Informative Rays Selection for Few-Shot Neural Radiance Fields

Marco Orsingher[1,3], Anthony Dell'Eva[2,3], Paolo Zani[3], Paolo Medici[3] and Massimo Bertozzi[1]

[1] *University of Parma, Italy*

[2] *University of Bologna, Italy*

[3] *VisLab, an Ambarella Inc. company, Italy*

Keywords: Neural Radiance Fields, Novel View Synthesis, Few-Shot Learning, 3D Reconstruction.

Abstract: Neural Radiance Fields (NeRF) have recently emerged as a powerful method for image-based 3D reconstruction, but the lengthy per-scene optimization limits their practical usage, especially in resource-constrained settings. Existing approaches solve this issue by reducing the number of input views and regularizing the learned volumetric representation with either complex losses or additional inputs from other modalities. In this paper, we present *KeyNeRF*, a simple yet effective method for training NeRF in few-shot scenarios by focusing on key informative rays. Such rays are first selected at camera level by a view selection algorithm that promotes baseline diversity while guaranteeing scene coverage, then at pixel level by sampling from a probability distribution based on local image entropy. Our approach performs favorably against state-of-the-art methods, while requiring minimal changes to existing NeRF codebases.

1 INTRODUCTION

3D reconstruction and novel view synthesis from a set of calibrated images is a longstanding challenge in computer vision, with applications in robotics (Van der Merwe et al., 2020; Wang et al., 2019), virtual reality (Cao et al., 2021; Yeh and Lin, 2018) and autonomous driving (Orsingher et al., 2022a; Orsingher et al., 2022b). Recently, Neural Radiance Fields (NeRF) (Mildenhall et al., 2020) have been introduced to model 3D scenes with a small neural network that can be queried with any point in space to produce the corresponding density and view-dependent color. In order to enable end-to-end training from images, differentiable rendering is used to integrate a set of points along each camera ray.

The simple formulation of NeRF and its unprecedented rendering quality are arguably the main reasons of its popularity. However, it also suffers from long training times, since each pixel of each input view must be seen multiple times until convergence. To tackle this issue, a possible solution is to reduce the number of input views and learn a 3D scene representation from few sparse cameras. Existing few-shot methods focus on regularizing the volumetric density learned by NeRF with new loss functions (Kim et al., 2022; Yang et al., 2023) and additional inputs (Jain et al., 2021; Deng et al., 2022; Roessle et al., 2022; Wynn and Turmukhambetov, 2023; Niemeyer et al., 2022), thus introducing complexity in the pipeline.

Furthermore, they all assume to be given a random set of viewpoints, without control on how such cameras are selected. However, in common use cases, such as object scanning from videos acquired by a user with handheld devices, the input data consist in a dense and redundant set of frames with a known acquisition trajectory. Our insight is to better exploit such information in the input views. To this end, we propose a method, called *KeyNeRF*, to identify key informative samples to focus on during training.

Firstly, we select the best input views by finding a minimal set of cameras that ensure scene coverage. Secondly, this initial set is augmented with a greedy algorithm that promotes baseline diversity. Finally, we choose the most informative pixels for each view, in terms of their local entropy in the image. Our rays selection procedure is extremely flexible, as it operates directly at input level, and it can be implemented by only changing two lines of any existing NeRF codebase. The proposed approach outperforms state-of-the-art methods on standard benchmarks in the considered scenario, while not requiring additional inputs and complex loss functions. Our contribution to existing literature is threefold:

1. We present a view selection algorithm that starts from the minimal set of cameras covering the

Orsingher, M., Dell'Eva, A., Zani, P., Medici, P. and Bertozzi, M.
Informative Rays Selection for Few-Shot Neural Radiance Fields.
DOI: 10.5220/0012303600003660

In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 4: VISAPP*, pages 253-261
ISBN: 978-989-758-679-8; ISSN: 2184-4321

scene and iteratively adds the next best view in a greedy way.

2. We propose to sample pixels in a given camera plane by following a probability distribution induced by the local entropy of the image.
3. To the best of our knowledge, our framework is the first few-shot NeRF approach that operates at input level, without requiring additional data or regularization losses.

2 RELATED WORK

Few-Shot NeRF. The original formulation of NeRF (Mildenhall et al., 2020) requires a large set of cameras to converge, thus leading to long training times. For this reason, several methods (Niemeyer et al., 2022; Kim et al., 2022; Seo et al., 2023; Wynn and Turmukhambetov, 2023; Jain et al., 2021; Deng et al., 2022) have been proposed to allow learning radiance fields from few sparse views. All these approaches introduce new loss functions to regularize the underlying representation. However, such losses might be difficult to balance and they are in contrast with one of the main advantages of NeRF, which can be trained in a self-supervised way from images with a simple MSE loss. Moreover, most of them further assume to have additional inputs, such as depth measurements (Deng et al., 2022; Roessle et al., 2022) or other pre-trained networks (Jain et al., 2021; Niemeyer et al., 2022; Wynn and Turmukhambetov, 2023). Specifically, DS-NeRF (Deng et al., 2022) and DDP-NeRF (Roessle et al., 2022) require sparse depth to guide sampling along each ray and optimize rendered depth. DietNeRF (Jain et al., 2021) enforces high-level semantic consistency between novel view renderings with pre-trained CLIP embeddings, while RegNeRF (Niemeyer et al., 2022) and DiffusioNeRF (Wynn and Turmukhambetov, 2023) maximize the likelihood of a rendered patch according to a given normalizing flow or diffusion model, respectively.

View Selection. Another shortcoming of existing few-shot NeRF methods is that the input dataset is sampled at random, both in terms of cameras and pixels. While this is a general assumption, it is also suboptimal in typical use cases, such as object scanning from videos, since random sampling discards geometric information about the scanning trajectory. To this end, we propose a view selection algorithm that guarantees scene coverage in an optimal way and promotes baseline diversity with a greedy procedure. View selection for 3D reconstruction has been studied in literature mainly in the context of large-scale scenarios, such as city-scale (Orsingher et al., 2022a) or building-scale (Mauro et al., 2014; Furukawa et al., 2010; Ladikos et al., 2009) reconstruction. The usual approach is to build a visibility matrix between all the possible pairs of cameras, and to employ either graph theory (Furukawa et al., 2010; Ladikos et al., 2009) or integer linear programming (Mauro et al., 2014; Orsingher et al., 2022a) to find a relevant subset of the available views. Besides targeting a different use case, such methods have several drawbacks in the considered setting. Firstly, the visibility matrix formulation assumes to have a set of sparse keypoints as input, while our approach works only with calibrated cameras. Secondly, it requires the desired number of cameras to be specified a priori, whereas KeyNeRF schedules all the views at once. Furthermore, we improve upon existing methods by explicitly and iteratively enforcing baseline diversity.

Rays Sampling. The aforementioned few-shot approaches treat all pixels equally for a given input image and sample a random batch at each iteration. Other methods (Sucar et al., 2021; Pan et al., 2022) pioneered the use of uncertainty-based sampling of rays and estimate such uncertainty online, which leads to a computational overhead. On the other hand, we propose to compute the local entropy of the image offline and to draw pixels from such distribution, which represents by definition the most informative rays.

3 METHOD

We present a framework, based on NeRF (Section 3.1), for novel view synthesis and 3D reconstruction from a given set of N calibrated cameras. We assume to have a dense and redundant set of views, such as the frames of a video acquired by a user for object scanning. Our method, named *KeyNeRF*, identifies the key information in the given set of views by greedily selecting a subset of cameras (Section 3.2) and choosing the most informative pixels within such cameras (Section 3.3) with entropy-based sampling. The proposed approach improves the efficiency of NeRF, while requiring minimal code changes to its implementation. Assuming NumPy imported as `np`, both cameras and rays are drawn uniformly in NeRF:

```
pose_idx = np.random.choice(num_poses)
rays_idxs = np.random.choice(
    num_rays, size = B, p = None
)
```

In KeyNeRF, we simply reduce the set of input views and change the probability distribution for sampling pixels. Differences in the code are highlighted with bold characters:

```
pose_idx = np.random.choice(select_cams)
rays_idxs = np.random.choice(
    num_rays, size = B, p = entropy
)
```

Our rays selection procedure is extremely flexible, as it operates directly at input level and it can be seamlessly integrated with other any NeRF approach, since it is orthogonal to improvements in loss functions or field representations (Liu et al., 2020; Chen et al., 2022; Müller et al., 2022).

3.1 Preliminaries

A Neural Radiance Field (NeRF) (Mildenhall et al., 2020) is a continuous and implicit representation of a 3D scene. A small MLP F_θ maps any point $\mathbf{x} \in \mathbb{R}^3$ in space and a viewing direction $\mathbf{d} \in \mathbb{S}^2$ to its corresponding density $\sigma(\mathbf{x}) \in \mathbb{R}^+$ and view-dependent color $\mathbf{c}(\mathbf{x}, \mathbf{d}) \in \mathbb{R}^3$. This network is trained by minimizing a reconstruction loss between the ground truth colors in the input images and the rendered colors from the radiance field. For each camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ with origin $\mathbf{o}$ and oriented as $\mathbf{d}$, the corresponding color $\hat{I}(\mathbf{r})$ is computed as:

$$\hat{I}(\mathbf{r}) = \int_{t_n}^{t_f} \exp\left(-\int_{t_n}^{t} \sigma(s)ds\right) \cdot \sigma(t) \cdot \mathbf{c}(t)dt \qquad (1)$$

where $[t_n, t_f]$ is the integration boundary. In practice, both integrals are approximated by numerical quadrature with a discrete set of samples along each ray. More details can be found in (Mildenhall et al., 2020). Given the ground truth color $I(\mathbf{r})$ for the ray $\mathbf{r}$, NeRF optimizes a batch of B rays at each iteration. Given N cameras with M pixels each, an epoch needs $\frac{NM}{B}$ iterations and training requires multiple epochs. Our insight is to focus on the most informative cameras and pixels, thus significantly reducing N and M.

3.2 View Selection

The goal of a view selection procedure is to sample K views from a dense set of N available cameras ($K \ll N$) for efficient 3D reconstruction, while (i) maintaining the visibility of the whole scene and (ii) ensuring diversity within the selected subset. Inspired by (Orsingher et al., 2022a), we propose to satisfy the first constraint by solving a simple optimization problem to find the minimal set of cameras that guarantee scene coverage. Then, a greedy algorithm iteratively adds the camera with the most diverse baseline, until all the views have been scheduled.

3.2.1 Scene Coverage

In the first phase, cameras are represented by binary variables $x_i \in \{0, 1\}$ as in (Orsingher et al., 2022a), and the scene is approximated with a uniform 3D grid of M points within bounds $(\mathbf{p}_{min}, \mathbf{p}_{max})$. Differently from (Orsingher et al., 2022a), we do not assume to have sparse keypoints as input, but they can be added to the scene, if available. Let $\mathbf{A}_j \in \mathbb{R}^N$ be a visibility vector with elements $a_{ij} = 1$ if point j is visible in camera i, 0 otherwise. The following integer linear programming (ILP) problem is then formulated:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} x_i \\ \text{s.t.} \quad & \mathbf{A}_j^\top \mathbf{x} > 0 \quad \forall j = 1, \ldots, M \end{aligned} \qquad (2)$$

The set of cameras selected in this way ensures scene visibility, but some regions of interest might not be fully covered with sufficient baseline for 3D reconstruction (see Figure 1, left).

3.2.2 Baseline Diversity

In order to promote baseline diversity, we design a greedy view selection algorithm to choose the next best camera among the available ones with respect to the currently selected set. We generate a $N \times N$ symmetric baseline matrix $\mathbf{B}$, where $b_{ij} = b_{ji}$ is the angle between the optical axes of cameras i and j:

$$b_{ij} = \arccos\left(\frac{\mathbf{z}_i^\top \mathbf{z}_j}{|\mathbf{z}_i| \cdot |\mathbf{z}_j|}\right) \qquad (3)$$

Then, at each iteration step, until the desired number of cameras has been reached, we add to the selected subset the camera with the highest relative angle with respect to *all* currently selected cameras, as shown in Figure 1. Practically, for each remaining view, we query from $\mathbf{B}$ its smallest score against the selected views and add to the subset the camera with the *highest* smallest score. Assuming NumPy imported as `np`, let `select_cams` be the output of the first stage:

```
while remaining_cams:
    sub_mat = B[remain_cams][:, select_cams]
    idx = np.argmax(np.min(sub_mat, axis = 1))
    select_cams.append(remain_cams[idx])
    remain_cams.remove(remain_cams[idx])
```

This formulation is different from (Orsingher et al., 2022a), where matchability is a hard constraint. Moreover, the iterative nature of the greedy procedure induces an implicit ranking on the set of cameras and allows the user to choose flexibly the desired K. This is a significant improvement with respect to (Orsingher et al., 2022a), where baseline diversity is not explicitly enforced, and a different optimization problem must be solved from scratch for

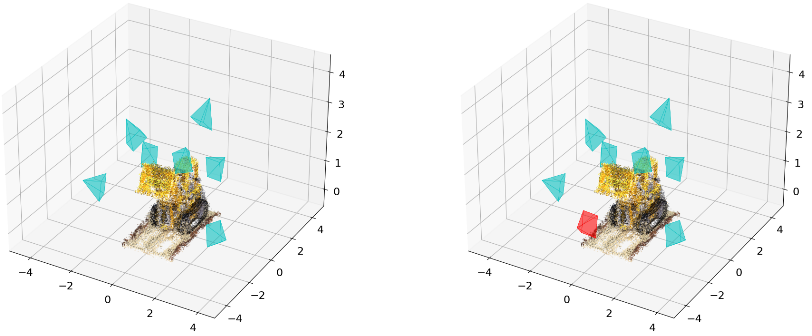


Figure 1: Illustration of the view selection procedure. The new camera (red, right) has the most diverse baseline with respect to the set of current cameras (blue, left). A proxy geometry of the scene is shown for reference.

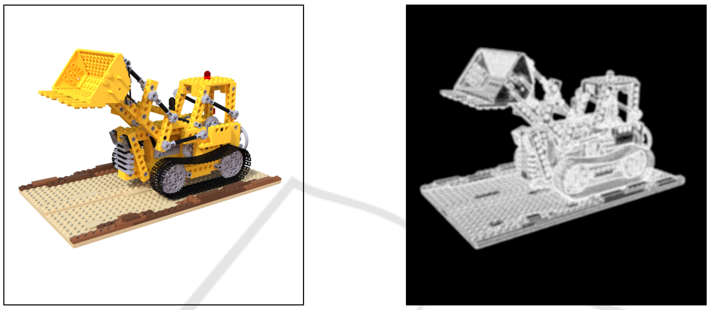


Figure 2: Probability distribution over pixels (right) for an example input image (left), induced by the local entropy of the image when sampling a batch of rays for training NeRF.

different values of K. We will show in the experiments that any $K \geq K_{min}$ leads to good results, where K_{min} is the cardinality of the minimal scene coverage set. Intuitively, more views progressively improve the performances, with diminishing returns towards the end, when cameras have large overlaps with the current set and do not add relevant information.

3.3 Rays Sampling

At each training iteration, NeRF (Mildenhall et al., 2020) samples a pose in the dataset and a batch of B pixels from such camera. Typically, rays are sampled uniformly from the whole set of available pixels. However, we observe that not all rays are equally informative about the scene. For example, the background or large textureless regions in the image could be covered with fewer samples, exploiting the implicit smoothing bias of MLPs (Ramasinghe et al., 2022). We propose to define a probability distribution over pixels and to focus on high-frequency details during training, in order to converge faster, especially in few-shot scenarios. The amount of information of a pixel p can be quantified by its local entropy:

$$e(p) = -\sum_{(u,v)\in\mathcal{W}} h_{uv} \log h_{uv} \tag{4}$$

where $\mathcal{W}$ defines a local window around p and h is the normalized histogram count. In order to allow random sampling, we normalize it to a probability distribution, which is the input to `np.random.choice()`. An example is shown in Figure 2.

4 EXPERIMENTS

4.1 Implementation Details

Dataset. We perform our experiments on two common benchmarks. The Realistic Synthetic 360° dataset (Mildenhall et al., 2020) contains 8 scenes of different objects with diverse materials and complex illumination, rendered by Blender from 400 random viewpoints. Moreover, we randomly select a subset of 8 scenes from the CO3D dataset (Reizenstein et al., 2021), which gathers a wide collection of object-centric videos from handheld devices. Due to the unavailability of pre-trained checkpoints of baseline methods on such data, a complete evaluation on this dataset would require a huge amount of computation, well beyond what we can access. To mitigate this issue, we randomly select a subset of 8 diverse scenes, both in terms of object category and individual sequence within each category. The rationale be-

Table 1: Quantitative results on the Blender dataset. Best and second results are **bold** and underlined, respectively.

Method	PSNR ↑	LPIPS ↓	SSIM ↑	Avg. ↓
NeRF (Mildenhall et al., 2020)	24.424	0.132	0.878	0.055
DietNeRF (Jain et al., 2021)	24.370	0.127	0.878	0.054
InfoNeRF (Kim et al., 2022)	24.950	0.117	0.884	0.050
KeyNeRF (w/o entropy)	25.568	0.109	0.895	0.046
KeyNeRF (ours)	**25.653**	**0.106**	**0.898**	**0.045**

Table 2: Quantitative results on the CO3D dataset. Best and second results are **bold** and underlined, respectively.

Method	PSNR ↑	LPIPS ↓	SSIM ↑	Avg. ↓
NeRF (Mildenhall et al., 2020)	20.708	0.491	0.744	0.128
DietNeRF (Jain et al., 2021)	19.994	0.511	0.728	0.138
InfoNeRF (Kim et al., 2022)	20.143	0.576	0.714	0.143
KeyNeRF (w/o entropy)	21.853	0.470	0.759	0.114
KeyNeRF (ours)	**22.183**	**0.463**	**0.762**	**0.109**

hind this choice is to establish a real-world equivalent of the synthetic dataset, with different objects, various aspect ratios and noisy camera poses from SfM (Schönberger and Frahm, 2016).

Parameters. For a given batch size B, at each iteration, we sample $B/2$ rays from the entropy-based distribution and $B/2$ rays at random to ensure full coverage. All the methods are trained for $N_{iter} = 50000$ iterations with $K = 16$ poses. Note that this is a different setup than the typical few-shot scenario, where $K \leq 8$ and training is much longer ($N_{iter} \geq 200000$). We argue that this setup is overlooked in the literature, despite having significant practical relevance. In the common case of object scanning from videos, it is reasonable to assume to have more than 8 frames and the actual goal is training efficiency. However, existing few-shot methods tend to saturate their contributions when $K > 8$ (Kim et al., 2022), as shown in Table 1 and Table 2, while our method shows improved results over a wide range of values of K (see Figure 3a). The influence of both the number of poses K and iterations N_{iter} is ablated in Section 4.3.

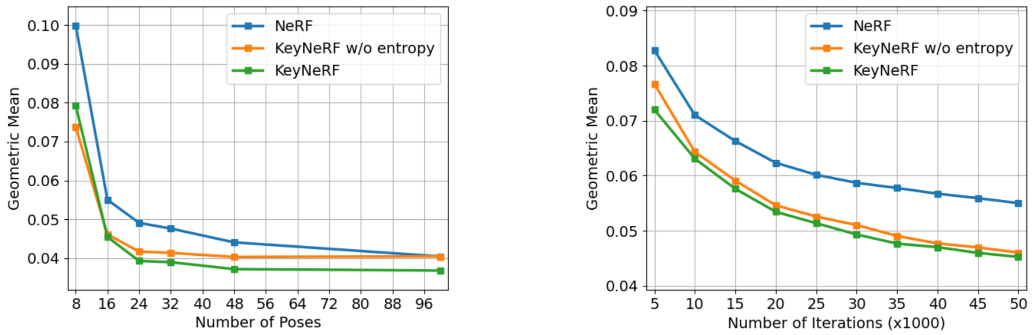
Code. The training code is based on a reference PyTorch version of NeRF (Yen-Chen, 2020). For the rays selection procedure, we solve the ILP with the OR-Tools library (Perron and Furnon, 2022) and compute the image entropy with the default implementation in scikit-image.

4.2 Quantitative Results

Following standard practice (Mildenhall et al., 2020), we evaluate the proposed approach in terms of the image quality of novel rendered views. Such quality can be measured by three common metrics: PSNR for pixelwise differences with ground truth images, SSIM for the perceived change in structural information, and LPIPS for the similarity between the activations in a pre-trained network. Moreover, we report the geometric mean of LPIPS, $\sqrt{1 - \text{SSIM}}$ and $10^{-PSNR/10}$ to combine them in a single metric for easier comparison (reported as Avg. in Table 1 and Table 2). We compare our KeyNeRF in two different versions (i.e. with and without entropy-based rays sampling) against the original NeRF (Mildenhall et al., 2020) and two state-of-the-art few-shot methods (Jain et al., 2021; Kim et al., 2022). Table 1 and Table 2 show that both versions of KeyNeRF outperform existing approaches on synthetic and real-world data, respectively, while being much simpler to implement and more flexible to integrate with any NeRF backbone. Moreover, note that the concurrent few-shot approaches fall behind the vanilla NeRF on real-world data (see Table 2), thus highlighting the complexity of loss weighting in such methods beyond controlled scenarios. On the other hand, KeyNeRF consistently outperforms them.

4.3 Ablation Studies

In this section, we analyze the impact of the number of poses K and the number of training iterations N_{iter} on the image quality metrics, as well as the separate role of selecting views and selecting informa-



(a) Ablation on the choice of K.

(b) Ablation on the choice of N_{iter}.

Figure 3: Quantitative comparison between our KeyNeRF and the original NeRF (Mildenhall et al., 2020) as a function of the number of poses (left) and iterations (right). Lower is better.

tive rays. We perform such ablations on the Blender dataset (Mildenhall et al., 2020).

Figure 3a shows that the proposed view selection method has more influence for low values of K and progressively decreases, as expected. In order to clearly visualize this difference, we provide a qualitative comparison for $K = 8$ in Figure 4 and as a function of the number of poses K in Figure 5. It can be seen that our approach converges faster and with better stability. Since the coverage constraint is satisfied optimally by the view selection algorithm, KeyNeRF allows to reconstruct an approximate scene even when $K = 8$. Crucially, our improvement is still significant up to $K = 48$, whereas concurrent few-shot methods only target the lowest end of this spectrum ($K \leq 8$).

Since we mainly focus on training efficiency, Figure 3b visualizes the convergence speed in steps of 5000 iterations each. Both versions of KeyNeRF show significant improvements across the whole training runs. Moreover, note how entropy-based sampling of rays is more effective in early iterations and then saturates after around 30000 steps. This confirms that selecting the most informative rays is important, especially with a limited training budget. The quantitative results in Table 1 and Table 2 underestimate the effect of this component. The lower quantitative impact is due to the fact that entropy-based sampling is most effective in fine-grained details and intricate structures, which are not well captured by numerical metrics. This is shown in Figure 6: sampling pixels uniformly discards crucial information, which leads to oversampling textureless areas and undersampling image regions with a lot of details.

4.4 Qualitative Results

The performance improvement of the proposed KeyNeRF in terms of rendering quality is visualized in Figure 7 for the Blender dataset and in Figure 8 for the CO3D dataset. The qualitative comparison against state-of-the-art methods shows that our informative rays selection strategy allows to render novel views with better details, especially in intricate structures such as the bulldozer wheels or the ship mast in Figure 7, and less hallucinated geometries (e.g. the teddybear and the hydrant in Figure 8). Moreover, our outputs are less blurry and preserve better the original colors of the scene. Finally, these results confirm that both DietNeRF (Jain et al., 2021) and InfoNeRF (Kim et al., 2022) tend to saturate their improvements over the original NeRF (Mildenhall et al., 2020) in the considered setup, while our approach presents significant advantages.

5 CONCLUSION

In this work, we present KeyNeRF, a method to select informative samples for training a few-shot NeRF. Given a dense and redundant set of views, we first select a subset of cameras to ensure both scene coverage and baseline diversity. Then, individual rays for each view are sampled from a probability distribution based on the local entropy of the image. This two-stage process allows the optimization to focus on relevant information in early iterations, thus speeding up convergence with a limited training budget. In this way, KeyNeRF outperforms other few-shot methods on both synthetic and real-world data, without requiring additional inputs or auxiliary pre-trained networks. Moreover, our approach can be easily integrated into any NeRF codebase, as it only requires to change the lines of code for sampling the batch of rays to optimize at each iteration. The current formulation is limited by the object-centric acquisition trajectory and the assumption of the object having higher entropy than the background. As future work, we plan

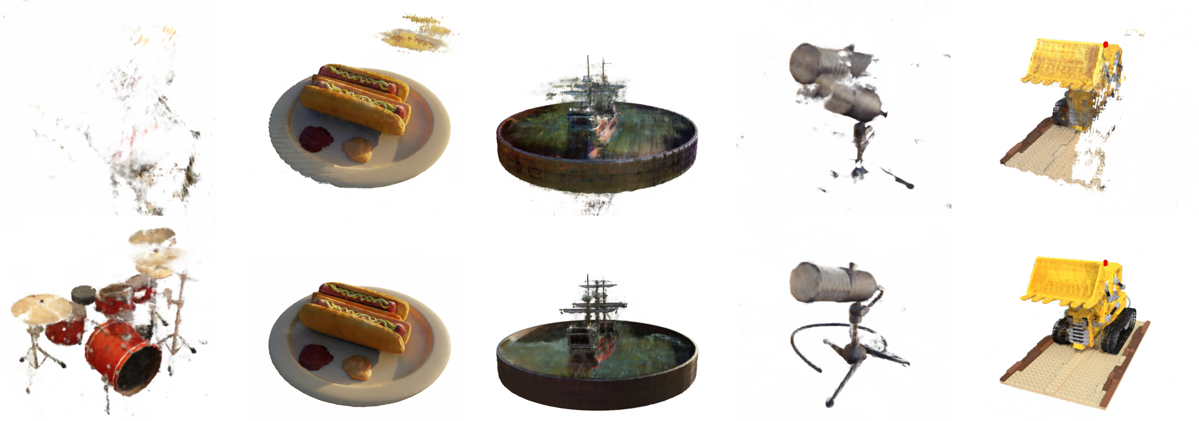
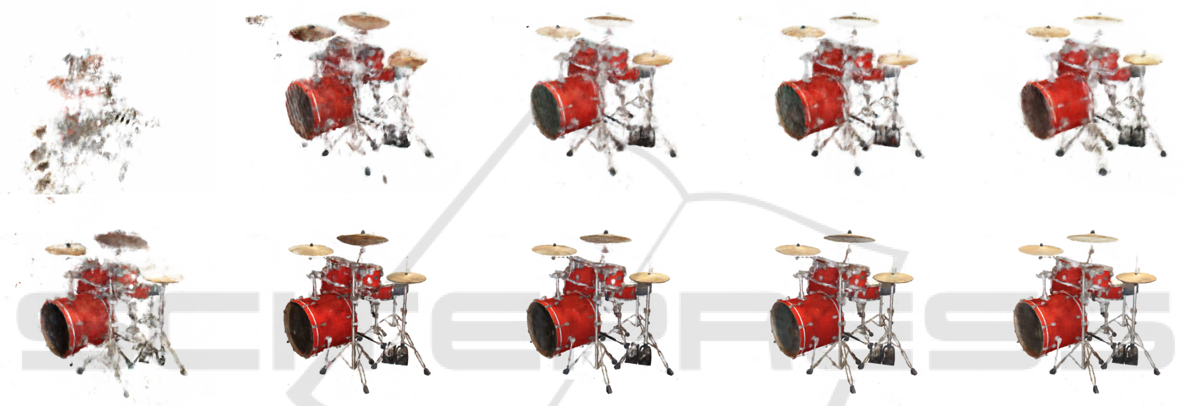


Figure 4: Qualitative comparison between choosing poses at random (top row) and using the proposed algorithm (bottom row) in a very few-shot setting ($K = 8$).



(a) $K = 8$. (b) $K = 16$. (c) $K = 24$. (d) $K = 32$. (e) $K = 48$.

Figure 5: Qualitative comparison between choosing poses at random (top row) and using the proposed algorithm (bottom row), as a function of the number of poses K. Zoom in for a better view.

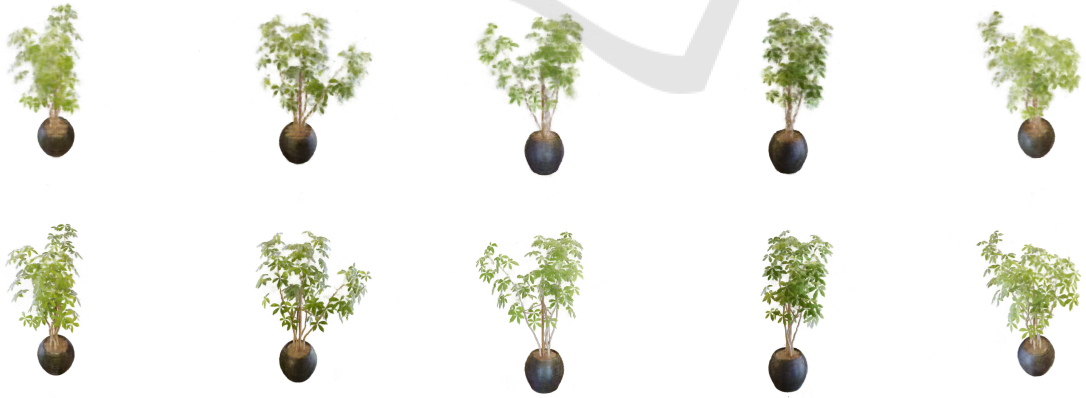


Figure 6: Qualitative comparison between sampling rays at random (top row) and using entropy-based sampling (bottom row) for different frames of the same scene. Zoom in for a better view.

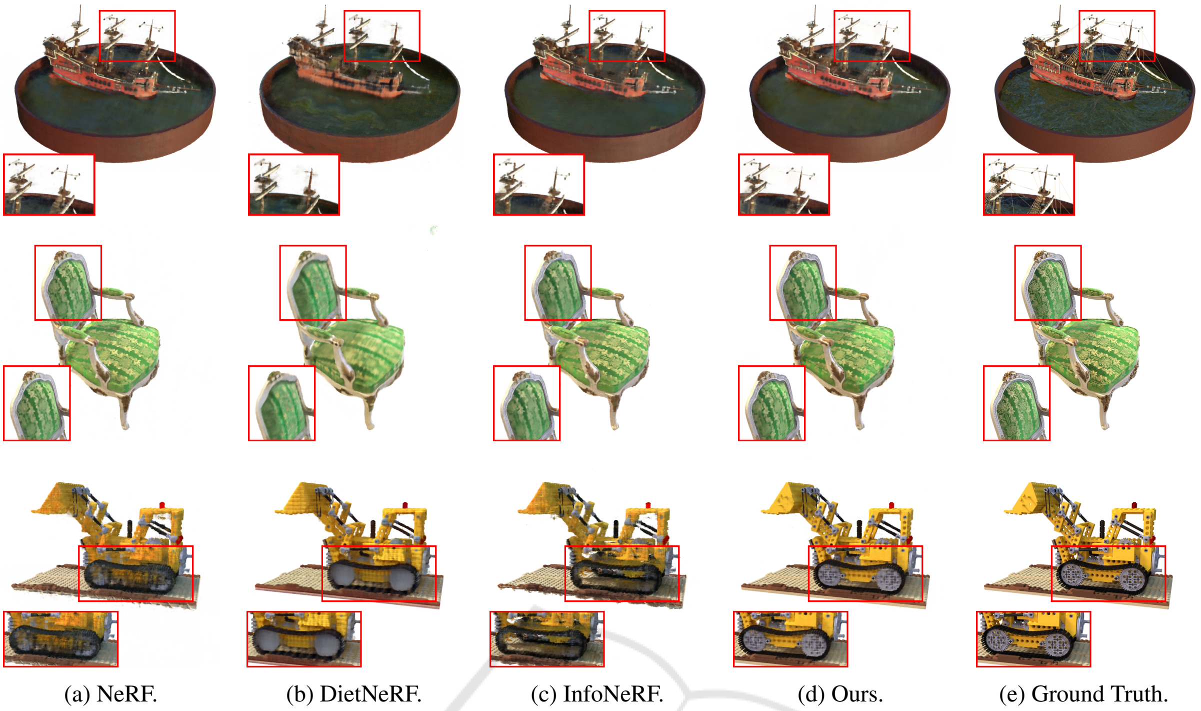

(a) NeRF. (b) DietNeRF. (c) InfoNeRF. (d) Ours. (e) Ground Truth.

Figure 7: Qualitative results on the Blender dataset. Zoom in for a better view.

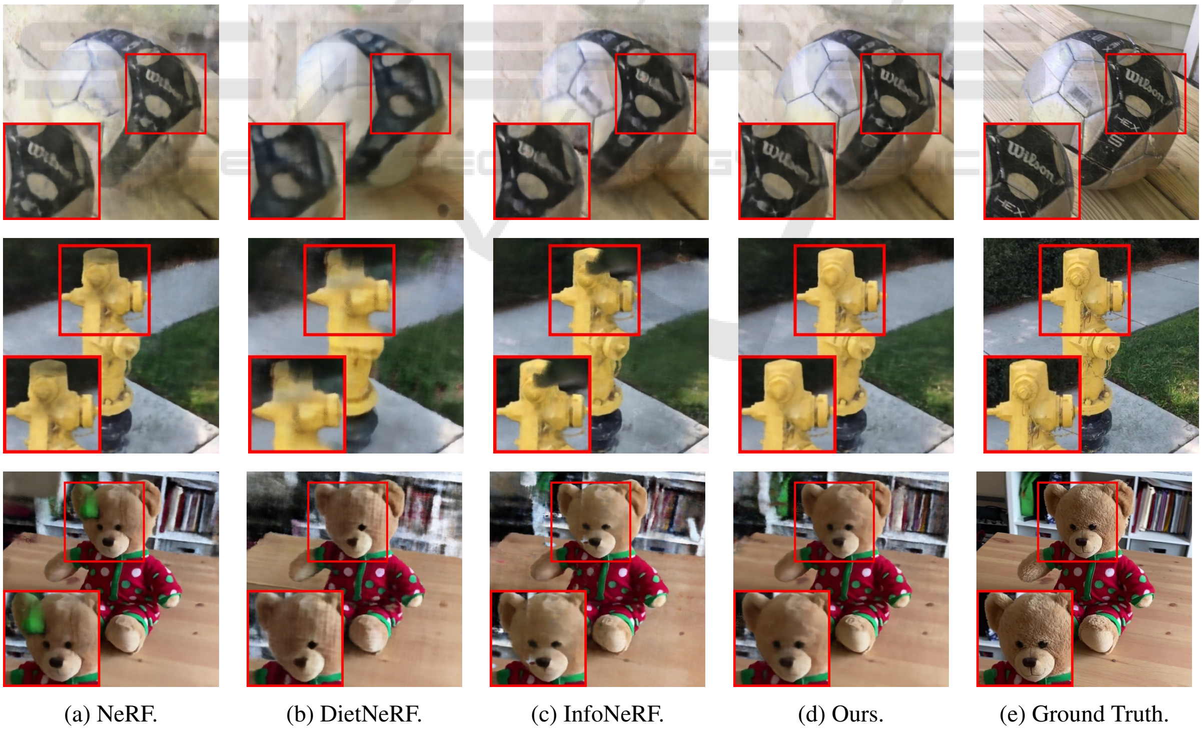

(a) NeRF. (b) DietNeRF. (c) InfoNeRF. (d) Ours. (e) Ground Truth.

Figure 8: Qualitative results on the CO3D dataset. Zoom in for a better view.

to tackle this issues and to integrate our selection procedure with other neural reconstruction methods.

REFERENCES

Cao, M., Zheng, L., Jia, W., Lu, H., and Liu, X. (2021). Accurate 3-d reconstruction under iot environments and its applications to augmented reality. *IEEE Transactions on Industrial Informatics*, 17(3):2090–2100.

Chen, A., Xu, Z., Geiger, A., Yu, J., and Su, H. (2022). Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*.

Deng, K., Liu, A., Zhu, J.-Y., and Ramanan, D. (2022). Depth-supervised NeRF: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. (2010). Towards internet-scale multi-view stereo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 1434–1441. IEEE.

Jain, A., Tancik, M., and Abbeel, P. (2021). Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5885–5894.

Kim, M., Seo, S., and Han, B. (2022). Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *CVPR*.

Ladikos, A., Ilic, S., and Navab, N. (2009). Spectral camera clustering. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 2080–2086. IEEE.

Liu, L., Gu, J., Lin, K. Z., Chua, T.-S., and Theobalt, C. (2020). Neural sparse voxel fields. *NeurIPS*.

Mauro, M., Riemenschneider, H., Signoroni, A., Leonardi, R., and Van Gool, L. (2014). An integer linear programming model for view selection on overlapping camera clusters. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 464–471. IEEE.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer.

Müller, T., Evans, A., Schied, C., and Keller, A. (2022). Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15.

Niemeyer, M., Barron, J. T., Mildenhall, B., Sajjadi, M. S. M., Geiger, A., and Radwan, N. (2022). Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Orsingher, M., Zani, P., Medici, P., and Bertozzi, M. (2022a). Efficient view clustering and selection for city-scale 3d reconstruction. In *Image Analysis and Processing–ICIAP 2022: 21st International Conference, Lecce, Italy, May 23–27, 2022, Proceedings, Part II*, pages 114–124. Springer.

Orsingher, M., Zani, P., Medici, P., and Bertozzi, M. (2022b). Revisiting patchmatch multi-view stereo for urban 3d reconstruction. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 190–196. IEEE.

Pan, X., Lai, Z., Song, S., and Huang, G. (2022). Activenerf: Learning where to see with uncertainty estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 230–246. Springer.

Perron, L. and Furnon, V. (2022). Or-tools.

Ramasinghe, S., MacDonald, L. E., and Lucey, S. (2022). On the frequency-bias of coordinate-mlps. In *Advances in Neural Information Processing Systems*.

Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., and Novotny, D. (2021). Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*.

Roessle, B., Barron, J. T., Mildenhall, B., Srinivasan, P. P., and Nießner, M. (2022). Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Schönberger, J. L. and Frahm, J.-M. (2016). Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Seo, S., Han, D., Chang, Y., and Kwak, N. (2023). Mixnerf: Modeling a ray with mixture density for novel view synthesis from sparse inputs. In *ArXiV*.

Sucar, E., Liu, S., Ortiz, J., and Davison, A. (2021). iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Van der Merwe, M., Lu, Q., Sundaralingam, B., Matak, M., and Hermans, T. (2020). Learning continuous 3d reconstructions for geometrically aware grasping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11516–11522.

Wang, Y., James, S., Stathopoulou, E. K., Beltrán-González, C., Konishi, Y., and Del Bue, A. (2019). Autonomous 3-d reconstruction, mapping, and exploration of indoor environments with a robotic arm. *IEEE Robotics and Automation Letters*, 4(4):3340–3347.

Wynn, J. and Turmukhambetov, D. (2023). Diffusionerf: Regularizing neural radiance fields with denoising diffusion models. In *ArXiV*.

Yang, J., Pavone, M., and Wang, Y. (2023). Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Yeh, Y.-J. and Lin, H.-Y. (2018). 3d reconstruction and visual slam of indoor scenes for augmented reality application. In *2018 IEEE 14th International Conference on Control and Automation (ICCA)*, pages 94–99.

Yen-Chen, L. (2020). Nerf-pytorch. https://github.com/yenchenlin/nerf-pytorch/.