DOI: 10.1111/cgf.14977
Pacific Graphics 2023
R. Chaine, Z. Deng, and M. H. Kim
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 7

A Post Processing Technique to Automatically Remove Floater Artifacts in Neural Radiance Fields

T. Wirth[1], A. Rak[1], V. Knauthe[1] and D. W. Fellner[1,2,3]

[1]Technical University of Darmstadt, Interactive Graphics Systems Group, Germany [2]Fraunhofer IGD, Darmstadt, Germany
[3]Graz University of Technology, Institute of Computer Graphics and Knowledge Visualization, Austria

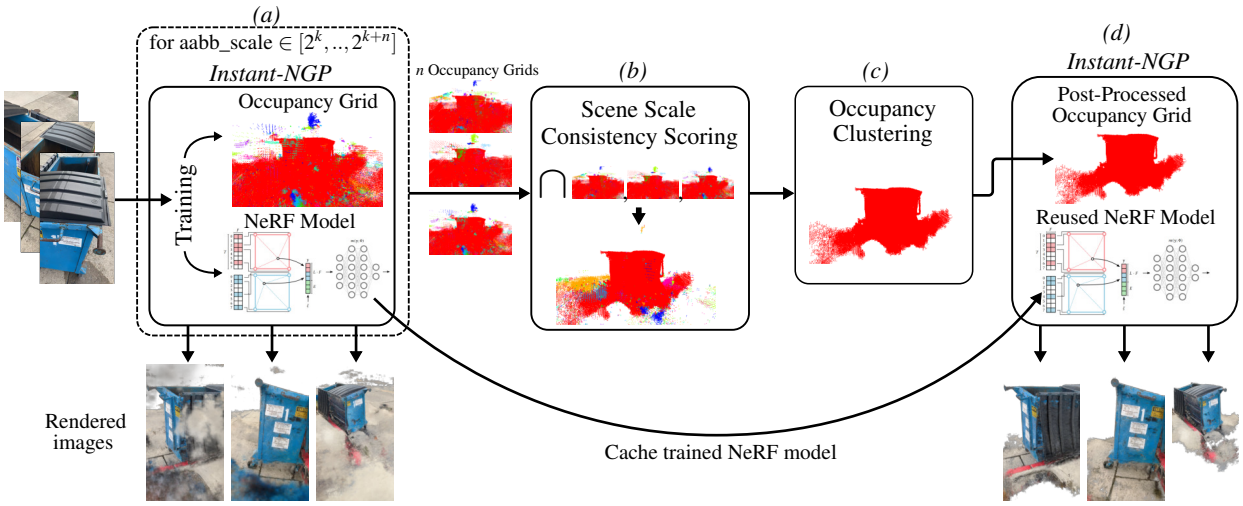

Figure 1: *Our post-processing pipeline and rendered evaluation views of an in-the-wild scene before and after post-processing. In (a), n Instant-NGP models are trained on the same set of training images with varying scene scales, which yields n occupancy grids, that contain the position of relevant densities in the scene. Rendered views distant to the training image's camera trajectory contain cloudy artifacts that obstruct the view and decrease visual quality. The occupancy grids are processed using our Scene Scale Consistency Scoring (SSCS) approach in (b), which merges them into a single filtered occupancy grid, where artifacts are severed from the canonical scene geometry. This allows our Occupancy Clustering approach to identify artifacts as separate clusters and prune their densities in (c). Reusing the initial NeRF model, the post-processed occupancy grid is leveraged to render novel views free of floater artifacts in (d).*

Abstract

Neural Radiance Fields have revolutionized Novel View Synthesis by providing impressive levels of realism. However, in most in-the-wild scenes they suffer from floater artifacts that occur due to sparse input images or strong view-dependent effects. We propose an approach that uses neighborhood based clustering and a consistency metric on NeRF models trained on different scene scales to identify regions that contain floater artifacts based on Instant-NGPs multiscale occupancy grids. These occupancy grids contain the position of relevant optical densities in the scene. By pruning the regions that we identified as containing floater artifacts, they are omitted during the rendering process, leading to higher quality resulting images. Our approach has no negative runtime implications for the rendering process and does not require retraining of the underlying Multi Layer Perceptron. We show on a qualitative base, that our approach is suited to remove floater artifacts while preserving most of the scenes relevant geometry. Furthermore, we conduct a comparison to state-of-the-art techniques on the Nerfbusters dataset, that was created with measuring the implications of floater artifacts in mind. This comparison shows, that our method outperforms currently available techniques. Our approach does not require additional user input, but can be be used in an interactive manner. In general, the presented approach is applicable to every architecture that uses an explicit representation of a scene's occupancy distribution to accelerate the rendering process.

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

delivered by EUROGRAPHICS DIGITAL LIBRARY
www.eg.org diglib.eg.org

1. Introduction

Neural Radiance Fields (NeRFs) [MST*21; BMV*22] have brought the realism of Novel View Synthesis based on multi-view images to a new level. Their expressiveness is based on a ray-wise optimization of a volumetric scene representation that is encoded in a Multi-Layer Perceptron (MLP).
Later contributions [MESK22; CFHT23] have accelerated the training of the underlying MLP to a few seconds and offer interactive frame rates at rendering time on end consumer hardware. However, especially in real-world scenes, NeRF architectures are prone to cloudy artifacts (*floaters*), which arise when the underlying MLP assigns non-zero optical densities to regions without objects or structures in the original scene. These artifacts can be caused by several circumstances like sparse inputs or strong view-dependant effects [PSB*21; PSH*21; PCPM21; TTG*21; MRS*21] that result in ambiguities [WWT*23; LTT23]. They are especially visible in novel views that are far away from from the training image's camera poses. Examples of these artifacts are illustrated in the rendered images on the left side of figure 1. Recent publications have addressed this issue by decomposing the scene into its view-dependant and independent components [LTT23], scoring scene regions based on the frequency of appearance in the input views [WWT*23] or suppressing them in a retaining step after interactive annotation of the relevant regions [JKK*23]. However, these contributions require retraining of the model, adaptation of the training process or manual annotation. Furthermore, they usually have inherent problems with processing transparent structures or texture hallucination.
In contrast to that, we propose a method that omits *floater* artifacts during the rendering process without altering the underlying MLP. Instant-NGP [MESK22] explicitly stores information about a scene's density distribution in an occupancy grid. During rendering, regions that do not contribute to the resulting image are identified using this density grid and therefore do not trigger an MLP query, accelerating the rendering process significantly.
Our method uses heuristics to identify regions with assigned density in the occupancy grid that are likely not to contain objects in the original scene. For this purpose we propose a neighborhood-based global clustering approach, that is optimized for the hierarchically arranged occupancy grid of Instant-NGP. Assuming that most of the scene consists of one coherent region of density, we flag identified clusters in descending order of size until they cover a predefined portion of the scene. The remaining clusters are likely to be *floater* artifact, which are then removed from the occupancy grid.
Furthermore, we leverage the inconsistent nature of artifacts under different scene scales. During the training process, a bounding box defines in which scene volume ray marching is performed, thereby defining in which region densities can arise in the scene. We train multiple models of a scene with different sizes of this bounding box - dictated by Instant-NGP's *aabb_scale* parameter - and omit regions for which the occupancy is inconsistent between these different NeRF models. This follows the intuition, that artifacts manifest in different areas for different bounding box scales, while objects that are present in the scene are expected to be modelled in the consistent locations.
Combining both methods, we show on a qualitative and quantitative basis, that our approach is suited to mitigate floater artifacts in NeRFs. The proposed approach does not alter the underlying MLP and therefore does not require retraining. Furthermore, it is easily extendable to any NeRF architecture that leverages an explicit density or occupancy representation to accelerate rendering. The overhead time of our post-processing step is negligible compared to the training time of Instant-NGP. Since we only alter the occupancy field of Instant-NGP, our approach has no influence on the rendering performance of the underlying NeRF technology. In summary, our main contributions are:

- a fast region-based clustering approach on multi-resolution occupancy fields that is consistent through different resolution levels,
- an approach to detect artifacts by comparing occupancy of models trained with different scene scales,
- a strategy to leverage these approaches to adapt occupancy fields in a way, that suppresses undesired NeRF artifacts during rendering time, without the requirement to alter the underlying MLP and negative performance implications,
- a quantitative evaluation on a state-of-the-art dataset [WWT*23] that is designed to address the problem of floaters.

To our knowledge, the presented approach is the first approach that mitigates *floater* artifacts without user input, that does not require MLP retraining. Our code is available under https://github.com/tristanwirth/floaty-removal.

2. Related Work

2.1. Neural Radiance Fields

NeRFs [MST*21] have brought the level of realism for novel view synthesis to a new level. This is achieved by training a neural network that models a 3D scene as a continuous color and density map. Given a 5D input vector consisting of 3D location (x, y, z) and a 2D viewing direction (θ, ϕ) this network returns a color and density estimate for a ray with the given origin and viewing direction. A novel view is generated by marching along the camera ray for each pixel by iterative querying this network and accumulating the color values weighted by the corresponding densities.
Since their proposal, subsequent contribution have addressed limitations of their approach and adapted it to other challenges. Recent publications extend the ability of NeRFs to support dynamic scenes [PSB*21; PSH*21; PCPM21; TTG*21; LNSW21], accelerating inference time [RPLG21; MESK22; YLT*21; FYT*22; CXG*22; LSS*21; WZL*22; CFHT23], making them robust against the challenges of in-the-wild image capture [MRS*21; TCY*22; MHM*22; RLS*22], reducing the required image count [YYTK21; DLZR22; NBM*22; YPW23; RMY*22] and enabling dynamic relighting [ZSD*21; SDZ*21; MHS*22].
We base our work on Instant-NGP [MESK22], which restricts our approach to rigid scenes under fixed lighting conditions.

2.2. Editing NeRFs

For multiple tasks, from removing undesired objects and artifacts to the composition of objects from different sources, it is desirable to edit scenes represented as NeRFs. Due to the implicit storage of scene geometry and color, this task is not straightforward compared to explicitly stored scenes like triangle meshes. A plethora of work

© 2023 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

addresses this issue, offering a diverse toolkit from object insertion and deletion to color, lighting and geometry alteration or translation and rotation of existing objects.

Niemeyer et al. [NG21] decompose a scene into multiple NeRFs, which contain one object each, allowing translation, rotation and stretching of individual objects. NeRF-Editing [YSL*22] and [JKK*23] establish a correspondence between an explicit mesh representation and the implicit neural representation of the NeRF. Object manipulations are performed by editing this mesh representation and bending or stretching the camera rays accordingly, while they travel through these cages during a retraining process. NeuMesh [YBZ*22] encode NeRFs by disentangling geometry and texture, allowing them to perform manipulations on the resulting mesh. Chen et al. [CLW23] propose NeuralEditor that stores a corresponding K-D tree-guided voxel representation, that produces high quality point clouds, on which object manipulations can be performed. Liu et al. [LZZ*21] extend the neural network architecture by a shape branch, allowing shape and color editing on object categories based on 2D user scribbles. Lazola et al. [LGO*23] propose Control-Nerf, that leverages scene-specific properties to seamlessly add objects into other NeRFs scenes.

Furthermore, recent work presents strategies for object manipulation in dynamic scenes [KYK*22; ZLX23], object removal that leverages 2D inpainting [YFYL23; WGM*23] and object and scene manipulation techniques, that are based on text prompts [HTE*23; YFYL23].

2.3. Floater Mitigation and Removal

NeRFs in general are prone to artifacts, that float around in the scene. Barron et al. [BMV*22] introduce the term *floater* for these kind of artifacts and define them as small disconnected regions of volumetric dense space which serve to explain some aspects of a subset of input views, but look like blurry clouds when viewed from a different angle. In the literature, their appearance is linked to several circumstances in rigid images with fixed lighting: sub-optimal camera registration [WWT*23], sparse inputs [WWT*23; LTT23], strong view-dependent visual effects [LTT23] as well as errors in the estimated scene geometry and divergent behavior in the beginning of the training process [NBM*22].

Several approaches have been published to mitigate the appearance of *floaters* by optimizing the NeRF training process. [LMTL21; WWX*21] optimize the camera pose estimation to mitigate artifacts in general. Mip-NeRF 360 [BMV*22] prevents the formation of *floaters* during the training process by introducing a distortion loss. Niemeyer et al. [NBM*22] reduce the number of artifacts in sparse input scenarios by regularizing the scene geometry and appearance of patches rendered from unobserved view points and annealing the ray sampling space, which reduces errors in the estimated scene geometry and divergent behavior at the start of the training. Nerfbusters [WWT*23] and ViP-NeRF [SS23] introduce information about the visibility of regions in the scene, to mitigate problems with sparse inputs. Liu et al. [LTT23] propose Clean-NeRF, which disentangles view-dependant and view-independent parts of the scene to gain a more robust geometry estimate. Thereby, they reduce the view-dependent effects, that create *floater* artifacts. There are existing approaches that remove *floaters* in post-processing procedures. Liu et al. [LKC*23] train a coarse NeRF model on a sparse image set, upon which they generate pseudo-observations that are used to produce a high-quality reconstruction. They achieve that by using a rectification latent diffusion model that generates image conditionals based on RGB images and depth maps from the coarse model. Nerfbusters [WWT*23] utilize a diffusion model that refines densities and color on randomly allocated local 3D cubes. To mitigate *floaters*, they penalize densities in regions that are only seen by few training views. However, due to the use of diffusion models, their approach suffers from texture hallucination. Warburg et al. [WWT*23] present a dataset, that aspires to incorporate the effect of *floater* artifacts into the quantitative analysis of NeRF architectures. This is achieved by providing two camera trajectories per scene. Training on one of the trajectories and evaluating on the second one, the difference between training and evaluation camera poses is higher, which is usually linked with a higher probability for *floater* appearance. We evaluate our technique in comparison to their approach on their dataset in section 4.2.

Jambon et al. [JKK*23] present NeRFshop, which is focused on manually editing NeRFs by manually annotating and manipulating scene volumes. During a retraining step the rays path is bend in these manipulated cages, to acquire a new model that incorporates the performed edits. By manually annotating *floaters* and ignoring their densities during retraining, they acquire a new NeRF model, that is free of *floaters*. In comparison to our fully automatic approach, however, they require manual annotation of artifacts.

The aforementioned post-processing strategies require retraining of the underlying MLP to remove *floater* artifacts. In contrast to that, we present an approach, that automatically removes these artifacts without altering the underlying MLP. Furthermore, our approach has no impact on the rendering performance.

3. Methodology

In this section, we describe our proposed method to suppress *floater* artifacts in a post-processing step leveraging an explicit occupancy structure. Our contribution is based on Instant-NGP [MESK22], which is the de facto standard for storing and rendering NeRFs, due to its low training times and interactive frame rates during rendering.

Instant-NGP uses multiscale occupancy grids, to skip empty space during ray marching. For real-world scenes, they use $K \in [1,5]$ grids, where each grid has a resolution of 128^3. Each grid spans an iteratively growing domain around the center point $(0.5,0.5,0.5)$ with the dimension $[-2^{k-1}+0.5, 2^{k-1}+0.5]^3$. For each grid cell they store the occupancy information in a single bit, where 1 indicates, that there is relevant density in the cell and 0 indicates, that there is no relevant density, thus ignoring the cell during rendering. Our approach suppresses *floater* artifacts by identifying cells in the density grid, that belong to a *floater* artifact and setting its occupancy value to 0. Thereby, while the information of the *floater* artifact is still encoded in the underlying MLP, it is ignored during rendering and therefore does not appear in the resulting images. This leads to a significant improvement in the resulting image quality.

To get a base model for our post-processing methodology, we train off-the-shelf Instant-NGP models based on the input images. We leverage the information from the resulting multiscale occupancy grid to identify regions that belong to *floater* artifacts by applying

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

two heuristics: First the occupancy grid is filtered based on a Scene Scale Consistency Score. Then, we apply our Occupancy Clustering step on the filtered occupancy grid. These heuristics each work exclusively with an occupancy grid without taking color information into account. They are described in the following.
Figure 1 depicts our system architecture with exemplary intermediate representations after each step.

3.1. Occupancy Clustering

For most static scenes, the relevant part of the scene consists of one coherent volume of density. This is based on the fact, that objects that are denser than air lie on other objects on the ground - which is consistent in itself - and objects that are less dense than air have to be locked into position by some object, in order to not float away - effectively leaving the scene. We consider scenes, that do not fulfill this coherency assumption out of scope. These scene properties are extensively discussed in section 5.
Assuming a coherent core of the scene, we leverage the multiscale occupancy grid of Instant-NGP to perform a neighborhood based global clustering, that is consistent through the different scales. The Instant-NGP multi scale occupancy grid consists of multiple occupancy grids G_k that cover the space $[-2^{k-1}+0.5, 2^{k-1}+0.5]^3$ respectively. Each of these grids consists of 128^3 cells $C_k(x_I, y_I, z_I)$ with indices $x_I, y_I, z_I \in [0, 127]$. A point P with coordinates $(x_P, y_P, z_P)^T$ is part of the cell $C_k(x_I, y_I, z_I)$ when [†]:

$$\begin{pmatrix} x_P \\ y_P \\ z_P \end{pmatrix} \in \begin{bmatrix} (-2^{k-1}+0.5)+2^{k-8}x_I, (-2^{k-1}+0.5)+2^{k-8}(x_I+1) \\ (-2^{k-1}+0.5)+2^{k-8}y_I, (-2^{k-1}+0.5)+2^{k-8}(y_I+1) \\ (-2^{k-1}+0.5)+2^{k-8}z_I, (-2^{k-1}+0.5)+2^{k-8}(z_I+1) \end{bmatrix}$$

Our clustering approach first compiles a single set of cells O, which consists of non-overlapping cells of all multi-scale grids G_k:

$$O = G_1 \cup \bigcup \{G_k \setminus G_{k-1} \mid k \in [2, .., K]\}$$

Any point P in the scene space $[-2^{K-1}+0.5, 2^{K-1}+0.5]^3$ is covered by exactly one cell in the set and this cell is part of the grid with the highest resolution (with the lowest level k). O is used as starting point for our clustering approach and for updating the lower resolution occupancy grids in a cascading, consistent manner after the clustering process.
We perform a neighborhood based clustering approach to identify coherent clusters of occupancy. In the following, we call cells with an occupancy value of 1 occupied. The clustering algorithm picks an occupied cell that is not assigned to a cluster from O at random and assigns this cell to a new cluster. Any time a cell is assigned to a cluster - including this first cell -, any occupied cell that is adjacent to it - meaning they share a non-zero area border - is assigned to the same cluster. When there are no more adjacent cells left, we repeat the process by picking a new unprocessed cell at random, assigning it to a new cluster and performing the same neighborhood based flood fill algorithm to this cluster. When there is no unassigned, occupied cell left, the clustering concludes.
For scenes that fullfil our coherence assumption, the cluster that covers the most volume depicts the relevant coherent scene geometry. Since the other clusters are not coherent with the scene geometry, they are omitted as artifacts. However, in reality Instant-NGP does not always produce perfectly coherent scene occupancies. To mitigate that fact, we iteratively label clusters as relevant in descending order of covered scene volume until at least 85% of the occupied volume of the scene is covered. The clusters that are left unlabeled, are then assumed to be artifacts.

Figure 2: *A NeRF artifact that is coherent with the scene geometry.*

We set the occupancy of the cells in the clusters that are assumed to be artifacts to 0. That way, the content of these regions is omitted during the rendering process. To ensure consistency between the differently scaled occupancy grids, each cell that is not part of O is iteratively updated. Each cell in $G_k \cap G_{k-1}$ for $k > 1$ was not part of set O, and is therefore not consistent with the applied changes to the multiscale occupancy grid. Each of these cells in G_k contains 8 cells in G_{k-1}. If none of those cells in G_{k-1} is occupied after the the update, the occupancy value of the parent cell in G_k is set to 0. This process is iteratively repeated for any $k \in [1, .., K]$ until the multiscale occupancy map is consistent.
Note that there are edge cases, in which floaters are close to relevant scene geometry. In these cases, it is possible, that while the clusters are removed from the occupancy grid with higher resolutions, they are not removed from occupancy grids with lower resolution. Therefore, this part of the *floater* artifact is not suppressed, when the lower resolution occupancy field is utilized for the rendering process. This effect is further discussed in section 5.

3.2. Scene Scale Consistency Scoring (SSCS)

Preliminary experiments showed, that while the proposed Occupancy Clustering approach is effective for the better part of the scenes, there are more challenging cases. These cases occur when there are *floater* artifacts, that are coherent with the relevant scene geometry (see figure 2). Our clustering approach is not able to detect these artifacts. We address this issue by proposing Scene Scale Consistency Scoring (SSCS). Instant-NGP [MESK22] uses an input parameter *aabb_scale* that controls, in which space of the scene ray marching is performed. In this context, *aabb_scale* depicts the scaling factor of the unit cube. The authors recommend to use powers of 2 in order to align its borders with an occupancy grid. The proposed SSCS follows the intuition, that the position of objects, that constitute the relevant scene geometries, is invariant to changes of the *aabb_scale* factor. Artifacts, on the other hand, are based on incorrect interpretation in regions with sparse input. These artifacts manifest in slightly different regions of the scene under different scaling factors. We utilize this property by comparing the occupancy grid of Instant-NGP models trained with different scales.

† Note that 2^{k-8} is equivalent to $\frac{2^{k-1}}{128}$

© 2023 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

(a) *aabb_scale* = 8 (b) *aabb_scale* = 16 (c) SSCS (d) Cluster (e) SSCS + Cluster

Figure 3: *Illustration of the effects of SSCS. (a, b) Renderings by two models trained on different aabb_scales (8 respectively 16), showing a consistent object in the center and differing artifacts in the surroundings, (c) after applying SSCS, (d) clustering without SSCS, showing persisting artifacts, (e) clustering after SSCS, successfully removing the floater artifacts.*

Using *aabb_scales* $s \in S = [8, 16, 32]$, for each cell $C_k(x_I, y_I, z_I)$ we compute their SSCS $\in [0, 1]$:

$$SSCS(C_k(x_I, y_I, z_I)) = \sum_{s \in S} C_{k,s}(x_I, y_I, z_I) \Big/ N(S, C_{k,s}(x_I, y_I, z_I))$$

where $C_{k,s}$ is C_k for the model with scale s and $N(S, C_{k,s})$ is the number of scene scales $s \in S$, whose corresponding extents contain the cell $C_{k,s}$. The score is equal to the fraction of scaled scenes in which the occupancy of the cell is set, taking into account the number of scaled scenes in which the cell is theoretically setable, i.e. the cell is inside of the training volume. We consider a cell as inconsistently set, if its $SSCS < 1$, meaning that there is at least one scaled model that does not carry density there. These cells occupancy is then set to 0, omitting them in the rendering process.

We perform this SSCS filtering step before Occupancy Clustering. In most cases, this step interferes with the coherence between relevant scene geometry and *floater* artifacts. Therefore, while SSCS filtering itself only detects parts of the occuring artifacts, it enables our Occupancy Clustering to detect *floaters* more reliably.

Figure 3 illustrates the behavior and benefits of employing SSCS. Especially on the garbage dataset the effects of SSCS become apparent. The models for $s = 8$ and $s = 16$ show a consistent depiction of the garbage can with a blue *floater* artifact in front. Our clustering approach is not able to prune these volumes due to their coherence with the scene geometry. However, the illustration depicts how most of these artifacts are inconsistent between the two *aabb_scale*s. Therefore, SSCS is able to remove most of these artifacts until only a small blue *floater* remains. This residual *floater* is then pruned by our Occupancy Clustering step due to its incoherence with the scene geometry, successfully removing the artifact at hand.

4. Evaluation

In this section, we perform a qualitative analysis of the novel view quality of our method in comparison to unprocessed Instant-NGP renderings, and evaluate our method quantitatively in comparison to the state-of-the-art on the Nerfbusters dataset [WWT*23], which is designed to give insight into performance of NeRF models when rendering novel viewpoints, that are distant from the training data. In these cases, the appearance of *floaters* is more likely. All experiments are conducted on an Nvidia RTX 3090 GPU.

4.1. Comparison to Instant-NGP

As stated in section 3.2, our Occupancy Clustering approach assumes that artifacts are structurally severed from the canonical scene geometry. This is often not the case for datasets with sparse training views or inconsistent illumination. Therefore, to qualitatively assess the effectiveness of our Occupancy Clustering approach, we employ the MIP-NeRF 360 dataset [BMV*22], which provides exhaustive views for the majority of the scene, uses fixed camera parameters and was shot on overcast days, assuring minimal illumination variances by cast shadows. While these measures are effective on their own to suppress artifacts in the learned model, renderings of novel views outside of the training image's camera trajectory still comprise floaters that negatively impact visual quality.

Implementation Details. We first train a vanilla Instant-NGP NeRF model on several scenes of the MIP-NeRF 360 dataset for a duration of one minute per scene. After training, we fixate both the trained NeRF model and the corresponding occupancy grid and render novel views from a predefined camera trajectory. The occupancy grid is then post-processed using our Occupancy Clustering approach, eliminating *floater* artifacts. Finally, the scenes are rendered again on the same camera trajectory. Videos of the scenes are supplied in the supplemental material.

Results. Figure 4 displays frames of the aforementioned camera trajectories for the Instant-NGP model before and after post-processing. Despite the exhaustive views provided in the dataset,

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

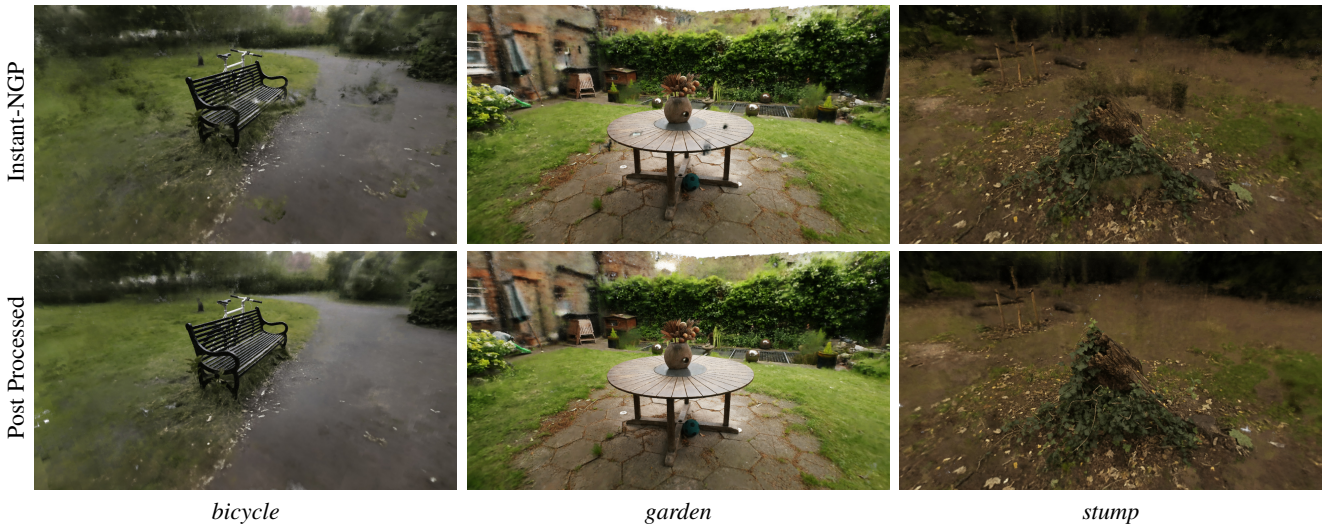

Figure 4: *MIP-NeRF 360 scenes 'bicycle', 'garden' and 'stump' shown before and after our post processing approach [BMV*22]. Floating artifacts can be observed occluding the street (left), the table (middle) and the stump (right). Our clustering approach successfully removes these artifacts without impacting the integrity of the remainder of the scenes, improving the resulting image quality.*

the trained model still contains floating artifacts of varying sizes and severity depending on the scene. In the *bicycle* scene, large floaters can be observed occluding the street. The *garden* scene contains fewer and smaller floaters. The background in the *stump* scene consists mainly of homogeneous foliage and soil and thus facilitates the formation of a multitude of floaters close to the object of interest. The floaters observed in these scenes are not coherent with the canonical geometry of the scene, i.e., they do not share neighboring cells in the corresponding occupancy grids. This allows our Occupancy Clustering approach to correctly identify them as artifacts and remove them from the occupancy grids consequently. As shown in figure 4, the resulting post-processed views do not contain the floaters, as the corresponding cells in the occupancy grid were omitted in the rendering process. This improves the overall visual quality of the rendered views.

4.2. Comparison on Specialized Dataset

The capturing constraints present in the MIP-NeRF 360 dataset are often not given for casually captured, in-the-wild scenes. Additionally, quantitative evaluation of *floater* removal using this or a similar dataset is futile, as there is no separate set of evaluation ground truth images. This necessitates the evaluative usage of a subset of the images from the training image camera trajectory, which drastically reduces the amount of visible *floater* artifacts in the rendered views. As a consequence, *floater* removal does not have a significant impact on the visual quality. Warburg et al. propose the Nerfbusters dataset [WWT*23], which attempts to mitigate both of these issues: The training images are a set of casually captured sparse views of various in-the-wild scenes. An additional set of evaluation images per scene is captured on a vastly different camera trajectory, providing previously unseen views of the scene. The authors also propose a *floater* mitigation approach and quantitative evaluation results comparing their post-processed renderings to a vanilla Nerfacto model [TWN*23].

Implementation Details. In accordance to Warburg et al. we first train a pseudo ground truth Instant-NGP model using both the training- and evaluation image sets [WWT*23]. This model serves as the ground truth for computing PSNR and SSIM metrics. We then train multiple Instant-NGP NeRF models on the training images with scales $aabb\_scale \in [8, 16, 32]$, each of which yield an unprocessed occupancy grid. The NeRF model trained with a scale of 16 serves as the evaluation model for the before and after post-processing renderings, as this is Instant-NGP's default scale and we chose to not manually tune parameters on a per-scene basis for this evaluation study. We leverage the occupancy grids to perform our SSCS strategy, as was described in section 3.2. Finally, the intermediate grid result from SSCS is processed using our Occupancy Clustering strategy to eliminate remaining *floater* artifacts. The evaluation views are rendered for each entire camera trajectory. For comparative purposes, we also render the evaluation views using the occupancy grids resulting from only applying SSCS or Occupancy Clustering, respectively. Similar to Warburg et al. we employ masked PSNR and SSIM metrics on a per-image basis, which consider only the image areas visible in the training views, and report the per-scene average.

Results. Figures 5 and 6 show rendered evaluation views of the aforementioned NeRF model variants and the corresponding Nerfbusters renderings side-by-side. Table 1 presents the PSNR and SSIM and coverage scores of each approach. The latter describes the percentage of the resulting image that is covered by pixels with density. As each of the post-processing steps prunes unwanted scene geometry, the coverage tends to decrease with each of them. In the unprocessed Instant NGP renderings, a multitude of artifacts can be observed. Some of these artifacts incoherently float in the air

© 2023 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

Figure 5: *Evaluation views of the Nerfbusters scenes 'car', 'century', 'garbage', 'pikachu' and 'picnic'. Renders of the Pseudo-GT Instant-NGP model, the unprocessed Instant-NGP model using only training views, renders using post-processed grids with only Occupancy Clustering and SSCS, the Nerfbusters renders by Warburg et al. [WWT*23] and our combined strategy are shown side-by-side.*

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

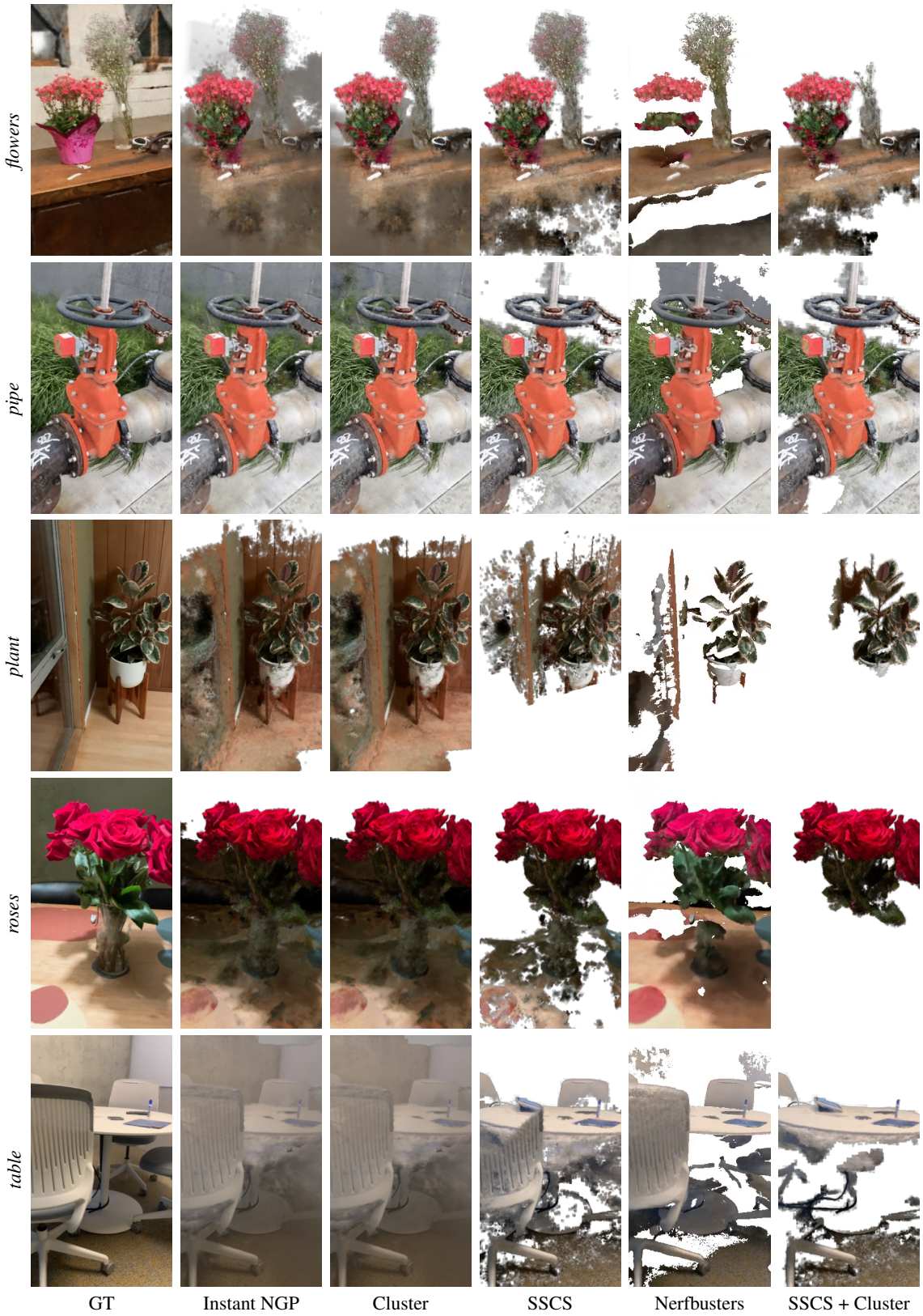

Figure 6: *Evaluation views of the Nerfbusters scenes 'flower', 'pipe', 'plant', 'roses' and 'table'. Renders of the Pseudo-GT Instant-NGP model, the unprocessed Instant-NGP model using only training views, renders using post-processed grids with only Occupancy Clustering and SSCS, the Nerfbusters renders by Warburg et al. [WWT*23] and our combined strategy are shown side-by-side.*

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

	Instant-NGP			Cluster			SSCS			Nerfbusters			SSCS + Cluster		
	PSNR	SSIM	Cvg	PSNR	SSIM	Cvg	PSNR	SSIM	Cvg	PSNR	SSIM	Cvg	PSNR	SSIM	Cvg
car	16.93	0.5607	0.97	16.99	0.6116	0.92	17.68	**0.6411**	0.50	17.40	0.5898	0.55	**17.83**	0.6389	0.36
century	16.12	0.5028	0.90	15.79	0.5151	0.87	16.53	0.6324	0.73	15.01	0.5028	0.72	**16.76**	**0.6541**	0.62
flowers	12.69	0.4311	0.86	13.46	0.5032	0.72	14.72	0.5508	0.67	**15.52**	0.5086	0.62	15.48	**0.5657**	0.48
garbage	16.04	0.5503	0.87	**16.49**	**0.6115**	0.83	16.37	0.5869	0.60	15.86	0.4466	0.63	16.44	0.5809	0.56
picnic	15.81	0.4375	0.72	15.70	0.4636	0.69	16.06	0.5443	0.54	15.72	0.4560	0.58	**16.22**	**0.5485**	0.39
pikachu	21.15	0.7865	0.84	25.18	0.9144	0.46	25.77	0.9369	0.48	25.71	0.9048	0.28	**28.40**	**0.9420**	0.25
pipe	18.84	0.6248	0.99	18.95	0.6502	0.97	19.14	0.7021	0.81	**19.23**	0.6165	0.82	19.03	**0.7039**	0.77
plant	18.08	0.6801	0.94	17.78	0.7183	0.90	20.40	0.7549	0.48	20.20	0.7535	0.25	**20.82**	**0.7889**	0.19
roses	16.25	0.6099	0.92	16.65	0.6725	0.91	17.54	0.6456	0.68	**19.14**	**0.7149**	0.87	18.44	0.6085	0.34
table	16.40	0.6618	0.85	16.31	**0.7451**	0.83	15.29	0.7347	0.62	**18.23**	0.7060	0.69	12.77	0.6562	0.42
Average	16.83	0.5846	0.89	17.33	0.6406	0.81	17.95	**0.6730**	0.61	18.20	0.6200	0.60	**18.22**	0.6688	0.44

Table 1: *Evaluation scores of the Nerfbusters dataset on renderings of the base Instant-NGP model, on the post-processed variants using our clustering, scene scale consistency scoring, and combined approaches and on the state-of-the-art Nerfbusters approach. On average, our combined approach outperforms Nerfbusters in visual quality measured by the PSNR and SSIM metrics. We achieve less coverage in the rendered images as we only render coherent clusters, while Nerfbusters tends to generate floating, incoherent chunks. The scenes 'aloe' and 'art' are omitted, since Instant-NGP was not able to train a coherent model with the provided camera poses.*

Figure 7: *Comparison of scene coherency between Nerfbusters and our approach [WWT*23]. Nerfbusters tends to remove scene geometry that is sparsely visible in the training data. This can result in floating objects (left) or textures with holes (right). Our Occupancy Coherency assumption avoids this behavior by clustering coherent regions.*

Figure 8: *Thin structures such as flower stalks can result in the underlying model assuming low densities in these areas. This results in our SSCS approach severing the structural connection between the stalks and the flowers in the 'flowers' and 'roses' Nerfbuster scenes. The subsequent cluster approach then removes the severed cluster(s), which may be important geometry of the scene.*

(*picnic*), some obstruct the entire view (*century*, *table*) and some are attached to the canonical scene geometry (*garbage*). The visual quality of the rendered images suffers from all of these artifacts. This manifests in the PSNR and SSIM scores, which are the lowest of all approaches for the vast majority of scenes.

Using only our Occupancy Clustering approach - analogous to Section 4.1 - a reduction of floating artifacts can be observed (most prominently in *car* and *picnic*). However, artifacts coherent with the scene geometry are not removed and remain having a negative impact on the visual quality. This also shows in the quantitative results, where Occupancy Clustering only moderately improves the results of the unprocessed renderings.

Our Scene Scale Consistency Scoring severs the structural connection between artifacts and the remainder of the scene geometry. This may leave many noisy artifacts behind (see *car*, *pikachu* and *picnic*). The subsequent Occupancy Clustering step prunes these artifacts and leaves behind a coherent scene geometry. This improves the image quality over Nerfbusters, as there are fewer holes in the scene's texture (*century*, *garbage*) and no additional incoherent floating geometry is introduced (*flowers*, *plant*). In the presence of thin structures such as flower stalks, our approach may remove densities that were part of the canonical scene geometry (*flowers*, *roses*). This is discussed in section 5. These qualitative observations manifest in the PSNR and SSIM scores, where our combined approach outperforms Nerfbusters for the majority of scenes, with notable exceptions being *flowers* and *roses*. Nerfbusters achieves a higher coverage for all scenes, which for the most part is due to the additional incoherent geometry floating around in most of the post-processed scenes. In contrast to that, our technique results in coherent scene geometry. A direct comparison of the *flower* and *plant* scene is illustrated in figure 7. On average, our combined approach is the only one with higher scores than Nerfbusters in both PSNR and SSIM.

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

5. Limitation and Future Work

Coherency Assumption. As Instant-NGP does not consider deformations between samples of the training data, a rigid scene is assumed. This premise is crucial for our coherency assumption: As floating or flying objects tend to move around and not stay in-place, they can be considered out-of-scope for this work. However, in particular capturing circumstances the coherency assumption may not be satisfied, although a coherent scene is captured. A practical example is a scene comprising two hill tips, where the terrain connecting the two hills is not captured. This would result in incoherent canonical geometry, violating our coherency assumption.

Transparent and Thin Structures. Another possible limitation leading to incoherent canonical scene geometry are transparent and thin structures, which constitute an inherently challenging task in a multitude of computer vision problems. Since our proposed post-processing approach is based on a binary representation of density, it is in itself not necessarily limited by the occurrence of transparent or thin structures, as long as they are recognized as a non-zero density by the underlying NeRF architecture.
However, transparent objects tend to have a low optical density, which often leads to scenarios in which Instant-NGP [MESK22] and other NeRF models adhering to the original NeRF architecture [MST*21] assign some regions that contain transparent objects no optical density at all. Due to this effect, these regions are not part of the occupancy grid, interfering with the coherency of the scene geometry, that our algorithms assume. Especially due to the applied SSCS, transparent regions that are only assigned an optical density in some cases, are omitted from the scene geometry. As is evident from the *roses* scene in figure 8, this can lead to the omission of major parts of the scene, due to the introduced incoherence.
Thin structures can lead to similar effect. With thin structures such as flower stalks, the corresponding occupancy grid cells may be sparsely connected or not share an immediate neighbor cell at all. This effect may be amplified by our SSCS approach, effectively introducing structural gaps in the canonical scene geometry. This is depicted in the *flowers* scene in figure 8, where the top part of the flowers, that consists of thin structures is ignored in the rendering process.
Future work could mitigate this effect by enhancing the SSCS technique in a way that takes general scene geometry into account by using additional scene information, e.g. depth priors that are capable of working with transparent structures. Alternatively, extending the neighborhood-based clustering algorithm into a density-based clustering approach with higher search radius, could mitigate the separation of the scene geometry. Furthermore, increasing the reconstruction robustness of Instant-NGP regarding transparent and thin structures could prevent this violation of our scene coherency assumption, resulting in better reconstructions.

Consistency between Occupancy Grids. Depending on the camera pose of the rendered novel view, Instant-NGP [MESK22] uses a different Occupancy Grid G_k to acquire density information. Since our approach does not alter the underlying MLP, the different resolutions of these Occupancy Grids can lead to inconsistencies when zooming in and out of the scene.
The content of an occupancy grid cell is rendered in full, if it contains any density information of the coherent relevant scene geometry. When the scene rendering utilizes a low-resolution occupancy grid, parts of some *floater* artifacts may reappear in the scene, if they are close to the relevant scene geometry. Figure 9 illustrates such a case, in which the purple floater artifact is not rendered, on scale $s = 4$, but is visible on a lower resolution occupancy grid.
While we did not observe this behavior in our experiments, it is reasonable to point out, that it is a theoretical scenario that might occur. Since these floaters only reappear when the camera has a considerable distance from the core region of the scene, future work should examine, if this effect influences the perceived novel view image quality.

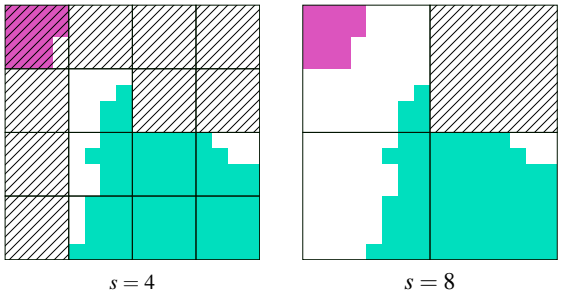

Figure 9: *Inter-scale-inconsistency due to Floater vicinity to Canonical Scene Geometry. While the floater artifact is pruned in grid scale $s = 4$, it shares an occupancy grid cell with the scene geometry in the occupancy grid with scale $s = 8$. As a consequence, the artifact will be rendered when $s = 8$ is queried in the rendering process, as for instance with sufficient camera distance.*

Compatibility. Our proposed technique that suppresses *floater* artifacts without retraining of the underlying MLP is generally suited as an extension for all NeRF architectures, that use an explicit density distribution as acceleration structure during rendering. There are multiple NeRF strategies, that leverage the occupancy grid introduced in Instant-NGP, e.g. Nerfacto that is used in Nerfstudio [TWN*23].
Furthermore, our SSCS and Occupancy Clustering approach could be applied in a more manually guided manner. This could be achieved by manually adapting the thresholds of the both techniques for a particular scene at hand. Increasing the threshold for the represented scene volume in the Occupancy Clustering technique could mitigate the effects of a violated coherency assumption for transparent or thin structures. Since this increases the probability of incorporating a big *floater* artifact, a manual annotation of remaining artifacts comparable to the concurrent work NeRFshop [JKK*23]. This could easily be achieved by intersecting a manually annotated pixel with the occupancy grid and annotating the corresponding cluster as a *floater*.

6. Conclusion

In this work, we presented a multi-step post-processing technique, that removes *floaters* from NeRFs with explicit occupancy representation as acceleration structure. We employ Scene Scale Consistency Scoring (SSCS) that detects artifacts by comparing mod-

© 2023 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

els trained with different scene scale parameters and employ a neighborhood-based clustering approach to detect the coherent scene geometry. Our post-processing strategy does not require retraining of the underlying MLP and has no implications regarding the rendering performance. The proposed technique runs automatically, but can be enhanced by user input if required. We have shown, that our approach efficiently removes *floater* artifacts on the MIP-NeRF 360 [BMV*22] and Nerfbusters [WWT*23] dataset, outperforming the state-of-the-art technique in *floater* removal during post-processing.

Acknowledgements. The project this publication is based on has been sponsored by the German Federal Ministry of Education and Research under contract number 01lS17050. The authors are responsible for its content. We thank the anonymous reviewers whose comments helped improve this manuscript.

References

[BMV*22] BARRON, JONATHAN T, MILDENHALL, BEN, VERBIN, DOR, et al. "Mip-nerf 360: Unbounded anti-aliased neural radiance fields". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 5470–5479 2, 3, 5, 6, 11.

[CFHT23] CHEN, ZHIQIN, FUNKHOUSER, THOMAS, HEDMAN, PETER, and TAGLIASACCHI, ANDREA. "Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, 16569–16578 2.

[CLW23] CHEN, JUN-KUN, LYU, JIPENG, and WANG, YU-XIONG. "NeuralEditor: Editing Neural Radiance Fields via Manipulating Point Clouds". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, 12439–12448 3.

[CXG*22] CHEN, ANPEI, XU, ZEXIANG, GEIGER, ANDREAS, et al. "Tensorf: Tensorial radiance fields". *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*. Springer. 2022, 333–350 2.

[DLZR22] DENG, KANGLE, LIU, ANDREW, ZHU, JUN-YAN, and RAMANAN, DEVA. "Depth-supervised nerf: Fewer views and faster training for free". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 12882–12891 2.

[FYT*22] FRIDOVICH-KEIL, SARA, YU, ALEX, TANCIK, MATTHEW, et al. "Plenoxels: Radiance fields without neural networks". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 5501–5510 2.

[HTE*23] HAQUE, AYAAN, TANCIK, MATTHEW, EFROS, ALEXEI A, et al. "Instruct-nerf2nerf: Editing 3d scenes with instructions". *arXiv preprint arXiv:2303.12789* (2023) 3.

[JKK*23] JAMBON, CLÉMENT, KERBL, BERNHARD, KOPANAS, GEORGIOS, et al. "NeRFshop: Interactive Editing of Neural Radiance Fields". *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 6.1 (2023) 2, 3, 10.

[KYK*22] KANIA, KACPER, YI, KWANG MOO, KOWALSKI, MAREK, et al. "Conerf: Controllable neural radiance fields". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 18623–18632 3.

[LGO*23] LAZOVA, VERICA, GUZOV, VLADIMIR, OLSZEWSKI, KYLE, et al. "Control-nerf: Editable feature volumes for scene rendering and manipulation". *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, 4340–4350 3.

[LKC*23] LIU, XINHANG, KAO, SHIU-HONG, CHEN, JIABEN, et al. "Deceptive-NeRF: Enhancing NeRF Reconstruction using Pseudo-Observations from Diffusion Models". *arXiv preprint arXiv:2305.15171* (2023) 3.

[LMTL21] LIN, CHEN-HSUAN, MA, WEI-CHIU, TORRALBA, ANTONIO, and LUCEY, SIMON. "Barf: Bundle-adjusting neural radiance fields". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 5741–5751 3.

[LNSW21] LI, ZHENGQI, NIKLAUS, SIMON, SNAVELY, NOAH, and WANG, OLIVER. "Neural scene flow fields for space-time view synthesis of dynamic scenes". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 6498–6508 2.

[LSS*21] LOMBARDI, STEPHEN, SIMON, TOMAS, SCHWARTZ, GABRIEL, et al. "Mixture of volumetric primitives for efficient neural rendering". *ACM Transactions on Graphics (ToG)* 40.4 (2021), 1–13 2.

[LTT23] LIU, XINHANG, TAI, YU-WING, and TANG, CHI-KEUNG. "Clean-NeRF: Reformulating NeRF to account for View-Dependent Observations". *arXiv preprint arXiv:2303.14707* (2023) 2, 3.

[LZZ*21] LIU, STEVEN, ZHANG, XIUMING, ZHANG, ZHOUTONG, et al. "Editing conditional radiance fields". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 5773–5783 3.

[MESK22] MÜLLER, THOMAS, EVANS, ALEX, SCHIED, CHRISTOPH, and KELLER, ALEXANDER. "Instant neural graphics primitives with a multiresolution hash encoding". *ACM Transactions on Graphics (ToG)* 41.4 (2022), 1–15 2–4, 10.

[MHM*22] MILDENHALL, BEN, HEDMAN, PETER, MARTIN-BRUALLA, RICARDO, et al. "Nerf in the dark: High dynamic range view synthesis from noisy raw images". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 16190–16199 2.

[MHS*22] MUNKBERG, JACOB, HASSELGREN, JON, SHEN, TIANCHANG, et al. "Extracting triangular 3d models, materials, and lighting from images". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 8280–8290 2.

[MRS*21] MARTIN-BRUALLA, RICARDO, RADWAN, NOHA, SAJJADI, MEHDI SM, et al. "Nerf in the wild: Neural radiance fields for unconstrained photo collections". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 7210–7219 2.

[MST*21] MILDENHALL, BEN, SRINIVASAN, PRATUL P, TANCIK, MATTHEW, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis". *Communications of the ACM* 65.1 (2021), 99–106 2, 10.

[NBM*22] NIEMEYER, MICHAEL, BARRON, JONATHAN T, MILDENHALL, BEN, et al. "Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 5480–5490 2, 3.

[NG21] NIEMEYER, MICHAEL and GEIGER, ANDREAS. "Giraffe: Representing scenes as compositional generative neural feature fields". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 11453–11464 3.

[PCPM21] PUMAROLA, ALBERT, CORONA, ENRIC, PONS-MOLL, GERARD, and MORENO-NOGUER, FRANCESC. "D-nerf: Neural radiance fields for dynamic scenes". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 10318–10327 2.

[PSB*21] PARK, KEUNHONG, SINHA, UTKARSH, BARRON, JONATHAN T, et al. "Nerfies: Deformable neural radiance fields". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 5865–5874 2.

[PSH*21] PARK, KEUNHONG, SINHA, UTKARSH, HEDMAN, PETER, et al. "Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields". *arXiv preprint arXiv:2106.13228* (2021) 2.

[RLS*22] REMATAS, KONSTANTINOS, LIU, ANDREW, SRINIVASAN, PRATUL P, et al. "Urban radiance fields". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 12932–12942 2.

[RMY*22] REBAIN, DANIEL, MATTHEWS, MARK, YI, KWANG MOO, et al. "Lolnerf: Learn from one look". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 1558–1567 2.

© 2023 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

[RPLG21] REISER, CHRISTIAN, PENG, SONGYOU, LIAO, YIYI, and GEIGER, ANDREAS. "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 14335–14345 2.

[SDZ*21] SRINIVASAN, PRATUL P, DENG, BOYANG, ZHANG, XIUMING, et al. "Nerv: Neural reflectance and visibility fields for relighting and view synthesis". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 7495–7504 2.

[SS23] SOMRAJ, NAGABHUSHAN and SOUNDARARAJAN, RAJIV. "ViP-NeRF: Visibility Prior for Sparse Input Neural Radiance Fields". *arXiv preprint arXiv:2305.00041* (2023) 3.

[TCY*22] TANCIK, MATTHEW, CASSER, VINCENT, YAN, XINCHEN, et al. "Block-nerf: Scalable large scene neural view synthesis". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 8248–8258 2.

[TTG*21] TRETSCHK, EDGAR, TEWARI, AYUSH, GOLYANIK, VLADISLAV, et al. "Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 12959–12970 2.

[TWN*23] TANCIK, MATTHEW, WEBER, ETHAN, NG, EVONNE, et al. "Nerfstudio: A Modular Framework for Neural Radiance Field Development". *ACM SIGGRAPH 2023 Conference Proceedings*. SIGGRAPH '23. 2023 6, 10.

[WGM*23] WEDER, SILVAN, GARCIA-HERNANDO, GUILLERMO, MONSZPART, ARON, et al. "Removing objects from neural radiance fields". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, 16528–16538 3.

[WWT*23] WARBURG, FREDERIK, WEBER, ETHAN, TANCIK, MATTHEW, et al. "Nerfbusters: Removing Ghostly Artifacts from Casually Captured NeRFs". *arXiv preprint arXiv:2304.10532* (2023) 2, 3, 5–9, 11.

[WWX*21] WANG, ZIRUI, WU, SHANGZHE, XIE, WEIDI, et al. "NeRF–: Neural radiance fields without known camera parameters". *arXiv preprint arXiv:2102.07064* (2021) 3.

[WZL*22] WANG, LIAO, ZHANG, JIAKAI, LIU, XINHANG, et al. "Fourier plenoctrees for dynamic radiance field rendering in real-time". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 13524–13534 2.

[YBZ*22] YANG, BANGBANG, BAO, CHONG, ZENG, JUNYI, et al. "Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing". *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVI*. Springer. 2022, 597–614 3.

[YFYL23] YIN, YOUTAN, FU, ZHOUJIE, YANG, FAN, and LIN, GUOSHENG. "OR-NeRF: Object Removing from 3D Scenes Guided by Multiview Segmentation with Neural Radiance Fields". *arXiv preprint arXiv:2305.10503* (2023) 3.

[YLT*21] YU, ALEX, LI, RUILONG, TANCIK, MATTHEW, et al. "Plenoctrees for real-time rendering of neural radiance fields". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 5752–5761 2.

[YPW23] YANG, JIAWEI, PAVONE, MARCO, and WANG, YUE. "FreeNeRF: Improving Few-shot Neural Rendering with Free Frequency Regularization". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, 8254–8263 2.

[YSL*22] YUAN, YU-JIE, SUN, YANG-TIAN, LAI, YU-KUN, et al. "NeRF-editing: geometry editing of neural radiance fields". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 18353–18364 3.

[YYTK21] YU, ALEX, YE, VICKIE, TANCIK, MATTHEW, and KANAZAWA, ANGJOO. "pixelnerf: Neural radiance fields from one or few images". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 4578–4587 2.

[ZLX23] ZHENG, CHENGWEI, LIN, WENBIN, and XU, FENG. "Editablenerf: Editing topologically varying neural radiance fields by key points". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, 8317–8327 3.

[ZSD*21] ZHANG, XIUMING, SRINIVASAN, PRATUL P, DENG, BOYANG, et al. "Nerfactor: Neural factorization of shape and reflectance under an unknown illumination". *ACM Transactions on Graphics (TOG)* 40.6 (2021), 1–18 2.

© 2023 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.