

# CSC2530 Project Report

Jason Tang (tangja10)

University of Toronto

jasontang@cs.toronto.edu

## Abstract

*In this paper, we explore the use of a stereoscopic depth sensor to estimate an initial point cloud for the Point-NeRF method, an extension of Neural Radiance Fields (NeRF) that optimizes the training and inference processes by sampling points near an estimated point cloud. We develop experimental modifications and conduct real-world experiments to analyze the impact of incorporating a coarsely estimated point cloud on the performance and efficiency of Point-NeRF. Our results suggest that incorporating stereoscopic depth sensors into the Point-NeRF 3D scene reconstruction pipeline has the potential to enhance the quality of novel view synthesized images, while still maintaining the sample efficiency improvements provided by Point-NeRF.*

## 1. Introduction

Neural Radiance Fields (NeRF) [2] have gained significant attention recently for their impressive ability to generate high-fidelity novel views and realistic 3D renderings of scenes using only a collection of 2D images captured from multiple viewpoints. In particular, NeRF employs a multilayer perceptron (MLP) network to predict color and density, based on the location and viewing direction, which is then used to reconstruct the entire scene.

To address the large number of empty points within the scene, NeRF first performs a coarse analysis to detect relevant parts of the scene, followed by a finer analysis on the selected portions. Despite this, a significant amount of computational time is still wasted on processing irrelevant, low-density points. As such, the computational requirements for training NeRFs pose a significant challenge for a wide range of practical applications and is an active area of research [1].

Faster NeRF training and inference times can have significant implications for a variety of applications, including film production, augmented reality, and autonomous systems. Improved sample efficiency can enable faster generation of realistic 3D scenes, enhanced immersive experiences for users, and precise environmental perception in autonomous vehicles.

Numerous extensions have been made to further optimize the training and inference processes of NeRF, usually by running the MLP at fewer, but higher quality points. In this project, we will analyze one such extension called Point-NeRF [6], which only samples points near an estimated point cloud. Specifically, we will assess the empirical performance of using a stereoscopic depth sensor (Intel RealSense Depth Camera D435) to estimate an initial point cloud compared to other point cloud estimation methods.

## 2. Related Work

### 2.1. NeRF

Mildenhall et al. presented the foundational NeRF method in 2020 [2], which proposed a new method for synthesizing novel views of a scene from a small number of input images by representing scenes as Neural Radiance Fields. This proposed method involves training a multilayer perceptron (MLP) network to learn a continuous 5D function that maps 3D spatial coordinates  $(x, y, z)$  and 2D viewing directions  $(\theta, \phi)$  to a volume density and RGB color.

To train the NeRF model, the authors sample 3D points along rays emanating from the input viewing angle and position for each pixel, generate output densities and colors, and integrate samples along each ray into a single pixel color using differentiable volume rendering techniques. They then minimize the L2 norm between the rendered images and the ground truth images from the training set.

In particular, the volume density  $\sigma(\mathbf{x})$  only depends on spatial location  $\mathbf{x} = (x, y, z)$ , whereas the output color  $\mathbf{c}(\mathbf{x}, \mathbf{d})$  depends on both spatial location  $\mathbf{x}$  and viewing direction  $\mathbf{d} = (\theta, \phi)$ . The pixel color of each ray  $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$  is then given by the integral of the product of the volume density and color along the ray:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t) \sigma(\mathbf{r}(t)) \mathbf{c}(\mathbf{r}(t), \mathbf{d}) dt \quad (1)$$

$$\text{where } T(t) = \exp \left( - \int_{t_n}^t \sigma(\mathbf{r}(s)) ds \right)$$

where  $t_n$  and  $t_f$  are the near and far bounds, respectively.

The authors use quadrature to numerically estimate this integral, with samples drawn uniformly from evenly spaced bins along the ray. This enables the MLP to learn a continuous representation as the samples are situated on a continuous domain throughout training.

To improve sampling efficiency, the authors begin by generating a coarse, evenly spaced set of points along each ray, which is then used to direct a finer sampling of points towards the more relevant and higher density segments of the ray. They also employ positional encoding of the 5D input to further enhance the resolution at which the MLP can learn to represent.

The empirical results demonstrated that NeRF was effective in generating new views, both in synthetic datasets such as DeepVoxels and a newly developed dataset in this paper, as well as complex real-world scenes such as LLFF and several manually captured scenes. Compared to prior state-of-the-art methods, NeRF showed significant improvement and did not require ground truth 3D geometry, enabling its use in higher complexity real-world datasets. Moreover, NeRF was able to achieve higher resolutions due to its improved sampling efficiency, which was not previously possible due to time and space constraints imposed by finer sampling requirements.

However, even with its improved efficiency, training NeRF still requires a significant amount of time, with convergence taking between 1 to 2 days on an NVIDIA V100 GPU.

## 2.2. Point-NeRF

Point-NeRF [6] is a method that leverages an estimated point cloud to enhance sample efficiency by selectively sampling points that are near the prior cloud and have a higher likelihood of being relevant. To generate the initial point cloud from the collection of input images, the authors utilize existing multi-view reconstruction methods, such as COLMAP [5] or the deep learning based Multi-View Stereo Networks (MVSNet) [7]. Afterward, learnable neural features are mapped to each point by a neural points model, which can utilize pretrained weights to accelerate training time and utilize auxiliary knowledge.

During inference for a given sample point, Point-NeRF combines the viewing angle with an aggregation of the neural features of the  $k$  nearest neighboring points, along with their confidence scores of being non-noisy and their relative positions with respect to the inference point. As a result, Point-NeRF achieves improved sampling efficiency, and is able to store local spatial information within neural points, thereby reducing the burden on the global model to memorize all the details within the scene.

Additionally, Point-NeRF also has the capability to remove points with low confidence (likely noisy) and add points to fill in holes which are commonly found in point

clouds estimated using stereopsis. This results in improved scalability, fast convergence, and higher rendering quality compared to the standard NeRF method.

## 2.3. Multi-View Reconstruction

Extensive research has been conducted in the area of converting multi-view image collections into 3D point clouds, with notable contributions including methods such as COLMAP [5] and Multi-View Stereo Network (MVSNet) [7].

COLMAP [5] is a photogrammetric 3D reconstruction method, which estimates the 3D position of points in a scene using triangulation from multiple input images. To achieve this, COLMAP first detects and matches features in the input images, then performs a sparse reconstruction by estimating the 3D position of these features, and finally refines the reconstruction by performing bundle adjustment to minimize the reprojection error between the estimated 3D points and their corresponding 2D image points. Following this, COLMAP is able to perform a dense point cloud reconstruction by estimating the 3D positions for all pixels in each image, using the computed sparse reconstruction and the camera calibration parameters.

MVSNet [7] is a popular deep learning approach to scene depth reconstruction. It begins by utilizing a convolutional neural network (CNN) to extract deep features from images, which are then warped through differentiable homographies and generates 3D feature volumes containing matching costs between two images. These feature volumes are then aggregated into a single cost volume using a variance-based cost metric to measure multi-view feature differences. This cost volume is then regularized utilizing a multi-scale 3D CNN similar to UNet [4], and a confidence

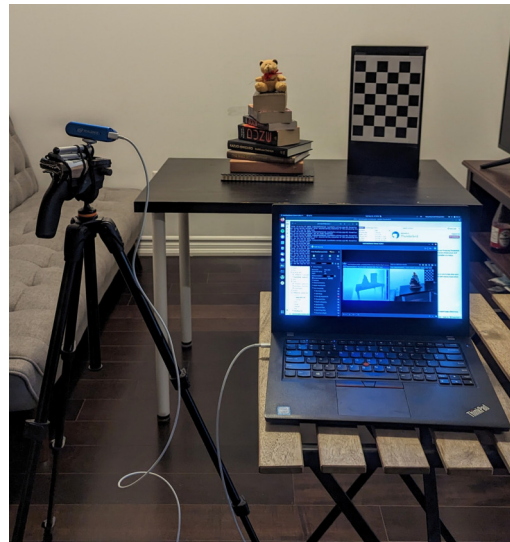


Figure 1. Scene capturing setup for the "books" scene.

map of depth hypotheses is generated. Lastly, the output depth maps are estimated using a differentiable soft argmin operation. The fully differentiable architecture of MVSNet enables end-to-end learning, leading to outstanding performance on standard benchmark datasets.

Despite their successes and popularity, these approaches still produce noisy outlier points or holes in the final point cloud, which are challenging to rectify through post-processing.

## 2.4. Similar NeRF Extensions

### 2.4.1 EfficientNeRF

EfficientNeRF [1] is another NeRF extension which also aims to improve sampling efficiency through two new strategies, Valid Sampling for the coarse stage and Pivotal Sampling for the fine stage. Overall, EfficientNeRF is able to speed up the training process by up to 10 times.

In vanilla NeRF, only a small proportion ( $\sim 10-20\%$ ) of the uniformly sampled coarse points contribute to the final rendering result. As such, EfficientNeRF only uses “valid samples”, which are defined as points with a non-zero density. To obtain these densities, the authors store and query a set of momentum density voxels, which reflect the latest density distribution over the whole scene and are smoothed using momentum. As a result, EfficientNeRF can quickly obtain the density of a given 3D point through querying rather than a forward pass in the MLP.

For the fine stage, the authors propose Pivotal Sampling, a method which only considers samples with a weight greater than a small threshold value. Each sample weight is inversely proportional to the previous coarse weights along the ray, and proportional to the density of the sample. Through this, the authors are able to finely sample points along each ray which are more likely to be relevant in the final rendering.

### 2.4.2 PixelNeRF

PixelNeRF [8] takes a different approach compared to Point-NeRF by first using a deep convolutional network to create spatial image features aligned to individual pixels, and then aggregating the traced pixel features from multiple views and feeding the result into the global MLP. This also allows for initialization with pretrained weights, but does not improve sampling efficiency over vanilla NeRF, meaning a significant amount of computation is still wasted on low density points.

## 3. Theory

Stereoscopic depth sensors are small and cost-effective devices that can be easily integrated into 3D scene reconstruction setups on devices such as drones, cameras, and

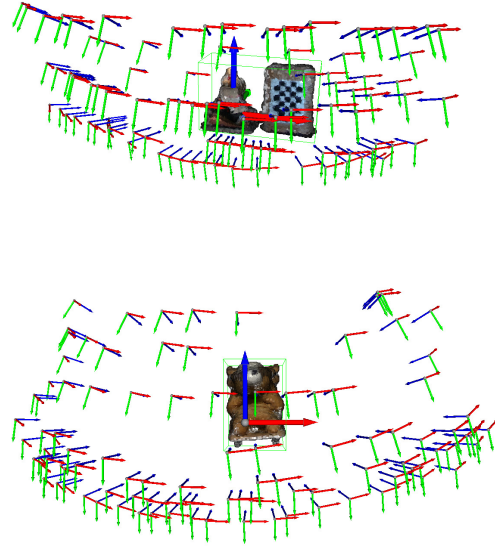


Figure 2. Camera poses for “books” (top) and “bear” (bottom) scenes.

robots. Given these practical advantages, we investigate the performance of Point-NeRF in various real-world scenarios and analyze the effects of possible improvements, such as initializing or regularizing the model using an initial estimated point cloud obtained from a stereoscopic depth camera.

To train the Point-NeRF model using an acquired point cloud, we develop our experimental modifications on top of the [official Point-NeRF Implementation](https://github.com/Xharlie/pointnerf)<sup>1</sup>. The Point-NeRF method requires processed RBGA images and calculated camera pose estimates as inputs to run.

In our experiments, we use the alpha channel to mask irrelevant image backgrounds, thereby allowing the model to focus solely on representing the target and improving training efficiency. To find the relevant parts of the image, we try two methods: a classical approach of separating the two modes of near and far depths, and a more recent neural-based object segmentation approach called “U<sup>2</sup>-Net” [3]. U<sup>2</sup>-Net is based on a nested version of U-Net [4], and has been shown to be highly performant at object segmentation. We use an existing implementation in [rembg](https://github.com/danielgatis/rembg)<sup>2</sup>.

For camera pose estimation, we experiment with both checkerboard alignment in openCV, as well as extracting camera pose estimates from [COLMAP](https://github.com/colmap/colmap)<sup>3</sup>.

Overall, the objective of our project is to analyze the im-

<sup>1</sup><https://github.com/Xharlie/pointnerf>

<sup>2</sup><https://github.com/danielgatis/rembg>

<sup>3</sup><https://github.com/colmap/colmap>

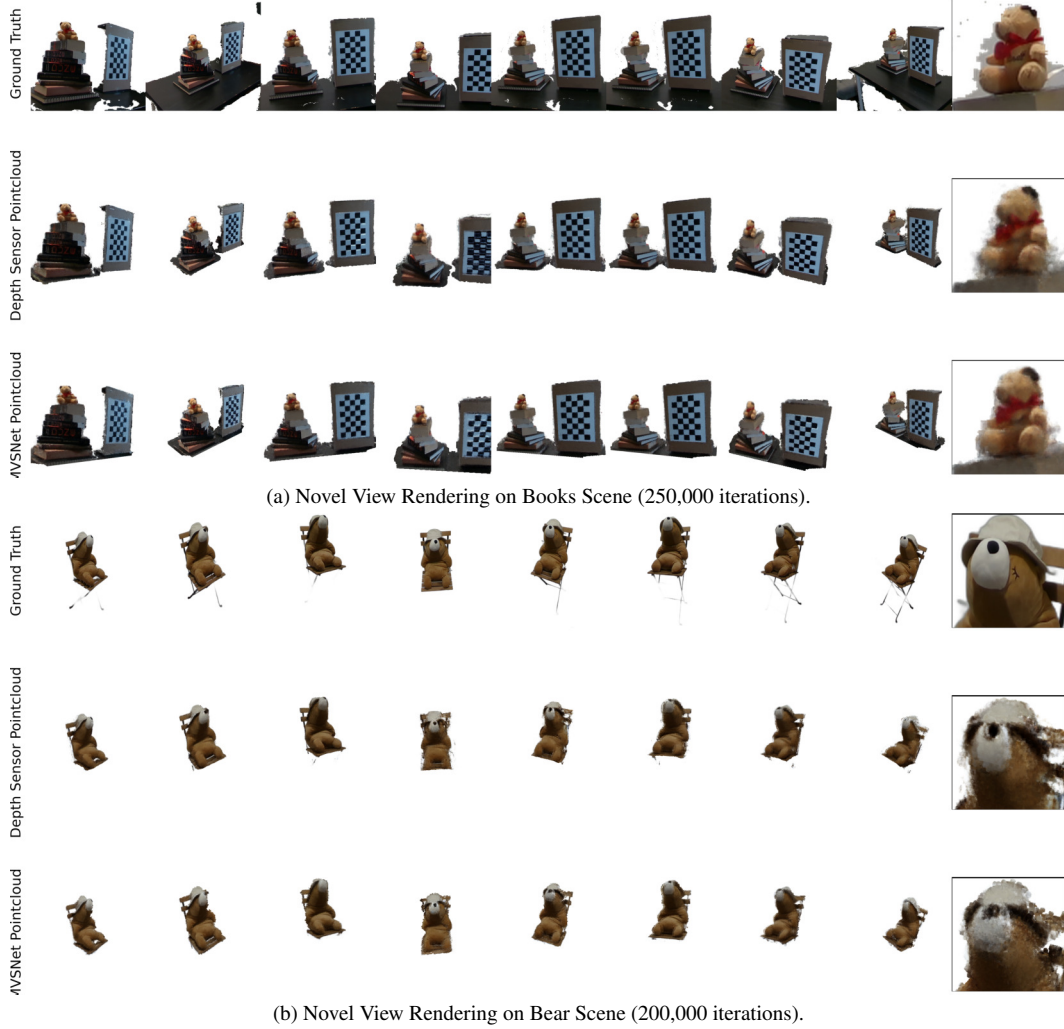


Figure 3. Ground Truth Test Images with Point-NeRF renderings under 2 different initializations.

pact of incorporating a coarsely estimated point cloud on the performance and efficiency of Point-NeRF. To achieve this, we plan to obtain coarse depth maps with a stereoscopic depth sensor, generate 3D point clouds, and conduct real-world experiments to assess the performance of different configurations.

#### 4. Analysis

In our experiments, we utilized an Intel RealSense Stereoscopic Depth Camera (model D435) to capture both RGB and depth images which were then aligned and processed into an estimated point cloud. The D435 sensor works at an ideal range of 0.3 - 3m, which we stay within for all experiments, and we set it to output 640x480 RGB and 848x480 depth images. Besides the two cameras necessary for stereo vision, this sensor also contains an infrared ( $\sim 850\text{nm}$ ) laser projector system which is used to improve

depth estimates of low texture surfaces such as walls. To reduce camera shake, we securely mounted the D435 sensor onto a camera tripod and used a wireless mouse to interact with the laptop when recording (see Fig 1).

We captured 2 scenes, “books” and “bear”, with 162 and 211 images taken from different angles facing the target object, respectively (see Fig 2). In “books”, we also placed a printed checkerboard pattern next to it for camera pose estimation (see Fig 1), whereas we relied only on COLMAP for pose estimation in “bear”.

For metrics, we measure peak signal-to-noise ratio (PSNR) and Structural Similarity Index (SSIM), where a higher value is better, as well as Learned Perceptual Image Patch Similarity (LPIPS) [9], where a lower value is better (see Fig 4). PSNR is a commonly used metric to measure image quality as the ratio of the maximum signal power to the power of corrupting noise. Similarly, SSIM



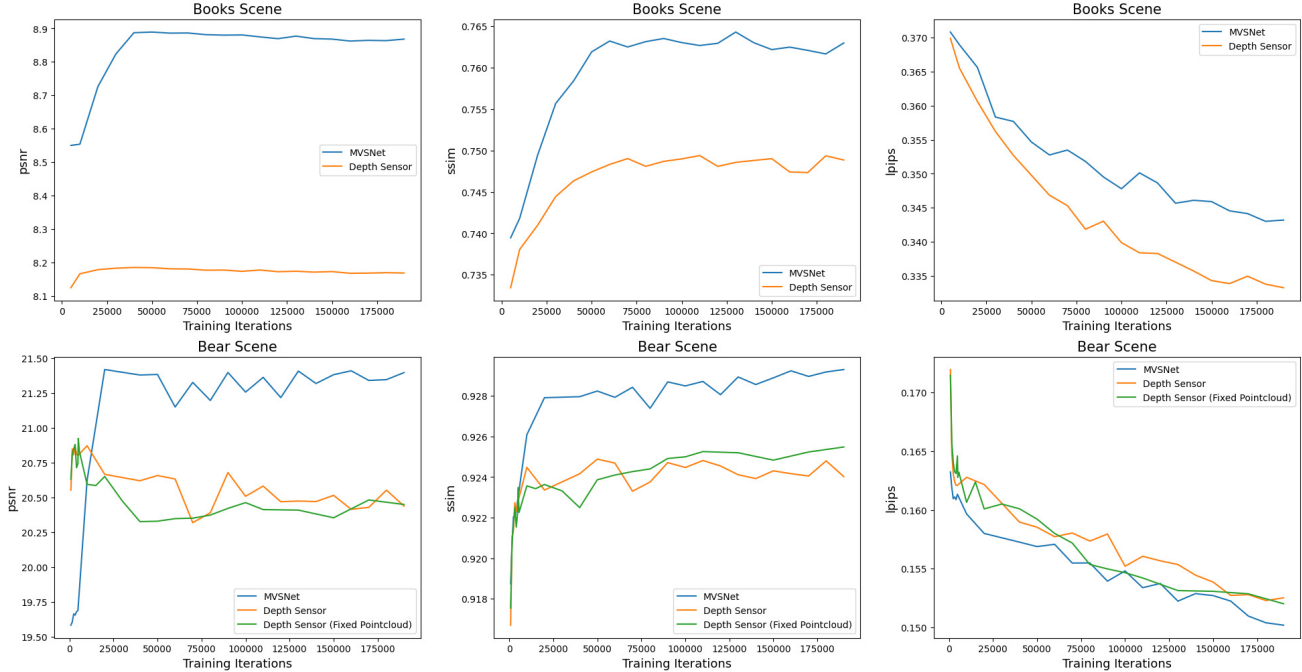


Figure 4. Metrics during experiments.

is commonly used to measure compression quality as structural similarities, such as luminance and contrast. LPIPS is a more recent metric which utilizes the similarity between activations of two images in a deep neural network, and has been shown to agree well with human perception.

All code for our experiments are available at [this GitHub repository](https://github.com/JasonTang99/csc2530_project)<sup>4</sup>. All collected images are available on [Google Drive](#), and all experiment run checkpoints are also available on [Google Drive](#).

## 5. Experimental Results

During our experiments, we observed that processing our scenes with COLMAP took around 30-60 minutes, whereas using MVSNet took around 10-15 minutes to complete. Following this, we trained each Point-NeRF experiment for at least 200,000 iterations, as per the recommendations in the research paper. We ran our experiments on an NVIDIA RTX 3060 Ti with 8GB VRAM whereas most researchers run their experiments on the much more powerful NVIDIA V100. Due to these hardware constraints, each run took between 12-16 hours to finish, which limited the number of experiments we could conduct.

In general, comparable quality results can be observed among the various configurations in generating new views from the test dataset, as depicted in Figure 3. We note the alignment issues that are observed in the “books” scene on

the checkerboard. We believe that this is likely due to alignment challenges when capturing images from the sides of the scene, resulting in a heavily distorted checkerboard.

In terms of empirical results (Fig 4), we can see that the MVSNet method used in the Point-NeRF paper outperforms our experiments using an initial point cloud estimated with the depth sensor in PSNR and SSIM metrics in both scenes. However, we can see that in terms of the LPIPS metric, our method performs comparably to MVSNet on the “bear” scene, and outperforms MVSNet on the “books” scene. We also consider fixing the initial depth sensor pointcloud without growing or pruning points as a form of regularization, but measures at around the same LPIPS and worse PSNR/SSIM compared to allowing points to grow and be pruned (see the “bear” scene in Fig 4). Moreover, the training process would occasionally crash due to nan values in this setting, likely due to the potential holes in the initialized pointcloud.

Another challenge we faced during our experimentation was that the U<sup>2</sup>-Net object segmentation method [3] frequently failed to identify the object of interest as the foreground when the camera was in close proximity to the chessboard, as illustrated in Fig 5. Therefore, we restricted the use of the neural method to situations where the chessboard was not visible in the frame, such as the “bear” scene, and instead utilized the separation of near and far modes in the depth distribution to estimate pose for the “books” scene.

<sup>4</sup>[https://github.com/JasonTang99/csc2530\\_project](https://github.com/JasonTang99/csc2530_project)



Figure 5. Examples of background removal using U<sup>2</sup>-Net.

## 6. Concluding Remarks

In this project, we have assessed the empirical performance of using a stereoscopic depth sensor to estimate an initial point cloud for Point-NeRF in a variety of real-world settings. Our results suggest that the integration of coarse depth images into the Point-NeRF pipeline could serve as a promising and effective approach to enhance the quality of novel synthesized views. This approach maintains the advantages offered by Point-NeRF, such as faster convergence, improved scalability, and superior rendering quality compared to the vanilla NeRF method.

One important avenue for future work is the optimization of hyperparameters, and the comparison of the proposed and original methods on a wider variety of scenes. This will allow for a comprehensive evaluation of which settings are necessary for depth sensor initialization to outperform MVSNet initialization. Another direction to explore is the incorporation of depth images as supplementary inputs to MVSNet or other multiview stereo techniques, as this may enhance the accuracy of the pose estimations and point cloud reconstructions which Point-NeRF processes. Lastly, future work could also investigate the potential of using other inexpensive depth sensors, like structured light projection, to acquire depth images with higher precision and resolution.

## References

- [1] T. Hu, Shu Liu, Yilun Chen, Tiancheng Shen, and Jiaya Jia. Efficientnerf - efficient neural radiance fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12892–12901, 2022. [1](#), [3](#)
- [2] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020. [1](#)
- [3] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 106:107404, 10 2020. [3](#), [5](#)
- [4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. [2](#), [3](#)
- [5] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, 2016. [2](#)
- [6] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5428–5438, 2022. [1](#), [2](#)
- [7] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *European Conference on Computer Vision*, 2018. [2](#)
- [8] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4576–4585, 2020. [3](#)
- [9] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. [4](#)