OmniVoxel: A Fast and Precise Reconstruction Method of Omnidirectional Neural Radiance Field

Qiaoge Li
University of Tsukuba
li.qiaoge@image.iit.tsukuba.ac.jp

Itsuki Ueda
University of Tsukuba
ueda.itsuki@image.iit.tsukuba.ac.jp

Chun Xie
University of Tsukuba
xiechun@ccs.tsukuba.ac.jp

Hidehiko Shishido
University of Tsukuba
shishido@ccs.tsukuba.ac.jp

Itaru Kitahara
University of Tsukuba
kitahara@it.tsukuba.ac.jp

Abstract—This paper proposes a method to reconstruct the neural radiance field with equirectangular omnidirectional images. Implicit neural scene representation with a radiance field can reconstruct the 3D shape of a scene continuously within a limited spatial area. However, training a fully implicit representation on commercial PC hardware requires a lot of time and computing resources (15 ~ 20 hours per scene). Therefore, we propose a method to accelerate this process significantly (20 ~ 40 minutes per scene). Instead of using a fully implicit representation of rays for radiance field reconstruction, we adopt feature voxels that contain density and color features in tensors. Considering omnidirectional equirectangular input and the camera layout, we use spherical voxelization for representation instead of cubic representation. Our voxelization method could balance the reconstruction quality of the inner scene and outer scene. In addition, we adopt the axis-aligned positional encoding method on the color features to increase the total image quality. Our method achieves satisfying empirical performance on synthetic datasets with random camera poses. Moreover, we test our method with real scenes which contain complex geometries and also achieve state-of-the-art performance. Our code and complete dataset will be released at the same time as the paper publication.

Index Terms—Human-centered computing, Virtual Reality, Immersive Experience, Free-viewpoint Videos, Image-based Rendering

I. INTRODUCTION

With the increasingly accurate reproduction of natural scenes by neural rendering technology, lifelike reconstruction of real scenes in Virtual Reality (VR) is gradually becoming possible. The most representative techniques to promote this research are Neural Radiance Field (NeRF) and Multi-Plane Image (MPI). In the foreseeable future, real scene reconstruction will increasingly be applied to the field of VR as well as many other multimedia fields like tele-education and virtual tourism. However, most cameras can only provide perspective views as input, and the number of images required to build a complete scene can be huge. Using omnidirectional shots to reconstruct the entire scene reduces the need for the number of pictures and the extra attention to the coverage area that needs to be considered for the shot.

Omnidirectional reproduction of real scenes has been put into use for several years. Years before, this procedure required the use of multiple perspective views for stitching. However, for hand-held photography, it is hard to strictly make the positions of different cameras identical when capturing multiple perspective images, which leads to the result that the stitching will produce displacements and distortions at the seams of adjacent images. Today, omnidirectional scenes are shot with ultra-wide-angle fish-eye lenses and built-in algorithms for real-time stitching. The quality of the omnidirectional images generated in this way is stable.

Our goal is to reconstruct the 3D space of the captured scene holistically with equirectangular omnidirectional inputs in a relatively short time. We focus on the fact that the omnidirectional image contains the ray information from all directions space to the camera position. Therefore, from the original hypothesis of NeRF, we assume that learning ray information from multiple spherical panoramas can generate a continuous omnidirectional radiance field that is fully capable of representing visual information within a whole space. However, training on new scenes based on this assumption takes a vast amount of time, also the limitations of NeRF in representing rays lead to a low quality reconstruction. Our proposed method modeled the 3D scene into latent voxels to accelerate the reconstruction speed for the radiance field and to increase the speed. When modeling rays from multi-view panoramas with original assumptions of NeRF, we find that the intersection among rays is not evenly distributed from the center to the edge of the scene. This property results in an uneven quality of the reconstruction of the scene from inside to outside for unbounded scenes. Therefore, we propose to use the spherical coordinate voxelization method instead of the traditional cubic voxel representation using Cartesian coordinates. This paper discusses the reconstruction quality as well as the processing speed of previous Omnidirectional NeRF and the two voxelization with the proposed method for different scenarios.

Our voxelization method adopt a tensor decomposition
approach to reduce spatial complexity, enabling us to train models with higher resolution. Due to the much higher frequency of omnidirectional images than perspective ones, the traditional positional encoding method is not enough to get satisfying results. We applied the axis-aligned positional encoding method to the color features to increase the detail quality for the final results on captured images. In addition, we produced a complete dataset that can be used for indoor reconstruction tasks, including 315 equirectangular photographs captured by a high-resolution omnidirectional camera under fixed lighting conditions and their camera parameters. We also provide ground truth depth information scanned by the LiDAR scanner. In summary, our main contributions are listed as follows:

- We propose a method that uses only RGB information from multiple captured panoramas to reconstruct the radiance field holistically within a short time.
- We elaborate and experimentally demonstrate that the reconstruction quality of the voxel-based partial explicit representation is better than the ray-based implicit representation when using panoramas for free viewpoint image generation.
- We provide a comprehensive dataset for omnidirectional novel view synthesis task including over 1500 equirectangular omnidirectional photographs with their camera parameters. This dataset contains four different scenes including indoor, outdoor and synthesized scenes. We also provide the ground truth depth information for indoor and synthesized scenes.

II. RELATED WORK

Novel view synthesis aims to solve this problem by synthesizing new views using a limited number of RGB images. In the past three years, various implicit neural scene representation methods using deep learning have achieved compelling results for the novel view synthesizing task. [1]–[4]. Among them, Neural Radiance Field (NeRF) [3] and its derivative methods [5], [6] receive wide range of attention. Unlike traditional scene reconstruction [7]–[9], which requires an explicit representation of the scene geometry as a first step, NeRF implicitly represents the scene as rays observed from the viewpoint using neural networks with the structure of multilayer perceptron (MLP). This representation allows the reconstruction of the scene to be continuous within space. Another essential feature of NeRF (i.e. “positional encoding”) enables low-dimensional inputs to be projected into higher dimensions [3], which not only allows the gradient of the interpolation function describing the volume density of the space to be calculated but also significantly improves the accuracy of the final function obtained by deep learning.

For boundless scenes, however, the performance of NeRF is relatively low. The intersection of light sampled from different cameras becomes increasingly sparse as the actual spatial distance gets farther away, which reduces the quality of NeRF when learning the representation of a distant scene or the objects in the background. NeRF++ [10] proposes to solve this problem by sampling based on disparity outside a specific range instead of sampling based on distance.

Another common problem that NeRF ignored is that the pixel on each image cannot actually be represented by the rays it modeled. This produces ambiguous features sampled on adjacent space points, thereby limiting NeRF’s performance significantly, especially for unbounded scenes. Mip-NeRF-360 [6] adopted cone sampling and integrated positional encoding to substantially increase the scene representation quality for unbounded scenes.

With the gradual unification of the structure of omnidirectional cameras, i.e., consisting of multiple wide-angle lenses, and the gradual decrease of their cost, the acquisition of spherical panoramic photos is no longer complicated. Therefore, research on synthesizing novel views directly on omnidirectional images comes out these years [11], [12]. Recently, some contemporary works have adopted equirectangular omnidirectional images as input for the reconstruction of the radiance field. Several recent works [13] show their attempt at omnidirectional radiance field representation. In their methods, the depth information of the scene is known, which allows them to obtain accurate results with almost no prediction of the transparency of the scene. Another work [14] used only RGB images and the camera parameters as input for reconstruction, but their results are very bad compared with SOTA methods. One common problem is that their methods take a lot of time (15 ~ 20 hours per scene) for reconstruction, while our method only requires a much shorter time (20 ~ 40 minutes per scene).

III. METHOD

A. Flexible Spherical Voxelization

Since it is impossible to choose the shooting direction with an omnidirectional camera, a spherical panorama is thus more likely to contain a boundless scene than a perspective view. In addition, commercial omnidirectional cameras require handheld or tripod shots, which will result in a significant portion of the panorama being occupied by near objects. These two points lead us to consider the reconstruction quality of both near and far scenes. If we use voxels uniformly distributed along the Cartesian coordinate system, then as the distance to the camera increases, less and less light will pass through the same voxel. As a result, the quality of the reconstruction will vary with the distance to the camera. Therefore, we devise a spherical voxelization approach to balance the reconstruction quality of distant and close views for unbounded scenes.

In the proposed method, the spherical voxel representation explicitly models the color and density features in spherical grid cells. We store these modalities separately in 3 tensors along r, θ, and ϕ coordinates. We optimize the partitioning on the r-axis in the voxelization process. The interval of voxelization along the r-axis decreases as the distance from the center increases as shown in the following equation:

\[ T_i = \ln (i + 1), i \in [0, N], \]  

(1)
where $T_i$ is the real sampling distance on the radius for $i$th voxel, and $t_i$ represents the distance between the $i$th evenly distributed voxel and the ray origin. $N$ is the total number of voxels sampled on the radius. This distribution effectively reduces the difference in the number of voxels per ray passed by cameras farther away from the scene center during sampling relative to cameras closer to the scene center. Then, the scene would be easily represented by interpolation. The trilinear interpolation method has been adopted to interpolate the queried 3D positions.

B. Tensor Decomposition.

Inspired by TensoRF [15], we applied Vector-Matrix (VM) decomposition on the tensors to decrease memory consumption. In our case, scene is represented by 3D tensor modes corresponding with $r, \theta,$ and $\phi$ axis. Given a 3D tensor $T \in \mathbb{R}^{R \times \Theta \times \Phi}$, VM decomposition factorizes a tensor into multiple vectors and matrices; the following equation shows this process:

$$T = \sum_{n=1}^{N_1} V_n^{R} \otimes M_n^{\Theta \Phi} + \sum_{n=1}^{N_2} V_n^{\Theta} \otimes M_n^{R \Phi} + \sum_{n=1}^{N_3} V_n^{\Phi} \otimes M_n^{R \Theta},$$

(2)

where $V_n^{R}, V_n^{\Theta}, V_n^{\Phi}$ corresponds to a rank-one tensor component, $M_n^{\Theta \Phi}, M_n^{R \Phi}, M_n^{R \Theta}$, are matrix factors for two (represented by superscripts) of the three modes that different from the tensor components denoted in the corresponding vector. In 3D representation with the Cartesian coordinate, a scene can distribute and appear equally complex along its three axes, and in that case, $N_1 = N_2 = N_3$. In our case, we also set $N_1 = N_2 = N_3$, the balance between $R$ ($N_1$ and $\Theta, \Phi$ ($N_2, N_3$) is adjusted by the scale of $R$. 3D tensor would be enough for representing the volume density, while color requires one more dimension for the representation of channels. This is represented by a vector $b$ multiplied by each color tensor. In addition, we use three component tensors to simplify notation and the following discussion in later sections: $A_R^{\Theta} = V^{R} \otimes M_{\Theta \Phi}^{\Phi}, A_\Theta^{\Theta} = V_{\Theta} \otimes M_{R \Phi}^{\Phi},$ and $A_\Phi^{\Theta} = V_{\Phi} \otimes M_{R \Theta}^{\Phi}$. Then the volume density and color of the 3D voxels can be expressed as

$$G_\sigma = \sum_{n=1}^{N_1} \sum_{m=1}^{\sigma \Phi} A_{m,n}^\sigma,$$

$$G_c = \sum_{n=1}^{N_1} A_{\sigma}^{R} \otimes b_{3n-2} + A_{\sigma}^{\Theta} \otimes b_{3n-1} + A_{\sigma}^{\Phi} \otimes b_{3n},$$

(3)

where $G_\sigma$ and $G_c$ represents the 3D geometric tensor for density and color. In total, we parameterize the voxels into $3N_\sigma + 3N_c$ matrices and $3N_\sigma + 6N_c$ vectors. Fig. 1 gives a more intuitive explanation of the whole procedure. An important benefit of representing voxels using tensor decomposition is that the computational effort of trilinear interpolation necessary to reconstruct neural radiation sites is greatly reduced. Interpolate the component tensor trilinearly is equivalent to interpolate the corresponding modes of its vector/matrix factors linearly/bilinearly. Therefore, it save a lot of time and computing costs which enables us to train higher resolution of images and voxels than other voxelization approaches.

C. Positional Encoding

In original NeRF, positional encoding is essential for performance improvement. It is attributed to the difficulty of MLPs to learn high-frequency functions due to spectral biases, which can make the network learned by MLPs with only 5D inputs unable to restore scene details adequately. The NeRF experiments obtained good results with a heuristic sinusoidal mapping of the input coordinates (i.e., “position encoding”) to allow MLPs to represent higher frequency content. Since the average amount of information per pixel contained in an equirectangular panorama is higher than that of a perspective view with the same number of pixels in most cases, the frequency of information required to be restored for our task is much higher than that of the original NeRF. We applied positional encoding along the direction of aligned axes to improve the reconstruction quality of high-frequency information. Our generic positional encoding mapping $\gamma$ inputs points $\mathbf{v} \in [0,1]^d$ to the surface of a hypersphere that has much higher dimensions with a set of sinusoids:

$$\gamma(\mathbf{v}) = [a_1 \cos(2\pi b_1^T \mathbf{v}), a_1 \sin(2\pi b_1^T \mathbf{v}), \ldots, a_m \cos(2\pi b_m^T \mathbf{v}), a_m \sin(2\pi b_m^T \mathbf{v})]^T.$$

(4)

For our axis aligned positional encoding:

$$a_i = J^d, b_i = \sigma^{j/m},$$

(5)

where $j = 0, \ldots, m - 1$. $\sigma$ is a hyperparameter that is different for various tasks. In our case, $\sigma = 2$. In both positional encoding methods, $a_i$ is a vector that only contains 1s with the same element number of the input dimension $d$. The embedding size for the positional encoding method is $m$. In our method, we applied the axis aligned positional encoding methods on the color features before input them to the MLP decoder.

D. Rendering and Learning

We render the image with volumetric differentiable renderer same as NeRF, for each pixel, the color result is integrated numerically by weighting the sum of the RGB values and the same as NeRF, for each pixel, the color result is integrated numerically by weighting the sum of the RGB values and the average amount of information per pixel contained in an equirectangular panorama is higher than that of a perspective view with the same number of pixels in most cases, the frequency of information required to be restored for our task is much higher than that of the original NeRF. We applied positional encoding along the direction of aligned axes to improve the reconstruction quality of high-frequency information. Our generic positional encoding mapping $\gamma$ inputs points $\mathbf{v} \in [0,1]^d$ to the surface of a hypersphere that has much higher dimensions with a set of sinusoids:

$$\gamma(\mathbf{v}) = [a_1 \cos(2\pi b_1^T \mathbf{v}), a_1 \sin(2\pi b_1^T \mathbf{v}), \ldots, a_m \cos(2\pi b_m^T \mathbf{v}), a_m \sin(2\pi b_m^T \mathbf{v})]^T.$$

(4)

For our axis aligned positional encoding:

$$a_i = J^d, b_i = \sigma^{j/m},$$

(5)

where $j = 0, \ldots, m - 1$. $\sigma$ is a hyperparameter that is different for various tasks. In our case, $\sigma = 2$. In both positional encoding methods, $a_i$ is a vector that only contains 1s with the same element number of the input dimension $d$. The embedding size for the positional encoding method is $m$. In our method, we applied the axis aligned positional encoding methods on the color features before input them to the MLP decoder.

We render the image with volumetric differentiable renderer same as NeRF, for each pixel, the color result is integrated numerically by weighting the sum of the RGB values and the volume density at a set of discrete sampling points on the ray as the following equation:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i,$$

(6)

$$T_i = \exp(-\sum_{j=1}^{i-1} \sigma_i \delta_i),$$

(7)

where $N$ is the number of the sampling points, $\delta_i$ is the distance between adjacent samples. The function adopts traditional alpha blending method with alpha values $\alpha_i = \cdots$
1 − \exp(−\sigma_i \delta_i) \cdot \delta_i \text{ represents the volume density at the sampled point. The } T_i \text{ function calculates the accumulated transmittance between the two samples along the ray.}

With the above rendering method, we can render the omnidirectional image spherically, then we compare the image sphere and the ground truth and calculate the photometric loss and L1 sparsity loss. Our total loss function is like the following:

$$L = \| C − \tilde{C}\|_2^2 + \omega \cdot L_1,$$

where \(\omega\) represents the weight for \(L_1\) loss, and \(\tilde{C}\) is the ground truth color.

IV. EXPERIMENTS

A. Datasets

We tested our method on the dataset made by our own. For synthesized data, we acquire camera parameters from Blender during rendering. For actual data, we use Pix4D mapper software to get the extrinsic parameters of the cameras. We use about 1/3 of them for training, 1/3 for validation, and 1/3 for testing.

B. Implementation Details

During the experiment, the indoor and Blender images were resized to the resolution of 1024 \times 2048 while the outdoor images kept their primitive resolution at 960 \times 1920. We set the batch size 4096, which is the number of rays we sampled for each unit sphere. AdamW optimizer is used during the training process. We initialize the learning rate at 5 \times 10^{-4} and exponentially reduce it to 5 \times 10^{-5} during all training steps. We set the weight of \(L_1\) loss as 8 \times 10^{-5} We train our voxelization models for 30 thousand epochs in all experiments on a single RTX 3090 GPU. As a comparison, we trained the omnidirectional NeRF model for 250 thousand epochs for their best performance.

C. Results

In Tab. I, we compare different encoding methods within our method quantitatively in PSNR, SSIM, and LPIPS. Our experimental results in Fig. 2 show that voxel representation achieved much better result than NeRF-based approach. The performance of spherical voxelization and cubic voxelization method are similar many different evaluation metrics. Axis-aligned positional encoding method has advantages in scenes with complex colors. Axis-aligned positional encoding performs well in real scenes because our reconstructed data are real-world 3D voxels, so the frequency domain of the infor-

---

**TABLE I QUANTITATIVE RESULTS**

| Metric | Scene | Indoor | Outdoor1 | Outdoor2 | Blender |
|--------|-------|--------|----------|----------|---------|
| PSNR   | Omninerf | 18.77  | 21.80    | 23.39    | 26.30   |
|        | Omninerf APE | 19.73  | 22.32    | 23.79    | 27.01   |
|        | Omnivoxel Cubic | 22.33  | 27.24    | 27.91    | 33.14   |
|        | Omnivoxel Cubic APE | 26.70  | 27.51    | 28.00    | 32.96   |
|        | Omnivoxel Sphere | 23.38  | 27.10    | 27.95    | 33.23   |
|        | Omnivoxel Sphere APE | 26.87  | 27.38    | 27.94    | 33.19   |
| SSIM   | Omninerf | 0.752  | 0.764    | 0.815    | 0.902   |
|        | Omninerf APE | 0.729  | 0.780    | 0.826    | 0.916   |
|        | Omnivoxel Cubic | 0.805  | 0.824    | 0.893    | 0.936   |
|        | Omnivoxel Cubic APE | 0.815  | 0.808    | 0.892    | 0.932   |
|        | Omnivoxel Sphere | 0.796  | 0.798    | 0.892    | 0.937   |
|        | Omnivoxel Sphere APE | 0.805  | 0.802    | 0.891    | 0.936   |
| LPIPS (Alex) | Omninerf | 0.583  | 0.376    | 0.346    | 0.183   |
|        | Omninerf APE | 0.448  | 0.364    | 0.328    | 0.131   |
|        | Omnivoxel Cubic | 0.301  | 0.355    | 0.234    | 0.125   |
|        | Omnivoxel Cubic APE | 0.266  | 0.344    | 0.233    | 0.116   |
|        | Omnivoxel Sphere | 0.303  | 0.350    | 0.237    | 0.110   |
|        | Omnivoxel Sphere APE | 0.279  | 0.328    | 0.234    | 0.112   |
| LPIPS (VGG) | Omninerf | 0.538  | 0.418    | 0.407    | 0.373   |
|        | Omninerf APE | 0.420  | 0.396    | 0.375    | 0.306   |
|        | Omnivoxel Cubic | 0.339  | 0.379    | 0.319    | 0.294   |
|        | Omnivoxel Cubic APE | 0.329  | 0.369    | 0.327    | 0.303   |
|        | Omnivoxel Sphere | 0.349  | 0.376    | 0.326    | 0.289   |
|        | Omnivoxel Sphere APE | 0.341  | 0.361    | 0.323    | 0.291   |
mation is oriented along the spatial axes (for both Cartesian and Spherical coordinate systems), rather than isotropic within dimensions. Even though we cannot compare directly with other methods on perspective dataset, it was evident that our result shows that the proposed method reached state-of-the-art performance for unbounded scene reconstruction and has similar performance to what mip-NeRF-360 [6] and NeRF++ [10] have for their dataset.

Due to space limitations, we cannot show the full-size equirrectangular image results in the main text. They are included in the supplementary materials. As a result, spherical voxelization could balance the quality of the close and distant views of the center of the space. Also, the axis-aligned positional encoding method can reconstruct details of the tripod object while the original positional encoding can’t. We also developed some satisfying flying-through videos with the trained models attached in the supplementary materials.

Another notable point is that our approach is far faster than directly applying the NeRF model to train on the spherical ray representation. Our method takes only 40 minutes to train a full representation on the Indoor dataset while NeRF takes more than 15 hours. Our proposed method makes it possible to reconstruct the entire scene using omnidirectional photos quickly and in high quality.

V. CONCLUSION

We present a method for fast holistic reconstruction of the neural radiance field with multiple omnidirectional images. Our key idea is to use voxel grid representation and tensor decomposition to replace the fully implicit representation. We use the Unit Sphere model to sample the rays in different directions and adopt a spherical voxelization method to balance the quality of closer and distant views from the center of the scene. By modifying the positional encoding approaches, we quantitatively increase the quality of our result. Our method achieves satisfying empirical performance on synthetic datasets with random camera poses. Moreover, our experiments on real datasets show that we can continuously reconstruct the unbounded omnidirectional scene at state-of-art-performance.

REFERENCES

[1] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, “Local light field fusion: Practical view synthesis with prescriptive sampling guidelines,” ACM Transactions on Graphics (TOG), 2019.
[2] E. Susic, S. Liu, J. Ortiz, and A. Davison, “iMIP: Implicit mapping and positioning in real-time,” in Proceedings of the International Conference on Computer Vision (ICCV), 2021.
[3] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthy, and R. Ng, “NeRF: Representing scenes as neural radiance fields for view synthesis,” in ECCV, 2020.
[4] Q. Wang, Z. Wang, K. Genova, P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. Funkhouser, “Ibn-net: Learning multi-view image-based rendering,” in CVPR, 2021.
[5] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, “Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields,” in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2021, pp. 5855–5864.
[6] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, “Mip-nerf 360: Unbounded anti-aliased neural radiance fields,” CVPR, 2022.
[7] P. Hedman, T. Ritschel, G. Drettakis, and G. Brostow, “Scalable inside-out image-based rendering,” ACM Trans. Graph., vol. 35, no. 6, Nov. 2016. [Online]. Available: https://doi.org/10.1145/2980179.2982420
[8] T. Kanade, P. Rander, and P. Narayanan, “Virtualized reality: constructing virtual worlds from real scenes,” IEEE MultiMedia, vol. 4, no. 1, pp. 34–47, 1997.
[9] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, “Stereo magnification: Learning view synthesis using multiplane images,” in SIGGRAPH, 2018.
[10] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, “Nerf++: Analyzing and improving neural radiance fields,” arXiv:2010.07492, 2020.
[11] Q. Li, T. Oto, C. Xie, H. Shishido, Y. Kameda, H. Kim, and I. Kitahara, “Generative image quality improvement in omnidirectional free-viewpoint images and assessments,” IEEEJ Trans. on IEVC, vol. 10, no. 1, pp. 107–119, 2022.
[12] T. Bertel, M. Yuan, R. Lindroos, and C. Richardt, “OmniPhotos: Casual 360° VR photography,” ACM Transactions on Graphics, vol. 39, no. 6, pp. 266:1–12, Dec. 2020. [Online]. Available: https://richardt.name/omniphotos/
[13] C. Hsu, C. Sun, and H. Chen, “Moving in a 360 world: Synthesizing panoramic parallaxes from a single panorama,” CoRR, vol. abs/2106.10859, 2021. [Online]. Available: https://arxiv.org/abs/2106.10859
[14] Q. Li, I. Ueda, C. Xie, H. Shishido, and I. Kitahara, “Omnidirectional neural radiance field for immersive experience,” in 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), 2022, pp. 740–741.
[15] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, “Tensorf: Tensorial radiance fields,” in European Conference on Computer Vision (ECCV), 2022.