VIINTER: View Interpolation with Implicit Neural Representations of Images

Brandon Yushan Feng
yfeng97@umd.edu
University of Maryland, College Park
USA

Susmija Jabbireddy
jsreddy@umd.edu
University of Maryland, College Park
USA

Amitabh Varshney
varshney@umd.edu
University of Maryland, College Park
USA

ABSTRACT

We present VIINTER, a method for view interpolation by interpolating the implicit neural representation (INR) of the captured images. We leverage the learned code vector associated with each image and interpolate between these codes to achieve viewpoint transitions. We propose several techniques that significantly enhance the interpolation quality. VIINTER signifies a new way to achieve view interpolation without constructing 3D structure, estimating camera poses, or computing pixel correspondence. We validate the effectiveness of VIINTER on several multi-view scenes with different types of camera layout and scene composition. As the development of INR of images (as opposed to surface or volume) has centered around tasks like image fitting and super-resolution, with VIINTER, we show its capability for view interpolation and offer a promising outlook on using INR for image manipulation tasks.

KEYWORDS

implicit neural representation, coordinate network, view synthesis

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ACM Reference Format:
Brandon Yushan Feng, Susmija Jabbireddy, and Amitabh Varshney. 2022. VIINTER: View Interpolation with Implicit Neural Representations of Images. In SIGGRAPH Asia 2022 Conference Papers (SA ’22 Conference Papers), December 6–9, 2022, Daegu, Republic of Korea. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3550469.3555417

1 INTRODUCTION

Neural networks have become a prevalent component in various computational systems over the past decade. For the graphics and vision community, they have been an effective tool in tasks involving visual data, such as recognition, segmentation, and 3D reconstruction. In these classic tasks, neural networks are often deployed as a feature extractor from the input visual signal (e.g., image), but more recently, coordinate network has emerged as a new concept. Instead of extracting features from the signal, the network takes in a coordinate and produces the signal value at that coordinate. Such a network learns a continuous function that maps signal coordinates to values, and it is often referred to as an implicit neural representation (INR) of the signal. INR has led to remarkable success in representing visual signals such as images, videos, signed distance fields, and radiance fields.

In scenarios where only 2D images are available, INR has found two prominent applications. One of them is image fitting, where
INRs are trained to produce the color of each known image pixel. Along this line, much progress has been made to improve the accuracy and speed of fitting INR on images, as well as its ability for compression and super-resolution. The other prominent application is reconstructing 3D scenes from 2D images. Here, INRs produce the attribute values (e.g., radiance and opacity) at each spatial coordinate, which are then differentiably rendered into pixels. In this case, the INRs are optimized such that these rendered pixels reproduce the known image pixels. Once sufficiently trained, these INRs can synthesize plausible novel views outside the training set.

On both fitting and view synthesis, INRs achieve impressive visual results that closely resembled the original 2D images. However, it also appears that the development of INRs has gone into two orthogonal directions. On one hand, the quality of fitting images with INRs is improved by incorporating traditional signal processing techniques like multi-scale subsampling and filtering. On the other hand, the quality of view synthesis is improved by augmenting INRs with well-established 3D graphics techniques, such as spatial subdivision, parametric modeling, and level-set methods.

Although the exciting advancements towards these two directions are rapidly pushing the state of the art, we like to explore a different direction and ask a new question: Given multiple 2D image views of a 3D scene, can we use the INR of those 2D images alone to do view synthesis without any 3D reconstruction, pose, or correspondence? In this paper, driven by this question, we present an initial exploration towards view interpolation with INR of images (VIINTER). With randomly initialized INR weights and code vectors for individual images, we modify the standard INR training process such that the trained INR can both faithfully reproduce the given images and synthesize plausible novel views when we interpolate between those learned image codes.

It is nontrivial to obtain sensible novel views through code interpolation with standard training of INR. We experiment on a range of changes to the training of INR and provide details in Section 3. We present further evaluation results on different types of multi-view scenes in Section 4. Our work takes an important early step toward revealing new potential of INR of images, and we summarize our main contributions as the followings:

- We present a novel approach to view interpolation by interpolating INRs trained to fit 2D images without any knowledge of 3D structure, pose, or correspondence.
- We introduce several modifications to the common process of training image-fitting INRs, which significantly improve the view interpolation quality.
- We show that the proposed non-3D approach achieves smooth and photorealistic interpolation across several scenes with a variety of viewpoint layout and scene content.

2 RELATED WORK

In this section, we review recent work on implicit neural representation, as well as prior techniques for view interpolation.

2.1 Implicit Neural Representations.

Following seminal works [Chen and Zhang 2019; Mescheder et al. 2019; Park et al. 2019] showing successful applications of neural network to encode 3D shapes, many methods have been introduced to solve various vision and graphics tasks using INRs of 3D shapes. These INRs usually use the multilayer perceptron (MLP) architecture to encode geometric information of a 3D shape by learning the mapping from a given 3D spatial point and a scalar value denoting either the signed distance or occupancy.

2.1.1 3D Reconstruction. As differentiable rendering becomes more practical, researchers have succeeded in training INRs to learn, not just fit, the geometry and appearance of a 3D scene based on 2D image observations. The most prominent works is Neural Radiance Fields (NeRF) [Mildenhall et al. 2020], which learns an INR of the view-dependent radiance volume inside a 3D scene and naturally enables view synthesis. The success of NeRF sparked an enthusiastic trend of improving INRs for highly photorealistic view synthesis in terms of their training speed, rendering speed, and rendering quality. A wide range of techniques have been studied and incorporated to 3D INRs, including spatial subdivision or octree [Liu et al. 2020; Yu et al. 2021], parametric modeling with human body shape prior [Liu et al. 2021; Peng et al. 2021], level-set methods for more accuracy geometry [Bergman et al. 2021; Wang et al. 2021a], caching and distillation for faster rendering [Hedman et al. 2021; Yu et al. 2021], camera pose refinement [Lin et al. 2021; Meng et al. 2021; Wang et al. 2021c], and lighting and camera variation during capture to better extract physical attributes [Bi et al. 2020; Zhang et al. 2021]. Convolutional neural networks [Beman et al. 2020; Dosovitskiy et al. 2016; Eslami et al. 2018; Tafarenko et al. 2016] have also been trained to take camera pose as input and produce 2D renderings of simple 3D scenes.

2.1.2 Image Fitting. Images are arguably the most dominant form of visual data, and many efforts on INRs are to make them fit 2D images as accurately and quickly as possible [Müller et al. 2022; Tancik et al. 2021]. ACORN [Martel et al. 2021] applies spatial subdivision to more efficiently train INR of a single gigapixel image (with 1 billion pixels). Various signal processing techniques are also shown useful in making INRs fit images more accurately, such as image pyramids [Saragadam et al. 2022], sinusoidal and Fourier basis functions [Sitzmann et al. 2020; Tancik et al. 2020], and multiplicative filtering with Fourier or Gabor wavelet basis functions [Pathony et al. 2020; Huang et al. 2021; Lindell et al. 2021]. Many of these techniques are applicable for other signals like 3D MRI data or 3D signed distance fields, which are beyond the scope of this paper.

While most of these methods focus on training a single network as INR of a single image, a single network may also serve as INR of multiple images. For consecutive images at a fixed viewpoint, an extra time dimension is added to the coordinate input [Sitzmann et al. 2020]. For structured 4D light fields where the views lie on a 2D plane, those 2D coordinates can be re-parameterized as input [Attal et al. 2022; Feng and Varshney 2021]. A single network can further serve as a generalizable INR of arbitrary images, by concatenating the code with the 2D pixel coordinate as input to the INR [Mehta et al. 2021]. An alternative approach is to modulate network activation based on the code [Dupont et al. 2022; Mehta et al. 2021], which has success in fitting arbitrary image patches. We find the simple code concatenation is sufficient for our problem, and it has been successfully used to train an INR of light rays from different scenes [Feng et al. 2022; Sitzmann et al. 2021]. Unlike modulation, it
We provide details on the INR parametrization adopted in our study, without finding pixel-wise correspondences between images.

2.2 Image-based Rendering.

The early approaches of image-based rendering (IBR) achieve novel view synthesis through explicitly blending relevant pixels from known images [Debevec et al. 1996; Gortler et al. 1996; Levoy and Hanrahan 1996]. The visual quality of IBR is heavily dependent on the strategy of deciding the blending weights of images, and researchers have developed a line of techniques improving blending weights selection, such as ray-space proximity [Chai et al. 2000; Du et al. 2018], soft blending [Penner and Zhang 2017; Riegler and Dyer 1996; Wolberg 1998], but we achieve the morphing effect by 3D reconstruction nor the knowledge of 3D locations and orientations of each image and leverage the spatial relationship among the cameras to decide the blending weights. In contrast, we explore a different and more challenging problem setting which does not involve 3D reconstruction nor the knowledge of 3D locations and camera orientations. Our problem setup is similar to prior work on image morphing [Chen and Williams 1993; Liao et al. 2014; Seitz and Dyer 1996; Wolberg 1998], but we achieve the morphing effect without finding pixel-wise correspondences between images.

3 METHOD

We provide details on the INR parametrization adopted in our study, and we introduce the proposed modifications to INR training.

3.1 INR for Image Fitting

Let $F$ denote the INR of images. In the case of a single image, for all pixels $p$ of the image, the INR $F$ defines

$$F(p_x, p_y) = p_c. \tag{1}$$

where $(p_x, p_y)$ denotes the coordinate of the pixel $p$, with $p_x \in \mathbb{R}$ and $p_y \in \mathbb{R}$. $p_c \in \mathbb{R}^3$ denotes the value (often the RGB vector) associated with the pixel $p$. In itself, the INR formulation is invariant to different numeric ranges of $(p_x, p_y)$ or $p_c$, and for simplicity we rescale the pixel coordinates and values to be within $[0, 1]$.

We adopt the conventional MLP architecture to parameterize $F$ as a chain of fully connected layers, with activation function usually set as a ReLU or sinusoidal function. Various embedding functions of the input coordinate $(x, y)$ have been proposed, but in this work we apply no embedding and use sinusoidal activation [Sitzmann et al. 2020], which are sufficient for fitting single 2D images.

The primary training objective of INR $F$ for single 2D images is to minimize the reconstruction error between the predicted $p_c$ and ground truth $p_c^{GT}$ across all known pixels in a single image, namely

$$L_{SingleRecon} = \sum_p \| p_c - p_c^{GT} \|^2. \tag{2}$$

3.2 Extension to Multiple Images

Our goal is to use a single network $F$ as the INR for multiple images from the same scene. Prior methods assume the camera layout (for planar light fields [Feng and Varshney 2021]) or known camera poses in the pipeline (for general light fields [Attal et al. 2022; Sitzmann et al. 2021]), but we are interested in pushing the limit to where the camera pose of each image is unknown.

In our 3D-agnostic setup which does not consider camera poses, we assign a randomly initialized vector $z \in \mathbb{R}^M$ for each image, which serves as its identity code. We then modify the INR setup...
so that the operation on each pixel coordinate \((p_x, p_y)\) is now conditional on the code \(z\) of length \(M\). In practice, we concatenate \(z\) with \((p_x, p_y)\) to form the input vector to the network. Formally, with \(N\) images and \(n = 1, \ldots, N\),

\[
\mathcal{F}(p_x, p_y \mid z_n) = p_{c|n},
\]

(3)

where \(p_{c|n}\) stands for the predicted value of pixel \(p\) in image \(I_n\).

The training loss function can be easily modified as

\[
L_{\text{Recon}} = \sum_n \sum_p \| p_{c|n} - p_{c|n}^{\text{GT}} \|^2,
\]

(4)

such that the INR \(\mathcal{F}\) fits the pixels among all \(N\) images.

While it is easy to optimize \(\mathcal{F}\) and \(Z_N = \{z_n\}_{n=1}^{N}\) to reach a low \(L_{\text{Recon}}\) across all known pixels, it remains unclear how the learned \(\mathcal{F}\) would perform given a novel \(z \notin Z_N\). Of course, it would be too demanding to expect \(\mathcal{F}\) to always produce sensible results for any random \(z \in \mathbb{R}^M\). Nonetheless, we believe it is fair to inquire \(\mathcal{F}\) under a more relaxed setting: With \(z_i, z_j \in Z_N\), can \(\mathcal{F}\) produce sensible results given a novel \(z_{\text{Inter}} = \alpha z_i + \beta z_j\)? In other words, as \(\mathcal{F}\) is trained to produce good results with \(z_i\) and \(z_j\), what would it produce with a weighted combination of \(z_i\) and \(z_j\)?

### 3.3 Direct Regularization

The reason we are interested in the above problem is that, if \(\mathcal{F}\) would produce good results on weighted combinations of known \(z_i, z_j \in Z_N\), it could produce a smooth transition from image \(I_k\) to image \(I_j\). Effectively, it could achieve view interpolation without using any correspondence point or 3D information.

In this paper, we select the interpolation weights \(\alpha, \beta\) with the simple linear interpolation, inducing \(\alpha = 1 - t, \beta = t\) with \(0 \leq t \leq 1\). Unfortunately, as shown in Fig. 2, linearly interpolating between the two learned codes fails to let \(\mathcal{F}\) produce any meaningful result.

The initial failure is not really a surprise. The codes in \(Z_N\) are optimized only towards minimizing \(L_{\text{Recon}}\). It would make sense for them to end up with different scales so that \(\mathcal{F}\) can better distinguish them and reduce \(L_{\text{Recon}}\). Thus, interpolating between them would likely produce a noise vector which \(\mathcal{F}\) cannot meaningful decode.

To address this issue, we directly regularize the \(Z_N\) during training. In particular, we prevent the codes from having different scales by explicitly enforcing the learnable code \(z\) as unit \(p\)-norm. For our method we select \(p = 1\) and enforce

\[
z = \frac{z}{\|z\|_1}, \forall z \in Z_N
\]

(5)

in addition to training \(\mathcal{F}\) and \(Z_N\) based on \(L_{\text{Recon}}\).

Although a more instinctive option to many people is to rescale \(z\) based on its Euclidean norm (sum of squares), this Euclidean norm is only a special case for the general \(p\)-norm when \(p = 2\), or \(\|z\|_2\). We investigate the effect of varying \(p\) while computing the norm of
Although the unit norm constraint significantly improves visual quality, artifacts are observable as shown in Fig. 4. This shortcoming does not come as a shock, because there must be a limit as to how smoothly $\mathcal{F}$ can interpolate, given how little prior knowledge it has about the content. After all, $\mathcal{F}$ is only trained on a small set of images, and, unlike powerful generative networks, it does not possess domain knowledge (e.g., faces, cars) from a huge dataset.

However, there is still room for improvement if we were proactively alter the training process. Since the fundamental issue is that $\mathcal{F}$ produces poor results when decoding novel interpolated $z_{\text{Inter}}$ unseen during training, we should explicitly encourage good results with interpolated codes during training. As we only have access to those $N$ images and do not have ground truth interpolated frames, we propose computing the loss of interpolated results through an off-the-shelf pre-trained network $E$. Specifically, we hope the features extracted from the interpolated output are similar to the features extracted from the two source images $I_i$ and $I_j$.

Formally, with $z_{\text{Inter}} = (1 - t) \cdot z_i + t \cdot z_j$, $\Delta$ denoting all pixels in a full image frame, the interpolated output image is $I_{\text{Inter}} = \mathcal{F}(\Delta | z_{\text{Inter}})$. We then use feature extractor $E$ to compute

$$L_{\text{Inter}} = \|E(I_{\text{Inter}}) - [(1 - t) \cdot E(I_i) + t \cdot E(I_j)]\|^2,$$

and train the INR towards minimizing this loss.

While this setup is similar to the VGGNet-based perceptual loss widely used in tasks like style transfer and image reconstruction, we find that using VGGNet as the feature extractor $E$ is not adequate for our problem, as shown in Fig. 4. Instead, we employ the recently released CLIP network [Radford et al. 2021] as $E$. As shown by Fig. 4, we discover that the CLIP-based extractor significantly outperforms VGGNet, likely because CLIP benefits from being trained to extract semantically-consistent features from images (see [Radford et al. 2021] for more details), whereas VGGNet is trained to extract features mainly for image classification. Our results are analogous to previous findings [Jain et al. 2021] that show CLIP improves NeRF training on sparse views.

In short, in addition to the original objective of minimizing $L_{\text{Recon}}$, we introduce rescaling with unit norm and the CLIP-guided interpolation loss to the process of training INR of images. To compute $L_{\text{Inter}}$, the interpolation endpoints $z_i$ and $z_j$ are randomly selected from $Z_N$. We refrain from specifically sampling neighboring or adjacent viewpoints to avoid using the camera pose information and keep training as 3D-agnostic. We would only knowingly select

---

**Table 1**: Top: Effect of varying $p$ for code rescaling. Quantitative results reaffirm that rescaling based on 1-norm achieves the best quality. Bottom: Effect of varying the length of code $M$. Results indicate that the code length cannot be arbitrary, as a small $M$ can be detrimental to the interpolation quality. Setting $M$ too large is also not helpful, as the quality appears to peak at $M = 128$. More details about the experimental setting are in Sec. 4.

| $p$-norm (M = 128) | No | $\infty$ | 2 | 1.5 | 1 |
|------------------|----|----------|---|----|---|
| Known            | SSIM   | 0.901    | 0.903 | 0.951 | 0.955 | 0.962 |
|                  | PSNR   | 27.49    | 27.57 | 31.63 | 31.92 | 33.38 |
| Novel            | SSIM   | 0.595    | 0.583 | 0.937 | 0.952 | 0.958 |
|                  | PSNR   | 11.02    | 11.04 | 29.15 | 31.25 | 32.39 |

| $M$ ($p = 1$) | 16 | 32 | 64 | 128 | 256 | 512 |
|--------------|----|----|----|-----|-----|-----|
| Known        | SSIM | 0.953 | 0.958 | 0.961 | 0.962 | 0.961 | 0.960 |
|               | PSNR | 32.64 | 33.03 | 33.37 | 33.38 | 33.16 | 32.95 |
| Novel        | SSIM | 0.891 | 0.951 | 0.958 | 0.958 | 0.958 | 0.956 |
|               | PSNR | 25.27 | 31.37 | 32.56 | 32.49 | 32.39 | 32.10 |

$z$, and we present the results in Table 1 and in Fig. 2. With rescaling based on 1-norm, the interpolation is more natural and stable than with 2-norm. Moreover, as indicated by Table 1, 1-norm even leads to more accurate reconstruction at the known image views, which is the original task of fitting INR of images.

### 3.4 Indirect Regularization

Although the unit norm constraint significantly improves visual quality, artifacts are observable as shown in Fig. 4. This shortcoming does not come as a shock, because there must be a limit as to how smoothly $\mathcal{F}$ can interpolate, given how little prior knowledge it has about the content. After all, $\mathcal{F}$ is only trained on a small set of images, and, unlike powerful generative networks, it does not possess domain knowledge (e.g., faces, cars) from a huge dataset.

Figure 5: Stanford 4D Light Fields Results. We interpolate learned codes of views $i$ and $j$ as $(1 - t) \cdot z_i + t \cdot z_j$. The INR preserves the image details from the known views and smoothly transitions between them, such as bright speckles on the ball (row 1) and view-dependent reflections (rows 2 and 3). Images are zoomed in for easier evaluations.
$t = 0$

$\frac{3}{4}$

$1$

Figure 6: Unstructured Light Field Results. We interpolate learned codes of views $i$ and $j$ as $(1 - t) \cdot z_i + t \cdot z_j$. The camera movement between the two known views includes rotation and translation. The interpolation through INR smoothly transforms the perspective, despite having no knowledge of 3D scene structure or camera pose. Images are zoomed in for easier evaluations.

endpoints based on their viewpoint locations during evaluation or demonstration of the interpolation results between different viewpoints, after training is finished.

4 EXPERIMENTS

In this section, we provide more results on view interpolation and ablation studies on the techniques introduced in Section 3. We train VIINTER to encode real-world scenes captured under two different regimes: 4D light fields (viewpoints are on a 2D plane with the same orientation) and unstructured light fields (viewpoints are not aligned on a 2D grid and orientations might be rotated).

4D Planar Light Fields. We use scenes from Stanford Light Field Archive [Wilburn et al. 2005], with 17 × 17 camera viewpoints on a 2D grid. We use a 5×5 subset by taking every 4-th image horizontally and vertically. We render new views by selecting two trained codes and linearly interpolate them. The resulting interpolation results are shown in Fig. 5, with more in the supplements.

Unstructured Light Fields. To test VIINTER on scenes with irregular camera layout, we test on the LLFF dataset [Mildenhall et al. 2019] and our own volumetric dataset. The LLFF scenes are captured in natural indoor environments, while our own scenes come from a volumetric studio for human body captures. We present the interpolation results in Fig. 5, with more in the supplements.

Quantitative Evaluation. The unique challenge in evaluating our method is we cannot explicitly specify a camera pose to render at. Nonetheless, to provide a quantitative evaluation, we approximately render at testing viewpoints by interpolating the codes from nearby known viewpoints. For example, for the Stanford Light Field scenes, we select two viewpoints in the 5×5 training set, viewpoints (4, 4) and (4, 8), and interpolate their learned codes with $t = 0.5$. Then we render the full image with the interpolated code and compare it against the actual test image (withheld from training) captured at viewpoint (4, 6). Thanks to the well-aligned structure of these 4D scenes, we can compute metrics like peak signal-to-noise ratio...
Table 2: Quantitative results on real-world scenes with different viewpoint layouts. Our method can only render at the approximate viewpoints of ground truth, as discussed in Section 4, leading to lower PSNR and SSIM values for novel views of “Unstructured” where the viewpoint mismatch is severe. See visual results for more comprehensive quality assessments.

| Method   | 4D Planar | Unstructured |
|----------|-----------|--------------|
|          | NeRF | LFN | Ours | NeRF | LFN | Ours |
| Known    | SSIM 0.926 | 0.977 | 0.978 | 0.911 | 0.920 | 0.885 |
| PSNR     | 33.62 | 37.67 | 37.28 | 29.04 | 30.11 | 28.32 |
| Novel*   | SSIM 0.917 | 0.944 | 0.975 | 0.905 | 0.788 | 0.664 |
| PSNR     | 33.28 | 30.67 | 35.77 | 27.15 | 21.35 | 16.80 |

(PSNR) and structural similarity index measure (SSIM) against a reasonable ground truth image.

Table 1 provides the quantitative impact of our proposed techniques on known and novel viewpoints in the Lego scene. In Table 2 we present the evaluation results aggregated from all scenes. Although our method is not meant to outperform methods which use 3D information, we still provide quantitative comparisons with two recent methods: NeRF [Mildenhall et al. 2020], the most prevalent INR method for view synthesis, and LFN [Sitzmann et al. 2021], which uses camera pose information to train an INR of 5D rays. For results at novel viewpoints, although we can reasonably approximate the viewpoint in the 4D Light Field scenes, the approximation is very inaccurate in the Unstructured scenes due to sparse and irregular camera layouts. For more comprehensive assessment of the quality, please refer to the supplements.

5 DISCUSSION

In this section, we provide further discussion on the significance of the proposed method and presented results.

Why not NeRF or 3D approach? In general, image-based methods avoid certain issues unique to 3D approaches (e.g. properly setting 3D bounding box, # of samples per ray) that often complicate the training. However, this paper is not intended to present a better method than the state of the art in view interpolation and synthesis, but rather to explore a new direction where a classic image manipulation problem meets the modern implicit neural representation. We believe that 3D approaches like NeRF are currently still more appropriate in production, due to the abundance of techniques and optimizations developed to improve their performance. For sake of transparency and thoroughness, we provide comparisons in Section 4 with representative methods and datasets, and we hope they help readers better contextualize this new and untested approach.

How is this different than image morphing? Although many image morphing techniques do not invoke explicit 3D knowledge about the structure or viewpoints, they rely on finding correspondence points between the images being interpolated. Our proposed method is correspondence-free, and the interpolation happens in the space of the z codes, rather than the space of image pixels. Moreover, image morphing often applies to images of different scenes or identities (e.g. face morphing between two people), but this paper is concerned with multi-view images from the same scene. In our setting, if we do find correspondences like most morphing methods, we would essentially do keypoint matching.

How is this different than GAN interpolation? The code interpolation is seemingly similar to the latent space interpolation of GANs, but our method fundamentally differs from GANs in three ways. First, VIINTER does not train a discriminator that provides adversarial guidance to a generator. Second, GANs usually model a continuous Gaussian latent space while we only consider the pairwise interpolation between codes in $\mathbb{Z}_N$. Third, GANs are trained for domain-specific data and are unlikely to work for out-of-distribution data. For example, although we could project face images into the latent space of a powerful GAN trained on aligned human faces, but we cannot use it to interpolate the various categories of images used in Section 4. Our method is applicable for each separate scene, and the CLIP-based feature extraction is shown to generalize well both in prior work [Jain et al. 2021] and our experiments with scenes containing vastly different visual attributes.

What is the implication of the norm of $z$? We point out that our strategy is not adding a penalty on the norm of $z$, but rather strictly enforcing the norm of $z = 1$. Controlling the norm of $z$ ensures that the learned codes exist on a well-defined region in $\mathbb{R}^M$. In the case of $M = 2$, 2-norm ensures all 2D points lie on a circle, whereas 1-norm ensures all 2D points lie on a square inside that circle. As we increase $M$ to higher dimensions, the difference between 1-norm and 2-norm intensifies as the gap between that “circle” and “square” enlarges. As a result, the codes learned with 1-norm is more compact and likely more conducive for interpolation.

What are the limitations? The method may produce obvious artifacts when interpolating scenes with large disparity. Additional experiments on more varied data would be necessary for a comprehensive assessment of its performance on challenging data. Another limitation is it has no sense of 3D locations or camera pose. As a result, it can only interpolate known viewpoints and cannot render at arbitrary locations. Nonetheless, this limitation would not be a deal breaker for many practical use cases that only interpolate between known viewpoints, like 4D light field rendering where viewpoints are fixed on a 2D plane. Another notable use case is event replay for TV viewers, which produces a fly-through experience by interpolating between known cameras. Finally, the proposed method is limited by the training speed of INR, and training the basic INR implemented in this work (with loss from feature extractor)
We sincerely thank the anonymous reviewers for their valuable feedback. Our implementation of CIP, a trained deep network, to guide the INR training also suggests that future developments of INRs could further benefit from absorbing concurrent progresses in other areas of deep learning. As INRs of images evolve to be more accurate and efficient, with this paper, we offer a promising outlook on employing INRs for image manipulation tasks beyond fitting and super-resolving known images.

ACKNOWLEDGMENTS

We sincerely thank the anonymous reviewers for their valuable suggestions to improve the paper. We thank Jonathan Heagerty, Sida Li, Eric Lee, Barbara Brawn, and Maria Herd for developing our light field datasets. This work has been supported in part by the NSF Grants 18-23321 and 21-37229, 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

Benjamin Attal, Jia-Bin Huang, Michael Zollhofer, Johannes Kopf, and Changil Kim. 2022. Learning Neural Light Fields With Ray-Space Embedding Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 19815–19829.

Mojtaba Bemana, Karol Myszkowski, Hans-Peter Seidel, and Tobias Ritschel. 2020. Implicit Fields: Implicit Neural View-, Light- and Time-Image Interpolation. ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2020) 39, 6 (2020). https://doi.org/10.1145/3414685.3417827

Alexander Bergman, Petr Kellnhofer, and Gordon Wetzstein. 2021. Fast Training of Neural Lumigraph Representations Using Meta Learning. Advances in Neural Information Processing Systems 34 (2021). 5875–5885.

Benno Heigl, Reinhard Koch, Marc Pollefeys, Joachim Denzler, and L. Van Gool. 1999. Plenoptic Model and Rendering From Image Sequences Taken By A Hand-held Camera. In Mustererkennung 1999. Springer, 94–101.

Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. 2000. Plenoptic Modeling and Rendering From Image Sequences Taken By A Hand-held Camera. In Proceedings of the 28th annual Conference on Computer Vision and Pattern Recognition. 343–350.

Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. 2021. Baking neural radiance fields for real-time view synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 5875–5884.

REFERENCES

Benjamin Attal, Jia-Bin Huang, Michael Zollhofer, Johannes Kopf, and Changil Kim. 2022. Learning Neural Light Fields With Ray-Space Embedding Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 19815–19829.

Mojtaba Bemana, Karol Myszkowski, Hans-Peter Seidel, and Tobias Ritschel. 2020. Implicit Fields: Implicit Neural View-, Light- and Time-Image Interpolation. ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2020) 39, 6 (2020). https://doi.org/10.1145/3414685.3417827

Alexander Bergman, Petr Kellnhofer, and Gordon Wetzstein. 2021. Fast Training of Neural Lumigraph Representations Using Meta Learning. Advances in Neural Information Processing Systems 34 (2021). 5875–5885.

Benno Heigl, Reinhard Koch, Marc Pollefeys, Joachim Denzler, and L. Van Gool. 1999. Plenoptic Model and Rendering From Image Sequences Taken By A Hand-held Camera. In Mustererkennung 1999. Springer, 94–101.

Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. 2000. Plenoptic Modeling and Rendering From Image Sequences Taken By A Hand-held Camera. In Proceedings of the 28th annual Conference on Computer Vision and Pattern Recognition. 343–350.

Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. 2021. Baking neural radiance fields for real-time view synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 5875–5884.

Benno Heigl, Reinhard Koch, Marc Pollefeys, Joachim Denzler, and L. Van Gool. 1999. Plenoptic Model and Rendering From Image Sequences Taken By A Hand-held Camera. In Mustererkennung 1999. Springer, 94–101.

Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. 2000. Plenoptic Modeling and Rendering From Image Sequences Taken By A Hand-held Camera. In Proceedings of the 28th annual Conference on Computer Vision and Pattern Recognition. 343–350.

Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. 2021. Baking neural radiance fields for real-time view synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 5875–5884.

REFERENCES

Benjamin Attal, Jia-Bin Huang, Michael Zollhofer, Johannes Kopf, and Changil Kim. 2022. Learning Neural Light Fields With Ray-Space Embedding Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 19815–19829.

Mojtaba Bemana, Karol Myszkowski, Hans-Peter Seidel, and Tobias Ritschel. 2020. Implicit Fields: Implicit Neural View-, Light- and Time-Image Interpolation. ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2020) 39, 6 (2020). https://doi.org/10.1145/3414685.3417827

Alexander Bergman, Petr Kellnhofer, and Gordon Wetzstein. 2021. Fast Training of Neural Lumigraph Representations Using Meta Learning. Advances in Neural Information Processing Systems 34 (2021). 5875–5885.

Benno Heigl, Reinhard Koch, Marc Pollefeys, Joachim Denzler, and L. Van Gool. 1999. Plenoptic Model and Rendering From Image Sequences Taken By A Hand-held Camera. In Mustererkennung 1999. Springer, 94–101.

Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. 2000. Plenoptic Modeling and Rendering From Image Sequences Taken By A Hand-held Camera. In Proceedings of the 28th annual Conference on Computer Vision and Pattern Recognition. 343–350.

Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. 2021. Baking neural radiance fields for real-time view synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 5875–5884.

REFERENCES

Benjamin Attal, Jia-Bin Huang, Michael Zollhofer, Johannes Kopf, and Changil Kim. 2022. Learning Neural Light Fields With Ray-Space Embedding Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 19815–19829.

Mojtaba Bemana, Karol Myszkowski, Hans-Peter Seidel, and Tobias Ritschel. 2020. Implicit Fields: Implicit Neural View-, Light- and Time-Image Interpolation. ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2020) 39, 6 (2020). https://doi.org/10.1145/3414685.3417827

Alexander Bergman, Petr Kellnhofer, and Gordon Wetzstein. 2021. Fast Training of Neural Lumigraph Representations Using Meta Learning. Advances in Neural Information Processing Systems 34 (2021). 5875–5885.

Benno Heigl, Reinhard Koch, Marc Pollefeys, Joachim Denzler, and L. Van Gool. 1999. Plenoptic Model and Rendering From Image Sequences Taken By A Hand-held Camera. In Mustererkennung 1999. Springer, 94–101.

Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. 2000. Plenoptic Modeling and Rendering From Image Sequences Taken By A Hand-held Camera. In Proceedings of the 28th annual Conference on Computer Vision and Pattern Recognition. 343–350.

Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. 2021. Baking neural radiance fields for real-time view synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 5875–5884.

REFERENCES

Benjamin Attal, Jia-Bin Huang, Michael Zollhofer, Johannes Kopf, and Changil Kim. 2022. Learning Neural Light Fields With Ray-Space Embedding Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 19815–19829.

Mojtaba Bemana, Karol Myszkowski, Hans-Peter Seidel, and Tobias Ritschel. 2020. Implicit Fields: Implicit Neural View-, Light- and Time-Image Interpolation. ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2020) 39, 6 (2020). https://doi.org/10.1145/3414685.3417827

Alexander Bergman, Petr Kellnhofer, and Gordon Wetzstein. 2021. Fast Training of Neural Lumigraph Representations Using Meta Learning. Advances in Neural Information Processing Systems 34 (2021). 5875–5885.

Benno Heigl, Reinhard Koch, Marc Pollefeys, Joachim Denzler, and L. Van Gool. 1999. Plenoptic Model and Rendering From Image Sequences Taken By A Hand-held Camera. In Mustererkennung 1999. Springer, 94–101.

Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. 2000. Plenoptic Modeling and Rendering From Image Sequences Taken By A Hand-held Camera. In Proceedings of the 28th annual Conference on Computer Vision and Pattern Recognition. 343–350.

Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. 2021. Baking neural radiance fields for real-time view synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 5875–5884.

REFERENCES

Benjamin Attal, Jia-Bin Huang, Michael Zollhofer, Johannes Kopf, and Changil Kim. 2022. Learning Neural Light Fields With Ray-Space Embedding Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 19815–19829.
International Publishing, 405–421. https://doi.org/10.1007/978-3-030-58452-2_24
Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant
Neural Graphics Primitives with a Multiresolution Hash Encoding. ACM Trans. Graph. 41, 4 (2022).
Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanra-
han. 2005. Light Field Photography With A Hand-held Plenoptic Camera. Ph. D.
Dissertation.
Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Love-
grove. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape
Representation. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recog-
nition (CVPR) https://doi.org/10.1109/CVPR.2019.00025
Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei
Zhou, and Hujun Bao. 2021. Animatable Neural Radiance Fields for Modeling
Dynamic Human Bodies. In Proceedings of the IEEE/CVF International Conference
on Computer Vision. 14314–14323.
Eric Penner and Li Zhang. 2017. Soft 3D Reconstruction For View Synthesis. ACM
Transactions on Graphics (TOG) 36, 6 (2017), 1–11.
Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sand-
hin Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al.
2021. Learning Transferable Visual Models From Natural Language Supervision. In
International Conference on Machine Learning. PMLR, 8748–8763.
Gernot Riegler and Vladlen Koltun. 2020. Free View Synthesis. In European Conference
on Computer Vision. Springer, 623–640.
Robin Rombach, Patrick Esser, and Björn Ommer. 2021. Geometry-free View Synthe-
sis: Transformers and No 3D Priors. In Proceedings of the IEEE/CVF International
Conference on Computer Vision. 14356–14366.
Vishwanath Saragadam, Jasper Tan, Gouth Balakrishnan, Richard G Baraniuk, and
Ashok Veeraraghavan. 2022. MINER: Multiscale Implicit Neural Representations.
In The European Conference on Computer Vision (ECCV).
Steven M Seitz and Charles R Dyer. 1996. View Morphing. In Proceedings of the 23rd
annual conference on Computer graphics and interactive techniques. 21–30.
Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon
Wetzstein. 2020. Implicit Neural Representations With Periodic Activation Func-
tions. Advances in Neural Information Processing Systems 33 (2020), 7462–7473.
https://doi.org/10.1109/WACV51458.2022.00024
Vincent Sitzmann, Semon Rezchik, Bill Freeman, Josh Tenenbaum, and Fredo Durand.
2021. Light Field Networks: Neural Scene Representations With Single-Evaluation
Rendering. Advances in Neural Information Processing Systems 34 (2021).
Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P Srinivasan,
Jonathan T Barron, and Ren Ng. 2021. Learned Initializations for Optimizing
Coordinate-based Neural Representations. In Proceedings of the IEEE/CVF Conference
on Computer Vision and Pattern Recognition. 2846–2855.
Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Ragha-
van, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. 2020.
Fourier Features Let Networks Learn High Frequency Functions in Low Dimen-
sional Domains. Advances in Neural Information Processing Systems (2020).
Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. 2016. Multi-view 3D
Models From Single Images With A Convolutional Network. In European Conference
on Computer Vision. Springer, 322–337.
Justus Thies, Michael Zollhofer, and Matthias Niessner. 2019. Deferred Neural Render-
ing: Image Synthesis Using Neural Textures. ACM Transactions on Graphics (TOG)
38, 4 (2019), 1–12.
Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping
Wang. 2021a. NeRF: Learning Neural Implicit Surfaces by Volume Rendering for
Multi-view Reconstruction. In Advances in Neural Information Processing Systems.
Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou,
Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser.
2021b. NeRF-: Learning Multi-view Image-based Rendering. In Proceedings of the
IEEE/CVF Conference on Computer Vision and Pattern Recognition. 4690–4699.
Ziru Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. 2021c.
NeRF–: Neural Radiance Fields Without Known Camera Parameters. arXiv preprint
arXiv:2102.07064 (2021).
Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam
Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. 2005. High Performance
Imaging Using Large Camera Arrays. In ACM SIGGRAPH 2005 Papers. 765–776.
http://lightfield.stanford.edu/lfs.html.
George Wolberg. 1998. Image Morphing: A Survey. The visual computer 14, 8 (1998),
360–372.
Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. 2021.
Plenoxels for Real-time Rendering of Neural Radiance Fields. In Proceedings of
the IEEE/CVF International Conference on Computer Vision. 5752–5761.
Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. 2021.
Physg: Inverse Rendering With Spherical Gaussians For Physics-based Material Editing
And Relighting. In Proceedings of the IEEE/CVF Conference on Computer Vision and
Pattern Recognition. 5453–5462.