3D Motion Magnification:
Visualizing Subtle Motions with Time-Varying Radiance Fields

Brandon Y. Feng* University of Maryland

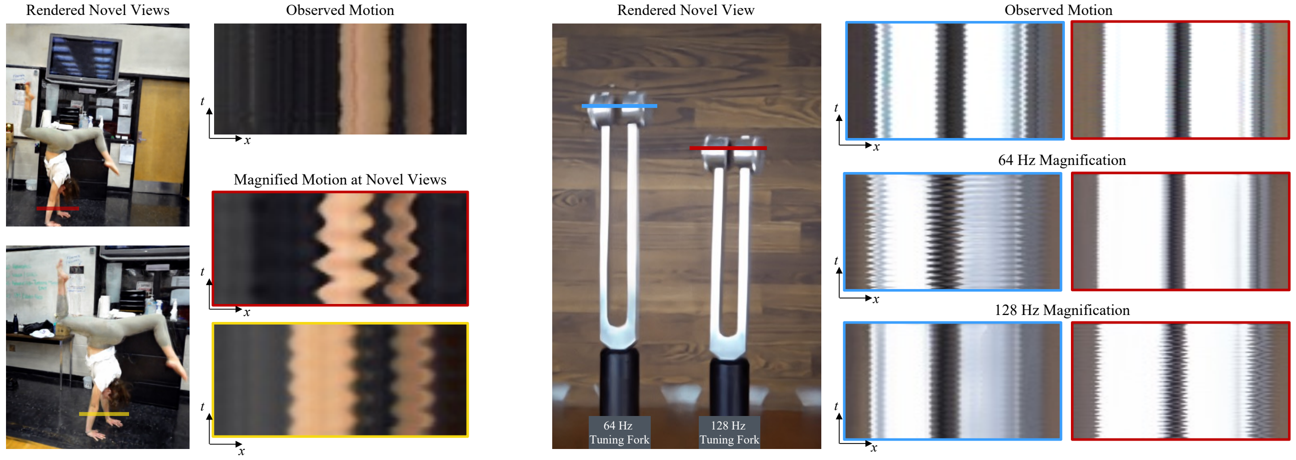
Hadi Alzayer* University of Maryland

Michael Rubinstein Google Research

William T. Freeman Google Research, MIT

Jia-Bin Huang University of Maryland

https://3d-motion-magnification.github.io/



(a) View synthesis with motion magnification

(b) Frequency-selective magnification with temporal filtering

Figure 1. **3D motion magnification**. (a) Novel view synthesis with a gymnast doing a handstand while magnifying the small movements of the arms needed to remain balanced. (b) Motion magnification based on targeted frequencies through temporal filtering, where the left tuning fork vibrates at 64Hz and the right at 128Hz. We visualize x-t (space-time) slices to demonstrate the motion.

Abstract

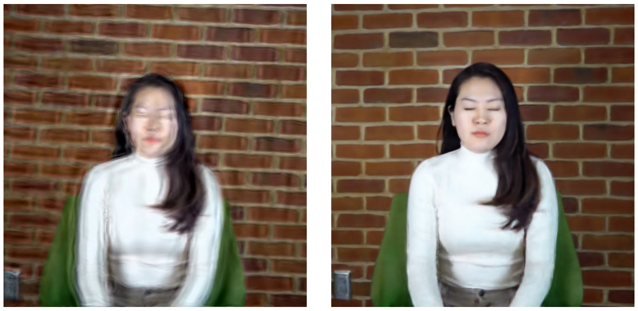
Motion magnification helps us visualize subtle, imperceptible motion. However, prior methods only work for 2D videos captured with a fixed camera. We present a 3D motion magnification method that can magnify subtle motions from scenes captured by a moving camera, while supporting novel view rendering. We represent the scene with time-varying radiance fields and leverage the Eulerian principle for motion magnification to extract and amplify the variation of the embedding of a fixed point over time. We study and validate our proposed principle for 3D motion magnification using both implicit and tri-plane-based radiance fields as our underlying 3D scene representation. We evaluate the effectiveness of our method on both synthetic and real-world scenes captured under various camera setups.

1. Introduction

We live in a big world of small motions. These motions, such as human respiration or object vibration, are hard to perceive with our naked eyes. Video processing techniques [29, 61, 56] have been developed to extract and magnify subtle motions captured in a 2D video to highlight and visualize those motions. These motion magnification techniques empower visual analytics tools like detecting the vibrations of buildings and measuring a person's heart rate using only a video, without the need for physical contact [58, 10, 46, 23].

However, we live in a *3D world* full of *3D motions*. Magnifying motion in 3D, as shown in Figure 1 allows us to perceive these motions from different views. Furthermore, modeling the motion in 3D provides a natural separation be-

*Equal contribution



2D Method [57]

Ours

Figure 2. **Motion magnification from a handheld video.** Prior 2D motion magnification approaches (e.g. [57]) cannot handle videos captured by a moving camera, producing severe artifacts. In contrast, our approach can naturally separate camera motion from object motion, allowing us to magnify only the motion of the subject of interest. See Figure 7 for our magnified output.

tween camera motion and the motion of subjects of interest. This enables magnifying the motion from *handheld videos*, as shown in Figure 2. In contrast, prior 2D motion magnification methods catastrophically fail in such scenarios.

In this paper, we propose a method for *3D motion magnification* using neural radiance fields (NeRF), with minimal modifications to standard NeRF backbones and training pipelines. Prior methods designed for 2D videos often leverage the *Eulerian* perspective, which analyzes and amplifies the color variations at each pixel location over time to magnify motion. In contrast, we bring the Eulerian analysis to a new domain beyond color space by designing a magnification method operating on the *feature embeddings* of NeRF. Our experimental results demonstrate that amplifying temporal variations in the *feature embedding of each 3D point* is highly effective in magnifying subtle 3D motion. We observe that magnifying the point embedding provides more accurate and robust magnified renderings than Eulerian magnification performed directly on rendered images.

Using images captured during a time window when *only subtle motion is visible*, we train NeRF to reconstruct the 3D scene with such subtle temporal variations. We ensure that the only element that changes *over time* is the point embedding function, while the MLP layers of NeRF *remain constant over time*. Although the linear Eulerian approach [61] is agnostic to data dimensionality and is extensible to point embeddings of NeRF, for the phase-based Eulerian approach [56], which showed superior properties over the linear approach, it remains unclear how it may be applied for NeRF as it specifically constructs a complex steerable pyramid over each 2D image frame. The recently introduced tri-plane representation for NeRF's embedding function naturally allows for 2D-specific magnification methods like the phase-based approach [7]. Instead of using the analytical *positional encoding* to generate point embeddings, we learn one feature tri-plane at each observed timestep. These tri-planes can be naturally organized as feature videos for 2D video-based magnification methods. Finally, the motion-magnified 3D scene is rendered using these motion-magnified feature triplanes as the point embedding functions.

To evaluate the performance of 3D magnification with NeRF, we first create a synthetic dataset of scenes with subtle motions and measure the magnification quality against synthetically magnified ground truth videos. The phase-based approach operating on tri-plane features leads to the best performance compared to other alternative approaches considered in our experiments. To further validate the practicality of the proposed method, we use our pipeline to process several real-world captured scenes with varying camera setups, scene compositions, and subject motions. Our results show that our proposed approach for 3D motion magnification achieves robust performance for real-world captures in the presence of image noise and camera poses.

To summarize, our contributions are:

- We introduce the problem of 3D motion magnification. We demonstrate the feasibility of applying Eulerian motion analysis for 3D motion magnification using standard NeRF backbones and training pipelines.
- We extend Eulerian analysis to a new domain beyond color space, exploring strategies to modify and filter point embedding and comparing their trade-offs.
- We demonstrate successful 3D motion magnification results on various real-world scenes with different motions, scene compositions, and even handheld videos unsupported by previous 2D methods.

2. Related Work

Video motion magnification. Prior approaches to video magnification fall under two categories, inspired by fluid dynamics: Lagrangian [29] and Eulerian [61, 56, 57, 37, 65]. The Lagrangian perspective tracks individual pixels as fluid particles and estimates their motion vectors to warp pixels in the image. Lagrangian-based approach to motion magnification computes the optical flow *explicitly* and uses the estimated flow to magnify the motions of the pixels [29]. The performance, however, is limited by the accuracy of flow estimation. On the other hand, the Eulerian perspective analyzes the changes at fixed pixel locations, amplifying the temporal variations at each pixel/location to magnify motion. This approach bypasses the need for explicit feature tracking or optical flow estimation, which can be inaccurate and costly. Two variants of the Eulerian approach are linear [61] and phase-based [56, 57]. Linear Eulerian [61] constructs Laplacian pyramids over the video frames and amplifies the color variation of each pixel over

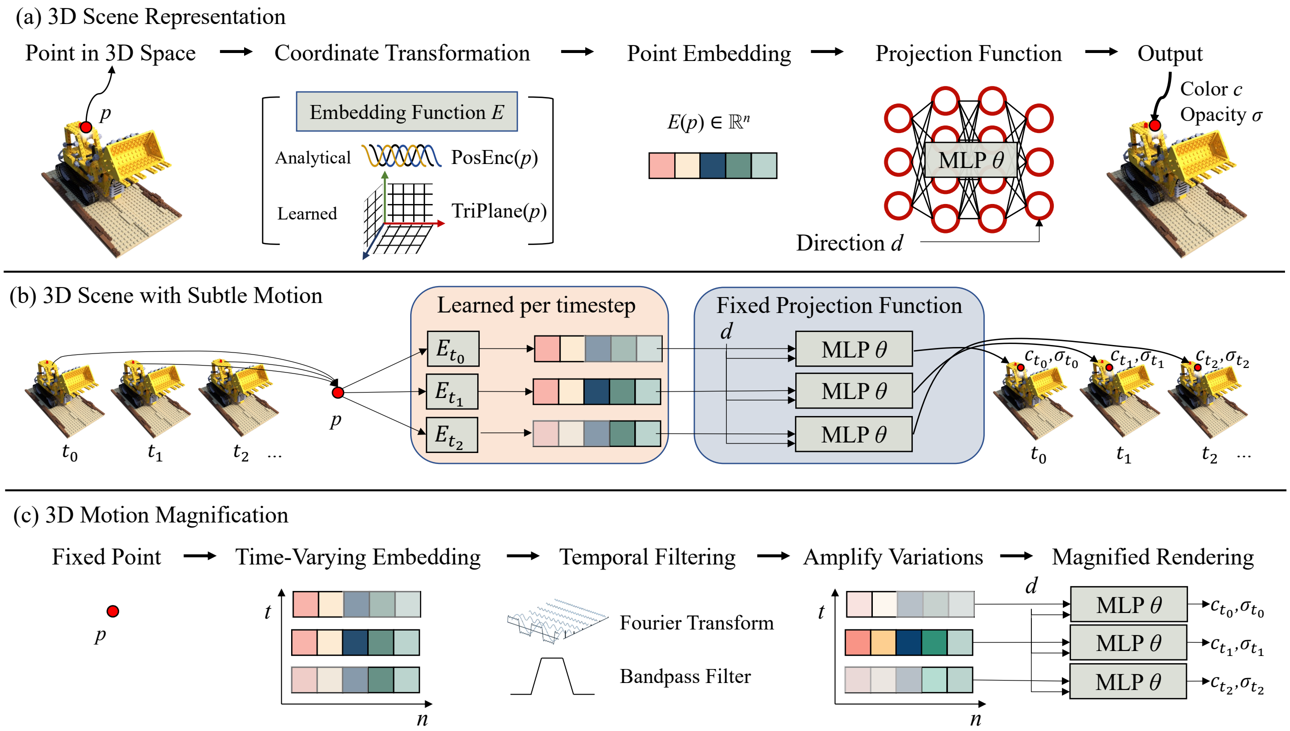


Figure 3. **Method overview.** (a) 3D scene representation with NeRF consists of two main components: 1) Coordinate Transformation uses an embedding function E to map the input point $p \in \mathbb{R}^3$ to a high-dimensional embedding vector $E(p) \in \mathbb{R}^n$. The embedding function can be analytical (positional encoding) or learned (tri-plane). 2) The Projection Function θ (usually an MLP) takes in the point embedding and viewing direction, and regresses them into the output color c and opacity σ at p. (b) We study scenes with **subtle motions**. To model the tiny variations with NeRF, we change E over time while fixing the projection function. (c) At a given point p, we analyze its embedding variations over time: $[E_{t_0}(p), \ldots, E_{t_{T-1}}(p)]$. We perform temporal filtering to isolate and amplify embedding variations within a certain frequency range and then send the amplified embedding to the MLP θ, resulting in motion-magnified 3D rendering.

time. Phase-based Eulerian [56, 57] operates on the phase variations at each pixel, extracted from a complex steerable pyramid [50, 17] decomposition of each video frame. Later work focuses on magnifying larger motion with affine transform and isolated regions of interest with matting [12], using linear-based methods instead of hand-designed filters [37], and adopting a second-order approximation (with acceleration) instead of first-order methods [65]. Video-based motion magnification has also been applied to extract signals like sound waves from videos recording objects, like a bag of chips, deform and oscillate [11, 48]. Our work builds upon classical Eulerian motion magnification but extends it 1) from 2D to 3D and 2) from color space to the point embedding space of radiance fields. Our results show that the Eulerian principle still holds in the point embedding space.

Static radiance fields. NeRF [35] has become the mainstream approach for representing 3D scenes and demonstrates high-quality view synthesis results. Various techniques have been introduced to improve NeRF in several aspects, including training and rendering acceleration [24, 36, 2, 18, 42, 8], reducing aliasing [3], unbounded scene modeling [4, 63], and optimizing poses [27, 34]. Factor Fields [9] present a unified framework summarizing various NeRF variants and other neural signal representations as mainly composed of two components: (1) a *Coordinate Transformation* that maps input coordinates into an embedding space, and (2) a *Projection Function* that maps the embeddings into a value in the field. In this paper, we adopt a similar perspective and focus on analyzing the relationship between point embedding and subtle motions. We propose magnifying subtle motions through Eulerian magnifications of point embeddings in NeRF. We demonstrate successful applications of this approach on NeRF with both positional encoding-based embedding [35, 24] and tri-plane embedding [7, 18].

Dynamic scene representations. Extensive research has been devoted to extending NeRF for modeling dynamic scenes. One line of work learns a deformation field and uses it to warp a canonical NeRF for each timestep [41, 39, 40, 55, 25]. Alternatively, one can directly learn a space-time radiance field with time as an additional coordinate [26, 62, 19, 1, 52, 31, 18, 6, 47]. A major challenge of reconstructing dynamic scenes is capturing time-synchronous multi-view observations. While a multi-camera setting is ideal for acquiring high-quality data [66, 5, 30], researchers have explored the more challenging but practical setting of single-camera captures, leveraging priors such as consistent depth [62, 32, 21], optical flow [26, 19, 31], or human prior [20, 60]. We demonstrate the applicability of 3D motion magnification in both multi-camera and single-camera setups.

Implicit representations. Implicit representations have emerged as powerful tools for modeling signals [51, 13, 38, 28, 43, 53, 54, 15, 16, 14, 49]. Mai and Liu [33] study *2D videos* with implicit neural representation and model motions in videos by learning *spatially invariant* phase shifts in the positional encoding function. Our experiment on magnification based on position encoding functions shares similar ideas. However, unlike Mai and Liu [33], we learn *spatially varying* phase shifts and magnify *3D motion*.

3. Preliminaries

Eulerian motion magnification. The Eulerian-based motion analysis focuses on the changes at a *fixed* spatial location over time instead of tracking a specific particle (pixel). The *linear Eulerian approach* converts the color variation over time at each pixel into a 1D vector, using the Fourier transform to obtain its temporal frequency components, and filters the frequencies corresponding to the desired motion. The color intensity changes within the desired frequency range are then amplified and added back to the original values to create a motion-magnified video (where subtle motions become more visible).

To offer an intuition on why amplifying per-pixel color intensity could magnify motion across the frame, let $f(x,t) = g(x+\delta(t))$ denote a signal with motion over time described by the shift $\delta(t)$. The first-order Taylor series expansion of $g(x+\delta(t))$ about the point x can be written as:

$$g(x) + g'(x)(x+\delta(t)-x) = g(x) + g'(x)\delta(t).$$

With observations at multiple timesteps t, we can easily filter out the static $g(x)$ term and keep the dynamic term $g'(x)\delta(t)$. If we multiply $g'(x)\delta(t)$ by α and add it back to the original signal, we get

$$g(x) + (1+\alpha)g'(x)\delta(t) \approx g(x+(1+\alpha)\delta(t)),$$

which is equivalent to magnifying the motion by α.

The *phase-based Eulerian approach* amplifies phase variations over time instead of color amplitude variations. The phase here is extracted from a complex steerable pyramid constructed from the original frames. The connection between phase and motion can be established through the Fourier shift theorem: if a function $f(x)$ is shifted by a distance δ in its domain, it would be equivalent to multiplying its Fourier component $\mathscr{F}(k)$ by a phase factor $e^{-i2\pi k\delta}$:

$$\mathscr{F}\{f(x-\delta)\}(k) = \mathscr{F}\{f(x)\}(k)e^{-i2\pi k\delta},$$

where $\mathscr{F}$ denotes the Fourier transform operator and k denotes the frequency component. In other words, extracting the phase changes over time reveals the motion-induced pixel shift in space by δ. After amplifying the phase changes, the motion-magnified signal can be generated with an inverse Fourier transform.

Neural radiance fields as 3D scene representations. NeRF models the radiance in a scene as a continuous function, which takes as input a 3D spatial coordinate $p \in \mathbb{R}^3$ and a viewing direction $d \in \mathbb{S}^2$, and outputs the radiance color c (observed from viewing direction d) and density σ at that point. Notably, the spatial coordinate p is transformed into a feature representation through some embedding function E, before a projection function (MLP) regresses it into the final prediction:

$$f(p,d) = \text{MLP}(E(p),d) = (c,\sigma).$$

With subtle and unknown scene motions, we assume this time-varying scene can be formulated as $f(p+\delta(p,t),d)$. If MLP is fixed across time, then the unknown motion $\delta(p,t)$ can be recovered by analyzing the temporal variations of $E(p,t)$. However, where do we access $E(p,t)$? Whereas in the 2D video case, the data of interest is directly recorded by a camera and is available for analysis, here we only have access to a collection of 2D images that may have observed the 3D subtle motions during capture. In the following subsection, we discuss how to reconfigure NeRF to model subtle 3D motions by varying the function $E(p,t)$.

4. Method

We assume the availability of: 1) Multi-view observations to reconstruct a static NeRF, and 2) video recording of the subtle scene motions, either with a time-synchronized multi-camera setup or a single moving camera.

The general workflow of our method is as follows: 1) We train a static NeRF from image observations that can be assumed as motionless. 2) For each timestep $t \in [0, T-1]$ in the video observations, we finetune the embedding function E_t so that the NeRF rendering matches with the observations at t. 3) After finetuning all T embedding functions E_t, we magnify motions by amplifying the temporal variations of each sampled point used in NeRF rendering.

In this section, we describe how we repurpose NeRF to capture subtle motions and perform magnification by analyzing the point embeddings learned by NeRF. We begin our discussion with the base case of the standard NeRF with positional encoding as the point embedding function. We then

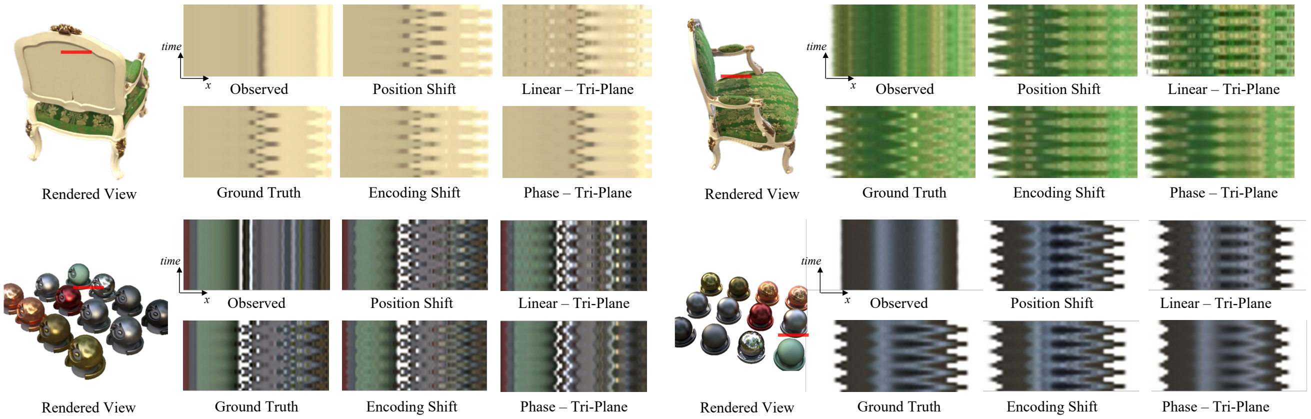


Figure 4. **3D motion magnification on synthetic scenes.** We generated each synthetic scene by periodically vibrating object parts. We magnify the subtle motion encoded in NeRF reconstruction using the approaches discussed in Sec. 4, and visualize the motion here as a 2D space-time slice image. The corresponding location of each space-time slice is indicated by a red line on the rendered view. All four approaches successfully capture and magnify the motion, although the linear Eulerian approach, *Linear - Tri-Plane*, is more prone to intensity overshooting [56], manifested as bright and dark spots in the space-time slice.

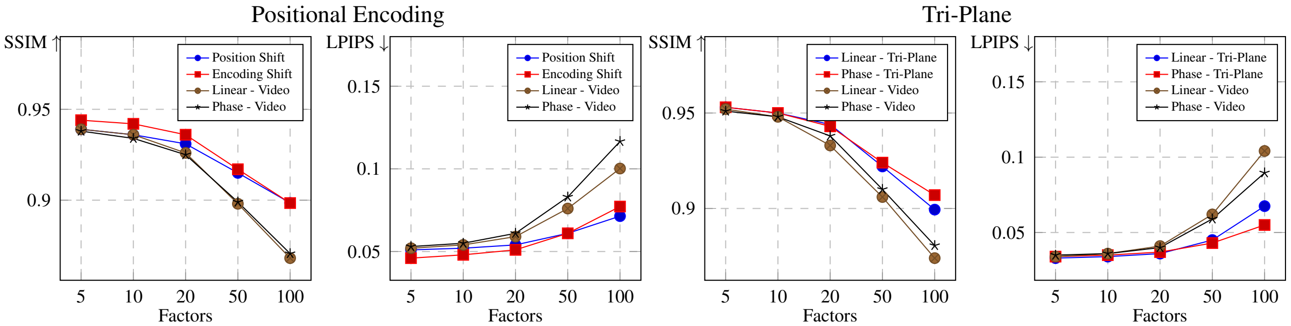


Figure 5. **Quantitative comparison.** We evaluate the quality of motion-magnified renderings as a function of the magnification factor used, using positional encoding (*Left*) and tri-plane (*Right*) as the point embedding function. With positional encoding, we evaluate two approaches to vary point embedding through phase shifts in the sine waves: *Position Shift* (shifting each 3D point) and *Encoding Shift* (shifting each frequency). With tri-plane, we evaluate two approaches to vary learned point embeddings: *Linear - Tri-Plane* (linear magnification on tri-plane) and *Phase - Tri-Plane* (phase-based magnification on tri-plane). For both embedding functions, we compare against two baseline methods for video motion magnification: *Linear - Video* (linear magnification on the NeRF-rendered video) and *Phase-Video* (phase-based magnification on the NeRF-rendered video). Results from two embedding functions are separated to enable better assessments of the impact of different magnification approaches and avoid confounding with the inherent performance gap between different embedding functions and MLP architectures.

describe our preferred approach with tri-plane as the embedding function for NeRF, which leads to a natural integration with the phase-based Eulerian magnification technique previously designed for videos.

4.1. NeRF with Positional Encoding

We first describe how Eulerian magnification in the embedding space can be achieved on standard NeRF with positional encoding. Motivated by prior work on motion-adjustable neural representations for video [33], we keep the main backbone of NeRF intact and separately train a small MLP g that learns to apply phase shifts in the positional encoding functions. However, different from prior work [33], with video observations from the scene, we organize the images by their captured time and train a separate MLP g for each timestep. Effectively, g learns to adjust the embedded representation of each point so that the NeRF output from the projection function (MLP θ) is consistent with the time-varying observations. Weights of the MLP θ are shared across all time steps, and the only difference lies in the point embeddings. There are two options to induce phase shifts to the positional encoding function: position shift and encoding shift.

Position shift. To model motion exclusively through changing the point embeddings, we let g directly **predict the 3D position shift of the queried point** p: $g(p,t) = \Delta p \in \mathbb{R}^3$. We add Δp to p before applying positional en-

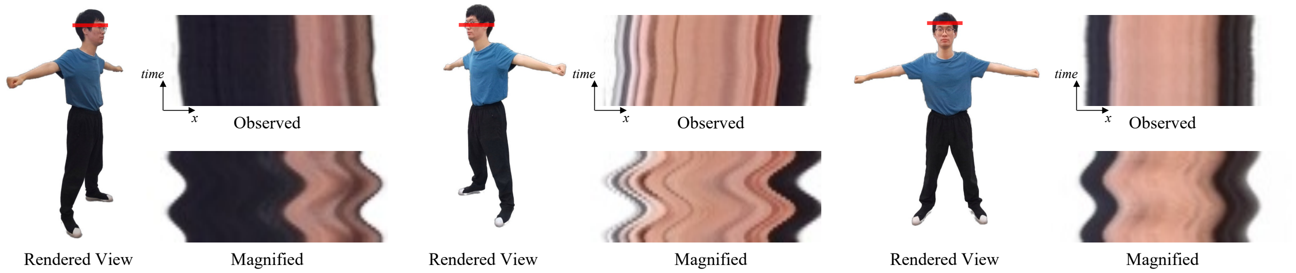


Figure 6. **Real-world multi-view motion magnification**. Using multi-view videos from the HumanNerf dataset [20], we can capture and magnify true 3D motion. We visualize the motion here as a 2D space-time image, where the corresponding location of each space-time slice is shown on the rendered view as a red line.

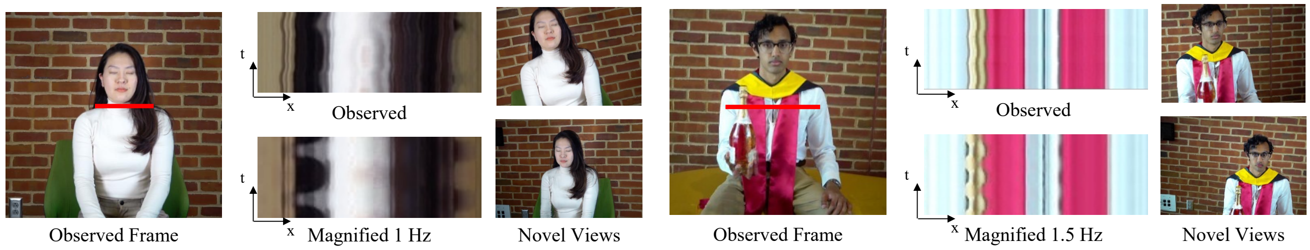


Figure 7. **Real-world single-camera motion magnification**. Despite the monocular view of the dynamic scene, we can render novel views while magnifying subtle motion. We visualize the motion as a 2D image slice through time.

coding, obtaining a time-varying point embedding function

$$E(p,t) = \text{PosEnc}(p + g(p,t)),$$

which is equivalent to applying a phase shift of $\omega \cdot g(p,t)$ within each sine wave $\sin(\omega x)$ used in positional encoding. Note that the phase shifts for all K frequencies have the same direction and only differ in magnitude, which is scaled by different $\omega \in [1,...,K]$.

Encoding shift. Note that motion does not only lead to geometric changes in the 3D scene since it would also cause appearance changes like shadows and reflections. Therefore, attributing all the scene variation to shifts in 3D position is not sufficient. Instead, we may let g learn **a separate shift for each encoding frequency**. In other words, with $\phi_\omega \in \mathbb{R}^3$, let $g(p,t) = [\phi_1, \phi_2, ..., \phi_K] \in \mathbb{R}^{3\times K}$, and the point embedding function becomes

$$E(p,t) = [\sin(\omega_1 p + \phi_1), ..., \sin(\omega_K p + \phi_K)].$$

This setup treats positional encoding as a feature generator that produces an embedding with $3K$ channels. As shown in later sections, this "motion-agnostic" approach still learns to capture the true motion while outperforming the approach that only accounts for position shifts.

With either of these two approaches to vary the point embeddings in a standard NeRF with positional encoding, we can render magnified motions by linearly amplifying the temporal variations of $g(p,t)$, and then rendering the point's color and opacity with NeRF.

4.2. NeRF with Tri-plane Learnable Embedding

We now describe our preferred approach using tri-plane as the embedding function for NeRF. Later experiments suggest that the tri-plane-based approach achieved better magnification quality than the positional encoding approach. Tri-plane [7, 18] has been recently proposed as an efficient way to obtain learnable embedding for points in NeRF rendering. Compared to NeRF with the analytical positional encoding, NeRF with tri-planes as the point embedding function can achieve similar representation capacity with far fewer MLP layers and, thus, faster inference.

In our use case of NeRF, the tri-plane formulation has a nice implication of reducing the 3D scene into a collection of 2D feature planes. Such a decomposition preserves the relative spatial relationship between points, instead of randomly hashing points into features. This observation suggests we may achieve 3D motion magnification by directly processing the feature planes and potentially outperforming the aforementioned linear magnification within the positional encoding function.

Specifically, we train a separate tri-plane embedding function for each timestep while the MLP-based projection function is shared across time. With this setup, all subtle temporal changes in the scene (motion or appearance) would need to be encapsulated in the temporal changes of the 2D feature images of the tri-plane. The point embedding

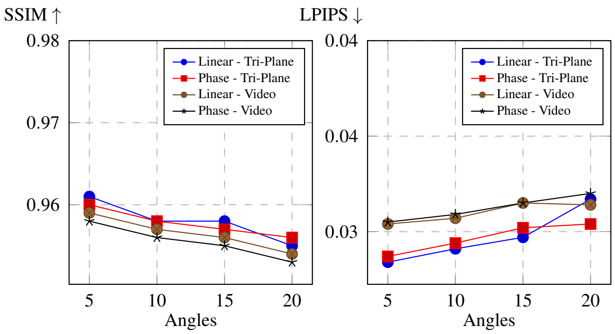


Figure 8. **Varying the angles of deviation from observed views**. As the deviation angle increases, magnifying through the embedding space consistently outperforms the baseline approaches that operate on the color space.

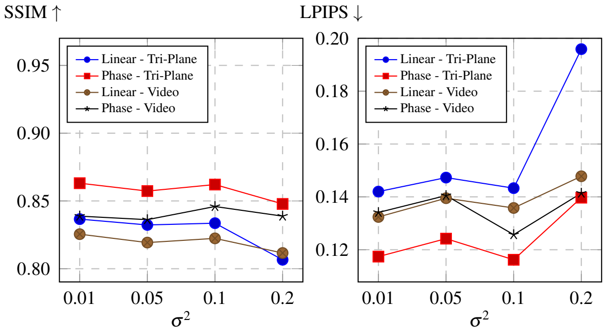


Figure 9. **Varying noise levels in training views**. When we perform Eulerian magnification on the tri-plane embedding space, *Phase - Tri-Plane* is more robust than *Linear - Tri-Plane* in the presence of noise. The finding is analogous to previous analysis [56] on color space magnification (validated by the two baseline results: *Linear - Video* and *Phase - Video*).

function here can be written as

$$E(p,t) = \text{Project}(p, \text{TriPlane}_t),$$

where the embedding of p is obtained by projecting p onto the tri-plane and aggregating the corresponding features.

With a separate tri-plane constructed for each timestep, we establish a key connection to prior video magnification methods: we essentially obtain a *video* for each tri-plane feature channel, on which we could either apply **linear magnification** and amplify the temporal changes of each pixel in the feature image, or **phase-based magnification** with complex steerable pyramids constructed over each channel of the 2D feature image. Our experimental results confirm the feasibility of this 2D-inspired approach. The performance comparison in Section 5.2 between the linear and phase-based approaches also validates the original findings from when these two approaches were applied to perform Eulerian processing of color spaces.

5. Experimental Results

We evaluate the performance of the proposed method using synthetic scenes in Section 5.2. We create ground truth sequences for the magnified motion and quantitatively compare the different approaches for 3D motion magnification. We then present our results on real-world captured data in Section 5.3. We first deploy our method on real-world multi-view video observations. After validating its effectiveness on real-world multi-view data, we further apply our method to real-world video sequences captured using a *single-camera*. As a result of extending motion magnification to 3D, our method successfully magnifies 3D motions from handheld-captured videos with camera shake, a scenario unattainable by prior work that focused on stabilized 2D videos.

5.1. Implementation Details

Positional encoding as point embedding. We train a 3-layer MLP with 32 hidden channels to predict $g(p,t)$ and apply the resulting phase shifts in positional encoding. We implement the network with nerfacc [24]. To train a static NeRF for the first timestep, we optimize for 50,000 steps, and for the remaining timesteps, we finetune the embedding function for 10,000 steps. After training, for each point p, we obtain its time-varying phase shifts $g(p,t)$. To render with magnified motion within a time window $[0,\dots,T-1]$, we Fourier transform $[g(p,0),\dots,g(p,T-1)]$ along the time dimension, use a bandpass filter to isolate the motions within the frequency range of interest, amplify the components within the passband range, and then apply inverse Fourier transform to obtain the magnified predictions. The magnified predictions are added inside the positional encoding as phase shifts, followed by the standard NeRF rendering with MLP inference.

Tri-plane as point embedding. Our implementation builds on K-Planes [18]. To aggregate the embedding from different planes, we adopt concatenation instead of the default Hadamard product, and we only set the triplanes at a single scale for simplicity. To train a static NeRF for the first timestep, we optimize for 30,000 steps, and for the remaining timesteps we finetune the embedding function for 10,000 steps. After training, we compose a video for each feature plane within $[0,\dots,T-1]$. Then, we apply 2D magnification methods directly on these *feature videos*. For the *linear* approach, we temporally filter and amplify the feature value variations within a frequency range. For the *phase-based* approach, we construct a complex steerable pyramid over the feature image, temporally filter and amplify the phase variations, and then collapse the pyramid back into the image space; the resulting feature image would exhibit magnified motions [56]. To produce motion-magnified rendering, we embed each sample point with the processed triplane features and then use the MLP to project the embedding into color and density output as usual.

5.2. Synthetic Scenes

Data Generation. We use the standard Blender scenes [35] and simulate motions in different object parts. We render each scene for one second at 30 frames per second. The simulated motions are periodic, ranging from 3 Hz to 5 Hz. We also render sequences with ground truth magnified motions under factors 5, 10, 20, 50, and 100.

Qualitative Evaluation. We magnify motions in the Blender scenes with the approaches described in Sec. 4. Results in Figure 4 show that all four approaches successfully model and magnify tiny motions in the observations. Consistent with previous findings in video magnification [56], the linear Eulerian approach leads to more artifacts, such as clipped color intensities and noise amplification.

Quantitative Evaluation. We evaluate the results against ground truth magnified frames using structural similarity index measure (SSIM) [59], and LPIPS [64] with AlexNet [22] as the backbone. We also render a video from the trained NeRF without magnification for baseline comparisons at each test view. We then apply classical 2D methods directly on the video to magnify subtle motions. As shown in Figure 5, our 3D magnification methods outperform the 2D baseline methods in producing motion-magnified renderings consistent with the ground truth renderings.

Sensitivity Aanalysis. In Figure 8, we analyze magnification quality as the test viewpoints deviate from the observed viewpoints. In Figure 9, we plot magnification quality as noise levels increase in the captured frames. Previous work [56] on color space magnification has found that the phase-based approach is less sensitive to noise than the linear approach; we observe similar phenomenon during our magnification in the embedding space.

5.3. Real-world Scenes

We test our methods on several real-world scenes captured with different camera setups.

Multi-Camera Setup. We first validate our method on the publicly available dataset from HumanNerf [66], comprising short videos from six cameras simultaneously capturing a scene with a person standing in the center. We extract a brief period where the person is relatively static but still exhibits subtle body movements. In Figure 6, we present the magnified results from different viewpoints; the full videos are available in the supplementary material.

Single-Camera Setup. As a multi-camera setup may be prohibitively inconvenient and expensive for many users, we further deploy our method on a single-camera setup. We design a capture procedure that consists of two stages. Step 1: we first capture a moving camera video of the static scene, which will be used to train a static NeRF. Step 2: we capture a single-view video of the dynamic scene, which is used to finetune the point embedding function in NeRF to model the time-varying scene. After the two-stage capture, we perform NeRF training and render magnified motions using the previously described pipeline. This two-step capture approach is prevalent in video-based motion magnification application scenarios, where people identify unwanted motion in static civil structures under normal conditions [46, 23]. We also highlight that our method **supports handheld capture and does not require a tripod**, unlike previous 2D-based approaches that would fail without a steady capture, as shown in Figure 2. This is possible since our method uses the estimated poses of each frame independently when updating the radiance fields on the dynamic sequence. Hence, an accurate pose estimation removes the need for a perfectly still capture. We show the results using monocular capture in Figure 1 of a gymnast holding a handstand and in Figure 7 on a person trying to balance on one leg and a person breathing. We also show observed space-time slices compared to our magnified ones with NeRF reconstruction.

6. Limitations

Data captured in real-world environments can be blurry due to defocus and camera shake, degrading the quality of NeRF. The training of NeRF assumes the knowledge of camera poses, so its performance depends heavily on the accuracy of pose estimation, often using RGB-based structure-from-motion (SfM) algorithms [44, 45]. The estimations are mostly reliable but not flawless due to lens distortions that require specific calibrations, which may not be performed by common pipelines for NeRF-based 3D reconstruction. More importantly, inaccurate pose estimation would exacerbate the ambiguity between *camera motion* and *scene motion*, which could hinder magnifying subtle scene motions or lead to false motion magnification. Therefore, real-world data should be captured under conditions where accurate camera intrinsic and extrinsic parameters are accessible, either from *reliable* RGB-based SfM with textured surfaces in the scene, or from cameras that support 6-DoF tracking during capture.

7. Conclusion

We present a 3D motion magnification method that applies Eulerain processing principles to analyzing NeRF embeddings over time. While classical magnification methods (developed originally for 2D videos) process pixel colors directly, we show that processing the point embeddings of NeRF successfully generalizes those approaches and allows magnifying motions in 3D renderings. We believe our work will motivate further research towards integrating traditional signal processing techniques into neural fields.

References

[1] Benjamin Attal, Jia-Bin Huang, Christian Richardt, Michael Zollhoefer, Johannes Kopf, Matthew O'Toole, and Changil Kim. HyperReel: High-Fidelity 6-DoF Video with Ray-Conditioned Sampling. *CVPR*, 2023. 3

[2] Benjamin Attal, Jia-Bin Huang, Michael Zollhöfer, Johannes Kopf, and Changil Kim. Learning Neural Light Fields with Ray-space Embedding. *CVPR*, 2022. 3

[3] 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. *arXiv*, 2021. 3

[4] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. *CVPR*, 2022. 3

[5] Zhongang Cai, Daxuan Ren, Ailing Zeng, Zhengyu Lin, Tao Yu, Wenjia Wang, et al. HuMMan: Multi-modal 4D Human Dataset for Versatile Sensing and Modeling. *ECCV*, 2022. 4

[6] Ang Cao and Justin Johnson. HexPlane: A Fast Representation for Dynamic Scenes. *CVPR*, 2023. 3

[7] Eric R. Chan, Connor Z. Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, et al. Efficient Geometry-aware 3D Generative Adversarial Networks. *CVPR*, 2022. 2, 3, 6

[8] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial Radiance Fields. *ECCV*, 2022. 3

[9] Anpei Chen, Zexiang Xu, Xinyue Wei, Siyu Tang, Hao Su, and Andreas Geiger. Factor Fields: A Unified Framework for Neural Fields and Beyond. *arXiv*, 2023. 3

[10] Justin G. Chen, Abe Davis, Neal Wadhwa, Frédo Durand, William T. Freeman, and Oral Büyüköztürk. Video Camera–based Vibration Measurement for Civil Infrastructure Applications. *Journal of Infrastructure Systems*, 23(3), 2017. 1

[11] Abe Davis, Michael Rubinstein, Neal Wadhwa, Gautham J. Mysore, Frédo Durand, and William T. Freeman. The visual microphone: Passive recovery of sound from video. *ACM Transactions on Graphics*, 33(4), 2014. 3

[12] Mohamed Elgharib, Mohamed Hefeeda, Fredo Durand, and William T. Freeman. Video Magnification in Presence of Large Motions. *CVPR*, 2015. 3

[13] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Multiplicative Filter Networks. *ICLR*, 2020. 4

[14] Brandon Y. Feng, Susmija Jabbireddy, and Amitabh Varshney. VIINTER: View Interpolation with Implicit Neural Representations of Images. *SIGGRAPH Asia*, 2022. 4

[15] Brandon Y. Feng and Amitabh Varshney. SIGNET: Efficient Neural Representation for Light Fields. *ICCV*, 2021. 4

[16] Brandon Y. Feng, Yinda Zhang, Danhang Tang, Ruofei Du, and Amitabh Varshney. PRIF: Primary Ray-Based Implicit Function. *ECCV*, 2022. 4

[17] William T. Freeman and Edward H. Adelson. The Design and Use of Steerable Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991. 3

[18] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-Planes: Explicit Radiance Fields in Space, Time, and Appearance. *arXiv*, 2023. 3, 6, 7

[19] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic View Synthesis from Dynamic Monocular Video. *ICCV*, 2021. 3, 4

[20] Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. NeuMan: Neural Human Radiance Field from a Single Video. *ECCV*, 2022. 4, 6

[21] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust Consistent Video Depth Estimation. *CVPR*, 2021. 4

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *NIPS*, 2012. 8

[23] Ricard Lado-Roigé, Josep Font-Moré, and Marco A. Pérez. Learning-based Video Motion Magnification Approach for Vibration-based Damage Detection. *Measurement*, 206, 2023. 1, 8

[24] Ruilong Li, Matthew Tancik, and Angjoo Kanazawa. NerfAcc: A General NeRF Accleration Toolbox. *arXiv*, 2022. 3, 7

[25] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, and Zhaoyang Lv. Neural 3D Video Synthesis. *CVPR*, 2022. 3

[26] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. *CVPR*, 2021. 3, 4

[27] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting Neural Radiance Fields. *ICCV*, 2021. 3

[28] David B. Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. BACON: Band-limited Coordinate Networks for Multiscale Scene Representation. *CVPR*, 2022. 4

[29] Ce Liu, Antonio Torralba, William T. Freeman, Frédo Durand, and Edward H. Adelson. Motion Magnification. *ACM Transactions on Graphics*, 2005. 1, 2

[30] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural Actor: Neural Free-View Synthesis of Human Actors with Pose Control. *ACM Transactions on Graphics*, 2021. 4

[31] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust Dynamic Radiance Fields. *CVPR*, 2023. 3, 4

[32] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics*, 2020. 4

[33] Long Mai and Feng Liu. Motion-Adjustable Neural Implicit Video Representation. *CVPR*, 2022. 4, 5

[34] Andreas Meuleman, Yu-Lun Liu, Chen Gao, Jia-Bin Huang, Changil Kim, Min H. Kim, and Johannes Kopf. Progressively Optimized Local Radiance Fields for Robust View Synthesis. *CVPR*, 2023. 3

[35] 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. *ECCV*, 2020. 3, 8

[36] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Mul-

tiresolution Hash Encoding. *ACM Transactions on Graphics*, 2022. 3

[37] Tae-Hyun Oh, Ronnachai Jaroensri, Changil Kim, Mohamed Elgharib, Frédo Durand, William T. Freeman, and Wojciech Matusik. Learning-based Video Motion Magnification. *ECCV*, 2018. 2, 3

[38] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. *CVPR*, 2019. 4

[39] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable Neural Radiance Fields. *ICCV*, 2021. 3

[40] 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 Transactions on Graphics*, 2021. 3

[41] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. *CVPR*, 2021. 3

[42] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance Fields without Neural Networks. *CVPR*, 2022. 3

[43] Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G Baraniuk, and Ashok Veeraraghavan. MINER: Multiscale Implicit Neural Representations. *ECCV*, 2022. 4

[44] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. *CVPR*, 2016. 8

[45] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. *ECCV*, 2016. 8

[46] Zhexiong Shang and Zhigang Shen. Multi-point Vibration Measurement and Mode Magnification of Civil Structures Using Video-based Motion Processing. *Automation in Construction*, 93, 2018. 1, 8

[47] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4D: Efficient Neural 4D Decomposition for High-fidelity Dynamic Reconstruction and Rendering. *CVPR*, 2023. 3

[48] Mark Sheinin, Dorian Chan, Matthew O'Toole, and Srinivasa G. Narasimhan. Dual-Shutter Optical Vibration Sensing. *CVPR*, 2022. 3

[49] Shayan Shekarforoush, David B. Lindell, David J. Fleet, and Marcus A. Brubaker. Residual Multiplicative Filter Networks for Multiscale Reconstruction. *arXiv*, 2022. 4

[50] Eero P. Simoncelli and William T. Freeman. The Steerable Pyramid: A Flexible Architecture for Multi-scale Derivative Computation. *ICIP*, 1995. 3

[51] Vincent Sitzmann, Julien Martel, Alexander Bergman, David B. Lindell, and Gordon Wetzstein. Implicit Neural Representations with Periodic Activation Functions. *NeurIPS*, 2020. 4

[52] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. NeRFPlayer: A Streamable Dynamic Scene Representation with Decomposed Neural Radiance Fields. *arXiv*, 2022. 3

[53] Kushagra Tiwary, Akshat Dave, Nikhil Behari, Tzofi Klinghoffer, Ashok Veeraraghavan, and Ramesh Raskar. ORCa: Glossy Objects as Radiance-Field Cameras. *CVPR*, 2023. 4

[54] Kushagra Tiwary, Tzofi Klinghoffer, and Ramesh Raskar. Towards Learning Neural Representations from Shadows. *ECCV*, 2022. 4

[55] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of A Dynamic Scene from Monocular Video. *ICCV*, 2021. 3

[56] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T. Freeman. Phase-Based Video Motion Processing. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2013)*, 32(4), 2013. 1, 2, 3, 5, 7, 8

[57] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T. Freeman. Riesz Pyramids for Fast Phase-based Video Magnification. *ICCP*, 2014. 2, 3

[58] Wenjin Wang, Sander Stuijk, and Gerard de Haan. Exploiting Spatial Redundancy of Image Sensor for Motion Robust rPPG. *IEEE Transactions on Biomedical Engineering*, 62, 2015. 1

[59] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4), 2004. 8

[60] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint Rendering of Moving People from Monocular Video. *CVPR*, 2022. 4

[61] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William T. Freeman. Eulerian Video Magnification for Revealing Subtle Changes in the World. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2012)*, 31(4), 2012. 1, 2

[62] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time Neural Irradiance Fields for Free-Viewpoint Video. *CVPR*, 2021. 3, 4

[63] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and Improving Neural Radiance Fields. *arXiv*, 2020. 3

[64] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *CVPR*, 2018. 8

[65] Yichao Zhang, Silvia L. Pintea, and Jan C. van Gemert. Video Acceleration Magnification. *CVPR*, 2017. 2, 3

[66] Fuqiang Zhao, Wei Yang, Jiakai Zhang, Pei Lin, Yingliang Zhang, Jingyi Yu, and Lan Xu. HumanNeRF: Efficiently Generated Human Radiance Field From Sparse Inputs. *CVPR*, 2022. 4, 8