Exploration of Capabilities and Limitations with View Changes in the X-Fields Model

Jack Wang, Groton School, Groton, MA

Mentor: Guillaume Leclerc, MIT, Cambridge, MA

# Introduction

## Motivation

Generating images of the same scenes from different perspectives—whether that is from different points (video), from different angles (light fields), under varying illumination (reflectance fields), or with other parameters—has a myriad of use cases stretching from creating debug models to smooth render videos.

[Bemana 20] proposes an approach to interpolation across view, light and time for any set of images, a process built on parameters defined as part of an "X-Field" (where X may be any combination of view, light, time, or other parameters). Their research shows how the right neural network (NN) can be used to create a universal and compact X-Fields representation.

Although X-Fields performs well on the given example datasets relative to alternative interpolation methods, the current X-Fields model struggles in adapting to broader use cases. Presently, compatible X-Fields training datasets contain very specific training examples that are not easy to capture in real-world situations and involve an unnecessarily complicated file-naming process. I seek to make X-Fields compatible with more data and simplify the user experience. Also, while X-Fields are shown to be effective for interpolating viewpoints spread out on an XY plane, I add another dimension (Z) to the model that allows interpolation in 3D space.

## Background

In this section, I introduce the X-Fields model proposed by [Bemana 20] and the debugging framework for computer vision models (3DB) [Leclerc 21]. I extend the X-Fields model by adding another coordinate feature and test the performance of the generalized program using samples generated from the 3DB framework.

In the X-Fields model, [Bemana 20] proposes an approach to represent an X-Field of the same scene from different views (video), from various angles (light fields), under varied illumination (reflectance fields), and under a variety of other conditions that allow users to explore view, light, and time changes by learning a neural network (NN) mapping coordinates to 2D images. By knowing the “basic tricks” of graphics (e.g., lighting, 3D projection, and occlusion) in a hard-coded and differentiable form, the X-Fields NN encodes the input as an implicit map such that for any view, light, or time coordinate, it can quantify how it will move if view, time or light coordinates change for any pixel.

The 3DB model is aimed to identify and evaluate the failure modes in computer vision models. It leverages a 3D simulator to render realistic scenes that can be fed into any computer vision system. Users can customize a set of transformations to apply to the scene, such as viewpoint positions, background changes, etc. The 3DB framework is used to generate training and testing datasets by specifying a set of pose changes on a predefined object.

## Related work

This section summaries the previous techniques used to interpolate discrete images in terms of view. View interpolation also refers to the concept of light fields (LFs), which is the set of all images of a scene for all views. [Levoy 96] and [Gortler 96] first formalize the concept of light fields and develop the hardware to capture the views by generalizing from observers’ positions and orientations. Results show that simple linear blending for view interpolations may lead to ghosting effects.

The distance between captures/viewpoints can influence interpolation results. Sparse captures mean less number of images with greater distinction among neighboring images. Examples of sparse capture include 34 views on a sphere [Lombardi 19] and 40 ones on a hemisphere [Malzbender 01]. On the other hand, dense captures refer to a larger number of similar or concentrated images. [Kalantari 16] applies dense images for a Lytro camera.

Early on, Unstructured Lumigraph Rendering (ULR) was used to create proxy geometry to warp multiple images into a target view and blend them with corresponding weights [Buehler 02]. Recent work includes per-view geometry [Hedman 16] and learned ULR blending weights [Hedman 18] to allow for sparse input and shade effects.

Later, researchers focused on learning synthesized novel views for LF data. [Kalantari 16] learned depth maps in an unsupervised way and interpolated views via a Lytro camera. [Flynn 16] decomposed LFs into multiple depth planes of output views and built a plan sweep volume (PSV) mechanism with dependent views. [Zou 18] constructed multi-plane image (MPI) representations via learning how neighboring views can impact the output one.

Another attractive idea has been to use a volumetric occupancy representation. [Penner 17] inferred a good volumetric/MPI representation that can be facilitated with learned gradient descent, where the gradient components directly encode visibility and effectively inform the NN of occlusion relations in the scene. The advantage of MPI techniques is in avoiding explicit depth reconstruction, which allows for softer and better results. The X-Fields model involves a learning route as well but explains the entire X-Field and uses a NN to represent the scene implicitly. Deployment only requires a few additional kilobytes of NN parameters on top of the input image data, and rendering is real-time.

[Zou 18]’s work on Appearance Flow combines the idea of warping pixels with learning how to warp. They typically take a single input view in account and [Sun 18] utilizes multiple views to improve warped view quality. Both methods use an implicit representation of the warp field   
(i.e. a NN in which every pixel in one view predicts where to replicate the value of the pixel in the new view). While those approaches worked best for training with fixed camera positions, [Chen 19] introduced an implicit NN of per-pixel depth that allows for variable view interpolations. All these methods require extensive training for certain types of scenarios such as cars, chairs, or urban city views. The X-Fields model expands on this line of work further by creating an implicit NN representation that generalizes entire geometry, motion, and illumination changes. Its task is on one hand simpler since it does not generalize across different scenes yet is at the same time more difficult as it generalizes over more dimensions and strives for state-of-the-art visual quality.

# Proposed Method

In my model, the original X-Fields model is improved by generalizing the input information of image datasets and adding the 3D view coordinates.

The original X-Fields model derives parameter (view, light, time, etc) values from image file names that implicitly contain the coordinates of view and light and steps in time with a fixed sequence of numbers. It may not be convenient for users to label the image file names manually. My program extracts information from JSON files with a specified format. In addition to being more intuitive, this change makes further generalization with more parameters feasible.

Also, I added another coordinate Z to construct a 3D view interpolation option, expanding on the 2D one from the original model. I made sure that my extension preserves the functionality of the original X-Fields program by manually inputting the coordinates of the example “island” dataset and replicating the resulting interpolation video after setting Z as a dummy variable. After exploring the tugboat dataset containing viewpoints from the surface of an entire sphere surrounding the boat, I found that the interpolation results are relatively poor. This prompted me to perform more controlled experiments with less sparse viewpoints. I generated new image datasets that lessened the view change and separated the background and foreground (in some datasets I replace the default ocean background with pure black). I also vary the resolutions of such datasets to find the resolution’s impact over interpolation and video rendering. In addition, I change different parameters in training—including learning rate, learning stop threshold, sigma and scaling down factor—to study these parameters’ impacts on results.

# Results

I generated datasets containing images of a tugboat from different viewpoints using the 3DB framework. Altering factors like dataset size (dimensions per axis), viewpoint angle range (defined by the surface area on which the viewpoints are spread out, which is in my experiment the surfaces of a whole sphere or the part of the sphere located in the positive octant), image resolution, and backgrounds, I study the parameters’ effect on interpolation quality and computational complexity. In addition to experimenting with 3DB input rendering settings, I investigate how changing X-Fields parameters like output resolution, learning rate, learning stop threshold, sigma, and factor influenced model results and computational demands. Training the model on datasets of 5, 10, and then 15 dimensions per axis consistently increased quality at the tradeoff of greater necessary computational power and memory. When learning rate was reduced, training time increased along with interpolation quality. Larger datasets required more memory, and my attempt to train my revised X-Fields on a high-resolution 20x20x20 dataset failed because the model required more than the 10GB of memory available to it.

The largest 3DB datasets in my experiment include 3375 (15x15x15) images spread out an equal distance in the X, Y, and Z directions. In my experiment, I first tried different learning rate values. As shown in Figure 1, the interpolation results with learning rate 0.00001 (the lowest setting) are best.

Learning rate = 0.005 Learning rate = 0.001



Learning rate = 0.0001 Learning rate = 0.00001



Figure 1: Sample Interpolation With Different Learning Rates

I also tried changing sigma, the bandwidth parameter in soft blending, from 0.1 to 0.5, which improved the interpolation quality. The sample interpolations are provided in Figure 2.

Sigma = 0.1 Sigma = 0.5



Figure 2: Sample Interpolation With Different Values of Sigma

Adding water to the reference frames increased the ghosting effect. One likely reason for this is the highly-variable and unpredictable reflections on the surface of the water. Sample interpolations are shown in Figure 3. Note that one reason for the images’ lower quality overall (including the one featuring the tugboat with the black background) is that these results come from 10x10x10 datasets because using 3DB to generate renders with the water background is very computationally expensive.

Background = Black Background = Ocean



Figure 3: Sample Interpolation With Different Backgrounds

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| dims/axis | learning rate | learning stop threshold | scope | background | input resolution | factor | output resolution  (input res/factor) | sigma | nfg | num\_n |
| 5 | 0.0005 | 0.01 | full | black | 224 | 2 | 128 | 0.1 | 16 | 6 |
| 5 | 0.00001 | 0.01 | positive | black | 512 | 1 | 512 | 0.1 | 16 | 6 |
| 10 | 0.0005 | 0.01 | full | black | 224 | 2 | 128 | 0.1 | 16 | 6 |
| 10 | 0.0005 | 0.01 | positive | black | 224 | 2 | 128 | 0.1 | 16 | 6 |
| 10 | 0.0005 | 0.01 | positive | water | 224 | 2 | 128 | 0.1 | 16 | 6 |
| 15 | 0.005 | 0.001 | positive | black | 2160 | 6 | 360 | 0.1 | 16 | 6 |
| 15 | 0.001 | 0.001 | positive | black | 2160 | 6 | 360 | 0.1 | 16 | 6 |
| 15 | 0.0001 | 0.001 | positive | black | 2160 | 6 | 360 | 0.1 | 16 | 6 |
| 15 | 0.00005 | 0.001 | positive | black | 2160 | 6 | 360 | 0.1 | 16 | 6 |
| 15 | 0.00001 | 0.001 | positive | black | 2160 | 6 | 360 | 0.5 | 16 | 6 |
| 15 | 0.00001 | 0.001 | positive | black | 2160 | 4 | 560 | 0.1 | 16 | 6 |

Table 1: Record of Different Experiments

# Conclusions & Future Plans

In conclusion, by controlling dataset generation and adjusting separate parameters, I have found that the X-Fields model has the following capabilities and limitations:

1. It can generate promising interpolation results with relatively sparse datasets and with large view angle changes.
2. Parameters such as learning rate and the bandwidth parameter in soft blending have impacts over the interpolation quality and construct trade-offs between training cost and interpolation quality.
3. Certain highly-variable backgrounds can pose a challenge for interpolation quality.

Computer vision is a dynamic area of research in which interpolation is an important application. New types of neural networks, such as the Transformer Network [Choi 20][Khan 21][Ye 21], can be combined with CNN to reduce training cost and improve interpolation accuracy. The combination of X-Fields with Transformer Networks is worth further investigation and experimenting. I have generated some sample frames from Choi’s [Choi 20] pretrained CAIN model with the default interpolation ratio of 3:1 as an initial exploration. Comparison of interpolation results with X-Fields with the same reference frames is shown below:

|  |  |
| --- | --- |
|  |  |
| X-Fields | CAIN |

Figure 4: Comparison of X-Fields and CAIN Interpolation Samples

# References

[Bemana 20] Bemana, M., Myszkowski, K., Seidel, H. P., & Ritschel, T. (2020). X-Fields: implicit neural view-, light-and time-image interpolation. ACM Transactions on Graphics (TOG), 39(6), 1-15.

[Buehler 02] Buehler, C., Bosse, M., McMillan, L., Gortler, S., & Cohen, M. (2001, August). Unstructured lumigraph rendering. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (pp. 425-432).

[Chen 19] Chen, X., Song, J., & Hilliges, O. (2019). Monocular neural image based rendering with continuous view control. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 4090-4100).

[Choi 20] Choi, M., Kim, H., Han, B., Xu, N., & Lee, K. M. (2020). Channel Attention Is All You Need for Video Frame Interpolation. Proceedings of the AAAI Conference on Artificial Intelligence, 34(07), 10663-10671. <https://doi.org/10.1609/aaai.v34i07.6693> <https://github.com/myungsub/CAIN>

[Flynn 16] Flynn, J., Neulander, I., Philbin, J., & Snavely, N. (2016). Deepstereo: Learning to predict new views from the world's imagery. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5515-5524).

[Gortler 96] Gortler, S. J., Grzeszczuk, R., Szeliski, R., & Cohen, M. F. (1996, August). The lumigraph. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (pp. 43-54).

[Hedman 16] Hedman, P., Ritschel, T., Drettakis, G., & Brostow, G. (2016). Scalable inside-out image-based rendering. ACM Transactions on Graphics (TOG), 35(6), 1-11.

[Hedman 18] Hedman, P., Philip, J., Price, T., Frahm, J. M., Drettakis, G., & Brostow, G. (2018). Deep blending for free-viewpoint image-based rendering. ACM Transactions on Graphics (TOG), 37(6), 1-15.

[Kalantari 16] Kalantari, N. K., Wang, T. C., & Ramamoorthi, R. (2016). Learning-based view synthesis for light field cameras. ACM Transactions on Graphics (TOG), 35(6), 1-10.

[Khan 21] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah, Transformers in Vision: A Survey, arXiv:2101.01169v2 [cs.CV] 22 Feb 2021.

[Leclerc 21] Leclerc, G., Salman, H., Ilyas, A., Vemprala, S., Engstrom, L., Vineet, V., ... & Madry, A. (2021). 3DB: A Framework for Debugging Computer Vision Models. arXiv preprint arXiv:2106.03805.

[Levoy 96] Levoy, M., & Hanrahan, P. (1996, August). Light field rendering. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (pp. 31-42).

[Lombardi 19] Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., & Sheikh, Y. (2019). Neural volumes: Learning dynamic renderable volumes from images. arXiv preprint arXiv:1906.07751.

[Malzbender 01] Malzbender, T., Gelb, D., & Wolters, H. (2001, August). Polynomial texture maps. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques(pp. 519-528).

[Penner 17] Penner, E., & Zhang, L. (2017). Soft 3D reconstruction for view synthesis. ACM Transactions on Graphics (TOG), 36(6), 1-11.

[Sun 18] Sun, S. H., Huh, M., Liao, Y. H., Zhang, N., & Lim, J. J. (2018). Multi-view to novel view: Synthesizing novel views with self-learned confidence. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 155-171).

[Ye 21] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, Fei Sha, Few-Shot Learning via Embedding Adaptation with Set-to-Set Functions, arXiv:1812.03664v6, 13 Jun 2021.

[Zou 18] Zou, Y., Luo, Z., & Huang, J. B. (2018). Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In Proceedings of the European conference on computer vision (ECCV) (pp. 36-53).