Improving NeRF Quality by Progressive Camera Placement for Unrestricted Navigation in Complex Environments

Georgios Kopanas, George Drettakis

Vision, Modeling, and Visualization (2023)
T. Grosch and M. Guthe (Eds.)

Improving NeRF Quality by Progressive Camera Placement for Free-Viewpoint Navigation

Georgios Kopanas[1] and George Drettakis[1]

[1] Inria & Université Côte d'Azur, France

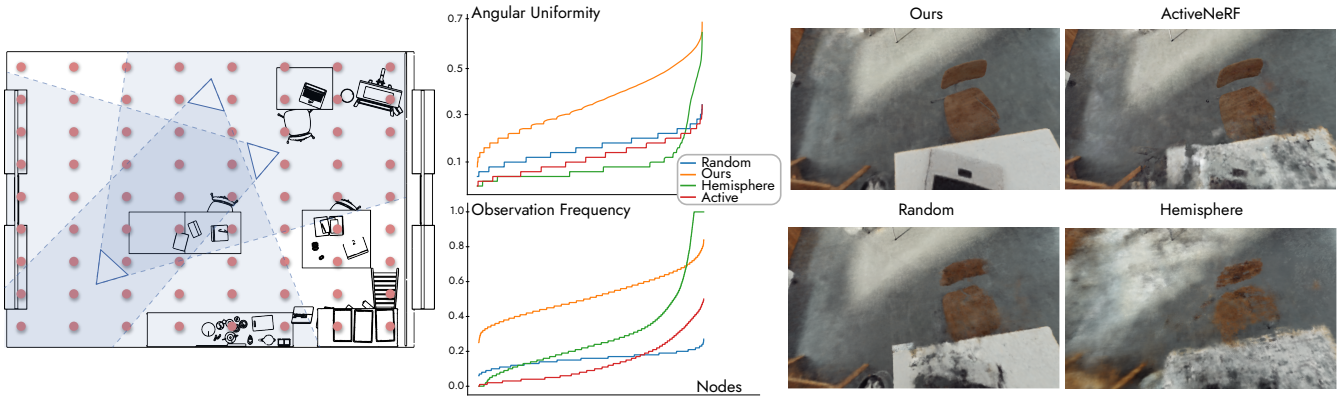

Figure 1: *We present a new method that proposes the next best camera placement for NeRF capture (left). We introduce two metrics that can be easily computed, observation frequency and angular uniformity (middle). On the right, we show that our approach outperforms two baseline camera placement strategies,* HEMISPHERE *which is the typical approach used in most NeRF methods and* RANDOM*, as well as recent related work [PLSH22].*

Abstract

Neural Radiance Fields, or NeRFs, have drastically improved novel view synthesis and 3D reconstruction for rendering. NeRFs achieve impressive results on object-centric reconstructions, but the quality of novel view synthesis with free-viewpoint navigation in complex environments (rooms, houses, etc) is often problematic. While algorithmic improvements play an important role in the resulting quality of novel view synthesis, in this work, we show that because optimizing a NeRF is inherently a data-driven process, good quality data play a fundamental role in the final quality of the reconstruction. As a consequence, it is critical to choose the data samples – in this case the cameras – in a way that will eventually allow the optimization to converge to a solution that allows free-viewpoint navigation with good quality. Our main contribution is an algorithm that efficiently proposes new camera placements that improve visual quality with minimal assumptions. Our solution can be used with any NeRF model and outperforms baselines and similar work.

CCS Concepts

• ***Computing methodologies*** → ***Computer graphics; Rendering;*** *Active learning settings;*

1. Introduction

In recent years, Neural Radiance Fields (NeRFs) [TTM*22] have emerged as a powerful approach allowing high-quality novel view synthesis, for scenes captured with photos taken from many different viewpoints. These methods also provide an alternative to Multi-View-Stereo [SZFP16] solutions that are slow and fail to produce faithful visual and geometric reconstruction. For both MVS and NeRF, capturing a scene typically starts with users taking many photos or video of the scene. Usually, users follow instructions to loop around an object a few times at different heights and to make sure to also capture top views [Lum23] which we call *hemispheri-*

cal capture. This works well for "object-centric scenes", i.e., scenes that have a main object that the users want to be able to view freely, while the rest is considered background. There has been little previous work on how to capture more general scenes such as rooms, buildings etc. that have no central point of interest, especially when the goal is to allow *free-viewpoint* navigation in the environment. Users typically place the cameras based on their intuition and empirical knowledge about which camera placements usually work, often leading to the failure of the reconstruction and consequently, forcing users to recapture the scenes in a costly and time consuming trial-an-error process.

Intuitively, hemispherical capture works for object-centric scenes because it samples the space containing the object *uniformly* both for camera positions and in the angular domain. This uniform coverage is a dense sampling of a complete radiance field, since rays from the camera centers through the pixels in each view frustum cover space uniformly. Such coverage provides multi-view information that is used in the optimization to disambiguate depth, allowing accurate reconstruction.

Achieving such uniform ray coverage both in positions and angles is much more challenging in the context of general scenes, where there is no single central object. With an infinite number of cameras, it could theoretically be possible to densely sample the light field, but in practice the number of cameras is limited and in addition given the geometry of the scene cameras cannot be placed everywhere (i.e., inside objects). The problem we try to solve is: given a camera budget and physical limitations of space, how can we *efficiently* choose the next camera that will allow the resulting ray sampling to be as close as possible to uniform in space and angle.

There has been little previous work on this problem; most methods that have been proposed require either modification on the training of the NeRF model making them unsuitable for generalization to other NeRF variants other than the ones that it was specifically designed for, or very expensive calculations based on the current state of the NeRF model. These properties make the process slow and cumbersome.

In our solution, we first develop a metric to evaluate uniformity in space and angle that is fast to evaluate. We then propose an algorithm that uses this metric to select the next best camera such that the overall distribution will be closer to uniform in positions and angle. We evaluate the metric and the algorithm on synthetic data and compare to baselines and previous work, demonstrating that our solution works well, and we also run our algorithm on a real dataset as a proof of concept. From a practical perspective, our algorithm can be used for automated capture using robotic or drone capture; we leave this as future work, but we discuss a practical future use case in Sec. 3.4.

In summary, our contributions are:

- The definition of an efficient metric for *observation frequency* and *angular uniformity* that can be computed on the fly during NeRF capture, without requiring additional images.
- An algorithm to quickly estimate reconstruction quality of a scene and that proposes the next camera placement that maximizes the improvement in quality of capture based on our metrics.

Our solution can work with any NeRF model without changes to the optimization loop, and only introduces a small performance overhead to the training. We performed extensive testing on synthetic scenes, and our method achieves the best quality against multiple baselines and other algorithms that we tried given a limited budget of cameras. We also present a first preliminary evaluation with real data, in which our method also performs well.

2. Related Work

In recent years a huge number of publications on Neural Radiance Fields (NeRFs) have been published; we will start by reviewing only a few papers that are closely related to our method and design choices. Recent comprehensive surveys on NeRFs can be found in [TTM*22] and [XTS*22]. We then review camera selection for reconstruction, both traditional and for NeRFs.

2.1. Neural Radiance Field Basics

Neural Radiance Fields were introduced by [MST*21]. They fundamentally changed how we can reconstruct 3D scenes from 2D images by introducing a continuous volumetric representation of the scene, encoded in a Multi-Layer Perceptron (MLP) which we can optimize using Stochastic Gradient Descent (SGD) to fit the input images, solving the reconstruction with a data-driven optimization.

Follow-up work improved the reconstruction quality by allowing for better extrapolation when dealing with cameras observing objects from different distances [BMT*21] but also generalized the algorithms for unbounded and realistic scenes [BMV*22]. Most of the results demonstrated in these papers focus on object-centric camera placements which has become the "typical NeRF-style capture", i.e., a hemisphere of cameras around the object and looking directly at the center of the object.

Another line of work focuses on performance, both for fast training and fast rendering. Most solutions achieve good results by encoding the radiance field in voxel grids with limited spatial extent [MESK22, FKYT*22, SSC22, CXG*22] or point clouds [XXP*22]. For our experiments we will built on top of Instant-NGP [MESK22], since it is currently the NeRF method with the best quality/performance trade-off. Instant-NGP uses a hash function to map a 3D hierarchical voxel-grid of high dimensional features to a compact 1D representation. This grid is later queried in an optimized way along the ray to produce the final color for each pixel. The optimization includes a very efficient *occupancy grid* that marks the voxels as occupied or free and results in an efficient way to skip empty space during ray marching.

Also, there are numerous papers that try to reconstruct NeRFs from a very limited amount of input views [YYTK21, JTA21, KDSB22]. These models could potentially be benefited greatly by an optimal selection of cameras.

While all these algorithms significantly improve the state-of-the-art, in the vast majority of cases they use datasets in which the

cameras are placed on a hemisphere over a region of interest. This allows for good quality reconstruction only in that specific region (typically an object of interest). It is not clear how one would place cameras for more complicated environments, when allowing the user to navigate freely. Camera placement is an important factor that controls the final quality of the reconstruction and the ability to *navigate freely* in the scene without artifacts. In this context we propose a solution that will automate and standardize the way of capturing NeRFs, removing the burden of trial and error from the user.

2.2. Camera placement for reconstruction

We discuss representative previous work in camera placement for reconstruction. In the vast majority of cases we assume that the scene is captured with photographs, and that the cameras of these photographs have been calibrated. Camera calibration is typically performed using Structure-from-Motion (SfM), using systems such as COLMAP [SF16].

2.2.1. Traditional Reconstruction

Multi-View Stereo (MVS) is an offline process that recovers the 3D geometry from a set of images and is very computationally expensive. The user first captures the images, performs camera calibration and then runs MVS. After a few hours of computation, the user may come to realize that the images are not good enough for a good 3D reconstruction, requiring the scene to be recaptured from scratch. This is a tedious process, especially if accessing the capture site is difficult.

To improve this cumbersome process the field of Next-Best-View (NBV) estimation [ISDS16,DF09,BKA*16] predicts the next view that will provide more information to the reconstruction process given a set of already captured views. In the field of volumetric reconstruction [ISDS16] focuses on sensors with depth and creates a set of heuristics to estimate the next best view that will maximize the information gain of each newly acquired sample. While this work is inspiring it lies outside the scope of optimizing a model from a set of data-samples with SGD because they use a depth sensor that directly observes the geometry information of the scene, while in neural radiance fields the geometry representation is being optimized to fit the scene.

Other works in camera selection that focus on MVS can be separated in heurstic-based methods [MRFB16, SMGH18] or data-based [LFR22]. They mostly focus on estimating or predicting the uncertainty of the MVS reconstruction process without actually running it. In contrast to MVS, fast NeRF models [MESK22, FKYT*22] open the door for new approach in the field of next best view estimation that allows online reconstruction and camera placement prediction, especially if camera calibration can be provided online by the capture device (e.g., augmented reality helmet).

2.2.2. Neural Radiance Fields

Automatic camera placement for Radiance Fields is an emerging topic of research. A popular approach is to modify the NeRF model to be able to predict it's own uncertainty [RZH*23,ZLR*22, PLSH22] which later is used in various ways to choose the views which maximize it. The uncertainty is modeled in two ways, either by converting the MLP that encodes the scene to a Bayesian MLP [RZH*23,ZLR*22,PLSH22] that also predicts it's own uncertainty or by using the physical properties of the volumetric representation along a ray based on the entropy of the density function [ZZX*22]. All methods that use the NeRF model to predict uncertainty are computationally intensive since they need many MLP evaluations for each candidate camera. In addition, it is hard to train an MLP that predicts it's own uncertainty. That is why [RZH*23] uses a depth sensor to stabilize the training. Some methods [PLSH22] focus on selecting views when there is a very limited budget of cameras allowed and [LCW*22] presents a solution that evaluates the uncertainty based on the spread of density along a ray. This needs a full rendering step per candidate camera which means when the space of candidates grows in unconstrained environments it comes with an increased cost. All the above methods are not demonstrated on non object-centric scenes, making them unsuitable for our context which focuses on free-viewpoint navigation in complex scenes.

3. Method

The goal of our method is to dynamically suggest new camera positions such that we create a dataset that will achieve a good quality reconstruction. This can be used to guide a robotic agent or a human to acquire new images when capturing a NeRF. NeRFs trained on object-centric datasets achieve excellent quality when observing the object from a camera that matches the distribution of the training cameras, but easily break when moving away from them, see Fig. 3. We are interested in constructing a carefully designed placement of cameras that will allow the final user of the NeRF to navigate freely in the scene, while avoiding strong visual artifacts.

We want to generalize the simple assumptions of the object-centric capture style to more complex scenes and viewing scenarios, in particular when we allow the viewer to navigate freely.

3.1. Observation Frequency and Angular Uniformity

The object-centric capture style of NeRF [MST*21] and MipNeRF [BMT*21] has two main properties. First, all cameras observe the object and second, the cameras are distributed along different directions to cover the angular domain uniformly. If we constrain the user to view the scene on the hemisphere, this capture style naturally provides a good reconstruction since it covers the space all the possible cameras uniformly.

We next provide a formal definition of this observation, and in particular a measure of *observation frequency* and a measure of *angular uniformity* of these observations.

Given a set of cameras $\mathcal{C}$, a point p in space will be well reconstructed if it is observed often from the input cameras and if these cameras are distributed uniformly in the angular domain of directions. We next formalize this mathematically and generalize it for multiple points p.

We define a function that describes how frequently each point is observed. For a point p we define the frequency $O_f(p)$ of observation as follows:

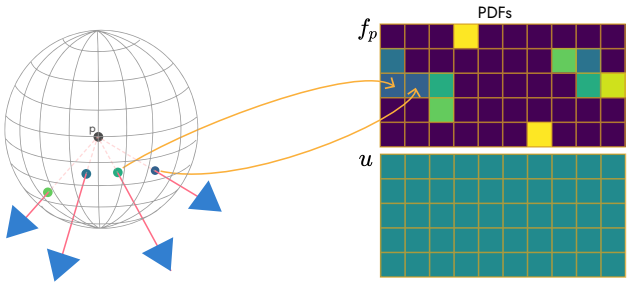

Figure 2: *To compute the distance of the empirical distribution in the angular domain of cameras that observe a point p, we assign the cameras to bins based on their direction from point p into a histogram in polar coordinates. We then convert this to a PDF, for which we account for the non-uniform surface area of the spherical coordinate system. Then we use the total variation distance in Eq.(2) to get the value for node point p.*

$$O_f(p) = \frac{\sum_{i=0}^{N} \mathbb{1}_{obs}(C_i, p)}{N} \tag{1}$$

Where $\mathbb{1}_{obs}$ is an indicator function that is 1 if point p lies inside the frustum of camera C_i and N is the total number of cameras. This equation describes a simple relationship between cameras and points in space: If all the available cameras observe a point, then $O_f(p) = 1$, while if no cameras observe it $O_f(p) = 0$.

For the directions of the observations, we next define a metric to measure *angular uniformity*, since more uniform angular distribution of observed directions results in better resulting visual quality.

We define f_p the distribution of camera directions in the angular domain that observe a point p and the uniform distribution u in the same angular domain. To determine the quality of the angular distribution of cameras, we will compute the total variation distance between the two distributions.

$$TV(f_p, u) = \frac{1}{2} \sum_{\theta,\phi \in \Omega} |f_p(\theta,\phi) - u(\theta,\phi)| \tag{2}$$

where u is a uniform PDF. We construct the piece-wise constant PDF f_p^C in the angular domain by computing the histogram of the directions of the cameras that observe point p. Every bin in the histogram contains the number of cameras that observe this point from the solid angle that corresponds to the bin. Similarly to Eq. (1), the angular metric is 0 if our point p is observed from a uniform distribution, while it approaches 1 as it moves further from the uniform distribution.

We provide a visual illustration of this process in Fig.2, showing the histogram of F_p and U in bins with polar coordinates. The direction are represented in polar coordinates, that not area preserving, so we weight the bins of the histogram based on the surface area of each bin.

3.2. Estimating Reconstruction Quality

We define the area where we want to estimate the reconstruction quality, and for simplicity we use an axis-aligned bounding box to define it. This is the area which the user wishes to observe; We refer to this area as $\mathcal{B}$.

Ideally, we would like to evaluate the quality of reconstruction of every point in $\mathcal{B}$. In practice however, we discretize the problem by constructing a regular grid in $\mathcal{B}$ with resolution of 32^3 and refer to it as $\mathcal{B}_{32}$; we will evaluate reconstructability on the *nodes* of the grid. In discrete space it is easier to measure the total reconstructability E of $\mathcal{B}$ given by the set of cameras $\mathcal{C}$ by summing over all the nodes p of the grid:

$$E(\mathcal{C}, \mathcal{B}_{32}) = \sum_{p \in \mathcal{B}_{32}} (1 - TV(f_p^{\mathcal{C}}, u)) + O_f(p)^{\gamma}, \tag{3}$$

where $f_p^{\mathcal{C}}$ is the empirical PDF in the angular domain for point p and set of cameras $\mathcal{C}$ and u is the uniform PDF in the same domain, while γ is a non linear scaling factor that modulates how much more important it is to observe points that are less frequently observed than points that have already been observed frequently.

Our formulation has several advantages. First, it is easily interpretable by a user, making it easy to modify and specify regions that have more importance than others. In the limit case of a single point and if C is constrained on a hemisphere our method reduces to the typical "object-centric" capture setup of NeRF and MipNeRF/360. Second, a key advantage is that our camera proposal does not require additional image acquisition to estimate reconstruction quality, as for other methods, e.g., based on uncertainty estimation [PLSH22]. Our combined observation frequency and angular uniformity metrics can be seen as a proxy for uncertainty, while being relatively cheap to compute.

3.3. Optimization

Our goal is to find the set of cameras $\mathcal{C}$ that maximize the quantity in Eq. (3):

$$\underset{\mathcal{C}}{\operatorname{argmax}}\, E(\mathcal{C}, \mathcal{B}_{32}) \tag{4}$$

To do this we optimize the NeRF model while we choose the new cameras. Cameras can only be placed in empty space and some NeRF models, including Instant-NGP [MESK22] that we use, provide an occupancy grid that is used to skip empty space while rendering. If the implementation does not provide one it is trivial to compute it on the fly by sampling 3-D space and storing the density and using a threshold to binarize the value. We use the occupancy grid to place candidate cameras in free space. For simplicity, we also constrain the cameras to lie inside $\mathcal{B}$.

In the beginning of our run we have no images nor a trained occupancy grid, and one needs the other to initialize the process. To overcome this, we ask the user to create a bounding box in the scene that is empty and call it $\mathcal{B}^f$. This allows us to sample cameras safely so we can start the process. This is somewhat of a chicken-and-egg problem, since we need enough of an initial reconstruction

Algorithm 1 Summary of the greedy optimization algorithm to minimize Eq. (3)

$T \leftarrow 100$ ▷ Total number of cameras we want to sample.
$N \leftarrow 1000$ ▷ Number of cameras we sample in each iteration.
$nodes \leftarrow \mathcal{B}_{32}$
$\mathcal{C} \leftarrow [\,]$
while $|\mathcal{C}| <= T$ **do**
  $aabb \leftarrow \mathcal{B}$ **if** $|\mathcal{C}| > 20$ **else** $\mathcal{B}^f$
  $C_p \leftarrow sampleRandomCameras(N, aabb)$
  for all $c \in C_p$ **do**
    $E_c \leftarrow E(\mathcal{C} \cup c, nodes)$ Eq. (3)
  end for
  $c_{max} \leftarrow \text{argmax}_c E_c$
  C.insert$(AcquireImage(c_{max}))$
  if $|\mathcal{C}| > 20$ **then**
    $train\_nerf(iterations = 250)$
  end if
end while

to have a coordinate system to define this initial box $\mathcal{B}^f$. In a real-world scenario, the user will simply take 20 photos of the scene – evidently in free space – that will initialize the reconstruction and the coordinate system, and imply the definition of $\mathcal{B}^f$. For the synthetic examples we used for evaluation, we have the coordinate system beforehand, and we defined $\mathcal{B}^f$ manually.

Now that we know how to sample random cameras safely, we want to maximize the quantity in Eq. (3). We use a greedy maximization technique: every 250 training iterations we acquire a new camera given a set of already chosen $\mathcal{C}_k$. We do this by sampling a set of N=1000 candidate cameras that lie inside $\mathcal{B}$, or inside $\mathcal{B}^f$ for the first 20 cameras. Each camera gets a random direction, and we filter out all cameras whose center lies in occupied space or observes occupied space from too close. Then we compute $E(\mathcal{C}_k \cup c_n, \mathcal{B}_{32})$ for each of the N cameras, choose the camera with the highest score and add it in $\mathcal{C}_{k+1} = \mathcal{C}_k \cup c_n$. We repeat until we acquire as many cameras as our budget allows. The algorithm is summarized in Alg. 1. Our current unoptimized implementation requires a few seconds to propose a new camera; further optimization would allow truly interactive use.

3.4. Future Practical Usage Scenario

The method above provides all the elements for online camera selection that can be used in future work by a robotic or drone-based system. In this paragraph we describe how such a system could function to motivate the utility of our results.

As discussed above, a user would first take 10-20 photos of the environment for an initial camera calibration, providing a reference frame, and allowing the definition of an initial box $\mathcal{B}^f$. In a fully operational system, we would then run a fast NeRF such as InstantNGP [MESK22] that reconstructs a first approximation of the scene volumetric representation in a few seconds. We then would run our algorithm to choose the next camera, and move the robot or drone to the next position; the new capture is then incrementally added to the NeRF optimization until the budget of cameras is reached. Given the latency of moving the capture agent and the fact that we need a few training iterations between the two captures, a slightly optimized implementation of our algorithm is perfectly suited to such a scenario, since it can provide the next best camera faster than the agent can actually move to the next position. As a consequence, a full system using our algorithm would allow automatic and high quality capture with a small number of photos, both reducing capture time and optimizing resulting image quality.

4. Results & Comparisons

Evidently, the goal of our method is to provide guidance for a human or robotic acquisition system while capturing a NeRF. Designing the user interface for human guidance or interfacing with an automated acquisition system are complex tasks that we leave for future work. Instead, we provide a thorough evaluation of synthetic scenes, in which image acquisition is achieved simply by rendering a new image from the camera proposed by our system. We also provide a preliminary evaluation of our method on a real capture, by capturing a large number of views that we can then sample.

This system was implemented by interfacing together Instant-NGP [MESK22] with Blender's python API [Com18] and Cycles renderer[†]. We extended the python interface of Instant-NGP to allow us to query the occupancy grid efficiently and we also linked Instant-NGP with Blender's python environment. We will provide all source code and data, that will be available here `https://github.com/anonymized`. The full pipeline that allows the treatment of complex synthetic scenes interfaced with NeRF systems such as InstantNGP is a powerful tool in itself, and was very useful for this project. We hope it will also be helpful to others experimenting with NeRFs in full, realistic scenes.

4.1. Evaluation on Synthetic Scenes

For the first set of synthetic scene comparisons, we evaluate our method against two baseline camera placement strategies: HEMISPHERE where we place the cameras on a hemisphere around an object of interest[‡] in the scene (this is the standard NeRF [MST*21] capture style) and RANDOM where we place the cameras at a random position and a random orientation by also making sure that the camera placement is not in occupied space (i.e., not inside objects), see also Sec. 3.3.

As discussed in Sec. 2, there are few methods that treat our specific problem; of all NeRF camera selection methods, only ActiveNerf [PLSH22] provides code, and we thus include a comparison to this approach. We use the authors' implementation which is based on an implementation of the original NeRF [MST*21] method. To allow a best-effort fair comparison we used their code to extract the cameras, and then trained the same Instant-NGP [MESK22] model as we did with all other baselines and our method. To extract the cameras we pre-render 1000 random cameras which act as the pool from which ActiveNeRF can choose

[†] https://www.cycles-renderer.org/

[‡] We manually select a point and a radius for the hemisphere such that it makes for that specific scene configuration, i.e a table with vases etc

Table 1: *Per Scene Quantitaive evaluation of our method. We provide the PSNR for each test set separately and the total average of each algorithm for each scene.*

Scene	LIVINGROOM1				LIVINGROOM2				OFFICE6				OFFICE9				KITCHEN5			
Train|Test	Random	Hsphere	Ours	Avg	Random	Hsphere	Ours	Avg	Random	Hsphere	Ours	Avg	Random	Hsphere	Ours	Avg	Random	Hsphere	Ours	Avg
Hsphere	17.43	**31.48**	14.63	21.18	19.13	**30.12**	13.04	20.76	20.34	**30.48**	15.22	22.01	15.17	**36.79**	13.23	21.73	18.92	**33.28**	16.36	22.85
ActiveNerf	20.57	17.91	18.59	19.02	25.51	22.76	21.28	23.18	24.71	20.78	19.65	21.71	25.75	27.52	24.40	25.89	25.00	22.83	21.42	23.08
Random	25.77	23.86	22.81	24.14	26.65	24.48	22.00	24.37	28.57	25.63	22.06	25.42	26.20	28.29	22.65	25.71	25.24	24.30	22.70	24.08
Ours	**27.46**	27.80	**26.86**	**27.37**	**28.40**	27.02	**26.02**	**27.14**	**31.35**	27.01	**27.12**	**28.49**	**28.38**	30.59	**28.91**	**29.29**	**27.74**	27.06	**25.87**	**26.89**

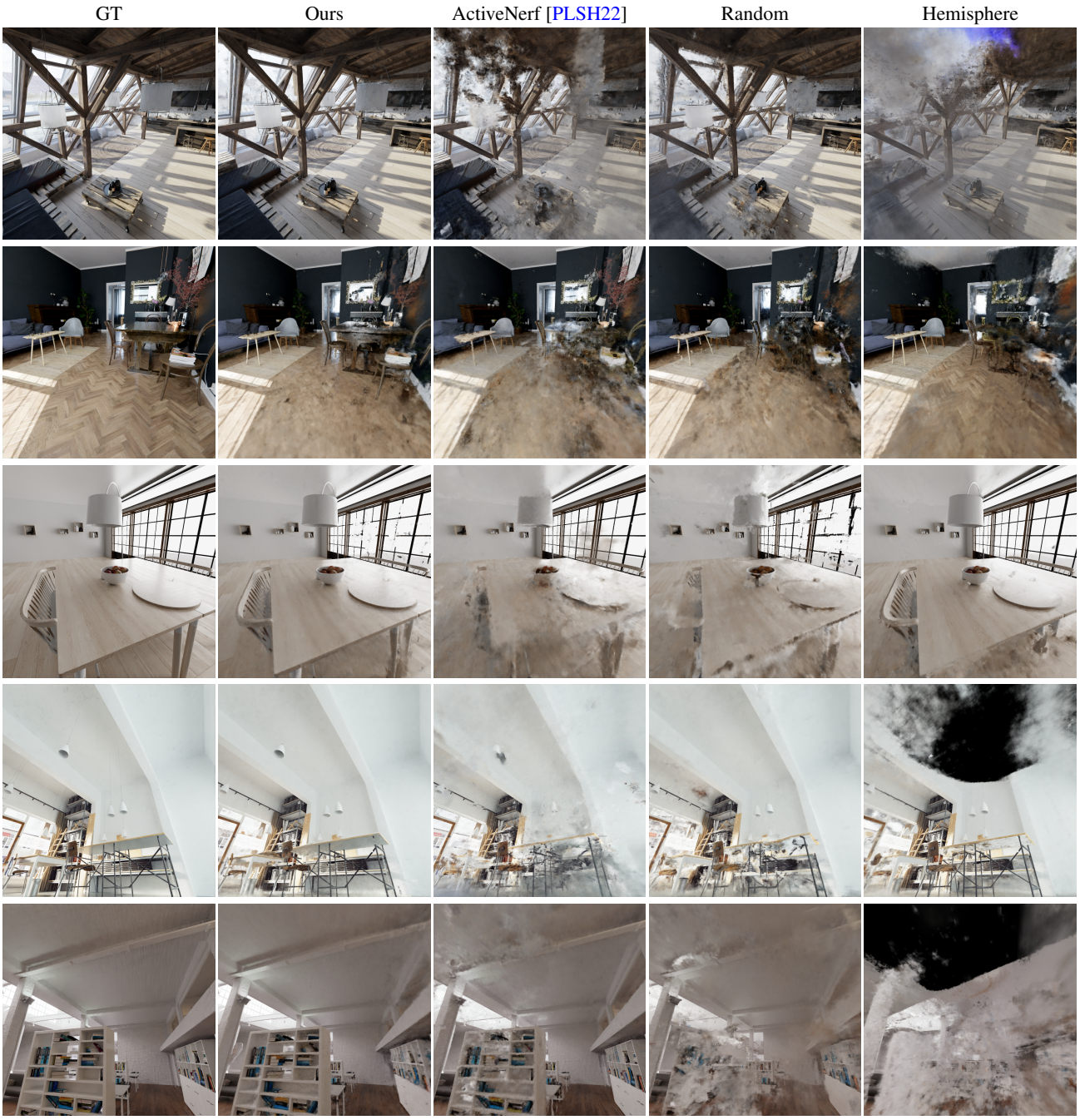

Figure 3: *Images from our sampling test set. We present a visual comparison to baselines(*RANDOM,HEMISPHERE*) and ActiveNerf. The first column shows the ground truth image. The scenes shown are* LIVINGROOM1, LIVINGROOM2, KITCHEN5, OFFICE6 *and* OFFICE9.

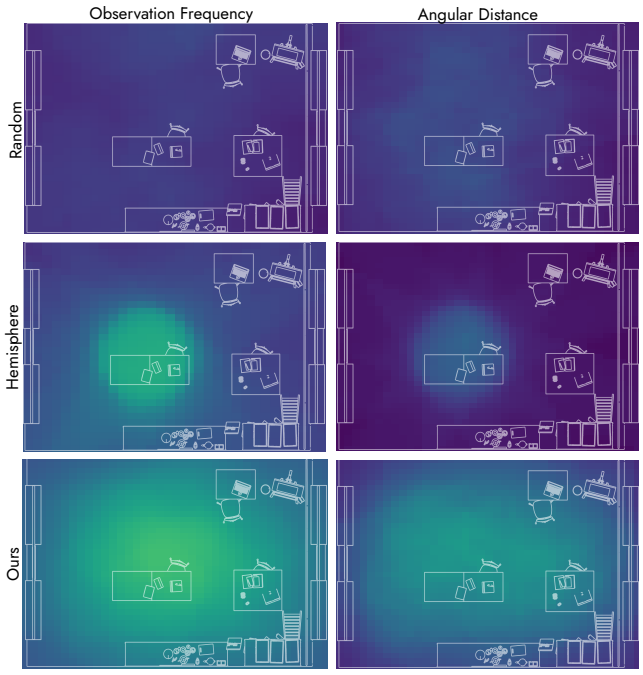

Figure 4: *"Floorplan" visualisation of the seperate elements that construct our energy term in Eq. (3). The values have been averaged along the Y axis.*

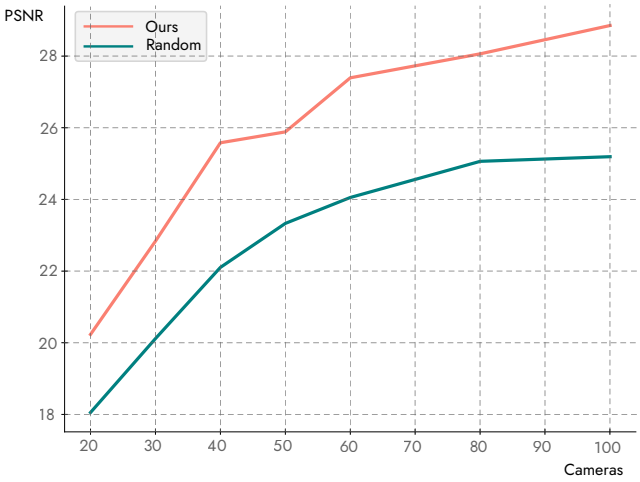

Figure 5: *In this plot we show the test-set PSNR with relation to the number of cameras in the training set. We plot two sampling algorithms, random and ours. We evaluate the PSNR across all 150 images of the test set for the scene Office6*

cameras. The ActiveNeRF implementation is very computationally and memory intensive so we tuned ActiveNerf to start from 20 random cameras, and choose 8 cameras every 50k iterations. These 8 cameras are the best cameras chosen from their algorithm from a random subset of 100 out of the 1000 cameras.

We used 5 synthetic scenes modeled by professional artists to represent realistic indoor environments[§]. For each scene we construct a training set corresponding to each one of the algorithms we want to evaluate and multiple test sets that provide a good overview of the total quality throughout the scene.

Our test-sets contain a total 150 views that are distinct from the training views. The test-sets are split in 3 sub-sets: 1) 50 random views using the HEMISPHERE capture style 2) 50 views using RANDOM and 3) 50 views using our sampling process. The purpose of the multiple test sets is to evaluate each algorithm fairly throughout different camera distributions such that the quantitative metric evaluate the total quality throughout the scene. This avoids bias towards one of the aforementioned distributions, and allows a more comprehensive overall evaluation of our algorithm. We provide all the renderings for all views and all algorithms in our supplemental material. We also rendered free-view point paths which we provide in the supplemental video.

As we observe in Fig. 3 and Fig. 6 the standard hemisphere captures of NeRF and MipNeRF fail to generalize in test sets coming from other distributions. Hemisphere views have a specific structure and objects that lie outside of the hemisphere are observed only from constrained angular directions and this allows for the NeRF model to overfit to the set of input cameras. While the random capture significantly outperforms the hemisphere capture in the generalized setting, we can still see significant artifacts because of the unstructured nature of the dataset. In theory, an infinite number of random views should allow for perfect reconstruction, but this is impractical since it is labor and computationally intensive.

As we can see from the quantitative and qualititave evaluation (Fig. 3,6), ActiveNerf [PLSH22] does not always successfully choose the cameras that would allow for a good reconstruction. This happens for many reasons. First, the original NeRF models used has a hard time to converge in complicated scenarios that are not similar to the synthetic blender dataset, and it becomes even harder with the Bayesian model of Active-NeRF. Active-NeRF needs to get a notion of the scene to allow for good camera placements and in complicated scenes it can be challenging just from 20 initial cameras. Second, ActiveNeRF chooses cameras that maximize the uncertainty for the specific model they are training for, this does not guarantee that this uncertainty metric will generalize to other NeRF models and finally the memory and speed requirements do not allow for a huge number of candidate cameras similar to our method.

We can see that our capture style outperforms all other algorithms across the scenes and views both quantitatively in Tab. 1 and qualitatively at Fig. 3 and Fig. 6. This supports our hypothesis that if we observe all parts of the scene while maintaining a uniform set of directions we will get an ideal reconstruction.

We also perform a visual analysis to provide insight on how different methods score against the energy function in Eq. (3). In Fig. 4 we provide a visualization of the scores of each of the two terms of Eq. (3). We see that our method clearly observes all the

§ The scenes are available for purchase at www.evermotion.com and are compatible with the Blender Cycles renderer.

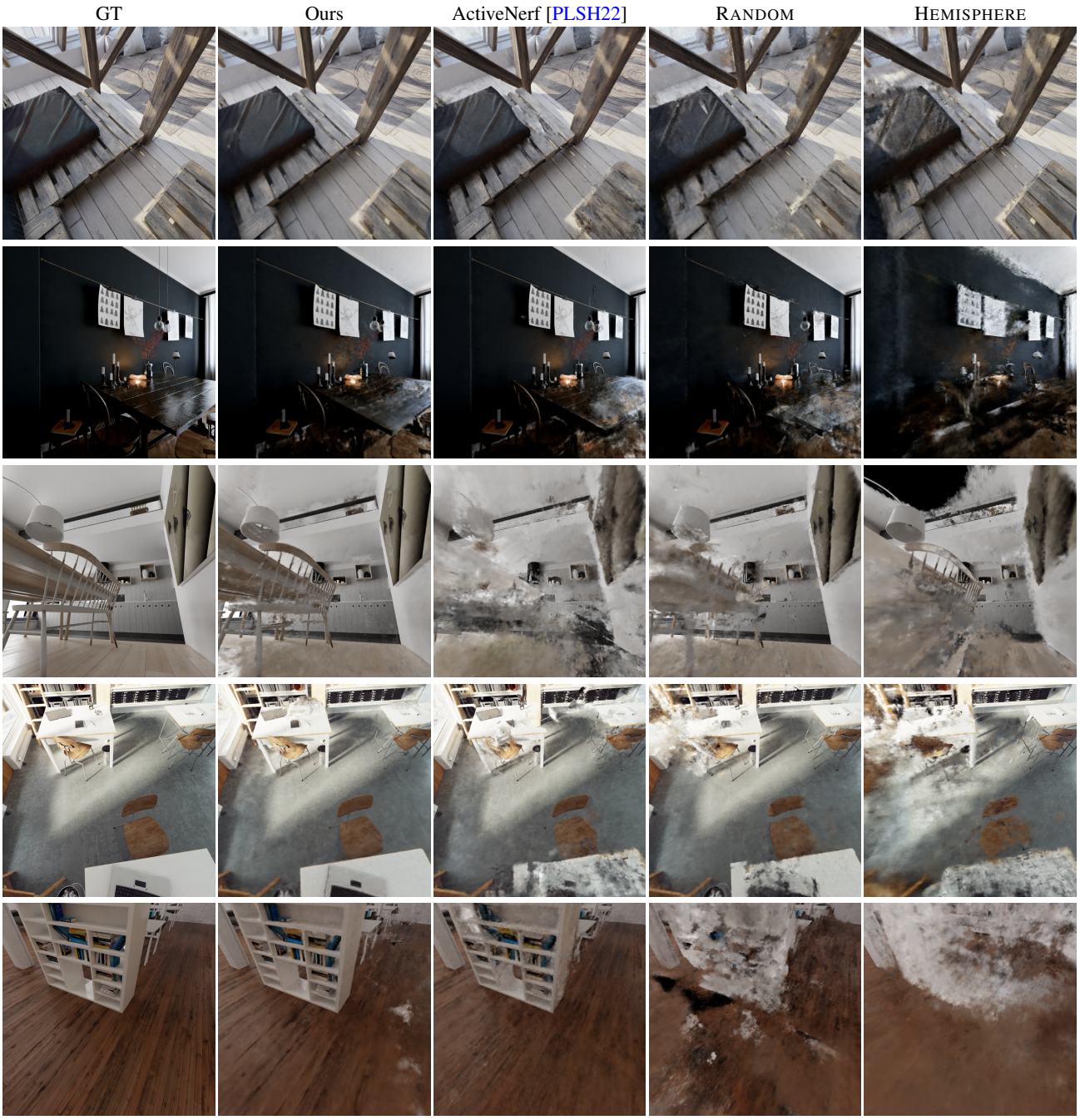

Figure 6: *Images from the random camera sampling test set. We present a visual comparison to baselines (*RANDOM,HEMISPHERE*) and ActiveNerf. The first column shows the ground truth image. The scenes shown are* LIVINGROOM1, LIVINGROOM2, KITCHEN5, OFFICE6 *and* OFFICE9.

nodes more often that the other baselines and we achieve better angular coverage for each node.

We also show in Fig. 5 how our camera sampling improves in the test-sets as we introduce more and more cameras against the random cameras.

4.2. Preliminary Real Scene Evaluation

As discussed earlier (Sec. 3.4, we leave the actual integration of our method into a full capture system as future work. Such a system would require either a user interface (e.g., on a phone) or interfacing with a robotic capture system (e.g., a drone). However, it

Figure 7: *We also do a prelimary simulated evaluation of our method with a real scene, in which we captured approximately 1300 images, from which we exclude every 14th image and create a test-set. We use our algorithm to select the best 200 and we compared it against choosing 200 images at random. On average Our selection scored 16.4 PSNR while the random selection scored 13.8 PSNR.*

is instructive to see how well our method works on real data, so we present a very preliminary test on a real scene. Since we are lacking a capture agent, we instead *simulate* the ability to select new cameras as a proof of concept. Specifically, we achieve this by taking approximately 1300 photos (removing every 14th image to create a left-out test set), simulating "random coverage" of the scene. We then use our algorithm to select 200 cameras from this pool of randomly distributed images. This is evidently a very preliminary test, but the results shown in Fig. 7 show that our method performs significantly better than random selection.

5. Conclusions

We presented an efficient method for selecting cameras for NeRF capture in complicated environments, targeting free-viewpoint navigation. Our key contributions are the introduction of the angular and coverage metrics, and our fast optimization to propose the next best camera for NeRF reconstruction. Our method outperforms baselines and one previous method in overall perfomance; it is also faster than other methods and without significant overhead over baseline methods. An important attribute of our solution is that it is easily interpretable and can provide meaningful guidance and understanding to users without requiring additional images. One other benefit from the simplicity of our methods is that it could be adapted to vary the importance of the scene spatially; we leave this as future work.

Our method is not without limitations. One issue is that we have not investigated if our sampling is biased. If this is the case, no matter how many cameras we sample, we might not reach a "perfect" reconstruction and visual quality. Also, even though the method is efficient, it would benefit from even faster performance allow truly interactive capture.

There are numerous possibilities for future work. From a theoretical perspective, we are interested in studying other metrics of reconstruction quality in a more extensive and complete fashion. We are also very excited about the idea of integrating our approach in a mixed Augmented/Virtual Reality (AR/VR) context: for example we can guide an on-site (AR) user to take photos of a scene so that the remote VR user can very quickly be immersed in the same environment. Using our method in the context of drone capture would allow NeRF captures to be performed with high quality with little human intervention, rendering the approach much more useful and easy-to-use.

References

[BKA*16] BIRCHER A., KAMEL M., ALEXIS K., OLEYNIKOVA H., SIEGWART R.: Receding horizon" next-best-view" planner for 3d exploration. In *2016 IEEE international conference on robotics and automation (ICRA)* (2016), IEEE, pp. 1462–1468. 3

[BMT*21] BARRON J. T., MILDENHALL B., TANCIK M., HEDMAN P., MARTIN-BRUALLA R., SRINIVASAN P. P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5855–5864. 2, 3

[BMV*22] BARRON J. T., MILDENHALL B., VERBIN D., SRINIVASAN P. P., HEDMAN P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5470–5479. 2

[Com18] COMMUNITY B. O.: *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL: http://www.blender.org. 5

[CXG*22] CHEN A., XU Z., GEIGER A., YU J., SU H.: Tensorf: Tensorial radiance fields. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII* (2022), Springer, pp. 333–350. 2

[DF09] DUNN E., FRAHM J.-M.: Next best view planning for active model improvement. In *BMVC* (2009), pp. 1–11. 3

[FKYT*22] FRIDOVICH-KEIL S., YU A., TANCIK M., CHEN Q., RECHT B., KANAZAWA A.: Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5501–5510. 2, 3

[ISDS16] ISLER S., SABZEVARI R., DELMERICO J., SCARAMUZZA D.: An information gain formulation for active volumetric 3d reconstruction. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (2016), IEEE, pp. 3477–3484. 3

[JTA21] JAIN A., TANCIK M., ABBEEL P.: Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5885–5894. 2

[KDSB22] KULHÁNEK J., DERNER E., SATTLER T., BABUŠKA R.: Viewformer: Nerf-free neural rendering from few images using transformers. In *European Conference on Computer Vision* (2022), Springer, pp. 198–216. 2

[LCW*22] LEE S., CHEN L., WANG J., LINIGER A., KUMAR S., YU F.: Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields. *IEEE Robotics and Automation Letters 7*, 4 (2022), 12070–12077. 3

[LFR22] LIU C., FISCHER M., RITSCHEL T.: Learning to learn and sample brdfs. *arXiv preprint arXiv:2210.03510* (2022). 3

[Lum23] LUMAI.COM: *Getting started with Luma AI: Guided Capture Mode*, 2023. URL: https://www.youtube.com/watch?v=JXb_a3ZIGnI. 1

[MESK22] MÜLLER T., EVANS A., SCHIED C., KELLER A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG) 41*, 4 (2022), 1–15. 2, 3, 4, 5

[MRFB16] MOSTEGEL C., RUMPLER M., FRAUNDORFER F., BISCHOF H.: Uav-based autonomous image acquisition with multi-view stereo quality assurance by confidence prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2016), pp. 1–10. 3

[MST*21] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: Representing scenes as neural

radiance fields for view synthesis. *Communications of the ACM 65*, 1 (2021), 99–106. 2, 3, 5

[PLSH22] PAN X., LAI Z., SONG S., HUANG G.: 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* (2022), Springer, pp. 230–246. 1, 3, 4, 5, 6, 7, 8

[RZH*23] RAN Y., ZENG J., HE S., CHEN J., LI L., CHEN Y., LEE G., YE Q.: Neurar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations. *IEEE Robotics and Automation Letters* (2023). 3

[SF16] SCHÖNBERGER J. L., FRAHM J.-M.: Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). 3

[SMGH18] SMITH N., MOEHRLE N., GOESELE M., HEIDRICH W.: Aerial path planning for urban scene reconstruction: A continuous optimization method and benchmark. 3

[SSC22] SUN C., SUN M., CHEN H.-T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5459–5469. 2

[SZFP16] SCHÖNBERGER J. L., ZHENG E., FRAHM J.-M., POLLEFEYS M.: Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14* (2016), Springer, pp. 501–518. 1

[TTM*22] TEWARI A., THIES J., MILDENHALL B., SRINIVASAN P., TRETSCHK E., YIFAN W., LASSNER C., SITZMANN V., MARTIN-BRUALLA R., LOMBARDI S., ET AL.: Advances in neural rendering. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 703–735. 1, 2

[XTS*22] XIE Y., TAKIKAWA T., SAITO S., LITANY O., YAN S., KHAN N., TOMBARI F., TOMPKIN J., SITZMANN V., SRIDHAR S.: Neural fields in visual computing and beyond. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 641–676. 2

[XXP*22] XU Q., XU Z., PHILIP J., BI S., SHU Z., SUNKAVALLI K., NEUMANN U.: Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5438–5448. 2

[YYTK21] YU A., YE V., TANCIK M., KANAZAWA A.: pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 4578–4587. 2

[ZLR*22] ZENG J., LI Y., RAN Y., LI S., GAO F., LI L., HE S., YE Q., ET AL.: Efficient view path planning for autonomous implicit reconstruction. *arXiv preprint arXiv:2209.13159* (2022). 3

[ZZX*22] ZHAN H., ZHENG J., XU Y., REID I., REZATOFIGHI H.: Activermap: Radiance field for active mapping and planning. *arXiv preprint arXiv:2211.12656* (2022). 3