Smoothness in Distilled Feature Fields

Sruthi Srinivasan and Umangi Jain

Abstract—Neural radiance fields (NeRFs) are a popular approach for rendering novel views and have been extended to a wide range of tasks such as editing, segmentation, and language-driven applications. Feature field distillation is a technique that facilitates editing and zero-shot segmentation by utilizing knowledge from large-scale 2D extractors such as CLIP and LSeg to learn a 3D feature field that is optimized in parallel to the radiance field. However, naive distillation can contain unwanted high-frequency artifacts, hampering fine-grained control and resulting in imprecise scene decomposition. In our work, we address this challenge by generating smoother Distilled Feature Fields (DFF) through the incorporation of explicit regularizers and off-the-shelf segmentation masks. Subsequently, we assess the effectiveness of these smoothened features through scene editing. Our experimental results demonstrate that the feature fields generated through our proposed method exhibits better smoothness compared to existing approaches. Our code is available at: link.

Index Terms—Distillation, Radiance field, Representation learning, Scene editing

1 Introduction

Neural radiance fields (NeRFs) [1] have emerged as a compelling approach for novel view synthesis of 3D scenes. However, the outputs of NeRFs themselves are low-level representations of the geometry and color of the scene, devoid of context and useful high-level semantics [2]. Recent works in label transfer, including Panoptic NeRF [3] and Semantic NeRF [4] have shown progress towards more holistic semantic understanding of the scene. These approaches have been further extended to transferring dense image features to neural renderers in [5] [2] [6]. Since utilizing dense representation for scenes by directly using off-the shelf 2D feature extractors results in features which are not view-consistent, these approaches distill the knowledge of feature extractors into a 3D student network. The concept of feature field distillation helps in representing 3D scenes in terms of semantically meaningful features in addition to the underlying geometry and color.

Distilled feature fields (DFF) are 3D neural feature fields that map every 3D coordinate in a scene to a semantic feature descriptor of that coordinate. Using the concept of teacher-student distillation [7], a DFF is learned and optimized in parallel to the radiance field by utilizing knowledge from pretrained 2D feature encoders for supervision. While DFFs have shown promising results for segmentation and editing, they contain high-frequency artifacts as a result of the underlying connectionist approach of the architecture and the stratified sampling, critically required by NeRF models, for training accurate radiance fields. In low spatial frequency tasks such as editing and segmentation of natural scenes, high-frequency noise resulting from fine-grained sampling is undesirable, emphasizing the importance of smoothness in distilled feature fields. However, the problem remains largely unaddressed. Kobayashi *et al.* [5] address this issue and adapt an ad hoc approach by only using coarse sampling for the feature fields. Nonetheless, as shown in Figure 1, even the segmentation results from the coarse MLP suffer from high-frequency artifacts.

In this work, we address the critical challenge of enhancing feature field distillation for NeRFs. The primary objective is to promote smoothness and minimize high-frequency artifacts prevalent in DFFs, ultimately leading to more precise and visually appealing scene decomposition. The proposed method leverages the Segment Anything Model (SAM) [8] to generate segmentation masks for every rendered feature in the DFF pipeline. Gaussian blur is then applied inside these segments obtained from the SAM model. The key idea behind this approach is to make the model more object-aware and facilitate edge-preserving blurring. We test our approach against Total Variation (TV) [9] and Bilateral Filtering [10] baselines since these two techniques are also known to enforce smoothness while preserving edges. Subsequently, we assessed the effectiveness of the DFFs obtained from the proposed approach on editing experiments, drawing a comparison against the naive distillation approach.

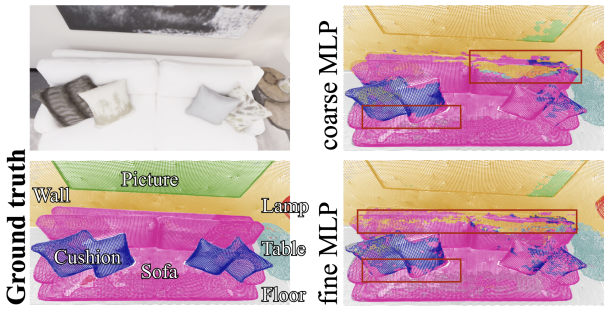


Fig. 1. High-frequency artifacts in distilled feature fields. Image taken from [5]. Boxes have been added around the pillow and sofa edges for highlighting the irregular decomposition.

2 Related Work

2.1 Semantic and Object-aware Neural Rendering

There have been some efforts to introduce semantics into radiance fields. Semantic-NeRF [11] incorporates multi-view semantic fusion of 2D labels, while NeSF [12] utilizes density fields as input to a 3D segmentation model. However, these methods are demonstrated on synthetic scenes with

limited shapes and categories. Further Kundu *et al.* [3] proposed an object aware panoptic approach to handle dynamic scenes but this method required 3D labels for training.

2.2 Feature Field Distillation

Unlike label distillation, feature distillation provides denser representation for a scene and also enables wider downstream applications. It leverages the immense developments in 2D scene understanding for 3D feature fields. In the context of feature fields for 3D scenes, the output of a 2D teacher network (pre-trained 2D foundational vision models) is distilled into a student network (that implements a 3D feature field) thus allowing for label free scene understanding. The student network is in the 3D domain and the teacher network in 2D domain. Kobayashi *et al.* [5] employed LSeg [13] and DINO [14] as teacher networks, distilling their outputs into 3D feature fields for tasks like 3D semantic segmentation and editing. Kerr *et al.* proposed LERF [6], which integrates 2D CLIP [15] embeddings into NeRF, producing 3D CLIP embeddings for generating relevancy maps in response to text queries. Tschernezki *et al.* [2] explore distilling 2D semantic features from teacher networks (DINO, MoCo-v3 [16], DeiT [17]) into their 3D model (NeuralDiff [18]). Although naive distillation in DFFs, built on NeRF and its variants, tends to suffer from high-frequency noise, that can make scene decomposition irregular.

2.3 Smoothness Approaches and Priors

In the domain of neural networks, Ramasinghe *et al.* [19] investigate regularization in coordinate-MLPs. Another method [20] applies regularization on the geometry of a NeRF for view synthesis from sparse inputs. Notably, there is a lack of established techniques specifically addressing the smoothness issue in distilled feature fields. Building on the work by Kobyashi *et al.* [5], our approach aims to generate smoother Distilled Feature Fields through the incorporation of explicit regularizers and segmentation masks.

3 Proposed Method

In this section, we briefly review the distillation process to learn 3D feature fields for scenes from 2D-image teacher model and describe our approach to smooth the rendered featured fields obtained from distillation. Section 3.1 provides the general set-up of feature distillation from 2D to 3D. Segment Anything (SAM) is described in section 3.2 and sections 3.3 and 3.4 discusses our method and implementation details, respectively. Overview of our approach can be found in Figure 2.

3.1 Review of feature distillation

Feature fields for 3D scenes are built on top of existing NeRF (and its variants) models. Given a point coordinate $\mathbf{x} = (x, y, z)$ and a view direction $\mathbf{d}$ in a 3D scene, NeRF maps it to an output density $\sigma(x)$ and color $\mathbf{c}(\mathbf{x}, \mathbf{d})$ using multilayer perceptions. To learn a student feature field model, NeRF produces, in addition to $\sigma(x)$ and $\mathbf{c}(\mathbf{x}, \mathbf{d})$, a feature vector $\mathbf{f}(\mathbf{x})$. Similar to the volume rendering for the color, a feature rendering is applied to the predicted feature field to produce feature maps for the rendered scene $\hat{\mathbf{F}}(\mathbf{r})$ (where $\mathbf{r}$ is the pixel camera ray).

$$\hat{\mathbf{F}}(\mathbf{r}) = \sum_{k=1}^{K} \hat{T}(t_k)\alpha(\sigma(\mathbf{x}_k)\delta_k)\mathbf{f}(\mathbf{x}_k) \tag{1}$$

where $\hat{T}(t_k) = exp(-\sum_{k'=1}^{k-1}\sigma(\mathbf{x}_{k'})\delta_{k'})$, $\alpha(x) = 1 - exp(-x)$ and $\delta_k = t_{k+1} - t_k$ (distance between adjacent point samples. Rendered colors, $\hat{\mathbf{C}}(\mathbf{r})$, are also produced using the same rendering equation.

These rendered features are supervised by the teacher's feature $\mathbf{f}_{img}(I, r)$, obtained from a 2D feature extractor. Along with the photometric loss L_p that NeRF minimizes, an additional feature loss L_f is minimized for learning the feature field. The total loss is given by:

$$L = L_p + \lambda L_f \tag{2}$$

$$L_p = \sum_{\mathbf{r}\in R} ||\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})||_2^2, L_f = \sum_{\mathbf{r}\in R} ||\hat{\mathbf{F}}(\mathbf{r}) - \mathbf{f}_{img}(I, r)||_1 \tag{3}$$

where $\mathbf{C}(\mathbf{r})$ is the ground truth pixel color of ray $\mathbf{r}$ and λ is a hyper-parameter to weigh the two losses. A stop-gradient is applied to density when feature rendering. As the teacher's features are not 3D consistent, it could potentially harm the learnt geometry of the scene.

3.2 Segment Anything Model

Segment Anything model (SAM) [8] is a foundational model for the task of image segmentation. It is trained on a dataset of 11 million images with over 1 billion masks. As a consequence of being trained on a large and diverse dataset, SAM efficiently adapts to a diverse image distributions and tasks without any further fine-tuning. Extensive evaluation of segmentation masks produced by SAM has shown that the zero-shot segmentation outputs are remarkable - even surpassing fully supervised approaches for certain settings.

For a given image, SAM produces multiple masks and their corresponding confidence score. These masks are often objects or sub-parts of objects. We apply SAM to the rendered images $\hat{\mathbf{C}}(\mathbf{r})$ to obtain segments in the image. We also apply non maximal suppression to remove overlapping masks and only consider masks with an area above a certain threshold (to reduce any additional noise that might be introduced). A subset of all the SAM masks for a rendered image in a drum set is shown in Figure 3. The resulting object decomposition is of a much lower spatial frequency than the geometry of the rendered image.

3.3 Feature smoothening

Owing to the underlying connectionist representation in NeRFs, the volume decomposition of scene using DFFs might not always be smooth. To eliminate high-frequency artifacts from the decomposition, we propose smoothing using masks obtained from SAM.

We follow [5] by first training the NeRFs without a feature branch and optimizing only for the radiance field (L_p). We then switch to fine-tune the feature fields in conjunction with the radiance fields ($L_p + L_f$) as the radiance field converges much faster when the geometry is well-trained. When fine-tuning for the feature fields, since the geometry

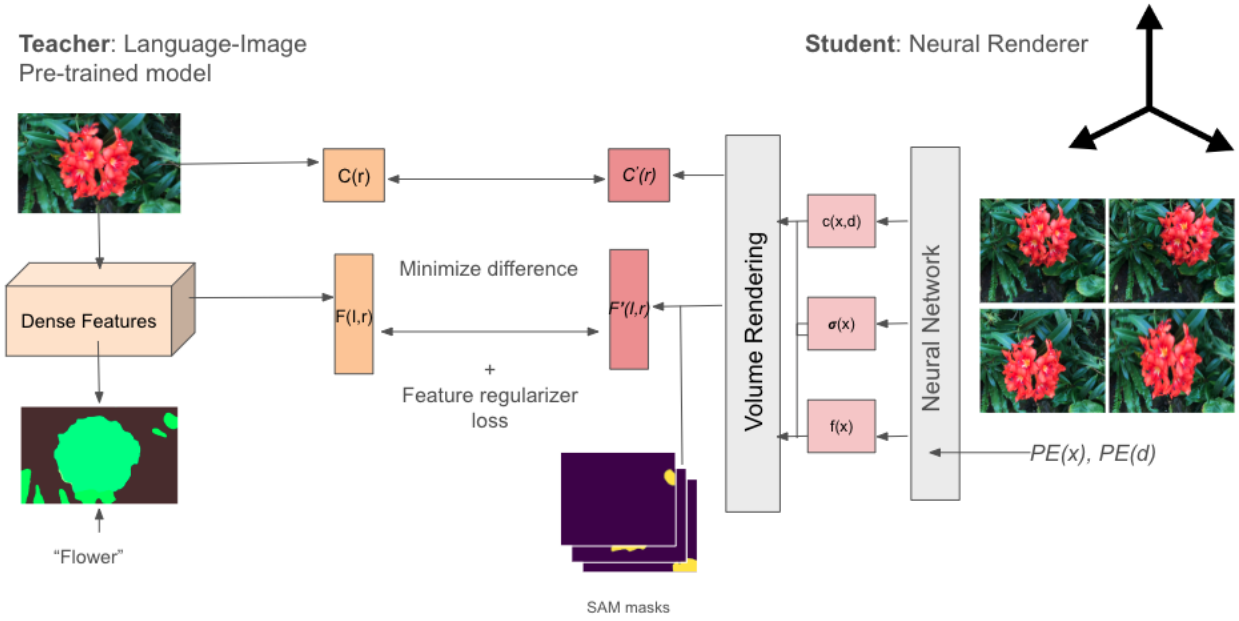


Fig. 2. Overview of the distillation process from language-2D image teacher model to 3D student model. Masks from Segment Anything Model (SAM) used for smoothing the rendered features.

Fig. 3. An image from the rendered scene (left) is passed through the Segment Anything Model to obtain semantic masks. Shown above are four samples from all the masks.

and color of the scene is already sufficiently trained, we pass the rendered image from the model to SAM to generate non-overlapping masks of low-spatial frequency for the image. We then use these masks to selectively blur features falling in that particular mask. The Gaussian blurring helps features from the same object or parts of the same object to eliminate high-frequency noise. This smoothed features are then used for the feature loss (Figure 2).

We also test our model on bilateral filtering and explicit regularizers such as anisotropic total variation to help in cohesive decomposition.

- **Total Variation (TV)** [9]: TV regularization is a technique introduced for image denoising and reconstruction. It serves as a measure of the variation in input intensity over its domain. TV helps in achieving a more cohesive decomposition of features by encouraging spatial continuity. This approach is particularly effective in maintaining edge information while smoothing out noise and small fluctuations in uniform regions of the image. For each dimension of the rendered features, we add an additional regularizer, given by:

$$
\begin{aligned}
L_{ani\_tv} &= \sum_{ij} ||(D_h \hat{\mathbf{F}})_{ij}||_1 + ||(D_v \hat{\mathbf{F}})_{ij}||_1 \\
L_{iso\_tv} &= \sum_{ij} \sqrt{(D_h \hat{\mathbf{F}}^2_{ij}) + (D_v \hat{\mathbf{F}}^2_{ij})} \\
L &= L_p + \lambda_f L_f + \lambda_{tv} L_{tv}
\end{aligned} \tag{4}
$$

where D_h and D_v are horizontal and vertical finite difference operators respectively.

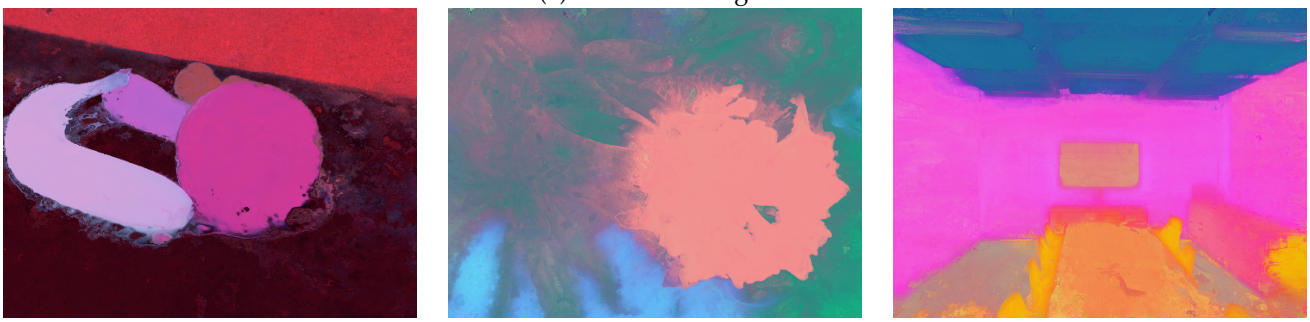
- **Bilateral Filtering** [10]: Bilateral filtering is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. It differs from typical Gaussian blurring by considering both the spatial distance and the intensity difference when performing the smoothing. This ensures that edges are preserved while reducing noise. We apply bilateral filter on the rendered features as another baseline.

3.4 Implementation Details

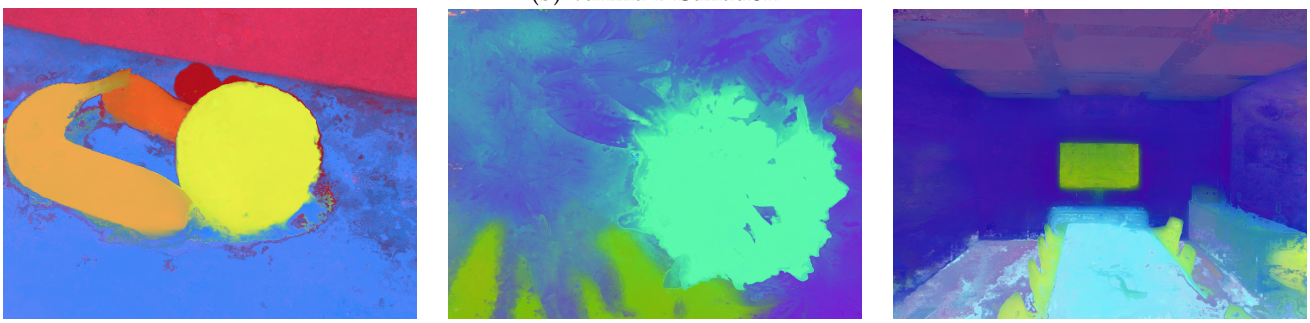
In this section, we discuss the training details and design choices for the model.

Teacher model: For the teacher model we use LSeg [13], a model developed for zero-shot semantic segmentation by

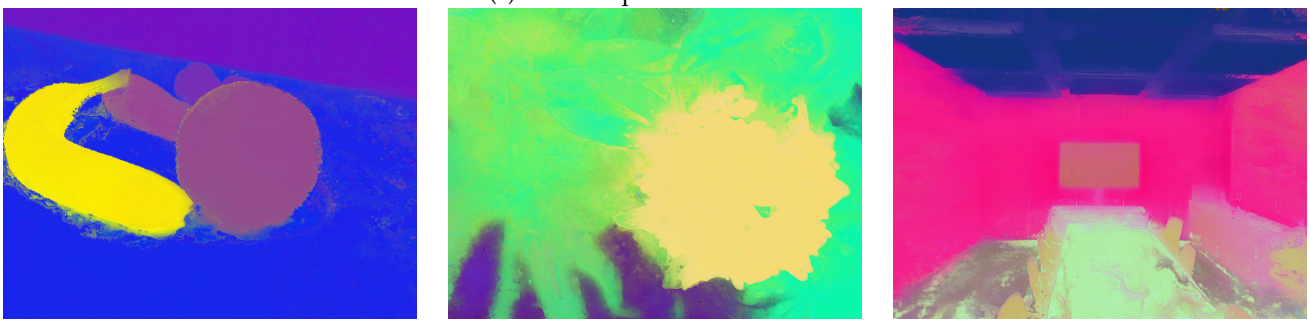
(a) Rendered Images



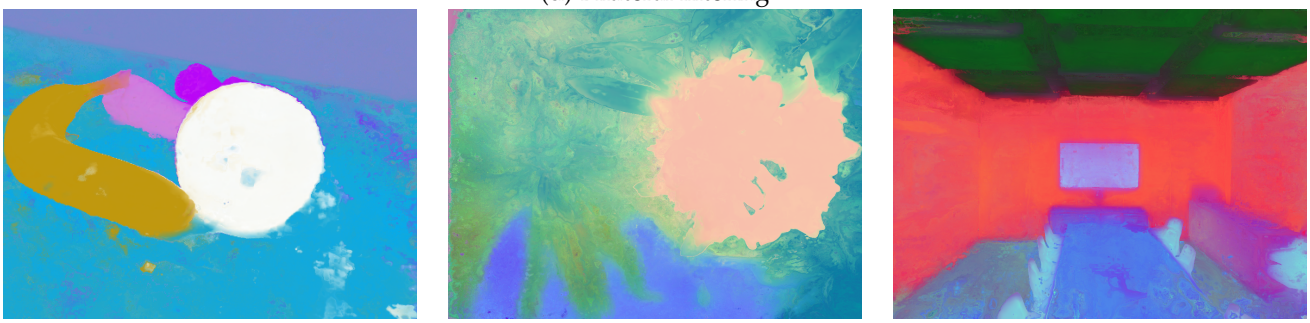
(b) Vanilla Distillation



(c) Anisotropic total variation



(d) Bilateral filtering



(e) SAM-guided Gaussian blurring

Fig. 4. Visualization of different smoothing techniques on three scenes of vegetables, flower, and room.

aligning pixel and text query feature. It used an image encoder with DPT architecture [21] and a CLIP-based [15] text encoder, $\mathbf{f}_t$. For downstream applications, given a pixel r in an image I, the probability of a text l is given by

$$p(l|I,r) = \frac{exp(\mathbf{f}_{img})\mathbf{f}_t(l)^T}{\sum_{l' \in L} exp(\mathbf{f}_{img})\mathbf{f}_t(l')^T} \quad (5)$$

where L is the set of possible labels. This is used for obtaining parts of the scene aligned with the text prompt for segmentation/editing. The features obtained from the teacher model are of a reduced size and are resized to the original size of the image for training. The feature length for each pixel in the image is 512 by default.

Student model: While the general idea of feature field distillation can be used with several NeRF-like models, we use a variation of Torch-NGP[1]. We keep all the other parameters (depth and size of MLP for color, density, and features) similar to [5].

Regularizers: For SAM-guided Gaussian blurring, we use a kernel of size 3 and sigma is computed using the default PyTorch setting of $0.3 \times ((kernel\_size - 1) \times 0.5 - 1) + 0.8$. We set the IoU cutoff used by non-maximal suppression to be 0.2 to avoid computing over the same pixels repeatedly. The hyper-parameter for weighting features λ_f is kept at 10^{-2}. For the TV regularizer, the weight of TV loss is set at $\lambda_{tv} = 10^{-3}$. For the bilateral filter, the diameter for each pixel neighbourhood is 15. No fine-tuning has been performed on these choices.

Model training: We train the model for 5 epochs. Compared to the DFFs trained in [5] for 20 epochs, our rendered features are therefore inferior in quality. To compare against their approach, we train all the scenes for only 5 epochs. While most scenes can benefit from further training, this choice is set due to the computational and time constraints.

4 Experimental Results

In this section, we detail the dataset and downstream application in section 4.1 and the results obtained are discussed in section 4.2.

4.1 Experimental Setup

4.1.1 Dataset

We test the smoothness priors on the dataset released in [1] and [5]. Each scene has between 34-100 images, allowing us to test the models for performance with varied number of view availability.

4.1.2 Downstream task and metrics

We test the model on the task of editing that includes extraction, deletion, and colorization. For editing, the selected region is found using query-based scoring and an appropriate transformation is applied. For the task of deletion, the density of the selected points is set to zero and for the task of colorization, color is editing by a coloring function. Since editing is a subjective task, we present qualitative results in Figure 5.

1. https://github.com/kwea123/ngp_pl

TABLE 1
Quantitative metrics on the rendered image and rendered features. PSNR, SSIM, LPIPS are measured on the rendered and the ground truth image. Cosine similarity and MSE are calculated between the rendered feature vector and the teacher feature vector.

	PSNR	LPIPS	SSIM	Cos	MSE
Vanilla [5]	26.19	0.3717	0.8373	0.9633	1.38e-4
Total Variation	26.86	0.3622	0.8505	0.9804	7.65e-4
Bilateral	26.38	0.3689	0.843	0.9698	1.14e-4
SAM-guided	26.18	0.3676	0.8496	0.9764	9.03e-4

We use SSIM [22], LPIPS [23], PSNR for the rendered image and and show mean squared error and cosine similarity between the rendered features and the features obtained from the teacher network. Although, it is worth noting that these metrics for rendered features are not extensive and the performance of the feature depends on the downstream application.

4.2 Results

To qualitatively assess the features obtained from distillation through different approaches, we reduce the dimensionality of the rendered features using Principal Component Analysis (PCA). Figure 4 shows the quality of rendered features for three different scenes. For all the three scenes, it can be seen that the regularizations and blurring provides sharper features. Compared to the baseline of vanilla distillation, all approaches result in smoother segments. For instance, for the vegetables scene, the curves around the *apple* are smoother and there is also reduced noise next to the *banana*. Similarly, for the flower scene, the petals on the left side of the flower are more clearly demarcated. In the room scene, the edges of the table and television also looks sharper. Table 1 shows quantitative results between the different approaches. It can be seen that feature smoothing does not hurt the geometry of the underlying scene as the rendered images metrics do not go down.

While the smoothing techniques shows improvement, a quantitative comparison between them is difficult as the feature quality depends on the downstream application. For bilateral filtering, it was observed that it is prone to over smoothing (can be seen the flower scene where the petals have been blurred). Total variation and SAM-based blurring both perform better on different scenes.

We use these feature representation for editing. Figure 5 compares performance of vanilla distillation, total variation, and SAM-guided blurring for the vegetables and flower scene. Editing the room scene was difficult as it is a complex scene and our models were only trained for 5 epochs. As seen in the figure, the vanilala distillation scene has more noise in the background, compared to our approaches which reduce these noises. For the deletion editing, total variation shows better inpained results than SAM-based and vanilla approach.

5 Conclusion and Future Work

In this work, we propose a SAM-guided method for the rendering of smoother Distilled feature fields (DFFs). Additionally, we also test our scenes on simpler baselines of

(a) Task: **Extract** *"apple"* from the scene. (left) Vanilla distillation, (center) Total variation, (right) SAM-guided

(b) Task: **Delete** *"flower"* from scene. (left) Vanilla distillation, (center) Total variation, (right) SAM-guided

Fig. 5. Extraction and deletion of objects in the scenes using different smoothing techniques.

total variation (TV) and bilateral filtering. Through editing experiments, we show that the SAM-guided method and TV produces smoother feature fields as compared to naïve DFFs. On the other hand, bilateral filtering over-smooths the segments. Our observations, supported by PCA-based feature map visualizations and editing experiments, indicate that additional smoothing techniques improve the baseline naïve DFF (especially evident for the backgrounds in the scene). Moreover, quantitative metrics reveal that the application of these smoothness techniques does not compromise the geometry of the scene and demonstrates high similarity with the teacher model features.

Future endeavors include extending the qualitative and quantitative evaluation of these feature fields to different downstream tasks, such as segmentation. An intriguing avenue for further exploration involves studying the influence of various 2D extractors on the smoothness of the rendered feature fields. An ablation study on varying the SAM masks granularity and testing the effects on the learnt geometry is also an interesting future work.

6 Acknowledgements

We express our gratitude to Dr. David Lindell and the course staff for organizing the CSC2529 course and for helping and guiding us through the project.

References

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[2] V. Tschernezki, I. Laina, D. Larlus, and A. Vedaldi, "Neural feature fusion fields: 3d distillation of self-supervised 2d image representations," in *2022 International Conference on 3D Vision (3DV)*. IEEE, 2022, pp. 443–453.

[3] A. Kundu, K. Genova, X. Yin, A. Fathi, C. Pantofaru, L. Guibas, A. Tagliasacchi, F. Dellaert, and T. Funkhouser, "Panoptic neural fields: A semantic object-aware neural scene representation," 2022.

[4] S. Zhi, T. Laidlow, S. Leutenegger, and A. J. Davison, "In-place scene labelling and understanding with implicit scene representation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 838–15 847.

[5] S. Kobayashi, E. Matsumoto, and V. Sitzmann, "Decomposing nerf for editing via feature field distillation," 2022.

[6] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "Lerf: Language embedded radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 729–19 739.

[7] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[8] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[9] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[10] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*. IEEE, 1998, pp. 839–846.

[11] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Addison-Wesley, 1999.

[12] S. Vora, N. Radwan, K. Greff, H. Meyer, K. Genova, M. S. M. Sajjadi, E. Pot, A. Tagliasacchi, and D. Duckworth, "Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes," 2021.

[13] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," 2022.

[14] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2023.

[15] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.

[16] X. Chen and S. Xie, "and kaiming he. an empirical study

of training self-supervised vision transformers," *arXiv preprint arXiv:2104.02057*, vol. 2, no. 5, p. 6, 2021.

[17] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International conference on machine learning*. PMLR, 2021, pp. 10 347–10 357.

[18] V. Tschernezki, D. Larlus, and A. Vedaldi, "Neuraldiff: Segmenting 3d objects that move in egocentric videos," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 910–919.

[19] S. Ramasinghe, L. MacDonald, and S. Lucey, "On regularizing coordinate-mlps," *arXiv preprint arXiv:2202.00790*, 2022.

[20] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. Sajjadi, A. Geiger, and N. Radwan, "Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5480–5490.

[21] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 179–12 188.

[22] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[23] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.