Continuous Levels of Detail for Light Field Networks

David Li
https://davidl.me
Brandon Y. Feng
https://brandonyfeng.github.io
Amitabh Varshney
https://www.cs.umd.edu/~varshney/

University of Maryland, College Park
Maryland, USA

Abstract

Recently, several approaches have emerged for generating neural representations with multiple levels of detail (LODs). LODs can improve the rendering by using lower resolutions and smaller model sizes when appropriate. However, existing methods generally focus on a few discrete LODs which suffer from aliasing and flicker artifacts as details are changed and limit their granularity for adapting to resource limitations. In this paper, we propose a method to encode light field networks with continuous LODs, allowing for finely tuned adaptations to rendering conditions. Our training procedure uses summed-area table filtering allowing efficient and continuous filtering at various LODs. Furthermore, we use saliency-based importance sampling which enables our light field networks to distribute their capacity, particularly limited at lower LODs, towards representing the details viewers are most likely to focus on. Incorporating continuous LODs into neural representations enables progressive streaming of neural representations, decreasing the latency and resource utilization for rendering.

1 Introduction

In the past few years, implicit neural representations [29, 31] have become a popular technique in computer graphics and vision for representing high-dimensional data such as 3D shapes with signed distance fields and 3D scenes captured from multi-view cameras. Light Field Networks (LFN) [40] are able to represent 3D scenes with support for real-time rendering as each pixel of a rendered image only requires a single evaluation through the neural network.

In computer graphics, levels of detail (LODs) are commonly used to optimize the rendering process by reducing resource utilization for smaller distant objects in a scene. LODs prioritize resources to improve the overall rendering performance. In streaming scenarios, LODs can prioritize and reduce network bandwidth usage. While LODs for implicit neural representations are beginning to be explored [6, 7, 22, 24, 28], most existing work focuses on offering a few discrete LODs which have three drawbacks for streaming scenarios. First, with only a few LODs, switching between them can result in flicker or popping effects as

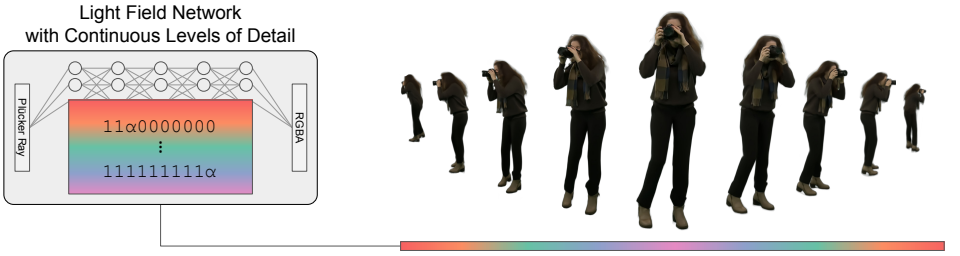


Figure 1: Our light field network features continuous levels of detail, enabled by training with summed-area table filtering and saliency-based importance sampling. Continuous levels of detail enable the interactive streaming of light fields with smooth transitions and finely tuned adaptivity.

there are significant jumps from one LOD to the next. Second, discrete LODs require streaming in larger model deltas which take longer and create spikes in network activity. Third, transitioning across successive LODs can require rendering multiple levels and impact the rendering performance. Continuous LODs resolve these challenges by allowing smoother transitions with finer quality adaptation and size differences across LODs. Additionally, they allow rendering engines to dynamically adjust the rendering quality based on real-time bandwidth and compute resource availability without needing the viewer to select the LOD.

In this paper, we develop a method to achieve continuous levels of detail for neural representations, focusing on light field networks. In summary, our light field networks with continuous LODs have the following benefits:

- Continuous LODs allow us to smoothly transition across LODs without popping artifacts,
- In streaming scenarios, the LOD can be increased or decreased by downloading only one additional row and column of weights per layer, allowing lower latency transitions with smoother network patterns, and
- Importance sampling allows LFNs to focus their capacity on the most salient regions of the light field, allowing details to resolve at lower LODs.

2 Background and Related Works

In this section, we provide some background on general neural fields and more specifically light field networks. We then overview recent methods for achieving multiple levels of detail with neural fields. For a general overview of neural rendering, we refer readers to Tewari *et al*. [44].

2.1 Implicit Neural Representations

Coordinate-based neural networks have been used to encode various signals such as images [39], signed distance functions [31], radiance fields [1, 3, 29], and light fields [12, 40]. These neural network-based models are often referred to as implicit neural representations

or neural fields. Among these representations, neural radiance fields (NeRFs) and light field networks (LFNs) are both able to represent colored 3D scenes with view-dependant appearance effects.

Neural radiance fields (NeRFs) [29] employ differentiable volume rendering to encode a 3D scene into a multi-layer perceptron (MLP) neural network. By learning the density and color of the scene and using a positional encoding, NeRF can perform high-quality view synthesis, rendering the scene from arbitrary camera positions, while maintaining a very compact representation. However, the original NeRF implementation has many drawbacks, such as slow rendering times, which has limited its practicality. With an incredible amount of interest in neural rendering, many follow-up works have been proposed to improve NeRFs with better rendering performance [9, 35, 46], better quality [1], generalizability [17], and deformations [32, 33, 34]. Additionally, feature grid methods [30, 46] enable learning scenes in seconds and rendering in real-time. Importance sampling [48] can achieve faster learning with fewer training rays.

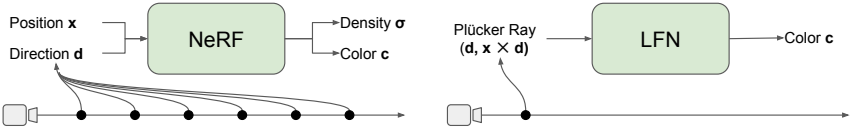


Figure 2: LFNs directly predict the RGB color for each ray in a single inference using Plücker coordinates, avoiding the dozens to hundreds of inferences required by NeRFs.

Light Field Networks Light Field Networks (LFNs) [4, 5, 12, 27, 40] encode light fields [16, 23] by directly learning the 4D variant of the plenoptic function for a scene. Specifically, LFNs directly predict the emitted color for a ray which eliminates the need for volume rendering, making light fields much faster to render compared to other neural fields. Earlier work in light field networks focus on forward-facing scenes using the common two-plane parameterization for light fields. SIGNET [12, 13] uses Gegenbauer polynomials to encode light field images and videos. NeuLF [27] proposes adding a depth branch to encode light fields from a sparser set of images. Plücker coordinates have been used [15, 40] to represent 360-degree light fields.

2.2 Levels of Detail

Several methods have been proposed for neural representations with multiple levels of detail. NGLOD [41] encode signed distance functions into a multi-resolution octree of feature vectors. VQAD [42] adds vector quantization with a feature codebook and presents results on NeRFs. BACON [28] encodes LODs with different Fourier spectrums for images and radiance fields. PINs [22] develop a progressive Fourier feature encoding to improve reconstruction and provide progressive LODs. MINER [36] trains neural networks to learn regions within each scale of a Laplacian pyramid representation. Streamable Neural Fields [7] propose growing neural networks to represent increasing spectral, spatial, or temporal sizes. Progressive Multi-Scale Light Field Networks [24] train a light field network to encode light fields at multiple resolutions.

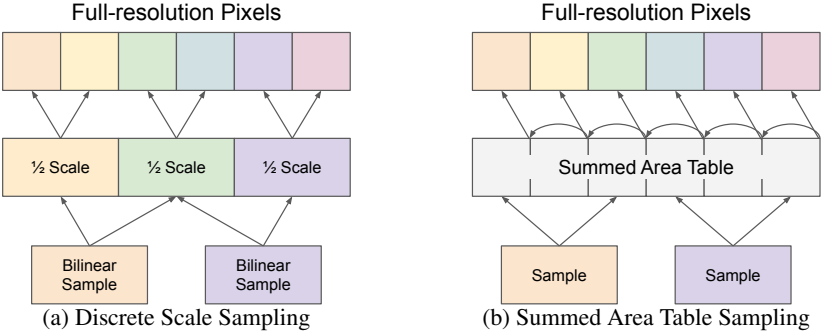
To generate arbitrary intermediate LODs, existing methods blend outputs across discrete LODs. With only a few LODs, the performance does not scale smoothly since the next dis-

crete LOD must be computed entirely. Our method offers continuous LODs with hundreds of performance levels allowing for finer adaptation to resource limitations.

3 Method

Our method primarily builds upon *Light Field Networks* (LFNs) [40]. Specifically, we represent rays $\mathbf{r}$ in Plücker coordinates $(\mathbf{r}_d, \mathbf{r}_o \times \mathbf{r}_d)$ which are input to a multi-layer perceptron (MLP) neural network without any positional encoding. The MLP directly predicts RGBA color values without any volume rendering or other accumulation. Each light field network is trained to overfit a single static scene.

3.1 Arbitrary-scale Arbitrary-position Sampling with Summed Area Tables



(a) Discrete Scale Sampling (b) Summed Area Table Sampling

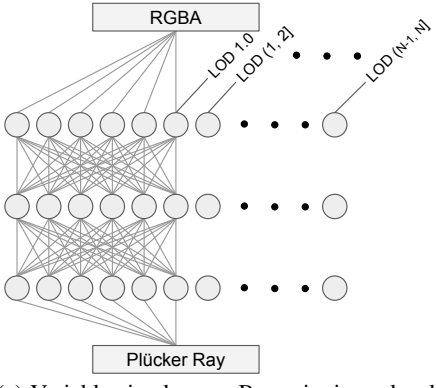
Figure 3: An illustration of discrete and summed-area table sampling. (a) Sampling from a discrete resolution requires linear interpolation from a downsampled image to the target scale and position. (b) Summed area tables allow us to sample at both arbitrary scales and positions without significant additional memory or compute.

In order to reduce aliasing and flickering artifacts when rendering at smaller resolutions, e.g. when an object is far away from the user, lower levels of details need to be processed with filtering to the appropriate resolution. In prior work, multi-scale LFNs [24] are trained on images resized to $1/2$, $1/4$, and $1/8$ scale using area downsampling. During training, rays are sampled from the full-resolution image while colors are sampled from lower-resolution images using bilinear sampling. While training on lower-resolution light fields yields multi-scale light field networks, the bilinear subsampling of the light field may not provide accurate filtered colors for intermediate positions. As shown in Figure 3, colors for higher-resolution rays get averaged over a larger area when performing bilinear subsampling in between low-resolution pixels.

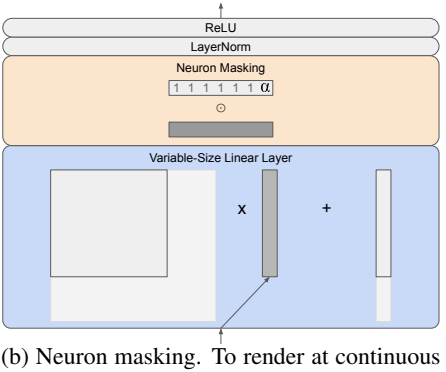
Another method for generating multi-scale light fields is to apply to filter at full resolution to get a spatially accurate anti-aliased sample for each pixel location. Naively precomputing and caching full-resolution copies of each light field image at each scale would significantly increase memory usage. Computing the average pixel color for each sampled ray at training time would require additional computation. Summed area tables [8, 26] can be used to efficiently sample pixels at arbitrary scales and positions, allowing us to sample from filtered

versions of the training image without caching multiple copies. Sampling from a summed area table is a constant time operation, giving us an average over any axis-aligned rectangular region with only four samples. With additional samples, summed-area tables can also be used to apply higher-order polynomial (e.g. cubic) filters [18, 19] or Gaussian filters [20] for even better anti-aliasing, though we only use box filtering in our implementation.

3.2 Continuous Levels of Detail



(a) Variable-size layers. By assigning a level of detail to every network width, we can achieve hundreds of performance levels.



(b) Neuron masking. To render at continuous LOD ℓ, we use weights corresponding to LOD $\lceil \ell \rceil$ and apply alpha-blending to continuously fade-in features produced by the new neuron with $\alpha = \ell - \lfloor \ell \rfloor$ at each layer.

Figure 4: Illustrations of our method to achieve continuous levels of detail.

While previous neural field methods offer static levels of detail corresponding to fixed scales [6, 24, 47] or fixed spectral frequency bands [7, 28], our goal is to generate a finer progression with continuous levels of detail. Continuous levels of detail enable smoother transitions and more precise adaptation to resolution and resource requirements.

Following existing work [1, 24, 47], we encode levels of detail using different widths of a single multi-layer perception neural network. Unlike Mip-NeRF [1, 2], this enables optimized performance with smaller neural networks at lower levels of detail. However, for continuous levels of detail, we propose two changes. First, we map the desired level of detail to every available width to extend a few levels of detail to hundreds of levels of detail as shown in Figure 4a. Second, we propose neuron masking which fades in new neurons to enable true continuous quality adjustments.

LOD to Scale Mapping Li *et al.* [24] train multi-scale LFNs which use width factors 1/4, 2/4, 3/4, and 4/4 (128, 256, 384, 512 widths) to encode 1/8, 1/4, 1/2, and 1/1 scale light fields respectively. To extend this to arbitrary widths, we formulate the following equations which describe the correspondence between network width w and light field scale s:

$$s = 2\hat{}(4w - 4) \tag{1}$$

$$w = (1/4) * (\log_2(s) + 4) \tag{2}$$

By using the above equations, we can assign a unique scale to each width sub-network in our multi-scale light field network. Since this is a one-to-one invertible mapping, we can

also compute the ideal level of detail to use for rendering at any arbitrary resolution. In our experiments, we use a minimum width of 25% of nodes corresponding to a scale of $1/8$ to ensure a reasonable minimum quality and training image size. As an example, for a network with 512-width hidden layers, the lowest level of detail uses only 128 neurons of each hidden layer while the highest uses 512.

Neuron Masking Since neural networks have discrete widths, it is necessary to map continuous levels of detail to discrete widths. Hence, we propose to use neuron masking to provide true continuous levels of detail with discrete-sized neural networks. As weights corresponding to each new width become available, we propose to apply alpha-blending on neurons corresponding to the width. This alpha-blending enables features from existing neurons to continuously transition, representing any intermediate level of detail between the discrete widths. Given feature $\mathbf{f}$ and fractional LOD $\alpha = l - \lfloor l \rfloor$, the new feature $\mathbf{f}'$ with neuron masking is the element-wise product:

$$\mathbf{f}' = (1, ..., 1, \alpha)^\top \odot \mathbf{f} \tag{3}$$

3.3 Saliency-based Importance Sampling

With continuous LODs representing light fields at various scales, the capacity of the LFN is constrained at lower LODs. Hence, details such as facial features may only resolve at higher levels of detail. To maximize the apparent fidelity, the capacity of the network should be distributed towards the most salient regions, *i.e.* the areas where viewers are most likely to focus. We propose to use saliency-based importance sampling which focuses training on salient regions of the light field. For all foreground pixels, we assign a base sampling weight λ_f and add a weight of $\lambda_s * s$ based on the pixel saliency s. Specifically, for a given foreground pixel x in a training image with saliency s, we sample from the probability density:

$$p(x) = \lambda_f + \lambda_s * s \tag{4}$$

In our experiments, we use $(\lambda_f, \lambda_s) = (0.4, 0.6)$ which yields reasonable results. At each iteration, we sample 67% of rays in each batch from foreground pixels using the above density. The remaining 33% of rays are uniformly sampled from background pixels.

4 Experiments

We conduct several experiments to evaluate whether our light field networks with continuous LODs overcome the problems with discrete LODs. We also conduct quality and performance evaluations to determine the compute and bandwidth overhead associated with continuous LODs.

4.1 Experimental Setup

We conduct our experiments using five light field datasets. Scenes are captured using 240 cameras with 40×6 layout around the scene and a 4032×3040 resolution per camera. Each dataset includes camera parameters extracted using COLMAP [37, 38] and is processed with background matting. Of the 240 images, we use 216 for training, 12 for validation, and 12

for testing. We generate saliency maps using the mit1003 pretrained network[1] of Kroner *et al.* [21].

For our model, we use an MLP with nine hidden layers and one output layer. Each hidden layer uses LayerNorm and ReLU. We use a minimum width of 128 and a maximum width of 512 for variable-size layers. Our models are trained using a squared L2 loss for the RGBA color with 8192 rays per batch. In all of our experiments, we train using the Adam optimizer with the learning rate set to 0.001 and exponentially decayed by $\gamma = 0.98$ after each epoch. We train for 100 epochs. Each of our models is trained using a single NVIDIA RTX 2080 Ti GPU. Our PyTorch implementation and processed datasets are available at https://augmentariumlab.github.io/continuous-lfn/.

4.2 Ablation Experiments

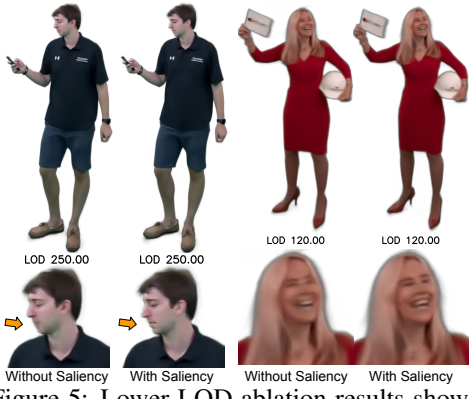


Figure 5: Lower LOD ablation results show the effects of our saliency-based importance sampling. With importance sampling, features such as eyes and mouths in the salient regions resolve at earlier, lower LODs.

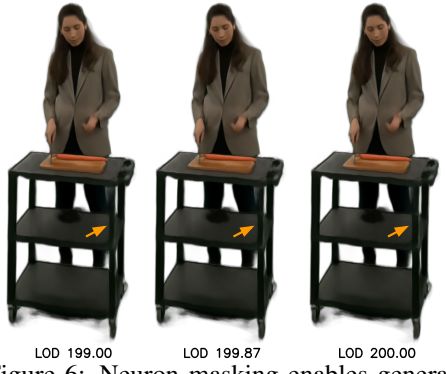


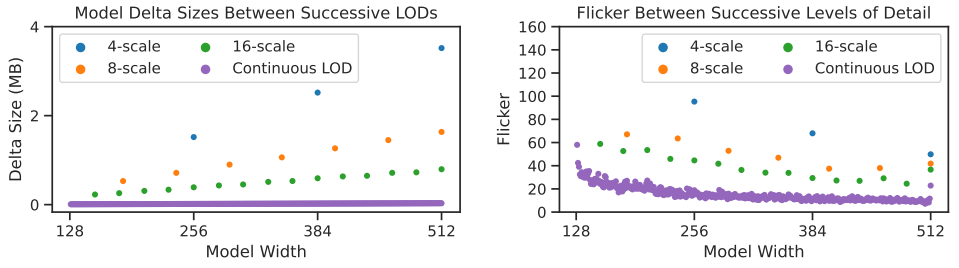
Figure 6: Neuron masking enables generating continuous levels of detail from discrete model widths. In the figure above, we see that one leg of the cart expands as the fractional level of detail α fades in new neurons. This effect is better seen in the accompanying supplementary video.

Our ablation experiments evaluate how each aspect of our method affects the final rendered quality. First, we replace the discrete resolution sampling in discrete-scale light field networks [24] with our summed area table sampling. Next, we add continuous LODs training which is enabled by arbitrary-scale filtering with summed-area tables. Finally, we compare the prior two setups with our full method which also includes saliency-based importance sampling.

4.3 Transitions across LODs

With continuous LODs, our method allows smooth transitions across LODs as additional bytes are streamed over the network or as the viewer approaches the subject. To quantitatively evaluate the smoothness of the transitions, we use the reference-based temporal flicker metric of Winkler *et al.* [45]. This flicker metric first computes the difference d between the

[1] From https://github.com/alexanderkroner/saliency



(a) Model Delta Sizes (b) Average Flicker

Figure 7: Plots showing the effects of transitioning across LODs. Transitioning with discrete LODs leads to larger network traffic spikes and more flickering.

processed images and reference images for two consecutive frames. Next, a difference image $c = d_n - d_{n-1}$ is computed across consecutive frames. The 2D discrete Fourier transform of the image c is computed and values are summed based on the radial frequency spectrum into low and high-frequency sums: s_L and s_H. Finally, the flicker metric is computed by adding these together: Flicker $= s_L + s_H$.

We compare against three discrete-scale baselines with 4, 8, and 16 levels of detail, with 8 and 16 LODs trained using summed-area table sampling. In our continuous LOD case, we render views at the highest LOD corresponding to each discrete width (i.e. LOD 1.0, 2.0, ..., 385.0), using the static ground truth view as the reference frames. Flicker values are computed for each LOD using the transition from the next lower LOD and then averaged across all test views. Our flicker results are shown in Figure 7b. With only four LODs, the discrete-scale LFN method has three transitions, each with large model deltas (up to 3.5 MB) and high flicker values. Additional levels of detail reduce the model delta sizes and the flicker values with our continuous LOD method minimizing the model delta sizes and the flicker values. With our method, the LOD can be transitioned in small ($\leq$ 32 KB) gradual steps.

4.4 Rendering Quality

Model	1/8	1/4	1/2	1/1
Discrete-scale LFN	29.34	29.68	28.39	27.41
+ SAT filtering	29.77	29.99	28.54	27.46
+ Continuous LOD	27.87	29.77	28.38	27.38
+ Importance Sampling	28.06	29.79	28.44	27.40

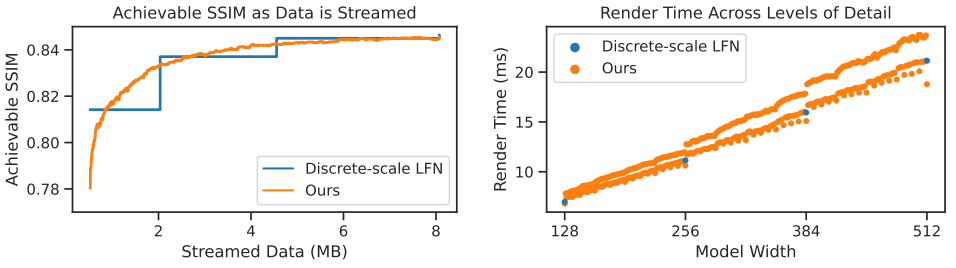
(a) PSNR (dB)

Model	1/8	1/4	1/2	1/1
Discrete-scale LFN	0.8809	0.8763	0.8503	0.8465
+ SAT filtering	0.8898	0.8806	0.8513	0.8466
+ Continuous LOD	0.8370	0.8743	0.8484	0.8460
+ Importance Sampling	0.8380	0.8751	0.8487	0.8455

(b) SSIM

Table 1: Quantitative Training Ablation Results at 1/8, 1/4, 1/2, and 1/1 scales. Each scale is evaluated at its corresponding LOD.

Quantitative PSNR and SSIM results are shown in Table 1. First, we see that adding summed-area table filtering to discrete-scale light field networks with four scales results in slightly improved PSNR and SSIM results while enabling arbitrary-scale sampling. Training a continuous LOD network impacts the performance at the original four LODs but allows us to have continuous LODs. Adding importance sampling allows us to focus on salient regions without significantly impacting the quantitative results.



(a) Average SSIM quality at the full resolution.

(b) Average render times at $1/4$ scale.

Figure 8: Plots showing our quantitative evaluation results. With continuous LODs, the LOD can be dynamically adjusted to maximize the quality based on available resources.

Qualitative results of our saliency-based importance sampling ablation are shown in Figure 5. We see that details along faces appear at earlier LODs when using saliency for importance sampling. All of these details resolve at the highest LODs with and without using importance sampling.

4.5 Rendering Performance

We evaluate the rendering performance by rendering training views across each LOD. For our rendering benchmarks, we use half-precision inference and skip empty rays with the auxiliary network which evaluates ray occupancy. Rendering performance results across the LODs are shown in Figure 8b. We observe that as the LOD increases according to the width of the neural network, rendering times increase as well. When rendering from a discrete-scale light field network with only four LODs, the user or application would need to select either the next higher or lower LOD, compromising on either the performance or the quality. With continuous LODs, software incorporating our light field networks would be able to gradually increase or decrease the LOD to maintain a better balance between performance and quality. In cases where the ideal model size is not known, continuous LODs allow dynamic adjusting of the LOD to satisfy a target frame rate. In our PyTorch implementation, we observe that LODs with odd model widths have a slower render time than LODs with even model widths. LODs with model widths that are a multiple of eight perform slightly faster than other even model widths.

5 Discussion

By requiring light field networks to output reasonable results at each possible hidden layer width and incorporating neuron masking, we can achieve continuous of LODs. However, this applies additional constraints on the network as it needs to produce additional outputs. In our experiments, we observe slightly worse PSNR and SSIM results at the specific LODs corresponding to the $1/8$ and $1/4$ scales compared to the discrete-scale LFN which is trained with only four LODs. This is expected due to the additional constraints and less supervision at those specific LODs. The goal of our importance sampling procedure is to improve the quality of the salient regions of the light field rather than to maximize quantitative results.

Light field networks require additional cameras compared to neural radiance fields due to the lack of multi-view consistency prior provided by volume rendering. Hence, training light field networks requires additional cameras or regularization [14] compared to NeRF methods. Furthermore, light field networks do not use positional encoding [43] and represent high-frequency details as faithfully as NeRF methods. As the primary goal of our work is to enable highly granular rendering trade-offs with more levels of detail, we leave these limitations to future work.

6 Conclusion

In this paper, we introduce continuous levels of details for light field networks using three techniques. First, we introduce summed area table sampling to sample colors from arbitrary scales of an image without generating multiple versions of each training image in a light field. Second, we achieve continuous LODs by combining arbitrary-width networks with neuron masking. Third, we train using saliency-based importance sampling to help details in the salient regions of the light field resolve at earlier LODs. With our method for continuous LODs, we hope to make light field networks more practical for 6DoF desktop and virtual reality applications [10, 11, 25].

Acknowledgments

We would like to thank Jon Heagerty, Sida Li, and Barbara Brawn for developing our light field datasets as well as the anonymous reviewers for the valuable comments on the manuscript. This work has been supported in part by the NSF Grants 18-23321, 21-37229, and 22-35050 and the State of Maryland's MPower initiative. Any opinions, findings, conclusions, or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the research sponsors.

References

[1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields, 2021.

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5460–5469, 2022. doi: 10.1109/CVPR52688.2022.00539.

[3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-NeRF: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023.

[4] Junli Cao, Huan Wang, Pavlo Chemerys, Vladislav Shakhrai, Ju Hu, Yun Fu, Denys Makoviichuk, Sergey Tulyakov, and Jian Ren. Real-time neural light field on mobile devices, 2022. URL https://arxiv.org/abs/2212.08057.

[5] Paramanand Chandramouli, Hendrik Sommerhoff, and Andreas Kolb. Light field implicit representation for flexible resolution reconstruction, 2021. URL https://arxiv.org/abs/2112.00185.

[6] Zhang Chen, Yinda Zhang, Kyle Genova, Sean Fanello, Sofien Bouaziz, Christian Häne, Ruofei Du, Cem Keskin, Thomas Funkhouser, and Danhang Tang. Multiresolution deep implicit functions for 3D shape representation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13067–13076, 2021. doi: 10.1109/ICCV48922.2021.01284.

[7] Junwoo Cho, Seungtae Nam, Daniel Rho, Jong Hwan Ko, and Eunbyung Park. Streamable neural fields. In *Computer Vision – ECCV 2022*, pages 595–612, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20044-1.

[8] Franklin C. Crow. Summed-area tables for texture mapping. *SIGGRAPH Comput. Graph.*, 18(3):207–212, jan 1984. ISSN 0097-8930. doi: 10.1145/964965.808600. URL https://doi.org/10.1145/964965.808600.

[9] Nianchen Deng, Zhenyi He, Jiannan Ye, Budmonde Duinkharjav, Praneeth Chakravarthula, Xubo Yang, and Qi Sun. FoV-NeRF: Foveated neural radiance fields for virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2022. doi: 10.1109/TVCG.2022.3203102.

[10] Ruofei Du, David Li, and Amitabh Varshney. Geollery: A mixed reality social media platform. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. doi: 10.1145/3290605.3300915. URL https://doi.org/10.1145/3290605.3300915.

[11] Ruofei Du, David Li, and Amitabh Varshney. Project geollery.com: Reconstructing a live mirrored world with geotagged social media. In *The 24th International Conference on 3D Web Technology*, Web3D '19, page 1–9, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367981. doi: 10.1145/3329714.3338126. URL https://doi.org/10.1145/3329714.3338126.

[12] Brandon Yushan Feng and Amitabh Varshney. SIGNET: Efficient neural representation for light fields. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14204–14213, 2021. doi: 10.1109/ICCV48922.2021.01396.

[13] Brandon Yushan Feng and Amitabh Varshney. Neural subspaces for light fields. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2022. doi: 10.1109/TVCG.2022.3224674.

[14] Brandon Yushan Feng, Susmija Jabbireddy, and Amitabh Varshney. VIINTER: View interpolation with implicit neural representations of images. In *SIGGRAPH Asia 2022 Conference Papers*, SA '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450394703. doi: 10.1145/3550469.3555417. URL https://doi.org/10.1145/3550469.3555417.

[15] Brandon Yushan Feng, Yinda Zhang, Danhang Tang, Ruofei Du, and Amitabh Varshney. PRIF: Primary ray-based implicit function. In *European Conference on Computer*

Vision, pages 138–155. Springer, 2022. doi: 10.1007/978-3-031-20062-5_9. URL https://doi.org/10.1007%2F978-3-031-20062-5_9.

[16] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 43–54, New York, NY, USA, 1996. Association for Computing Machinery. ISBN 0897917464. doi: 10.1145/237170.237200. URL https://doi.org/10.1145/237170.237200.

[17] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleNeRF: A style-based 3d aware generator for high-resolution image synthesis. In *International Conference on Learning Representations*, 2022.

[18] Paul S. Heckbert. Filtering by repeated integration. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, page 315–321, New York, NY, USA, 1986. Association for Computing Machinery. ISBN 0897911962. doi: 10.1145/15922.15921. URL https://doi.org/10.1145/15922.15921.

[19] Justin Hensley, Thorsten Scheuermann, Greg Coombe, Montek Singh, and Anselmo Lastra. Fast summed-area table generation and its applications. *Computer Graphics Forum*, 24(3):547–555, 2005. doi: https://doi.org/10.1111/j.1467-8659.2005.00880.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2005.00880.x.

[20] Peter Kovesi. Fast almost-gaussian filtering. In *2010 International Conference on Digital Image Computing: Techniques and Applications*, pages 121–125, 2010. doi: 10.1109/DICTA.2010.30.

[21] Alexander Kroner, Mario Senden, Kurt Driessens, and Rainer Goebel. Contextual encoder-decoder network for visual saliency prediction. *Neural Networks*, 129: 261–270, 2020. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2020.05.004. URL http://www.sciencedirect.com/science/article/pii/S0893608020301660.

[22] Zoe Landgraf, Alexander Sorkine Hornung, and Ricardo S Cabral. PINs: Progressive implicit networks for multi-scale neural representations. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 11969–11984. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/landgraf22a.html.

[23] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 31–42, New York, NY, USA, 1996. Association for Computing Machinery. ISBN 0897917464. doi: 10.1145/237170.237199. URL https://doi.org/10.1145/237170.237199.

[24] David Li and Amitabh Varshney. Progressive multi-scale light field networks. In *2022 International Conference on 3D Vision (3DV)*, pages 231–241, 2022. doi: 10.1109/3DV57658.2022.00035.

[25] David Li, Eric Lee, Elijah Schwelling, Mason G. Quick, Patrick Meyers, Ruofei Du, and Amitabh Varshney. Meteovis: Visualizing meteorological events in virtual reality. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, page 1–9, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368193. doi: 10.1145/3334480.3382921. URL https://doi.org/10.1145/3334480.3382921.

[26] David Li, Ruofei Du, Adharsh Babu, Camelia D. Brumar, and Amitabh Varshney. A log-rectilinear transformation for foveated 360-degree video streaming. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2638–2647, 2021. doi: 10.1109/TVCG.2021.3067762.

[27] Zhong Li, Liangchen Song, Celong Liu, Junsong Yuan, and Yi Xu. NeuLF: Efficient Novel View Synthesis with Neural 4D Light Field. In Abhijeet Ghosh and Li-Yi Wei, editors, *Eurographics Symposium on Rendering*. The Eurographics Association, 2022. ISBN 978-3-03868-187-8. doi: 10.2312/sr.20221156.

[28] David B. Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16231–16241, 2022. doi: 10.1109/CVPR52688.2022.01577.

[29] 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 *ECCV*, 2020. doi: 10.1007/978-3-030-58452-8_24.

[30] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4), jul 2022. ISSN 0730-0301. doi: 10.1145/3528223.3530127. URL https://doi.org/10.1145/3528223.3530127.

[31] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. doi: 10.1109/CVPR.2019.00025.

[32] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5845–5854, 2021. doi: 10.1109/ICCV48922.2021.00581.

[33] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. HyperNeRF: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6), dec 2021. ISSN 0730-0301. doi: 10.1145/3478513.3480487. URL https://doi.org/10.1145/3478513.3480487.

[34] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10313–10322, 2021. doi: 10.1109/CVPR46437.2021.01018.

[35] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. KiloNeRF: Speeding up neural radiance fields with thousands of tiny mlps. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14315–14325, 2021. doi: 10.1109/ICCV48922.2021.01407.

[36] Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G. Baraniuk, and Ashok Veeraraghavan. MINER: multiscale implicit neural representations. *CoRR*, abs/2202.03532, 2022. URL https://arxiv.org/abs/2202.03532.

[37] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. doi: 10.1109/CVPR.2016.445.

[38] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. doi: 10.1007/978-3-319-46487-9_31.

[39] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[40] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 19313–19325. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/a11ce019e96a4c60832eadd755a17a58-Paper.pdf.

[41] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11353–11362, 2021. doi: 10.1109/CVPR46437.2021.01120.

[42] Towaki Takikawa, Alex Evans, Jonathan Tremblay, Thomas Müller, Morgan McGuire, Alec Jacobson, and Sanja Fidler. Variable bitrate neural fields. In *ACM SIGGRAPH 2022 Conference Proceedings*, SIGGRAPH '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393379. doi: 10.1145/3528233.3530727. URL https://doi.org/10.1145/3528233.3530727.

[43] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/55053683268957697aa39fba6f231c68-Paper.pdf.

[44] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi, T. Simon, C. Theobalt, M. Nießner, J. T. Barron, G. Wetzstein, M. Zollhöfer, and V. Golyanik. Advances in neural rendering. *Computer Graphics Forum*, 41(2):703–735, 2022. doi: https://doi.org/10.1111/cgf.14507. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14507.

[45] Stefan Winkler, Elisa Drelie Gelasca, and Touradj Ebrahimi. Toward perceptual metrics for video watermark evaluation. In Andrew G. Tescher, editor, *Applications of Digital Image Processing XXVI*, volume 5203, pages 371 – 378. International Society for Optics and Photonics, SPIE, 2003. doi: 10.1117/12.512550. URL https://doi.org/10.1117/12.512550.

[46] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks, 2021.

[47] Jiahui Yu and Thomas Huang. Universally slimmable networks and improved training techniques. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1803–1811, 2019. doi: 10.1109/ICCV.2019.00189.

[48] Wenyuan Zhang, Ruofan Xing, Yunfan Zeng, Yu-Shen Liu, Kanle Shi, and Zhizhong Han. Fast learning radiance fields by shooting much fewer rays. *arXiv preprint arXiv:2208.06821*, 2022.