Interactive Segmentation of Radiance Fields

Rahul Goel*   Dhawal Sirikonda*   Saurabh Saini   P J Narayanan

CVIT, Kohli Center on Intelligent Systems (KCIS), IIIT Hyderabad

{rahul.goel, dhawal.sirikonda, saurabh.saini}@research.iiit.ac.in, pjn@iiit.ac.in

Figure 1. We present ISRF, an interactive method to segment objects in radiance fields. Users can draw positive strokes to segment multiple objects at a time in 3D and negative strokes to remove unwanted regions repeatedly. In the figure, the WOODEN TABLE TOP is segmented using one positive and one negative stroke as shown.

Abstract

Radiance Fields (RF) are popular to represent casually-captured scenes for new view synthesis and several applications beyond it. Mixed reality on personal spaces needs understanding and manipulating scenes represented as RFs, with semantic segmentation of objects as an important step. Prior segmentation efforts show promise but don't scale to complex objects with diverse appearance. We present the ISRF method to interactively segment objects with fine structure and appearance. Nearest neighbor feature matching using distilled semantic features identifies high-confidence seed regions. Bilateral search in a joint spatio-semantic space grows the region to recover accurate segmentation. We show state-of-the-art results of segmenting objects from RFs and compositing them to another scene, changing appearance, etc., and an interactive segmentation tool that others can use.

1. Introduction

Scene representation is a crucial step for any scene understanding or manipulation task. Relevant scene parameters, be it shape, appearance, or illumination, can be represented using various modalities like 2D (depth/texture) maps, point clouds, surface meshes, voxels, parametric functions, *etc*. Each modality has its strengths and weak-

nesses. For example, shape correspondence is straightforward between point clouds compared to surface meshes but compromises rendering fidelity. Thus, choosing an appropriate representation has a major impact on downstream analyses and applications.

Neural implicit representations have emerged as a promising modality for 3D analysis recently. Although initially proposed only for shapes [28, 34], they have been extended to encode complete directional radiance at a point [30], other rendering parameters like lightfields, specularity, textual context, object semantics, *etc*. [1, 9, 11, 12, 16, 19, 51]. The representation was extended beyond static inward-looking and front-facing scenes to complex outward-looking unbounded $360°$ views, dynamic clips, occluded egocentric videos, and unconstrained images.

Radiance fields have also been used beyond Novel View Synthesis (NVS) for other applications [5, 26, 35, 44, 47, 49, 53, 56, 59]. Segmenting objects of the scene representation is a first step towards its understanding and manipulation for different downstream tasks. There have been a few efforts at segmenting and editing of radiance fields. Recently, N3F [48], and DFF [21] presented preliminary solutions to this in the neural space of radiance fields. Both use distillation for feature matching between user-provided cues with the learned 3D feature volume, with N3F using user-provided patches and DFF using textual prompts or patches as the segmentation cues. These methods struggle to segment objects with a wide appearance variation. The NVOS system provides segmentation with strokes but have poor quality and non-interactive computations [38].

Project Page: https://rahul-goel.github.io/isrf/
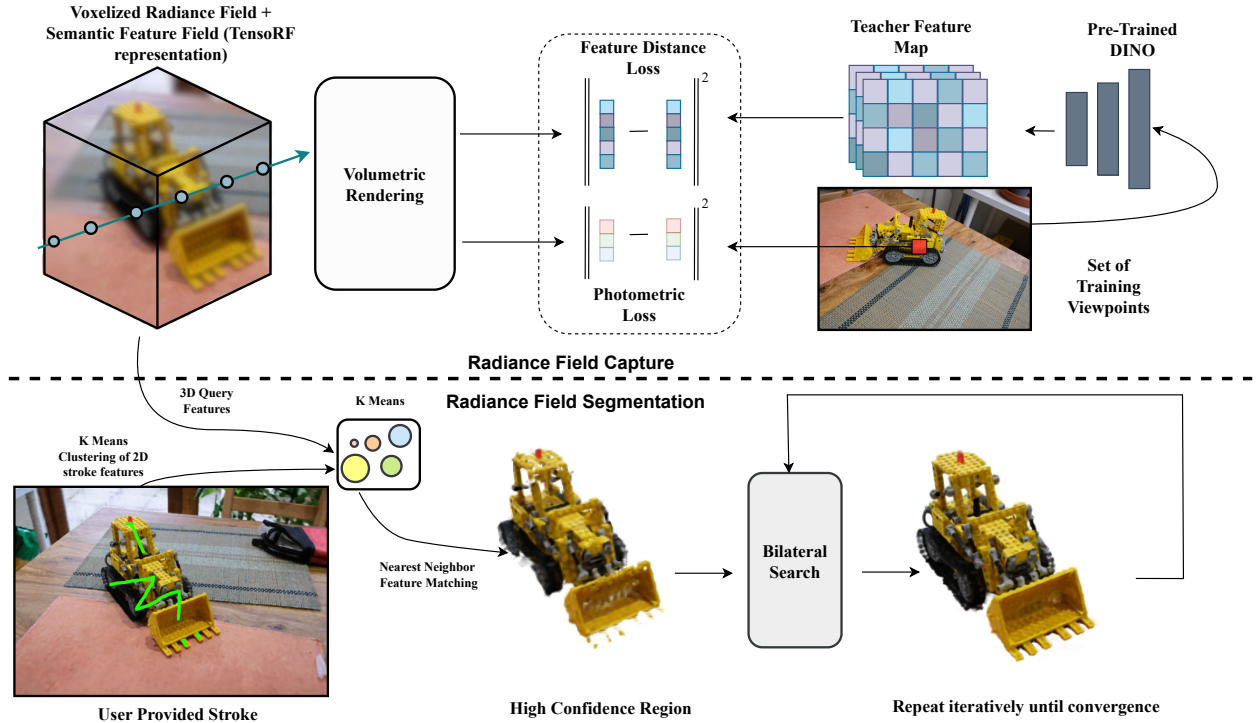* Equal Contribution

Figure 2. *ISRF System overview*: We capture a 3D scene of voxelized radiance field and distill the semantic feature into it. Once captured, the user can easily mark regions using a brush tool on a reference view (green[■] stroke). The features are collected corresponding to the marked pixels and clustered using K-Means. The voxel-grid is then matched using NNFM (nearest neighbor feature matching) to obtain a high confidence seed using a tight threshold. The seed is then grown using bilateral search to smoothly cover the boundaries of the object, conditioning the growth in the spatio-semantic domain.

In this paper, we present a simple and efficient method to interactively segment objects in a radiance field representation. Our ISRF method uses an intuitive process with the user providing easy strokes to guide it interactively. We use the fast and memory-efficient TensoRF representation [7] to train and render. TensoRF uses an explicit voxel representation that is more amenable to manipulation. We include a DINO feature [6] at every voxel to facilitate semantic matching from 2D to 3D. DINO features are trained on a large collection of images and are known to capture semantics effectively. We condense the DINO features from the user-specified regions to create a fixed-length set using K-Means. A nearest neighbor feature matching (NNFM) on this set in the 3D voxels identifies a high-confidence *seed region* of the object to be segmented. The seed region is grown using a bilateral filtering-inspired search to include neighboring proximate voxels in a joint feature-geometric space. We show results of segmenting several challenging objects in forward facing [29] and 360 degrees [2] scenes. The explicit voxel space we use facilitates simple modification for segmenting objects. We also show examples of compositing objects from one RF into another. In summary, the following are the core contributions of ISRF:

○ An easily interpretable and qualitatively improved 3D object segmentation framework for radiance fields.

○ Interactive modification of segmentation to capture fine structure, starting with high-confidence matching. Our representation allows a spatio-semantic bilateral search to make this possible. The framework can also use other generalized distances to grow the region for specific applications.

○ A hybrid implicit-explicit representation that is memory-efficient and fast to render also facilitates the distillation of semantic information for improved segmentation. Our results show improved accuracy and fine-grain object details in very challenging situations over contemporary efforts.

○ An easy-to-use, GUI based tool to interactively segment objects from an RF representation to facilitate object replacement, alteration, *etc*.

○ Consistent 2D/3D segmentation masks for a few scenes and objects created manually using our method to facilitate future work in segmentation, manipulation, and understanding of RFs.

2. Related Work

Radiance field research work is extensive and fast-growing. Hence, we restrict our related works discussion to three relevant topics, *i.e.*, hybrid representations, manipulation of radiance fields and feature-encoded semantics. For a more comprehensive background, we encourage the reader to refer to the latest surveys in this area [45, 54].

Hybrid Representations: In the past several years, various representations have been employed for the NVS applications [15, 23, 50]. The latest line of works based on implicit volumetric representations [2, 30, 60] specifically has shown great promise by leveraging the Radiance Fields (RF) [63] for comprehensive scene representation. NVS, from the perspective of radiance fields, involves volumetric rendering [30] from a particular viewpoint. However, despite its vast utility, NeRF demands tens of hours of training time per scene. The computational overhead issue has been the focus of subsequent works like PlenOctrees [57], KiloNeRF [37], *etc.* which borrows efficient techniques from the traditional 3D literature. Later, Plenoxels [10] and DVGO [42, 43] have advanced on this front by harnessing the lattice structure in a hybrid representation with implicit field features encoded on an explicit spatial grid. This significantly reduces the training time overhead to 5-10 minutes per scene by trading it off with increased storage requirements. InstantNGP [31] reduced the training time to few seconds using multilevel hash encodings. Recently, TensoRF [7] proposed a tensor-decomposed set of matrix-vector representation for the radiance feature lattice structure, which addresses both the storage and time overhead issues. We base our method on TensoRF representation to exploit this gained efficiency and explicit geometric information.

Editing: The advent of RFs has paved a principled way for altering the appearance of 3D scene content. Many extended this approach to solve problems in varied domains on the editing problem. More specifically, works like [3, 4, 32, 41, 61] have disentangled the photo-realistic rendering equation to account for the material and lighting edits. Others like [33] and [18, 58] have aimed to alter the appearance via post-hoc image-based stylization modules [13, 20]. Apart from such appearance edits, methods like [8], [55] have concentrated on geometric deformations of object-centric scenes represented as Radiance Fields. Our proposed method allows both appearance and geometric scene manipulations.

Semantics: For the scene, understanding the semantic information of the scene is essential; still, only a few solutions have been proposed in this area. Initial methods like Semantic NeRF [62] tried directly regressing semantic labels in the novel views from sparse priors. A few leveraged deep image features like DINO [6] and LSeg [24] to attribute semantics

to the 3D scene points. N3F [48] and DFF [21] demonstrate object-specific segmentation using deep semantics. Though these methods support segmentation, interactive content addition and removal are not supported by them, as the underlying scene representation is an implicit neural function that prohibits easy alterations and extensions into other application scenarios.

In this work, we use a 2D-3D distillation-based approach similar to N3F and DFF, but focus on fine-grained interactive segmentation. Initial works like GrabCut [39] and its variants utilized positive and negative user strokes to obtain the correct segmented objects. Subsequent variants [25, 40] have augmented this by leveraging deep learning in the temporal and non-temporal domains. NVOS [38] follows a 3D variant of GrabCut using the positive and negative strokes for segmentation of scenes represented as RFs and MPI [50] but struggles to produce faithful segmentation while incurring significant performance overhead. We draw inspiration from them and build upon the proven methods like semantic features, nearest neighbor matching, and voxel carving techniques [17, 22] by extending them to radiance fields. We experimentally prove how such simple techniques combined with the appropriate scene representation can improve result quality and fine details while simultaneously being quite intuitive, interpretable, and efficient ($80\times$ faster than NVOS).

3. Method

We first provide the basics on radiance fields and the feature distillation strategy related to our scene representation. We then detail our proposed interactive segmentation workflow comprising 2D-3D feature matching, region growing, and manipulation techniques on this learned representation.

3.1. Radiance Field Representation

A radiance field (RF) [63] $\mathcal{F}$ maps the scene radiance values as view dependent RGB color $c \in \mathbb{R}^3$, given a continuous point $x \in \mathbb{R}^3$ and viewing direction $d \in \mathbb{S}^2$ in space as inputs: $\mathcal{F}(x, d) : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}^3$.

NeRF [30], and its variants [2, 27, 60] encoded this mapping as the neural function using an MLP, with a low memory footprint but high training and rendering overhead. They also store scalar point density $\sigma \in \mathbb{R}$ which is used for differentiable volumetric rendering to train the network:

$$\hat{C}(r) = \left(\sum_{i=i}^{K} T_i \alpha_i c_i \right) \quad \text{where} \quad (1)$$

$$\alpha_i = 1 - e^{-\sigma_i \delta_i} \quad \text{and} \quad T_i = \prod_{j=1}^{i-1}(1 - \alpha_j). \quad (2)$$

Here for a given point i along a ray, δ_i is the distance to the sampled point, T_i is the accumulated transmittance, and c_i is the view-dependent color for the point. Later efforts like Plenoxels [10], and DVGO [42] stored the field variables in

a lattice structure akin to a 3D voxel grid, significantly improving the training and rendering times at the cost of high storage requirements. These quantized values are trilinearly interpolated and decoded to render color value at any point. The grid structure provides easy spatial context and explicit representation leading to higher efficiency. Recently, TensoRF [7] proposed a matrix-vector decomposition representation of this lattice, reducing storage requirements while facilitating efficient training and view generation. We use TensoRF as the basis of our work. The top part of Fig. 2 shows our radiance field capture step, with the volume represented using TensoRF. In the case of the quantized representation of radiance fields, the radiance is obtained as follows:

$$\sigma_i = \psi(V^\sigma, x_i) \quad \text{and} \quad c_i = \mu_\theta^{rbg}(\psi(V^f, x_i), d). \quad (3)$$

Here σ is the density of the volumetric space, V^f the radiance feature grid of appearance features f, and ψ indicates trilinear interpolation. While rendering a given sample point $x_i \in \mathbb{R}^3$ along the ray direction d, a small decoding MLP $\mu_\theta^{rbg}(f_i, d) \to c_i$ is evaluated. The final color of a ray is calculated by combining all sample colors c_i at every point x_i along it using the Eq. (1). This is used to reduce the photometric loss $\mathcal{L}^{(rgb)}$ optimizing for both the radiance feature lattice V^f and parameters θ of MLP (μ).

3.2. Semantic Features Distillation

Object segmentation requires knowledge of scene semantics. We include an additional feature into the radiance field for this. In order to attribute semantics to the radiance field, we distill contextual knowledge from a large pre-trained teacher model similar to the prior art [21, 48]. Specifically, our teacher is a vision transformer model trained using self-supervision and is shown to pay attention to semantically meaningful objects in the scene in a class-agnostic manner. This knowledge from the teacher is distilled into the student radiance field in addition to the color and density values as point semantic features $\phi \in \mathbb{R}^m$. Thus the mapping now becomes: $\mathcal{F}(x, d) : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}^m$. More concretely, we use 2D semantic features using the DINO ViT-b8 model [6] for each input posed image. Recent efforts [21, 48] also use DINO; unlike them, we directly optimize for the features on the voxel grid in the TensoRF representation without a neural network. We also do not encode the direction dependence in these semantic features since the object semantics are direction agnostic. We trilinearly interpolate the distilled semantic feature $\phi_i = \psi(V^\phi, x_i)$ for a point x_i from the learned feature lattice V^ϕ. We combine the ϕ_i along the ray using the Eq. (1) like color c_i. The TensoRF representation is optimized to minimize the total loss

$$\mathcal{L} = \mathcal{L}_{rgb} + \lambda \mathcal{L}_{feature} \quad (4)$$

to obtain the final radiance field with ϕ, V^f, and V^ϕ. Both losses $\mathcal{L}_{rgb}$ and $\mathcal{L}_{feature}$ are calculated using L^2 norm.

High-resolution feature rendering results in high-frequency feature fields similar to N3F [48]. (See the supplementary document for distilled feature field visualizations.) Explicit semantic features at every point open the way to adapt traditional 3D analysis techniques to radiance fields in a semantically meaningful fashion. Segmenting objects in 3D voxel space and using bilateral filtering inspired search are examples that go beyond what prior neural representations have shown.

3.3. 2D-3D Feature Matching

For object segmentation, the user picks a(few) reference views and annotates the regions of interest using a brush stroke. Semantic DINO features associated with the marked pixels are collected. DINO features were shown to fare well using 1-NN feature matching for good 2D semantic segmentation [6]. However, a single DINO feature will not suffice to segment complex objects with diversity. We cluster the input features using K-Means to obtain a fixed-size exemplar set of features for matching in 3D space. We use nearest neighbor feature matching (NNFM) on the exemplar set to label each voxel as foreground or background. The result is stored in a 3D bitmap. In this step, we use a tight threshold to identify a high-confidence seed region, which is processed further. Prior methods [21, 48] used a single averaged semantic feature from the user-specified patch to match 2D to 3D. Their implicit neural representation can only be segmented after ϕ values are rendered. Feature matching methods like NNFM are too *costly* to evaluate at every point on the ray using a neural representation.

The segmentation results can also be precomputed and stored, facilitating downstream tasks like view generation and editing on the fly without repeated processing.

3.4. Region Growing

The high confidence seed region (M^0) from the previous step is grown in the volume-space to delineate the complete object volume. We do this in joint spatio-semantic space to include proximate voxels that are also semantically close. We adopt a *Bilateral Filtering* [46] inspired search dubbed as *Bilateral Search* on the voxel grid using the spatial feature x and semantic feature ϕ values as filter's domain and range kernels, respectively. We iteratively grow the current bitmap region M^r till convergence, as given below.

$$M^{r+1}(x) = \mathcal{T}_\tau(\frac{1}{W} \sum_{x_i \in \Omega_x} M^r(x_i)\, g_{\sigma_\phi}(\phi_i^2)\, g_{\sigma_s}(s_i^2))$$
$$\text{where} \quad \phi_i = \|\phi_{x_i} - \phi_x\|, \quad s_i = \|x_i - x\|$$
$$\text{and} \quad W = \sum_{x_i \in \Omega_x} g_{\sigma_f}(\phi_i^2)\, g_{\sigma_s}(s_i^2).$$

4

Here M^r is the rth iteration of filtering; ϕ_x is the distilled semantic feature at point x in the volumetric space; g_σ is the Gaussian smoothing functions with variance σ; $\mathcal{T}_\tau$ is binary thresholding against value τ; and Ω_x is the immediate voxel neighbors of x. We find that $\tau = 0.2$ works well for our scenes. The seed region expands to the boundaries of the desired object in a few iterations of bilateral filtering.

3.5. User Interactivity

Region growing results in a stable voxel content based on the input strokes. The user can add or remove parts interactively if the extracted content misses out on a few details or when some extraneous content floods into the segmented region. We use positive and negative strokes to add and remove the content in the image space, as followed by methods like GrabCut [39]. The mask of the negative segment is subtracted from the mask of the positive segment to get the final segmented objects. We find practically that even complex objects can be segmented well with a few positive and negative strokes, as shown in the results in the paper and in the supplementary material. Additionally, our method provides interactive feedback for every stroke (as can be seen in Tab. 1) that allows users to segment interactively unlike methods like NVOS [38]. Implementation details have been reported in the supplementary document.

4. Results

In this section, we discuss the comparisons and results of our proposed method against the existing semantic features

(a) Stroke 1   (b) Output 1

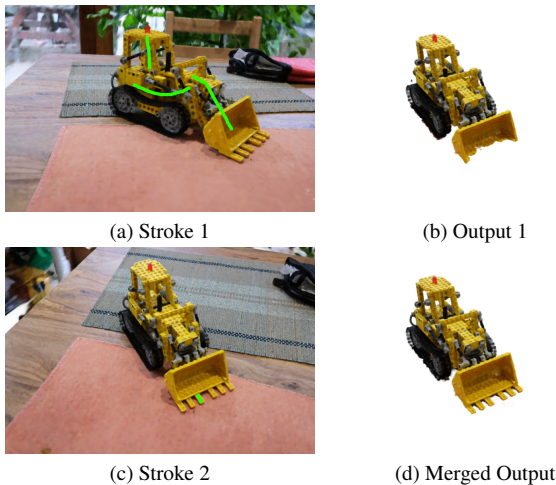(c) Stroke 2   (d) Merged Output

Figure 3. *Multiple Positive Strokes*: When the method fails to capture some of the details using initial set of strokes, the user can iteratively add more positive strokes to recover the desired object. (a) depicts the initial strokes which lead to missing teeth as shown in (b). Addition of a small stroke on one of the teeth (c) and followed by region grown captures full-details as shown in (d).

distillation-based Radiance fields segmentation approaches. Specifically, we focus on the two recent approaches: DFF-DINO [21] and N3F [48]. Both use extracted features from input images and fuse them into the volumetric space. DFF additionally concentrates on the language queries using LSeg [24], but both approaches are similar regarding semantic features. As the code of DFF is not publicly available, we compare our method against N3F, which is similar to DFF for this part.

4.1. Comparison

As discussed earlier, our approach supports region selection either by a patch or a hand-drawn brush stroke as shown in Fig. 1. To obtain the desired volumetric content, we follow the methods described in the Sections (Secs. 3.3 and 3.4). Fig. 4 shows our segmentation results on a few challenging scenes.

The usage of clustering followed by NNFM clearly outperforms the prior approach of average matching [21, 48]. The direct incorporation of nearest neighbor feature matching (NNFM) in these approaches leads to significant rendering delays, while the choice of neural space limits them from using elegant techniques like bilateral search.

In Fig. 4 it can be observed that in the case of the COLOR FOUNTAIN , the simple average feature matching technique faithfully recovers the region of interest albeit with some additional noise. However, as the scene's complexity and region of interest grows, the prior art fails to garner pleasing results. This can be observed clearly in the case of the three LLFF [29] scenes (CHESS TABLE , SHOE RACK , STOVE). When only simple averaging is employed, the CHESS TABLE scene suffers due to the erroneous feature matches. The clustered matching mitigates the errors and confines the segmented volume to the TABLE. A similar effect can be observed in the case of STOVE where the object of interest is sparingly covered in the input images but is faithfully recovered with distinct boundaries, unlike N3F. The last scene SHOE RACK is a *classic* example where recovering white-sole might be challenging even with the best feature matching scheme. This is where the bilateral search helps in exploiting multi-domain content by conditioning on the spatio-semantics.

We also qualitatively compare our results with another stroke-based approach NVOS [38] in Fig. 12. Quantitative evaluation of *mIOU/mAcc* scores on all the NVOS dataset also reflect similar behavior. Using the input strokes and GT masks of NVOS, we obtain an *mIoU* of **83.75%** (compared to **70.1%** of NVOS) and an *mAcc* of **96.4%** (compared to **92.0%** of NVOS), on the same LLFF dataset. Additionally, our interactive scheme allows for improving the segmentation in subsequent iterations. We achieved an *mIoU* of 90.8% and an *mAcc* of 98.2% on the same dataset using multiple strokes. A detailed depiction of results is discussed

Figure 4. *Our ISRF vs N3F/DFF [21, 48]:* Both N3F and DFF employ a similar strategy for segmentation. We tweak the threshold for their method and bring out the best results and show their respective results in the Row 2. Row 3 shows our results with the same queried patch (highlighted in green[■] in Row 1). Since our method works best on user provided strokes (shown in yellow[■] in Row 1), we show the corresponding results in Row 4. While N3F/DFF are able to recover simpler objects like COLOR FOUNTAIN , they fail to capture other objects. Our method faithfully recovers the queried objects with clear and smooth boundaries. For more details, please refer to Sec. 3.4.

in the supplementary document.

4.2. Interactive Segmentation with User Strokes

Our method allows both adding and removing content using positive and negative strokes. The cases where the single stroke fails to obtain the desired content in the extracted space, the user can add another positive stroke to add more content. Fig. 3 shows one such example where the excavator ('JCB') has missing teeth in the extracted region. Drawing an additional stroke and bilaterally growing the region again brings out the full desired result. This effectively grows the bit-map M^r by segmenting more desirable regions from the volumetric space.

Similar to adding new content, some scenarios demand the need to remove extraneous content from the extracted region. In such scenarios, we mark the region to be removed and grow it independently of the positive content. Once fully grown, the full extent of the negative/undesirable content is obtained which we subtract from the previously extracted regions obtaining the edited bit-map M^r. Fig. 1 shows one such example where the REFLECTIVE GRANITE floods into the TABLE region. We add a negative stroke (red) to remove this undesired region.

Incorporating these functionalities is not trivial in the case of the prior art, as an additional negative match or a positive match calculation at the time of rendering is a tedious task.

Figure 5. Results. Left: NVOS [38] (from their paper), Middle: reference masks from NVOS-dataset, Right: Our ISRF system. More results can be seen in the supplementary document.

5. Experiments

In this section, we discuss various feature-matching variants we used to obtain the high-confidence seed region. Additionally, we show some immediate applications of radiance field segmentation.

5.1. Ablations

In order to obtain a high-confidence region, which acts as a seed for the bilateral filter, a feature-matching technique is required to match the marked features with the distilled semantic features in the volumetric space. To this end, we experimented with three different feature matching techniques, namely (1) Average Feature Matching, (2) Nearest neighbor Feature matching(NNFM) (3) K means + NNFM, which are compared in the Fig. 6. It can be easily inferred from Fig. 6b that average feature matching performs poorly in this task. In order to improve these results, we resort to the nearest neighbor feature matching. Though this recovers a good high confidence region, it is accompanied by additional noise as seen in Fig. 6c Furthermore, as the marked region's size grows, computation also becomes tedious in this case. To address this, we cluster the features using K-means clustering and then do an NNFM that reduces com-

Step	Time Taken
Pre-training radiance field	7 mins
Training feature field	2.5 mins
K-Means Clustering	2 secs
3D Feature Query	1 secs
Bilateral Region Growing	0.3 secs

Table 1. Timings of different steps of the ISRF pipeline

(a) Ground Truth   (b) Average Feature Matching
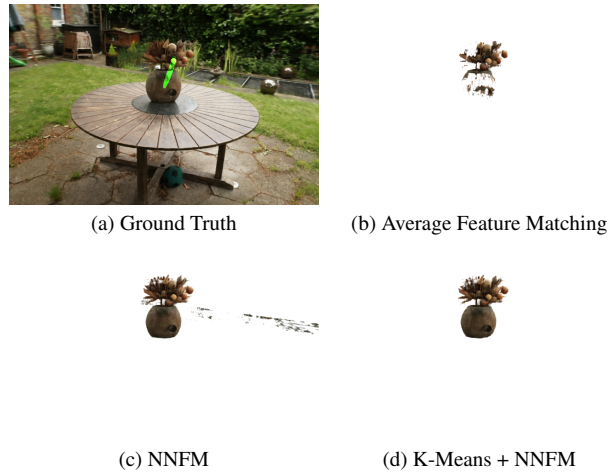
(c) NNFM   (d) K-Means + NNFM

Figure 6. *Feature Matching*: This figure shows the high confidence region of the RF (a) obtained using different feature-matching techniques for a particular stroke. While *Average feature matching* (b) fails to cover the entire object due to loss of information during the averaging process, *NNFM* (c) without clustering leads to noise bleeding. Use of *NNFM after clustering* (d) eliminates noisy regions while also considering multiple features at once.

putational overhead and avoids noisy matches, as seen in Fig. 6d. When $K = 1$, clustering results in mean features of the selected stroke, and as K approaches high values, the search approximates NNFM.

5.2. Editing

After obtaining a good segmentation, many appealing opportunities for editing open up:

Object Removal: The removal of an object from the scene

is a simple task and is shown in Fig. 7a where the POT in GARDEN scene is removed. Please note that we do not inpaint the scene post content removal.

Affine transformation: As we have good quality segmented volumetric content, we can perform affine transformations on voxel space (POT) for object position manipulation. We demonstrate this in Fig. 7b. One can look *behind* the TABLE on the GROUND to see the repositioned POT.

Geometric Scene Composition: With high-quality 3D segmentation masks, we can also composite two different radiance fields. We demonstrate this in Fig. 7c. We follow the composition technique of [52] to accomplish this task. The JCB is picked from the KITCHEN scene from the [2] dataset and placed in GARDEN . Please note that we do not take global illumination into account for these edits.

Appearance Editing: As the appearance vector is associated with each voxel in the grid, we can alter the appearance of the individual segmented objects. We demonstrate this by stylization the content using [14] in Fig. 7d.

5.3. Discussions and Limitations

Our method improves upon the prior art on several fronts but has its own shortcomings. Like prior works, we rely on DINO features to represent object semantics and this can result in artefacts if the features do not capture the semantics properly. Third column in the last row in Fig. 4 shows a small false appendage at the bottom of the utensil holder which can not be easily removed interactively without eating into object's body. Better semantic features can resolve this problem. Also, the leftmost example in Fig. 7 shows

(a) Removal   (b) Translation

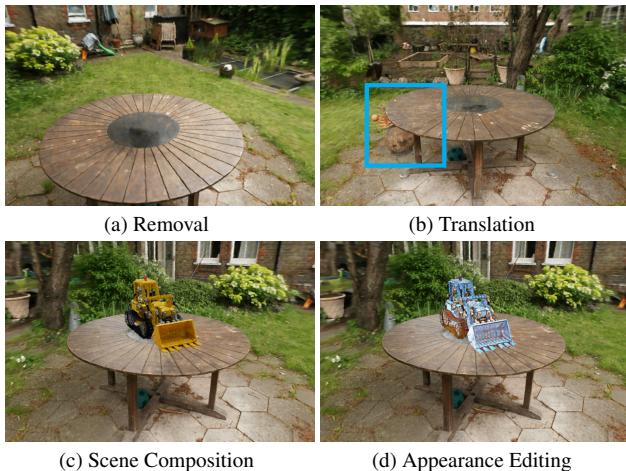(c) Scene Composition   (d) Appearance Editing

Figure 7. *Scene manipulation*: Segmented object(s) can be edited in different ways. In (a), we remove the POT from the center of the table. In (b), we translate the POT to the GROUND behind the TABLE. In (c), we replace the POT with the LEGO JCB obtained from a different scene (KITCHEN). We stylize the newly added LEGO JCB using [14] in (d). All scenes are from [2].

that the shadow of the pot is left behind on the granite center of the table even after the pot is edited out. Removing the pot from the geometric representation does not guarantee removal of its secondary effects on neighbouring objects like shadows or highlights, without elaborate geometric post-processing. Our method may also struggle in segmenting geometry well if the voxel resolution is low compared to the scale of object details as shown in supplementary results. Multiresolution voxel representations can solve this problem with additional overhead.

6. Conclusions and Future Work

In this paper, we presented an easy and accurate method to segment objects from a TensoRF representation of radiance fields and showed simple scene editing operations facilitated by this. The efficient voxel-based representation we use makes our method more versatile and simple compared to the prior works in this direction. We show several results on multiple challenging scenes (and present more in the supplementary document). Semantic segmentation is a first step towards interpretation, understanding, and manipulation of 3D scenes. This work provides high quality segmentation that can be the basis for several such downstream tasks. A simple extension to the current method would be to generalize the distance used for matching in the NNFM and region-growing steps to include other features like color latent vectors. Extending the current method to a InstantNGP [31] framework, while incorporating additional multi-domain explorations strategies like guided filtering [17] would be a good direction to explore.

In the future, multi-representation processing might be needed by combining parts of captured RFs, graphics models, SDFs, etc., to provide maximum flexibility in Virtual Reality and Augmented Reality applications. This requires processing parts of the RFs directly without going through the full learning process post-editing. This is a promising direction of work that we intend to pursue in the future.

Potential negative societal impact: Our work presents a tool to manipulate radiance fields captured casually. While ill-intentioned manipulation to create the appearance to fake scene content is possible using such a tool, the risk is negligible compared to the sophisticated image or geometry editing tools that are already prevalent. Our method needs very little additional data and doesn't directly use vast internet collections with or without consent.

References

[1] Benjamin Attal, Jia-Bin Huang, Michael Zollhöfer, Johannes Kopf, and Changil Kim. Learning Neural Light Fields with Ray-Space Embedding Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 8, 13

[3] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. NeRD: Neural Reflectance Decomposition from Image Collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[4] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan T. Barron, and Hendrik P.A. Lensch. Neural-PIL: Neural Pre-Integrated Lighting for Reflectance Decomposition. In *Adv. Neural Inform. Process. Syst.*, 2021. 3

[5] Arunkumar Byravan, Jan Humplik, Leonard Hasenclever, Arthur Brussee, Francesco Nori, Tuomas Haarnoja, Ben Moran, Steven Bohez, Fereshteh Sadeghi, Bojan Vujatovic, and Nicolas Heess. NeRF2Real: Sim2real Transfer of Vision-guided Bipedal Motion Skills using Neural Radiance Fields. 2022. 1

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 4, 12

[7] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial Radiance Fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2, 3, 4, 12

[8] Chong Bao and Bangbang Yang, Zeng Junyi, Bao Hujun, Zhang Yinda, Cui Zhaopeng, and Zhang Guofeng. NeuMesh: Learning Disentangled Neural Mesh-based Implicit Field for Geometry and Texture Editing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 3

[9] Akshat Dave, Yongyi Zhao, and Ashok Veeraraghavan. PANDORA: Polarization-Aided Neural Decomposition Of Radiance. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 1

[10] Fridovich-Keil and Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance Fields without Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[11] Xiao Fu, Shang-Wei Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic NeRF: 3D-to-2D Label Transfer for Panoptic Urban Scene Segmentation. In *International Conference on 3D Vision (3DV)*, 2022. 1

[12] Xuan Gao, Chenglai Zhong, Jun Xiang, Yang Hong, Yudong Guo, and Juyong Zhang. Reconstructing Personalized Semantic Facial NeRF Models from Monocular Video. *ACM Trans. Graph.*, 2022. 1

[13] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image Style Transfer Using Convolutional Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[14] Rahul Goel, Dhawal Sirikonda, Saurabh Saini, and P. J. Narayanan. StyleTRF: Stylizing Tensorial Radiance Fields. In *Proceedings of the Thirteenth Indian Conference on Computer Vision, Graphics and Image Processing*, ICVGIP '22, 2022. 8, 12

[15] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The Lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, 1996. 3

[16] Yudong Guo, Keyu Chen, Sen Liang, Yongjin Liu, Hujun Bao, and Juyong Zhang. AD-NeRF: Audio Driven Neural Radiance Fields for Talking Head Synthesis. 2021. 1

[17] Kaiming He, Jian Sun, and Xiaoou Tang. Guided Image Filtering. *IEEE Trans. Pattern Anal. Mach. Intell. (T-PAMI)*, 2013. 3, 8

[18] Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. StylizedNeRF: Consistent 3D Scene Stylization as Stylized NeRF via 2D-3D Mutual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[19] Ajay Jain, Matthew Tancik, and P. Abbeel. Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 3, 12

[21] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing NeRF for Editing via Feature Field Distillation. In *Adv. Neural Inform. Process. Syst.*, 2022. 1, 3, 4, 5, 6, 12, 13, 15

[22] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999. 3

[23] Marc Levoy and Pat Hanrahan. Light field rendering. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. 3

[24] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven Semantic Segmentation. In *Int. Conf. Learn. Represent.*, 2022. 3, 5

[25] Feng Li, Hao Zhang, Huaizhe xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask DINO: Towards A Unified Transformer-based Framework for Object Detection and Segmentation. *arXiv, abs:2206.02777*, 2022. 3

[26] Dominic Maggio, Marcus Abate, J. Shi, Courtney Mario, and Luca Carlone. Loc-NeRF: Monte Carlo Localization using Neural Radiance Fields. *ArXiv, abs/2209.09050*, 2022. 1

[27] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[28] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[29] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM Trans. Graph.*, 2019. 2, 5, 12, 14

[30] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 3

[31] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.*, 2022. 3, 8

[32] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Mueller, and Sanja Fidler. Extracting Triangular 3D Models, Materials, and Lighting From Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[33] Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao. SNeRF: Stylized Neural Implicit Representations for 3D Scenes. *ACM Trans. Graph.*, 2022. 3

[34] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[35] Keunhong Park, U. Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. HyperNeRF: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields. *ACM Trans. Graph.*, 2021. 1

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 12

[37] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[38] Zhongzheng Ren, Aseem Agarwala, Bryan Russell, Alexander G. Schwing, and Oliver Wang. Neural Volumetric Object Selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 3, 5, 7, 14, 15

[39] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabCut: interactive foreground extraction using iterated graph cuts. *ACM SIGGRAPH 2004 Papers*, 2004. 3, 5, 14

[40] Konstantin Sofiiuk, Ilya A. Petrov, Olga Barinova, and Anton Konushin. F-BRS: Rethinking Backpropagating Refinement for Interactive Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[41] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[42] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[43] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Improved Direct Voxel Grid Optimization for Radiance Fields Reconstruction. *arXiv,abs/2206.05085*, 2022. 3, 12

[44] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-NeRF: Scalable Large Scene Neural View Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[45] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edith Tretschk, Yifan Wang, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, Tomas Simon, Christian Theobalt, Matthias Nießner, Jon T. Barron, Gordon Wetzstein, Michael Zollhöfer, and Vladislav Golyanik. Advances in Neural Rendering. *Comput. Graph. Forum*, 2022. 3

[46] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision*, 1998. 4

[47] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1

[48] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural Feature Fusion Fields: 3D distillation of self-supervised 2D image representations. In *International Conference on 3D Vision (3DV)*, 2022. 1, 3, 4, 5, 6, 12, 13, 15

[49] Wei-Cheng Tseng, Hung-Ju Liao, Yen-Chen Lin, and Min Sun. CLA-NeRF: Category-Level Articulated Neural Radiance Field. *Proc. of the IEEE International Conference on Robotics and Automation*, 2022. 1

[50] Richard Tucker and Noah Snavely. Single-View View Synthesis With Multiplane Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[51] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. CLIP-NeRF: Text-and-Image Driven Manipulation of Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[52] Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. D^{2}NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video. In *Adv. Neural Inform. Process. Syst.*, 2022. 8, 12

[53] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. BungeeNeRF: Progressive Neural Radiance Field for Extreme Multi-scale Scene Rendering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 1

[54] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural Fields in Visual Computing and Beyond. *Comput. Graph. Forum*, 2022. 3

[55] Tianhan Xu and Tatsuya Harada. Deforming Radiance Fields with Cages. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 3

[56] Lin Yen-Chen, Peter R. Florence, Jonathan T. Barron, Tsung-Yi Lin, Alberto Rodriguez, and Phillip Isola. NeRF-Supervision: Learning Dense Object Descriptors from Neural Radiance Fields. In *Proc. of the IEEE International Conference on Robotics and Automation*, 2022. 1

[57] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for Real-time Rendering of Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[58] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. ARF: Artistic Radiance Fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 3

[59] Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. IRON: Inverse Rendering by Optimizing Neural SDFs and Materials from Photometric Images. 2022. 1

[60] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. NeRF++: Analyzing and Improving Neural Radiance Fields. *arXiv:2010.07492*, 2020. 3

[61] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. NeRFactor: Neural Factorization of Shape and Reflectance under an Unknown Illumination. *ACM Trans. Graph.*, 2021. 3

[62] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew Davison. In-Place Scene Labelling and Understanding with Implicit Scene Representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[63] Kun Zhou, Yaohua Hu, Stephen Lin, Baining Guo, and Heung-Yeung Shum. Precomputed Shadow Fields for Dynamic Scenes. *ACM Trans. Graph.*, 2005. 3

1. Implementation Details

All the methods proposed in the paper have been implemented using PyTorch [36] branching off the code provided by DVGOv2 [43]. All experiments are performed using a commodity hardware equipped with AMD Ryzen 5800x and a NVIDIA RTX 3090.

The feature components of the radiance fields namely radiance latent vectors and the learnt DINO features are stored using VM decomposition proposed by TensoRF [7]. For radiance latent vectors, we use *VM-48* representation of TensoRF and for DINO features, we use *VM-64* variant of TensoRF. The segmentation masks and densities have been stored as a full voxel grids.

The DINO *ViT-b8* [6] model provides 768 features for each patch of 8×8 pixels in an image. We reduce the dimensionality of these features by doing a principal component analysis reducing the effective dimension to 64. This is consistent with the prior works [21, 48]. For each pixel, the feature is calculated by referring to the feature of the respective patch that pixel corresponds to.

We first pre-train the model for the volumetric density and radiance for $20,000$ iterations. Once the radiance field is stabilized on the VM-48 TensoRF representation, we introduce distillation using *student-teacher* strategy similar to that of [21, 48] on the VM-64 TensoRF variant. Upon adoption, the resultant VM-48 variant of TensoRF along with its shallow MLP represents the radiance field, and VM-64 constitute the distilled features. It is to be noted that the distilled features are not accompanied by a shallow MLP. The features are store at voxel lattice locations and tri-linearly interpolated to be compared and optimized against the DINO features without the involvement of any non-linearity. The adoption is done with $\lambda = 0.001$ for the weighted loss function for $5,000$ iterations. The loss is taken on the features and radiance together to maintain consistency.

We choose $K = 10$ when applying K-Means to the set of features selected from the user's brush stroke. For the bilateral search, the value of σ_ϕ and σ_s are set to 10.0 and the 1.0 respectively while the threshold value τ is 0.1.

2. Scene Editing

In this section, we explain the procedures that were followed for editing the 3D scenes post segmentation. The segmentation procedure provides a 3D bit map representing the segmented voxels. Utilization of an additional bitmap also assists in faster rendering as the voxels with segmentation mask values of 0 can easily be filtered out. Fig. 8 shows the additional results of scene editing.

2.1. Object Removal

For removing a segmented object from the scene, we alter the evaluation of the density for a 3D point. We simultaneously evaluate the bit map value b_x at the queried point. To segmented the object of interest (foreground), the effective density σ'_x is $\sigma_x * b_x$. Similarly, to render the background the effective density σ'_x is $\sigma_x * (1.0 - b_x)$.

2.2. Translation

If an object needs to be moved to another location, the ray queries lying inside the object's voxel space can be shifted to the desired location. Let t be the translation vector for the object to be moved, then the object's ray-point query changes as shown below.

$$\sigma'_x, rgb'_x = \sigma_x, rgb_x \ \forall \ b_x = 0$$
$$\sigma'_x, rgb'_x = \sigma_{x+t}, rgb_{x+t} \ \forall \ b_x = 1$$

2.3. Scene Composition

To perform scene composition, we follow a similar strategy used by D^2NeRF [52]. We alter the volumetric rendering equation to account for density and color from both the scenes as shown below:

$$\hat{C}(r) = \int_{t_n}^{t_f} T(t) \left(\sigma_1(t) c_1(t) + \sigma_2(t) c_2(t) \right) dt$$
$$T(t) = exp \left(- \int_{t_n}^{t} \left(\sigma_1(s) + \sigma_2(s) \right) \right) ds$$

The results for scene composition have been shown in the main paper and Fig. 8 of the supplementary.

2.4. Appearance Editing

Here, we apply style transfer on an already composed scene. We first calculate a 3D bitmap for the JCB lego in the KITCHEN scene. Then, we generate a new set of stylized training images using the method proposed by [14, 20] using a reference image. The appearance latent vectors and the rendering MLP is fine-tuned according to the new training images while keeping the density and feature weights frozen. This transfers the style from a reference image to the 3D object.

3. Quantitative Analysis

To quantitatively compare our method on the LLFF Dataset [29], we hand-annotate the segmentation masks for the prominent objects in the CHESS TABLE , COLOR FOUNTAIN , STOVE and SHOE RACK scenes. Tab. 2 reports the segmentation metrics for the four scenes. In our method, to predict the segmentation mask, we threshold α to be greater than 0.1 while rendering. This removes the low volumetric density seeping in that contribute negligibly in the rendered visuals.

| (a) Original Rendered Image | (b) Removal of Pot | (c) Composition | (d) Style Transfer |

Figure 8. *Seamless Progressive Scene Editing*: Image (a) is the reference rendered viewpoint. In (b), the pot has been removed. Image (c) shows scene composition. The JCB from KITCHEN scene has been placed on the top of the table in the GARDEN scene. Image (d) shows appearance editing of specific objects. We apply style transfer on just the JCB. For more details please refer to Sec. 2.

Scene	Metric	N3F	Ours (Patch)	Ours (Stroke)
CHESS TABLE	Mean IoU ↑	0.344	0.864	**0.912**
	Accuracy ↑	0.820	0.985	**0.990**
	mAP ↑	0.334	0.874	**0.916**
COLOR FOUNTAIN	Mean IoU ↑	0.871	**0.927**	**0.927**
	Accuracy ↑	0.979	**0.989**	**0.989**
	mAP ↑	0.871	**0.927**	**0.927**
STOVE	Mean IoU ↑	0.416	**0.827**	0.819
	Accuracy ↑	0.954	**0.992**	**0.992**
	mAP ↑	0.387	**0.824**	0.817
SHOE RACK	Mean IoU ↑	0.589	0.763	**0.861**
	Accuracy ↑	0.913	0.965	**0.980**
	mAP ↑	0.582	0.773	**0.869**

Table 2. This table denotes the Mean IoU (Intersection Over Union), Accuracy and Mean Average Precision measurements for the four LLFF scenes shown in the main paper. The ground truth segmentation masks have been hand-annotated for comparison.

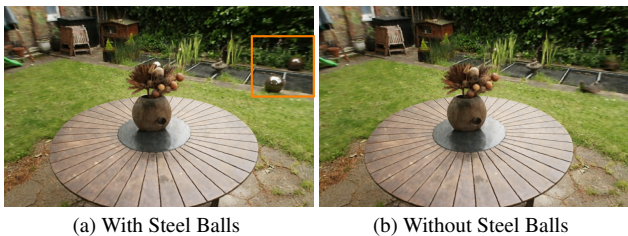| (a) With Steel Balls | (b) Without Steel Balls |

Figure 9. *Removal of Steel Balls*: We use the MipNeRF360 [2] formulation in voxel space for unbounded 360 degree scenes. This gives fewer number of voxels to the background objects compared to the central volume of interest. In this scene, we remove the steel balls appearing in the background region of the scene.

4. Region Growing: Bilateral Growth

In this section, we discuss the effect of bilateral filtering on the radiance fields and how it improves the final result. Even after employing an efficient feature-matching technique, we often obtain a high-confidence volumetric region with missing constituting parts. This is because the content search solely depends on feature distances while ig-

noring the spatial priors. To resolve this issue we resort to Bilateral search which exploits spatio-semantic domain priors resulting in accurate segmentation constituting all the desired regions of the semantic object. This is demonstrated in Fig. 10, where the initial high-confidence region misses the outer leaf of the dry plant. While the bilateral region is growing, we iteratively add more details into the extracted region, finally obtaining desired volumetric content. This content can be further used for various purposes as discussed in Sec. 2.

5. Evaluation strategies against SOTA techniques

5.1. N3F/DFF

As mentioned in the main document, we experiment with various thresholds in the case of N3F/DFF [21, 48]. We report the quantitative metrics (Tab. 2) of our method against the best results of their methods. N3F/DFF don't produce good results for any threshold as shown in Fig. 13.

(a) Rendered Image   (b) High Confidence Region
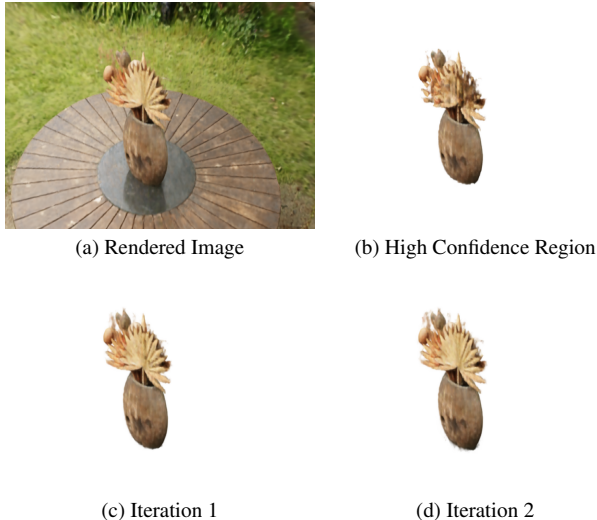
(c) Iteration 1   (d) Iteration 2

Figure 10. *Region Growing*: Image (a) is the reference rendered viewpoint. Image (b) is the high confidence region which misses out frontal region of the dry-leaf when extracting the content. Image (c) shows the result obtained after the first iteration of bilateral filtering, which captures most of the desired region of the leaf. Image (d) is the result of the bilateral filtering applied for the second time to include intricate details such as strands around the dry-leaf.

NVOS		Ours(NVOS Stroke)		Our best	
mIOU	mAcc	mIOU	mAcc	mIOU	mAcc
70.1	92.0	83.75	96.4	90.8	98.2

Table 3. Quantitative metrics(*mIOU and mAcc)* of NVOS against Ours using NVOS provided strokes and additional strokes using our interactive feedback tool

5.2. NVOS

To make a fair comparison against NVOS [38], we utilize the masks provided by NVOS and evaluate the quantitative numbers on their dataset. We observe that our method out performs NVOS both qualitatively and quantitatively as shown Fig. 12 and Tab. 3 even when using their strokes. Using our own interactive tool with additional strokes achieve much better results.

6. Interactive Segmentation

Our method provides interactive segmentation capabilities to the user with the incorporation of positive and negative brush strokes similar to GrabCut [39].

Upon the addition of a new positive stroke, a new segmentation mask b_p is calculated using the procedure described in the main paper. The user has the option to grow this new region using bilateral filtering until not required. The new segmentation mask b_{new} is given by $b \cup b_p$.

When the user adds a negative stroke, a new segmentation mask b_n is calculated. Similar to a positive stroke, the user has the option to grow this region using bilateral filter-

(a) Rendered View 1   (b) Rendered View 2
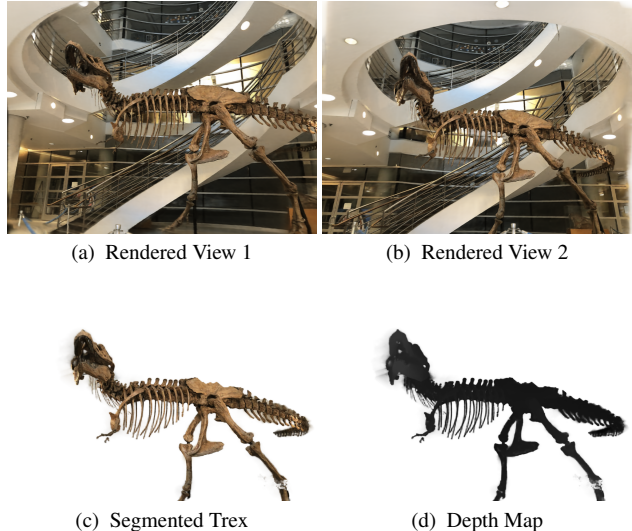
(c) Segmented Trex   (d) Depth Map

Figure 11. *Finer Segmentation*: Images (a) and (b) show rendered views of T-Rex from the LLFF dataset [29]. Image (c) shows the segmented output of T-Rex scene. Our method achieves fine-grained segmentation of objects such as the rib-cage bones of T-Rex. However, on close observation, the region near the tail bones background bleeds in. This is due to the wall and the tail-bone lie at the similar depth as shown in the depth map (d). This can be mitigated by having more 3D information (better training views) or higher voxel grid resolution.

ing until not required. The new segmentation mask b_{new} is given by $b \cap (b \cap b_n)'$ (X' denotes the complement of X).

7. Critical Analysis

7.1. DINO Features

The teacher DINO features calculated on the training set of images are for patches of size 8x8. This method associates a total of 64 pixels to the same feature vector. As shown in Fig. 15, the teacher features appear to be in low resolution due to this. When performing the teacher-student training using the joint loss function, the features learnt by the student are finer in detail due to assistance from volumetric density. Hence, the student surpasses the teacher during distillation. This is evident from Fig. 15 as features are allocated with distinct boundaries in the voxel space.

7.2. Finer Segmentation

Our method can segment out fine-grained details such as the ribs of a T-Rex as shown in Fig. 11. However, it requires accurate 3D information to achieve this. In the T-Rex scene, the tail-bones cannot be distinguished from the wall behind, since the training set images do not cover views which indicate the separation. Therefore, the optimized model containing the wall and the tail bones lie at similar depths as shown in Fig. 11d. Use of additional images covering more viewpoints can circumvent this issue.
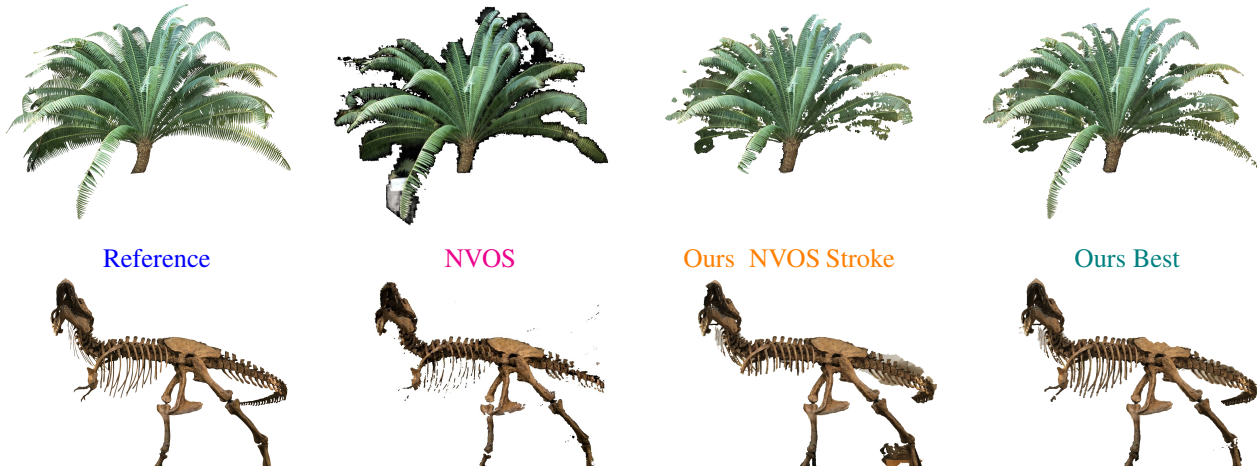
Figure 12. *left to right:* Reference segmentation using NVOS professionally segmented mask, Result of NVOS [38], Our result using NVOS stroke, Our result using additional strokes. The quantitative comparisons are mentioned in the main document where our method performs better than NVOS even when using NVOS strokes. Please zoom using *Adobe Acrobat/Okular* reader to see the details.

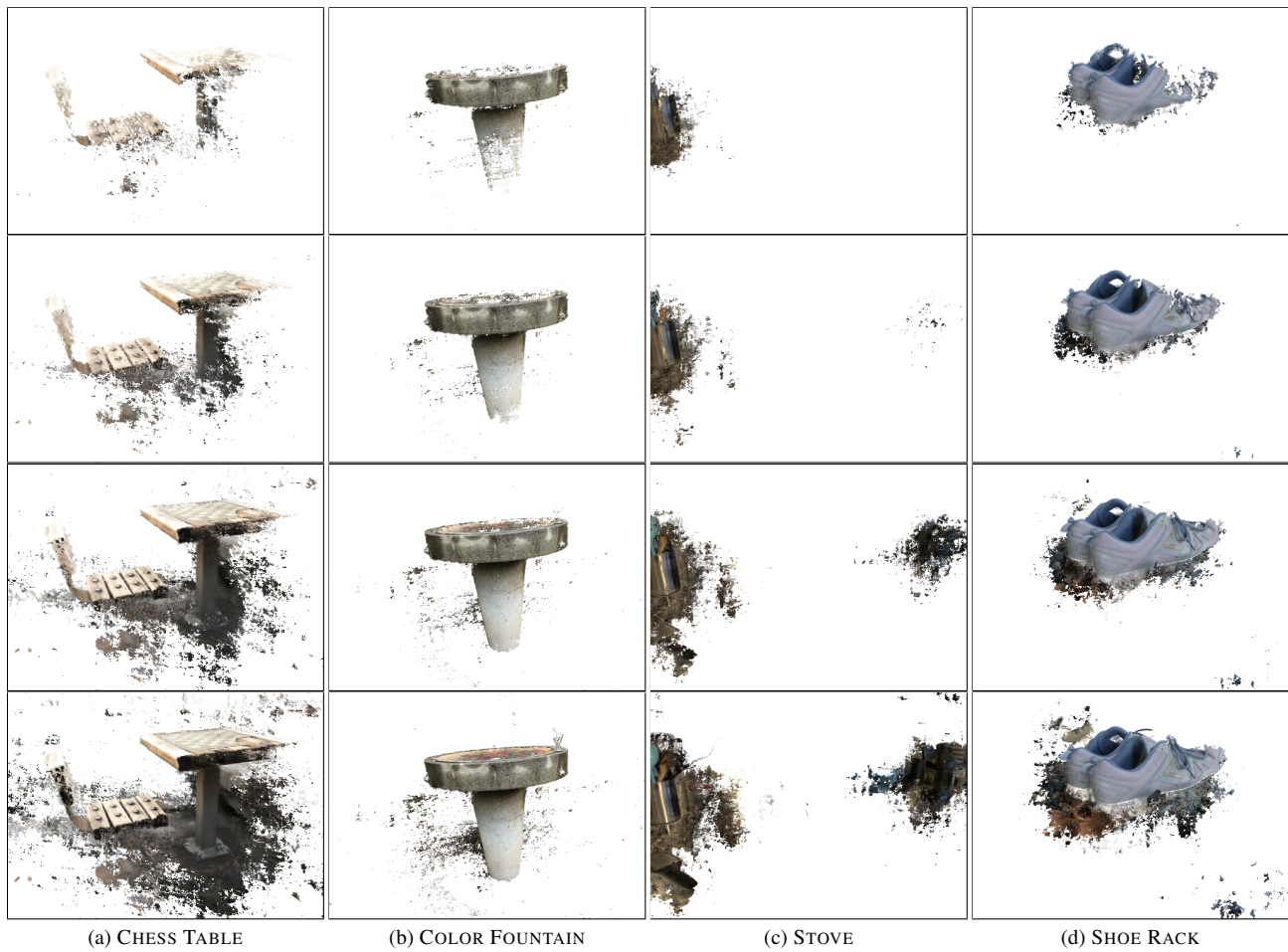(a) CHESS TABLE (b) COLOR FOUNTAIN (c) STOVE (d) SHOE RACK

Figure 13. *N3F/DFF Results*: In this figure we show result of DFF/N3F [21, 48] on different thresholds and we reported the best of their method in main document. It can be seen that despite varying the thresholds the result is poorly segmented. The background objects are starting to bleed into the foreground. For the results of our method on the same scenes, please refer to the main paper.
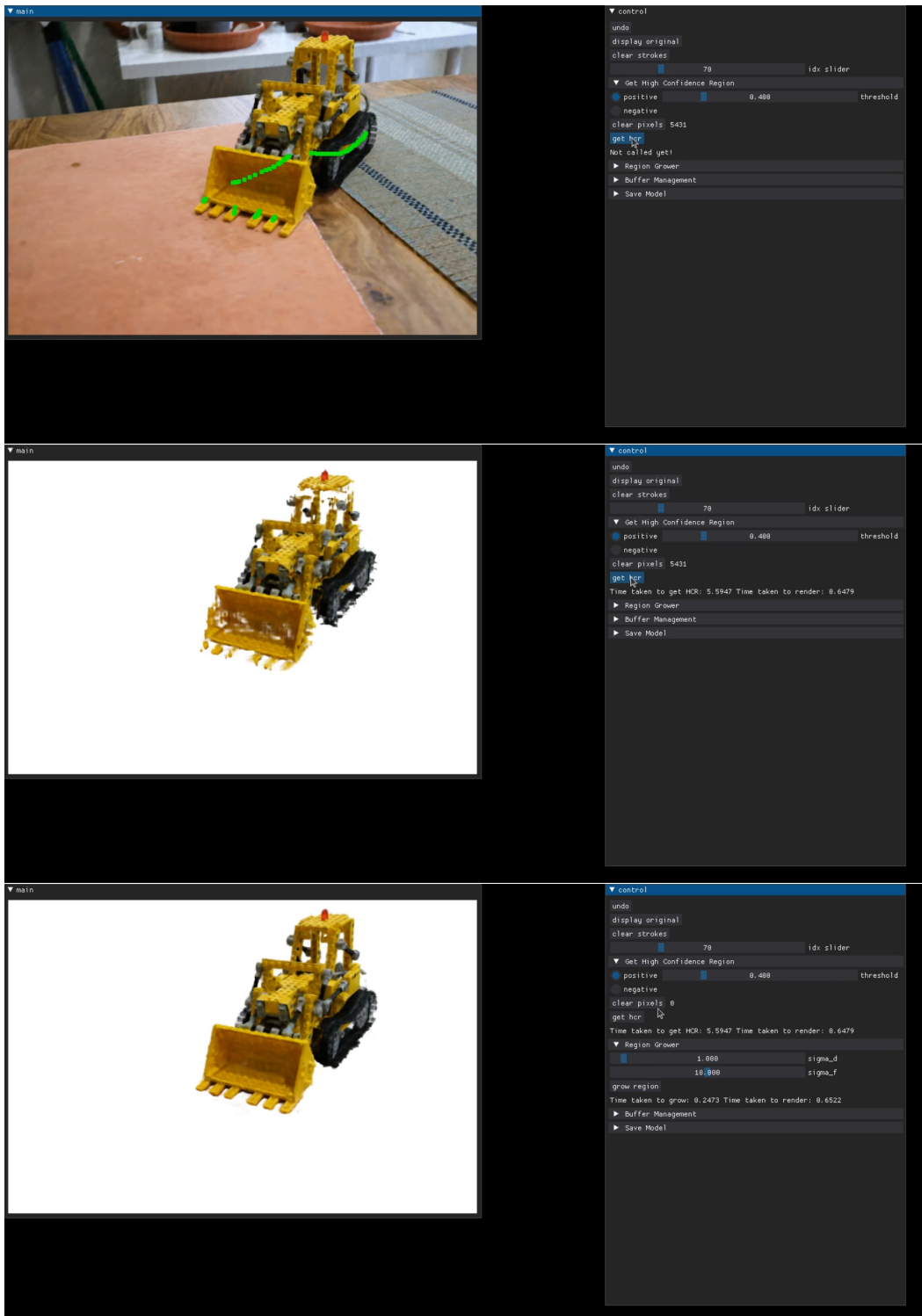
Figure 14. *Interactive GUI Tool:* We also release an easy-to-use interactive GUI tool which can be used to draw strokes and segment radiance fields.

Figure 15. *Student Surpasses Teacher*: The 4 columns of this figure shows the DINO features used as teacher vs the ones learnt by student post optimization. Since, the student learns finer features than the teacher due to assistance from the volumetric density, we can claim that the student surpasses the teacher. This is consistent with the prior art N3F and DFF.