0.296	0.335	24.228	0.848	0.130	0.345	0.288
Ours – depth predicted using [62]	26.540	0.895	0.087	0.112	0.118	25.010	$\underline{0.856}$	$\underline{0.128}$	0.142	0.140
Our method	**29.437**	**0.916**	$\underline{0.078}$	**0.069**	$\underline{0.096}$	$\underline{25.271}$	**0.859**	**0.125**	**0.071**	**0.044**

Table 1. **Comparison with baselines and state of the art methods.** Our method is either **best** or $\underline{\text{second-best}}$ compared to other novel-view synthesis baselines in inpainting the missing regions of the scene, by propagating multi-view information and leveraging 2D inpainting information. Notes: [†]These methods can't be evaluated on the proposed real dataset as they do not synthesize novel views. [‡]These methods do not produce depth maps. [*] [77] requires the actual object therefore it cannot be evaluated on the proposed synthetic dataset.

4. Experiments

4.1. Datasets

While previous approaches have tackled static object removal from videos, no standard dataset/metrics to evaluate these systems has been proposed, to our knowledge. This work introduces an RGB-D dataset of real scenes, designed to evaluate the quality of object removal. Our dataset has two variants, which are used differently when benchmarking. Please see the supplementary material for more details and visualisations and our website for the full dataset.

Ours — Real objects. This dataset comprises 17 scenes focusing on a small area with one object of interest. They vary in difficulty in terms of background texture, size of the object, and complexity of scene geometry. For each scene, we collected two sequences, one with and the other without the object that we want to remove. The sequences are collected using ARKit [2] on an iPhone 12 Pro with Lidar, and contain RGB-D images and poses. The masks are annotated and refined as described in Section 3.4. For each scene, we use the sequence with the object and corresponding masks for training the NeRF model, and the sequence without the object for testing. The use of real objects makes it easier to evaluate how the systems deal with real shadows and reflections, as well as novel view synthesis.

Ours — Synthetic objects. Most video and image inpainting methods, *e.g.,* [30, 69], do not perform novel view synthesis, meaning such methods cannot be fairly evaluated on our 'Real objects' dataset. We therefore introduce a separate synthetically-augmented variant of our dataset. This uses the same scenes as the real objects dataset, but we only use the sequence without the object. We then manually position a 3D object mesh from ShapeNet [7] in each scene. The object is placed so that it has a plausible location and size, *e.g.,* a laptop on a table. The masks are obtained by projecting the mesh into the input images, which is the only use we make of the 3D object mesh. For this synthetic dataset, following *e.g.,* [4], we use every 8th frame for testing and the rest of the frames for training the NeRF model.

ARKitScenes. We further validate our approach qualitatively on ARKitScenes [5]. This is an RGB-D dataset of 1,661 scenes, where depth was captured via iPhone Lidar.

4.2. Metrics

To evaluate the object removal and inpainting quality, we compare our system's output image against the ground truth image, for each test image in the dataset. All metrics in the paper are only computed inside the masked region. Metrics for the full image are provided in the supplementary material. We use the three standard metrics for NeRF evaluation [42]: PSNR [19], SSIM [73] and LPIPS [83]. To evaluate the geometric completion, we compute the L_1 and L_2 error between the rendered and the ground-truth depth maps inside masked regions. The metrics are averaged over all frames of a sequence, and then averaged over all sequences.

4.3. Ablations and comparison with baselines

In Table 1, we compare our approach with alternative methods for object removal, with a focus on methods that use an underlying NeRF representation.

Image and video inpainting baselines. We compare with two state-of-the-art methods for image ***LaMa*** [69] and video inpainting ***E*^{2}*FGVI*** [30]. In both cases, we use their reference implementation and provided trained network. Neither of these methods allows novel view synthesis, so they are only evaluated on the synthetic objects dataset.

3D scene completion baselines. We compare with several published works for 3D scene completion. Note that none

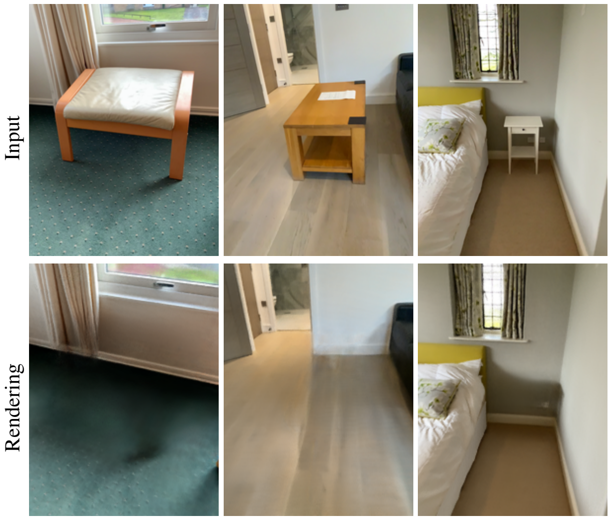

Figure 6. **Results on ARKitScenes [5].** We can successfully remove objects from casually captured sequences in indoor scenes.

of these baselines specifically targets inpainting, so results should be viewed with this in mind. For all of them, we use their publicly available implementation. ***PixelSynth*** [56] and ***CompNVS*** [29] were both proposed for scene *outpainting*. Given one or a few frames of a scene their goal is to complete the scene to enable novel view synthesis. Both of these methods rely on a generative model of indoor scenes and neither requires test-time optimization. Both methods are adapted to use our masks as input. We show qualitative results for ***CompNVS*** in the supplementary materials. ***Object compositional NeRF*** [77] editing of objects in NeRFs via pose transformations. We adapt their code for object *removal* by setting a transformation that moves the object outside the camera's field of view.

Ablations. We compare different ablations of our method, including different ways of training a baseline NeRF model. ***Masked NeRF*** corresponds to training a NeRF using the full input RGB-D data, but pixels and depths in the masked regions are ignored in the NeRF losses. ***Inpainted Images*** is a NeRF trained with all inpainted images, but not using the inpainted depth maps, while ***Inpainted Images + Inpainted Depth*** uses all the inpainted images and inpainted depth maps. This baseline corresponds to ***All views*** in Table 2. We also present results for our method, *i.e.,* training the NeRF with the confidence-based view selection step, but without using depth maps as input (***Ours - no depth***) and using depth maps from a state-of-the-art multi-view depth prediction method [62] to show that we do not necessarily rely on sensor depth (***Ours - depth predicted using*** [62]).

Finally, ***Our method*** is our proposed approach, which uses the method described in Section 3.

As shown in Table 1, our method is superior to other novel-view synthesis baselines across most appearance and depth metrics. Moreover, as opposed to the single image inpainting ***LaMa*** [69], our method is close to multi-view consistent, significantly reducing inter-frame flickering. To scrutinize the flickering, we refer to the supplementary video. Our method also outperforms the naive baselines, which train a NeRF with a masked version of the image, or all the inpainted images. Training our method without using depth maps leads to comparable performance in terms of image metrics, while the depth metrics are considerably worse indicating a degrade in the quality of the recovered 3D shape. Using predicted depth maps from [62] gives competitive results, while pointing to an interesting direction for future research.

Method	# of views	PSNR↑	SSIM↑	LPIPS↓	L_1 ↓	L_2 ↓
All views	82 - 382	27.568	0.898	0.094	0.231	0.318
1/10th	8 - 38	27.098	0.900	0.079	0.202	0.291
1/50th	1 - 7	26.718	0.893	0.087	0.229	0.309
Single view	1	26.232	0.892	0.079	0.133	0.198
Ours	10 - 185	**29.437**	**0.916**	**0.078**	**0.069**	**0.096**

Table 2. **Ablation on view selection methods.** We validate our view selection formulation by comparing to alternative approaches. Ours consistently produces better performing models.

Qualitative comparison. In Figure 7, we show that the proposed method successfully removes the selected object compared to the baselines. While ***Masked NeRF*** fails to complete large holes and ***Inpainted NeRF*** suffers from bad inpaintings in the training set, our method can leverage the 2D inpaintings, while avoiding integrating artefacts by removing those input frames. In contrast to ***Object compositional NeRF*** [77], leveraging the inpaintings also helps to mitigate the appearance of artefacts below the object's surface. Compared to [77] and [56], our method is better able to generate plausible scene completions. We also show results from the ARKitScenes dataset in Figure 6, and please also see our supplementary video for additional results.

View selection ablation. Here we validate that our view selection procedure from Section 3.3 contributes to improved results. We compare our method with different view selection strategies in Table 2 on our synthetic object dataset. ***All views*** uses all the inpainting views when training the NeRF model. The other baselines use a subset of views to train the NeRF model, spaced at regular intervals: every 10th frame for ***1/10th***; every 50th frame for ***1/50th*** and a single middle frame for ***Single view***. The number of views used for each sequence varies depending on the length of the sequence. We outperform these baselines, suggesting that our proposed strategy for view selection is effective in choosing a good set of views to include in the NeRF optimization.

4.4. Limitations

Our method is upper bounded by the performance of the 2D inpainting method. When the masks are too large along the entire trajectory, the 2D inpainting fails and no realis-

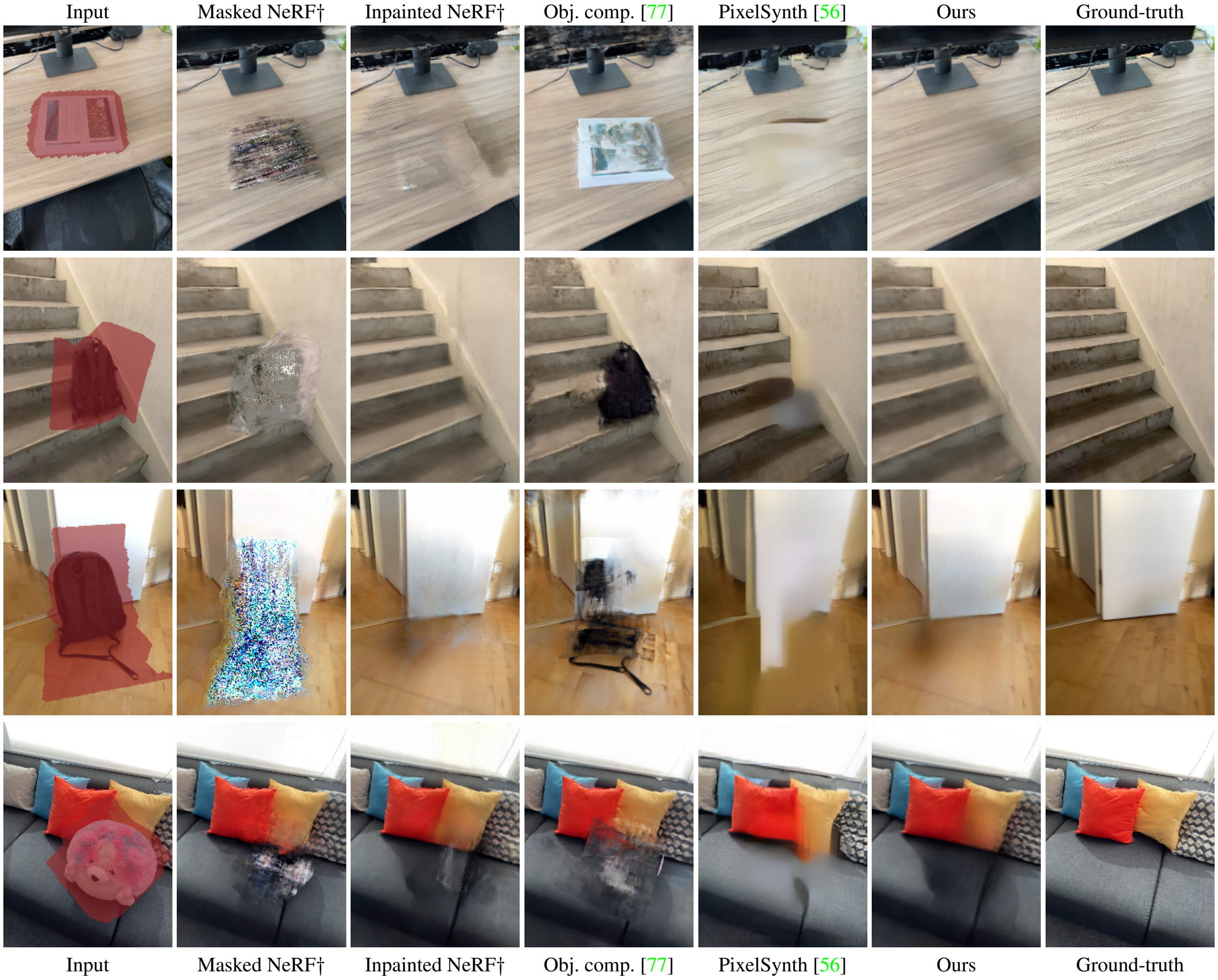

Figure 7. **Qualitative comparisons with baseline.** Our method significantly improves over 3D scene completion baselines for the task of removing objects from a scene using 3D inpainting. Further, it mitigates artefacts and stabilizes convergence compared to the inpainting baseline without automatic view selection. †: We also compare with two NeRF based baselines: ***Masked NeRF*** and ***Inpainted NeRF***.

tic views can be selected. Our renderings sometimes suffer from blurring, caused by flickering of high-frequency textures in the 2D inpaintings. Furthermore, cast shadows or reflections of the object are not handled well. We leave tackling these challenges for future work.

5. Conclusion

We have presented a framework to train neural radiance fields, where objects are plausibly removed from the output renderings. Our method draws upon existing work in 2D inpainting, and introduces an automatic confidence-based view selection scheme to select single-view inpaintings with multi-view consistency. We experimentally validated that our proposed method improves novel-view synthesis from 3D inpainted scenes compared to existing work, despite suffering from blurring. We have also introduced a dataset for evaluating this work, which sets a benchmark for other researchers in the field.

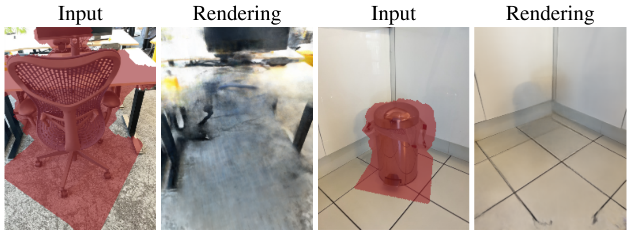

Figure 8. **Failure cases and limitations.** Our method can not recover when the 2D inpainting method fails all the frames, for example when the mask and covers a large part of the image. Further, our method keeps the shadows of the removed object, if they are not included in the object mask.

Acknowledgments. We thank Jamie Watson for fruitful discussions about the project, Paul-Edouard Sarlin for the constructive feedback on an early draft, and Zuoyue Li for kindly running CompNVS on our dataset.

References

[1] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *CVPR*, 2021. (page 5)

[2] Apple. ARKit. Accessed: 14 October 2022. (pages 3 and 6)

[3] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. (pages 1, 2, and 5)

[4] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. (pages 1, 2, 3, 4, and 6)

[5] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes - a diverse real-world dataset for 3D indoor scene understanding using mobile RGB-D data. In *NeurIPS*, 2021. (pages 6 and 7)

[6] Borna Bešić and Abhinav Valada. Dynamic object removal and spatio-temporal RGB-D inpainting via geometry-aware adversarial learning. *IEEE Transactions on Intelligent Vehicles*, 2022. (page 2)

[7] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012, 2015. (page 6)

[8] 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 *ICCV*, 2021. (pages 1 and 2)

[9] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3D shape reconstruction and completion. In *CVPR*, 2020. (page 2)

[10] Paolo Cignoni, Massimiliano Corsini, and Guido Ranzuglia. MeshLab: an open-source 3D mesh processing system. *ERCIM News*, 2008. (page 5)

[11] Abe Davis, Marc Levoy, and Fredo Durand. Unstructured light fields. In *Computer Graphics Forum*, 2012. (page 2)

[12] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *CVPR*, 2022. (pages 1 and 2)

[13] Terrance DeVries, Miguel Angel Bautista, Nitish Srivastava, Graham W Taylor, and Joshua M Susskind. Unconstrained scene generation with locally conditioned radiance fields. In *ICCV*, 2021. (pages 2 and 3)

[14] Ryo Fujii, Ryo Hachiuma, and Hideo Saito. RGB-D image inpainting using generative adversarial network with a late fusion approach. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, 2020. (page 2)

[15] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. Flow-edge guided video completion. In *ECCV*, 2020. (page 2)

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. (page 5)

[17] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (ToG)*, 37(6), 2018. (page 2)

[18] Jan Herling and Wolfgang Broll. Advanced self-contained object removal for realizing real-time diminished reality in unconstrained environments. In *ISMAR*, 2010. (page 2)

[19] Alain Hore and Djemel Ziou. Image quality metrics: PSNR vs. SSIM. In *ICPR*, 2010. (page 6)

[20] Ching-Yu Hsu, Cheng Sun, and Hwann-Tzong Chen. Moving in a 360 world: Synthesizing panoramic parallaxes from a single panorama. *arXiv:2106.10859*, 2021. (page 3)

[21] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Temporally coherent completion of dynamic video. *ACM Transactions on Graphics (ToG)*, 2016. (page 2)

[22] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 2017. (page 2)

[23] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *NeurIPS*, 30, 2017. (page 4)

[24] Joohyung Kim, Janghun Hyeon, and Nakju Doh. Generative multiview inpainting for object removal in large indoor spaces. *International Journal of Advanced Robotic Systems*, 2021. (page 2)

[25] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *CVPR*, 2022. (pages 1 and 2)

[26] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing NeRF for editing via feature field distillation. In *NeurIPS*, 2022. (pages 1 and 2)

[27] Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Pathdreamer: A world model for indoor navigation. In *ICCV*, 2021. (page 3)

[28] Vincent Lepetit, Marie-Odile Berger, and LORIA-INRIA Lorraine. An intuitive tool for outlining objects in video sequences: Applications to augmented and diminished reality. In *ISMAR*, 2001. (page 2)

[29] Zuoyue Li, Tianxing Fang, Zhenqiang Li, Zhaopeng Cui, Yoichi Sato, Marc Pollefeys, and Martin R. Oswald. CompNVS: Novel view synthesis with scene completion. In *ECCV*, 2022. (pages 2, 3, 6, and 7)

[30] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *CVPR*, 2022. (pages 5 and 6)

[31] Miao Liao, Feixiang Lu, Dingfu Zhou, Sibo Zhang, Wei Li, and Ruigang Yang. DVI: Depth guided video inpainting for autonomous driving. In *ECCV*, 2020. (page 2)

[32] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, 2018. (page 2)

[33] Hao-Kang Liu, I-Chao Shen, and Bing-Yu Chen. NeRF-In: Free-form NeRF inpainting with RGB-D priors. *arXiv*, 2022. (page 3)

[34] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3D supervision. *NeurIPS*, 2019. (page 2)

[35] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *ICCV*, 2021. (page 1)

[36] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping GAN: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *ICCV*, 2019. (page 2)

[37] Yu-Lun Liu, Wei-Sheng Lai, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Learning to see through obstructions. In *CVPR*, 2020. (page 2)

[38] Yu-Lun Liu, Wei-Sheng Lai, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Learning to see through obstructions with layered decomposition. *IEEE TPAMI*, 2021. (page 2)

[39] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. SparseNeuS: Fast generalizable neural surface reconstruction from sparse views. In *ECCV*, 2022. (pages 1 and 2)

[40] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. (page 2)

[41] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *CVPR*, 2019. (page 2)

[42] 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. In *ECCV*, 2020. (pages 1, 2, 3, and 6)

[43] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G. Derpanis, Jonathan Kelly, Marcus A. Brubaker, Igor Gilitschenski, and Alex Levinshtein. SPIn-NeRF: Multiview segmentation and perceptual inpainting with neural radiance fields. In *CVPR*, 2023. (page 3)

[44] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. LENS: Localization enhanced by NeRF synthesis. In *Conference on Robot Learning*, 2022. (page 1)

[45] Shohei Mori, Okan Erat, Wolfgang Broll, Hideo Saito, Dieter Schmalstieg, and Denis Kalkofen. InpaintFusion: incremental RGB-D inpainting for 3D scenes. *IEEE TVCG*, 2020. (page 2)

[46] Shohei Mori, Dieter Schmalstieg, and Denis Kalkofen. Good keyframes to inpaint. *IEEE TVCG*, 2022. (page 2)

[47] Thomas Müller, Alex Evans, Christoph Schied, Marco Foco, András Bódis-Szomorú, Isaac Deutsch, Michael Shelley, and Alexander Keller. Instant neural radiance fields. In *ACM SIGGRAPH 2022*. 2022. (page 1)

[48] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3D representations from natural images. In *CVPR*, 2019. (page 2)

[49] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, 2022. (pages 2 and 5)

[50] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *CVPR*, 2021. (page 2)

[51] Selcen Ozturkcan. Service innovation: Using augmented reality in the IKEA Place app. *Journal of Information Technology Teaching Cases*, 11(1), 2021. (page 1)

[52] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. (page 2)

[53] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. (page 2)

[54] Julien Philip and George Drettakis. Plane-based multi-view inpainting for image-based rendering in large scenes. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 2018. (page 3)

[55] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *CVPR*, 2022. (page 2)

[56] Chris Rockwell, David F Fouhey, and Justin Johnson. Pixel-Synth: Generating a 3D-consistent experience from a single image. In *ICCV*, 2021. (pages 2, 6, 7, and 8)

[57] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *CVPR*, 2022. (pages 2 and 3)

[58] Martin Runz, Kejie Li, Meng Tang, Lingni Ma, Chen Kong, Tanner Schmidt, Ian Reid, Lourdes Agapito, Julian Straub, Steven Lovegrove, et al. FroDO: From detections to 3D objects. In *CVPR*, 2020. (page 2)

[59] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, 2019. (page 2)

[60] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3D human digitization. In *CVPR*, 2020. (page 2)

[61] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2021. (page 2)

[62] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. SimpleRecon: 3D reconstruction without 3D convolutions. In *ECCV*, 2022. (pages 6 and 7)

[63] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. (pages 2 and 3)

[64] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. (pages 2 and 3)

[65] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3D-aware image synthesis. *NeurIPS*, 2020. (page 2)

[66] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3D photography using context-aware layered depth inpainting. In *CVPR*, 2020. (page 3)

[67] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew Davison. iMAP: Implicit mapping and positioning in real-time. In *ICCV*, 2021. (page 2)

[68] Mohamad Zaidi Sulaiman, Mohd Nasiruddin Abdul Aziz, Mohd Haidar Abu Bakar, Nur Akma Halili, and Muhammad Asri Azuddin. Matterport: virtual tour as a new marketing approach in real estate business during pandemic COVID-19. In *International Conference of Innovation in Media and Visual Design (IMDES)*, 2020. (page 1)

[69] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *WACV*, 2022. (pages 2, 4, 5, 6, and 7)

[70] Theo Thonat, Eli Shechtman, Sylvain Paris, and George Drettakis. Multi-view inpainting for image-based scene editing and rendering. In *3DV*, 2016. (page 2)

[71] Jonathan Tremblay, Moustafa Meshry, Alex Evans, Jan Kautz, Alexander Keller, Sameh Khamis, Charles Loop, Nathan Morrical, Koki Nagano, Towaki Takikawa, and Stan Birchfield. RTMV: A ray-traced multi-view synthetic dataset for novel view synthesis. *ECCVW*, 2022. (page 2)

[72] Qianqian Wang, Zhengqi Li, David Salesin, Noah Snavely, Brian Curless, and Janne Kontkanen. 3D moments from near-duplicate photos. In *CVPR*, 2022. (page 3)

[73] 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*, 2004. (page 6)

[74] Qianyi Wu, Xian Liu, Yuedong Chen, Kejie Li, Chuanxia Zheng, Jianfei Cai, and Jianmin Zheng. Object-compositional neural implicit surfaces. In *ECCV*, 2022. (pages 1 and 2)

[75] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *CVPR*, 2019. (pages 2 and 5)

[76] Xiaogang Xu, Ying-Cong Chen, and Jiaya Jia. View independent generative adversarial network for novel view synthesis. In *ICCV*, 2019. (page 2)

[77] 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 *ICCV*, 2021. (pages 1, 2, 6, 7, and 8)

[78] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021. (page 2)

[79] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018. (page 2)

[80] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. NeRF-editing: geometry editing of neural radiance fields. In *CVPR*, 2022. (page 1)

[81] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *ECCV*, 2020. (page 2)

[82] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Learning pyramid-context encoder network for high-quality image inpainting. In *CVPR*, 2019. (page 2)

[83] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. (page 6)

[84] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021. (page 2)