DOI: 10.1111/cgf.14888
Eurographics Symposium on Rendering 2023
T. Ritschel and A. Weidlich
(Guest Editors)

COMPUTER GRAPHICS *forum*
*Volume 42 (2023), Number 4*
*e14888* *pp. 1 - 13*

# ModalNeRF: Neural Modal Analysis and Synthesis for Free-Viewpoint Navigation in Dynamically Vibrating Scenes

Automne Petitjean[1,2], Yohan Poirier-Ginter[3,2], Ayush Tewari[4], Guillaume Cordonnier[2], George Drettakis[2]

[1] ENS de Lyon [2] Inria, Université Côte d'Azur [3] Université Laval [4] MIT CSAIL

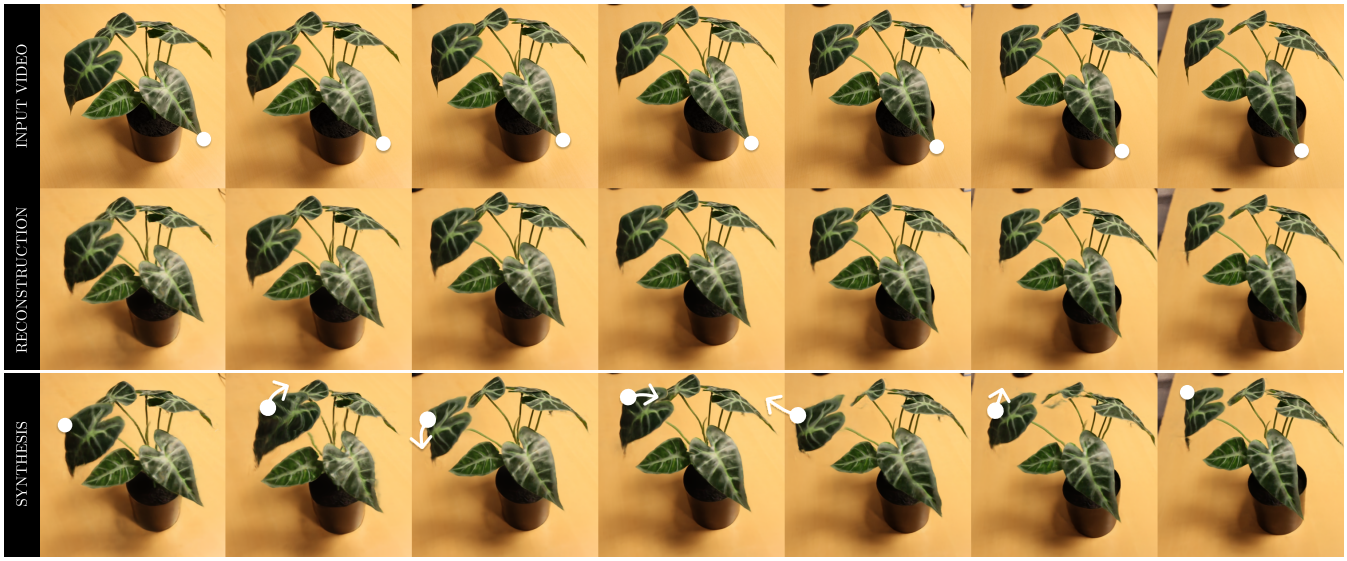


**Figure 1:** *The first row shows the input video; the second row shows a rendering of the same frames using our new Lagrangian particle-based representation and the third row shows the result of our edited motion. Notice how the leaves move with much higher amplitude than in the original video. Please see the supplemental video for the results in motion.*

**Abstract**

*Recent advances in Neural Radiance Fields enable the capture of scenes with motion. However, editing the motion is hard; no existing method allows editing beyond the space of motion existing in the original video, nor editing based on physics. We present the first approach that allows physically-based editing of motion in a scene captured with a single hand-held video camera, containing vibrating or periodic motion. We first introduce a Lagrangian representation, representing motion as the displacement of particles, which is learned while training a radiance field. We use these particles to create a continuous representation of motion over the sequence, which is then used to perform a modal analysis of the motion thanks to a Fourier transform on the particle displacement over time. The resulting extracted modes allow motion synthesis, and easy editing of the motion, while inheriting the ability for free-viewpoint synthesis in the captured 3D scene from the radiance field. We demonstrate our new method on synthetic and real captured scenes.*

## 1. Introduction

We live in a dynamic 3D world where objects exhibit interesting and complex natural motion. Several methods have been developed in recent years to reconstruct the 3D properties of such dynamic scenes from 2D video observations, enabling novel view synthesis [TTG*21, PSB*21, PCPMMN20, LNSW21]. However, these methods only reconstruct the 3D motion of the scene and do not reason about the underlying physical models of objects that lead to the observed motion. Thus, while rendering any observed state of

a scene from novel camera viewpoints is feasible, it is impossible to naturally interact with the scene and create novel, physically-plausible deformations.

In this work, we are interested in capturing properties of dynamic real-world scenes in a way that enables 3D interactions based on physics. We focus on scenes that exhibit vibrating or periodic motions. This covers a wide array of situations, including plants moving in the wind, a common occurence in outdoor captures. To this end, we develop a method that can learn physics-based models of the 3D scene dynamics and reconstruct the 3D structure, appearance, and motion of the scene. For the first time, a simple capture of a dynamic scene with a single hand-held camera allows a user to apply virtual forces and synthesize physically-plausible scene deformations. The deformed 3D scene can be rendered from any virtual camera – in the limit of the views captured – at photorealistic quality, thus, opening avenues for interesting applications in computer graphics.

Our dynamics model is inspired by modal analysis [DBC*15, HSO03], that has been used in computer graphics to compute the physical properties of moving objects from videos by extracting the periodic components of motion using spectral analysis. These physical properties can then be used to generate novel, plausible motion for the scene. The closest method to our paper is [DCD15], which uses modal analysis for editing objects in videos. However, this method is limited to 2D – it uses the optical flow of a video captured from a static camera to learn the physical dynamics. Since this method does not reason about the 3D scene geometry, it cannot account for scenes with complex motion and large disocclusions, and can also not synthesize 3D motion and novel camera-view renderings. Additionally, it requires the scene to be captured by a static camera.

In contrast, our method enables modal analysis and synthesis of 3D scenes only using 2D observations from a single moving hand-held video camera. We introduce several technical innovations that enable this. We present a novel particle-based Lagrangian representation for reconstructing the 3D scene motion. Unlike existing methods [TTG*21, PSB*21] that use neural fields to deform 3D points in the camera view to a learned canonical volume, we use a particle-based formulation. We regularly sample a set of particles in a canonical volume, and represent the motion in the scene as a displacement of those particles. The scene geometry and appearance in the canonical – or rest – space are learned using a factorized voxel grid, as introduced in TensoRF [CXG*22]. To render the scene at a given frame and alternate viewpoint, we deform the canonical space using the motion stored in the particles.

We develop a volume rendering formulation that uses the deformed particles to render an image from any novel camera viewpoint. This particle-based Lagrangian formulation is essential, as we can now persistently track a set of 3D points over the entire motion sequence, which was not possible with existing dynamic reconstruction methods. Using a voxel grid also makes our approach more efficient to train, compared to Multi-Layer Perceptrons (MLPs) used in existing methods [MST*20, TTG*21, PSB*21, PCPMMN20].

The set of deforming 3D points is used for modal analysis, where we recover the modal frequencies exhibited by the object. We select interesting modes and use them to generate new scene deformations, where a user can pick a point in the scene and apply a force in any 3D direction. Since we perform physical reasoning in 3D, we are not limited by complex occlusion effects in image-space like the method of [DCD15], and in addition, we can also render novel camera views of the synthesized motion. Our contributions can be summarized as follows:

- A Lagrangian, particle-based representation of motion that is trained together with a radiance field of a scene captured with a single hand-held camera.
- A modal analysis and synthesis method that uses the particle-based representation, enabling editing of motion while allowing free-viewpoint synthesis in the captured 3D scene.

We show the results of our method on a synthetic and three captured scenes, demonstrating both the quality of the captured motion and the ability to edit and synthesize new motions in the scene while rendering novel views.

## 2. Related Work

We first discuss 3D static and non-rigid reconstruction methods, and then discuss modal analysis and synthesis in the context of image/video manipulation.

### 2.1. Neural and Deformable Radiance Fields

Neural Radiance Fields (NeRFs) are a very active area of research [TTM*22], allowing for high-quality novel view synthesis for static scenes. Radiance fields are often stored as a volumetric representation; NeRF [MST*20] stores the radiance field in a Multi-Layer Perceptron (MLP) that is fit to a single scene. Numerous variants have been proposed, frequently attempting to address the high computational cost of training and rendering. Many such solutions have been presented recently, such as InstantNGP [MESK22] that uses a hash grid and fast low-level CUDA operations to accelerate computation, and Plenoxels [FKYT*22] that forgoes the neural network altogether and stores density and spherical harmonic coefficients in a sparse grid. Another solution is TensoRF [CXG*22], where the radiance field is stored as a 4D tensor. Specifically, it stores density and features in a factored 3D voxel grid. The method uses a specific factorization-based block term decomposition, that computes the 3D tensor as products of vectors and 2D tensors to limit the effective storage size to $\mathcal{O}(n^2)$. At inference, tri-linear interpolation of the density and features is performed in the factorized grid. The interpolated features and camera ray directions are directly fed to a shallow MLP that predicts the view-dependent local color. We adopt TensoRF as a basis for our method for its combination of simplicity and efficiency.

Several dynamic reconstruction methods have been developed to lift the restriction of NeRF to static scenes. These methods reconstruct time-varying 3D NeRFs from a monocular video of a dynamic scene. Some approaches directly condition a NeRF network on the timestep [XHKK21, GSKH21, GXH*22] while others learn a canonical NeRF volume that is deformed to create the reconstruction for each timestep [TTG*21, PSB*21, PCPMMN20,

PSH*21, GCD*22, FYW*22]. The latter methods learn a deformation field that deforms the canonical volume to the time-varying volume, enabling computation of correspondences between the different timesteps. However, while these methods can be used for time-varying novel view synthesis, they cannot perform edits using physics-based principles such as the modal analysis/synthesis we use. Some methods exist that provide initial solutions for editing NeRFs, such as CoNeRF [KYK*21] or HyperNeRF [PSH*21]. However, the editing of the NeRF relies on pre-trained latent codes that can be interpolated and thus, these methods cannot be used to generate completely novel motion. All these methods can be seen as Eulerian representations, that make it harder to perform synthesis. In contrast, our Lagrangian particle-based approach is more suitable for motion control, since we can easily define, track and modify motion over time.

Similar to us, several existing approaches enable controllable editing of motion but with different priors that do not enable physically-based interactions. Control over human or animal motion has been explored [LHR*21, WCS*22, JYS*22, PZX*21, XAS21, AXS*22, YVN*22, HLX*21]; however, these methods do not extract physical parameters for general scenes and use components designed specifically for humans or animals in their methods. Other approaches [ZLY*21, YSL*22] enable control over the geometry of general scenes; however, they do not discover the underlying physical motion parameters and thus rely entirely on user intervention to create plausible edits.

While primitive-based rendering has been explored for static scene reconstruction [KPLD21, LSS*21, LZ21, XXP*22], dynamic scene reconstruction has received little attention. Recently, two approaches have been developed for particle-based 3D reconstruction of dynamic scenes [LQC*23, ACDS22]. PAC-NeRF [LQC*23] uses a hybrid Eulerian-Lagrangian representation to model dynamic scenes where density and color information is stored in voxel grids that can be transformed into particles. A differentiable material point method (MPM) is used to model the motion of the particles, enabling estimation of their physical properties. PAC-NeRF uses a similar method as ours to convert between the Eulerian and Lagrangian representations; however there are some key differences. In particular, initialization of the complete geometry is required before modeling the scene motion. The Lagrangian representation is initialized using a NeRF trained over observations in the first frame. This requires access to multi-view data, as the reconstruction from the first frame needs to be complete. In contrast, we use monocular video data, and cannot rely on a single-frame reconstruction as initialization. We instead use all the frames to jointly reconstruct the geometry and motion in the scene. Our particles do not live in the camera space of the first frame, but live in a canonical space that is jointly discovered. Finally, our particles are allowed to move freely, subject to weak constraints, while PAC-NeRF uses MPM that requires significant information about the materials of the objects in the scene, and the scene complexity.

ParticleNeRF [ACDS22] uses a particle-based scene representation to model the scene geometry and motion. A position-based dynamics physics model adds collision constraints for the particles. Our method also uses particles to represent the scene motion. However, unlike these approaches that rely on synchronous *multi-view* *video* observations, our method only relies on a monocular training video. Further, we do not add any constraints on the kinds of materials allowed in the scenes, unlike PAC-NeRF, and enable the synthesis of novel controllable motion, unlike ParticleNeRF.

### 2.2. Modal analysis and synthesis

Modal analysis and synthesis have been used extensively in graphics and animation [PW89, NMK*06, JP02]. Modal analysis can be used to extract the periodic components of the motion in a scene by performing a Fourier Transform on the optical flow of a single viewpoint video sequence. These periodic components have been used to estimate an objects's material properties [DBC*15], synthesize sound from a 3D shape [JLQ*20], extract ambient sounds from a muted video [DRW*14] or deform a 3D mesh [HSO03]. The approach works best on scenes that are subject to an harmonic vibration. This can range from springs to trees or clothing moving in the wind.

We focus on the full 3D modal synthesis of oscillating objects captured in a video sequence from a moving camera. Davis et al. [DCD15] take video from a static camera as input, and propose the synthesis of new physically based motion from the properties extracted with modal analysis based on optical flow. Instead of relying on optical flow [Far03] to estimate the motion of a scene, we will estimate the deformation field of a dynamic NeRF, providing the required 3D motion, allowing depth and occlusion handling and novel view synthesis.

## 3. Lagrangian deformable radiance field

Existing dynamic NeRF solutions [TTG*21, PSH*21] typically store deformation in a MLP that can be seen as an Eulerian deformation of the sampling space. We show that our new particle-based Lagrangian representation is easier to edit and allows for efficient analysis and synthesis of the physical properties of the scene.

Radiance fields, and especially *neural radiance fields* represent view-dependent effects in a 3D scene as a mapping between the position in space $\mathbf{x}$, view direction $\theta$ and the corresponding density $\sigma(\mathbf{x})$ and color $\mathbf{c}(\mathbf{x},\theta)$. Rendering of the scene is performed with volumetric ray-marching. The color of a pixel $\mathbf{p}$ with view direction $\theta$ is computed as follows, with $s$ the distance along the ray:

$$c(\mathbf{p},\theta) = \int_0^\infty \tau(s)\,\mathbf{c}(\mathbf{p}+s\,\theta,\theta)\left(1-e^{-\sigma(\mathbf{p}+s\,\theta)}\right)\mathrm{d}s, \qquad (1)$$

where:

$$\tau(s) = \exp\left(-\int_0^s \sigma(\mathbf{p}+s'\,\theta)\right)\mathrm{d}s'. \qquad (2)$$

Such radiance fields are trained with standard stochastic gradient descent methods, minimizing the error between the predicted images and a set of user-provided, pre-calibrated input images. In our case, our inputs are all the frames of a monocular video, and the corresponding camera poses are extracted using SfM (e.g., with Colmap [SF16]).

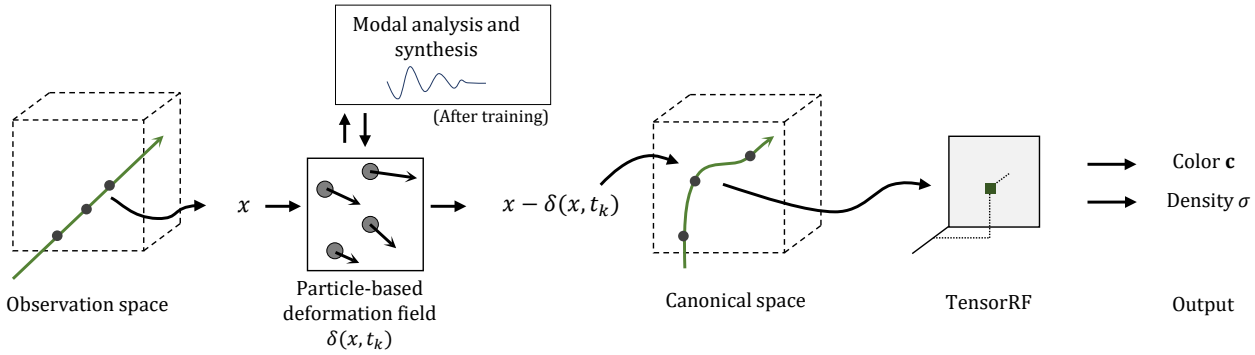


**Figure 2:** *Our pipeline for the particle-based deformable tensorf (in the particle box). Rays in the observation space are first deformed to the canonical space using our particle-based deformation field. Then we query the radiance field to get the color and density of each point along the ray. After training, we use the deformation field for modal analysis and synthesis. Outside the particle box, the deformation is assumed to be null and TensoRF additionally takes as input the view direction* $\theta$.

### 3.1. Particle-based deformation of radiance fields

Dynamic NeRF solutions [PSB*21, PSH*21, TTG*21] decouple the deformation and appearance of the scene. The appearance is handled in a standard NeRF-like fashion, in a *canonical* space that represents the original, undeformed version of the scene. The deformation $\delta$ is an auxiliary space-time field that warps the space between a frame $k$ and the canonical space: it is used in the rendering equation (Eq. 1) by moving any sampling point $\mathbf{x}$ in the space of frame $k$ to the warped position in canonical space, i.e., $\mathbf{x} - \delta(\mathbf{x}, t_k)$.

We can see this as an Eulerian representation of the motion in the scene: at each point $\mathbf{x}$ in space and time $t_i$, the deformation field expresses the origin of the information that arrives at that point in canonical space. Although this representation is straightforward to interface with the rendering equation of NeRF, it poses significant challenges in our case, where we need to analyze and synthesize the motion through physical laws defined in object space, which is inherently Lagrangian.

Instead, we propose to represent motion, not as a deformation of the space, but as the displacement of particles. The particles are seeded in canonical space and contain a displacement toward their position in each frame $t_k$. The advantage of this representation is that the displacement can be easily manipulated to obtain the physical properties of the scene and synthesize new, physically consistent motion. The main advantage compared to Eulerian representations is that we can directly define the motion of particles in canonical space and directly track them at each frame thanks to the *forward* mapping. For a point in canonical space, forward mapping allows us to know its position at each time step. Doing this with the backward mapping of the deformation fields in previous work would be much more challenging since the backward mapping gives for each point in the observation space its origin in the canonical space.

However, we need to be able to recover the displacement from the space of a given frame to canonical space and use it in the NeRF rendering equation. To do this, we define a continuous deformation field based on the particles.

For a frame at time $t$, each particle is moved from its position in canonical space $\mathbf{p}_i^c$ to a new position $\mathbf{p}_i(t) = \mathbf{p}_i^c + \mathbf{d}_i(t)$ by a displacement $\mathbf{d}_i(t)$. Whenever we need to determine the motion for a point in space $\mathbf{x}$ (e.g., samples along rays in NeRF), we interpolate the deformation associated with the particles in the neighborhood $N(\mathbf{x})$ of $\mathbf{x}$:

$$\delta(\mathbf{x}, t) = \frac{\sum_{i \in N(\mathbf{x})} \mathbf{d}_i(t) e^{-||\mathbf{p}_i(t) - \mathbf{x}||^2}}{\sum_{i \in N(\mathbf{x})} e^{-||\mathbf{p}_i(t) - \mathbf{x}||^2}} \tag{3}$$

We accelerate the neighborhood search by storing all the particles in a regular grid and limiting the neighbors to the particles in adjacent cells. This remains valid as long as the relative displacement remains small, which is ensured during training by our spatial regularizer.

We jointly optimize the weights of the NeRF that encode the appearance and the per-particle displacement ($\mathbf{d}_i(t_k)$). We use TensoRF as a lightweight NeRF representation of the time-invariant appearance in canonical space ($\sigma$ and $\mathbf{c}$ in Eq. 1), while each particle stores a time-indexed array of displacements. During the main optimization loop, the particles are moved using the corresponding displacement for a given time $t$ to create the continuous warp field defined in Eq. 3. We use this field to warp the samples in the evaluation of the integral in Eq. 1 to its position in canonical space. The entire system is trained end-to-end, relying on backpropagation to update the gradients of both the TensoRF weights and the particle displacements.

### 3.2. Training

The particles are sampled regularly in canonical space. We optimize the number of particles and particle spacing by limiting their positions to a box around the object, which we choose as the maximal bounding box of all the possible configurations of the moving object after 1h of training. We fix the canonical positions of the particles $\mathbf{p}_i^c$ and optimize for the displacement, initialized at $\mathbf{d}_i(t_k) = 0$.

In addition to the standard $L_2$ rendering loss using the ground truth pixel colors as supervision, we introduce regularizations on the particle displacements.

**Time regularization loss.** We observe that the motion - coming from real footage - should not allow for large changes over time. Therefore, we constrain the motion to a small velocity $d\mathbf{p}_i/dt = d\mathbf{d}_i/dt$:

$$L_{\text{time-reg}} = \frac{T}{(n-1)|P|} \sum_{t=1}^{n-1} \sum_{i \in P} ||\mathbf{d}_i(t_{k+1}) - \mathbf{d}_i(t_k)||_1 \quad (4)$$

With $P$ the set of particles and $\mathbf{d}_i(t)$ the displacement of a particle $i$ at frame $t$. The frames are numbered from 1 to $n$, for a total video length of $T$ seconds.

**Space regularization loss.** We target the reconstruction of oscillatory objects, which are almost rigid: the motion should not exhibit large spatial changes. Other works used a divergence loss [TTG*21]; we found that while this regularization effectively reduced local changes in volume, it did not prevent abrupt shearing or orthogonal deformations. Therefore, instead of using the divergence loss, we prefer a simplified elastic potential energy:

$$L_{\text{space-reg}} = \frac{1}{dx^2|P|} \sum_{i \in P} \sum_{j \in N_i} ||\mathbf{d}_i(t_k) - \mathbf{d}_j(t_k)||_2^2. \quad (5)$$

This formulation benefits from the regular sampling of the particles in the canonical space, with an inter-distance of $dx$. The set $N_i$ contains the 8 nearest neighbors of particle $i$ in canonical space, and the frame at time $t_k$ is randomly chosen at each iteration.

In static radiance field methods, the view direction affects the color of the points to account for specular effects. In our case, we have found that this feature was competing with the deformation field to explain the motion. As a first approximation, we disabled all view-dependent effects inside the box containing the particles.

## 4. Modal analysis and synthesis

Modal analysis makes the implicit assumption that a superposition of harmonic oscillators can explain the motion in the scene. Many types of motion respect this assumption: plants or clothing moving in the wind, springs, or pendulums. Conceptually, we approximate the scene and its motion as a set of points linked to each other with springs. Each spring has different rigidity and damping, and each point has a different mass. Without external forces, the system tends to return to a rest state, and the displacement $\mathbf{d}(t) = \{\mathbf{d}_i(t), i \in P\}$ from the rest state is derived from the conservation of momentum:

$$\mathrm{M}\ddot{\mathbf{d}}(t) + \mathrm{C}\dot{\mathbf{d}}(t) + \mathrm{K}\mathbf{d}(t) = \mathbf{f}(t), \quad (6)$$

where $P$ is the set of discrete elements of the system (here particles), $\dot{\mathbf{d}}$ denotes the time derivative of $\mathbf{d}$ and $\ddot{\mathbf{d}}$ its second derivative, $\mathbf{f}(t)$ stands for the external force applied to the system at time t, M, C and K are the parameter matrices for the mass, damping, and rigidity respectively [Sha91].

This law can be expressed in a modal basis, where these matrices are diagonal, reducing this equation to a set of $|P|$ independent equations (here we factor the mass into the other parameter and forces:)

$$\ddot{q}_i(t) + c_i\dot{q}_i(t) + k_i q_i(t) = f_i(t), \quad (7)$$

where $m_i$, $c_i$ and $k_i$ are the elements of the diagonalized matrices, while $q_i$ and $f_i$ are the elements of the vectors $\mathbf{d}$ and $\mathbf{f}$ transformed to the modal space. Note that we do not explicitly compute the matrices M, C and K nor the diagonalization matrices, but we rely on simple observations to transfer information from and to modal space. A solution to Eq. 7 with no external forces is [Bat06]:

$$q_i(t) = a_i e^{-c_i t} \sin(\omega_i t). \quad (8)$$

This result means that the motion of one point is the result of the sum of periodic displacements of amplitude $a_i$, damping $c_i$ and *damped natural frequency* $\omega_i$. We can extract the values of the parameters $a_i$ and $\omega_i$ by examining the motion of the scene, which is directly stored in the displacement of the particles trained in Sec. 3 if we assume that the canonical space of our deformable NeRF is the rest state for the system.

### 4.1. Mode Extraction

We perform a discrete-time Fourier transform independently on each $(x, y, z)$ component of the deformation of each particle. Then, the user picks the frequencies that are the most common across particles, similarly to Davis et al. [DCD15], and stores their indices in a set $F$. The intuition is that these frequencies are the most representative of the physical motion, and therefore that the corresponding *modal displacements* $\{q_i(t), i \in F\}$ are sufficient to simulate a similar movement. The very small number of selected modes (typically 1-10) makes this method an efficient model reduction.

Modal displacements can be interpreted as the displacement of a part of the object, with local weights given by the amplitudes in the Fourier spectrum. We denote the weights as $\mathbf{w}_{j \to i}$, or the influence of the frequency of the mode $j$ on particle $i$. Therefore, we express the displacement of a particle $i$ as the weighted sum of all the modal displacements:

$$\mathbf{d}_i(t) = \sum_{j \in F} \mathbf{w}_{j \to i} q_j(t) \quad (9)$$

### 4.2. ModalNeRF Synthesis

From the trained canonical radiance field and extracted mode, we can efficiently synthesize new physically-based motion from user interactions, by solving Eq. 7 in modal space. At each time-step, we can reproject the new modal displacements as particle displacements with Eq. 9, essentially recreating a new continuous deformation field (Eq. 3) to re-render the scene without any retraining.

The user sets the external forces $f(t)$ at different points in time and the initial modal deformation $q_i(0)$; otherwise initialized at 0. We further initialize the deformation velocity to $\dot{q}_i(0) = 0$ and solve

Eq. 7 with finite differences, discretizing time by small steps of duration $dt$.

First, we simplify the momentum equation (Eq. 7) following [DCD15] with the assumption of Rayleigh damping, which leads to $k_i = \omega_i^2 + c_i^2$, and update the deformation velocity from the previous time step at $t - dt$:

$$\dot{q}_i(t) = \dot{q}_i(t - dt) + dt \left( f_i(t) - \left( \omega_i^2 + c_i^2 \right) q_i(t) - c_i \dot{q}_i(t) \right), \quad (10)$$

and update the modal defomation with sympleptic Euler:

$$q_i(t) = q_i(t - dt) + dt\, \dot{q}_i(t) \quad (11)$$

The damped natural frequency $\omega_i$ was computed during mode extraction, while $c_i$ is the damping parameter of the mode. Modal analysis does not allow us to compute $c_i$, therefore this parameter is provided by the user. We found that the synthesized motions are more plausible when increasing the damping coefficient $c_i$ for higher frequencies $\omega_i$. These coefficients can be used to change the stiffness of the movement, providing more editability.

## 5. Results and Evaluation

We implemented our method in *PyTorch* and used a single NVidia A6000 GPU for training and motion synthesis. We will release all our code and data, which will be available at: *URL will be updated upon acceptance.*

We built our implementation on top of on the TensoRF codebase and we use the original hyper-parameters for the static NeRF part: Adam Optimizer with an initial learning rate of 0.02, training on 4096-pixel rays at each step, and using a small MLP with two fully-connected layers (with 128-channels) and ReLU activation for the view dependency. We jointly train TensoRF and the displacement of the particles, therefore we need significantly more iterations than what TensoRF uses for a static scene (120k iterations in our case, for approximately 5 hours). We found that the training time descreased if we delay the upsampling of the tensor grid so that the training spends more iterations on lower resolution radiance fields.

We ran our method on one synthetic scene, to validate the analysis step and have a simple baseline test case, and we also captured 3 real scenes with varying types of motion. The synthetic scene is a sphere bouncing in mid air with a specific frequency. PLANTA and PLANTB are plastic plants with a fan blowing on them producing motion, and RULER is an oscillating ruler (see Fig. 3 and Fig. 1 for PLANTB, see supplemental videos for a better appreciation of the synthesis).

For each scene, we selected the coordinates of a box around the moving object and sampled all the particles inside that box. Boxes contain approximately 400 particles regularly sampled. For better results, the weights of the regularizers have been manually set. The spatial regularizer loss weight is at 0.1 for all three real scenes while it is at 1e-5 for the synthetic one. Both plant scenes have a time regularizer weight of 1e-5 while the RULER and SYNTHETICBALL have a weight of 0.1. We use a learning rate of 3e-4 on the particles, decayed to 0.01 of its initial value alongside all other learning rates.

For efficiency, TensoRF skips evaluating color (Eq 1) using both an occupancy grid and by thresholding on the color's weights. We disable both of these checks inside the particle box.

### 5.1. Results

We first present results for motion reconstruction, and then results of modal analysis and synthesis which are our main contributions. All results are best viewed in the supplemental video where motion is easier to see and understand.

**Motion Reconstruction.** In Fig. 3, we show two rows for the four scenes. Above we show frames from the input video, and below we show renderings using our modified TensoRF that captures the motion. To better appreciate the captured motion, please see the supplemental video, where the motion is easier to see.

Please note that we are not attempting to improve motion capture in NeRF; this is not our contribution. Our goal is to allow modal analysis and synthesis with NeRF; we found that modal analysis requires a novel representation of motion in the form of our particle-based approach. Nonetheless, our method is competitive with (and in some cases even better than) existing methods, even for motion capture (see Sec. 5.2).

**Modal Analysis.** We performed modal analysis on the scenes. In Fig. 4 we show the result of modal analysis for the SYNTHETIC BALL, the RULER and the PLANTA scenes. The graphs show two accurately extracted periodic motions. The main frequencies found for the Synthetic Ball and for the (real) Ruler scenes are the correct frequencies of motion of the object in the video (respectively 1 and 0.7 Hz, measured from the video). This allows easy and meaningful manipulation of the motion in the scene. For PLANTA the modal analysis is more complex since more frequencies are present in the motion. For SYNTHETIC BALL and RULER, only one mode was selected since there is only one frequency in the video. For PLANTA and PLANTB, 2 and 3 modes were selected respectively.

To further test the ability of our approach to extract the frequency of motion, we modified the synthetic scene to have $\frac{1}{2}$, x2, and x4 the frequency of the original scene. We show the results in Fig. 5; we can see that our method extracts the correct frequencies accurately, demonstrating the ability to accurately model motion based on physics.

**Free-Viewpoint Modal Synthesis.** To perform synthesis, we start by selecting dominant modes for each scene and apply two types of motion manipulation. We first apply external forces $f(t)$ at some given times $t$. The forces $f(t)$ are defined in modal space, *i.e.,* one for each mode, which means that the user can chose to apply forces to some parts of the object only. As a second type of control, we select a particle $i$ and pull it on a user-given direction $\mathbf{d}_{\text{pull}}$. In that case, we do not set external forces but change the initial modal deformations to:

$$q_j = -\mathbf{d}_{\text{pull}} \cdot \mathbf{w}_{j \to i} \quad (12)$$

We show the result of these manipulations in Fig. 6 and the supplemental video.

As can be seen in the supplemental video, by applying forces

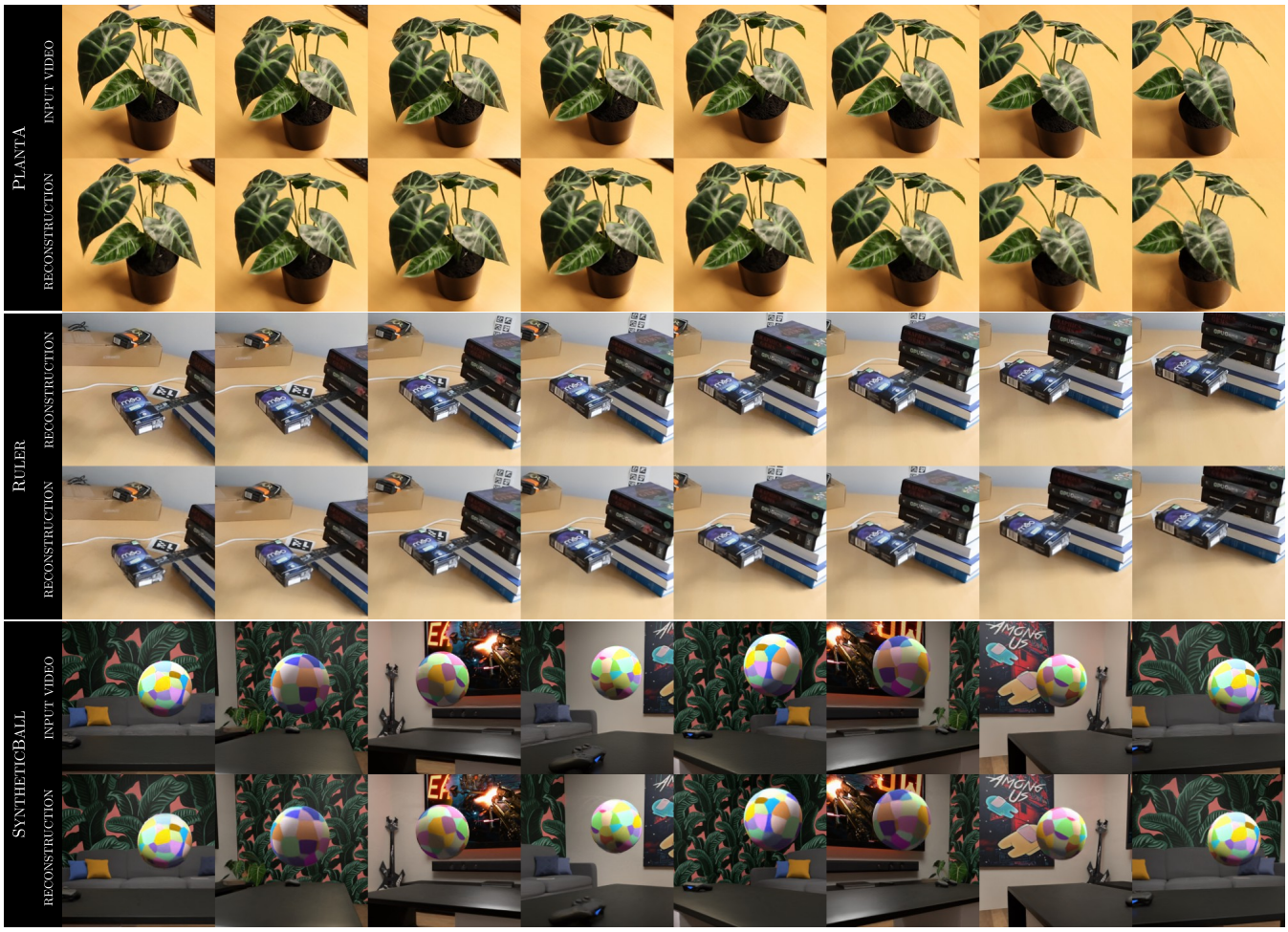


**Figure 3:** *Three of our test scenes:* PLANTA, RULER *and* SYNTHETICBALL. *For each scene the top row shows frames of the input video and below are renderings of the same motion with our method.*

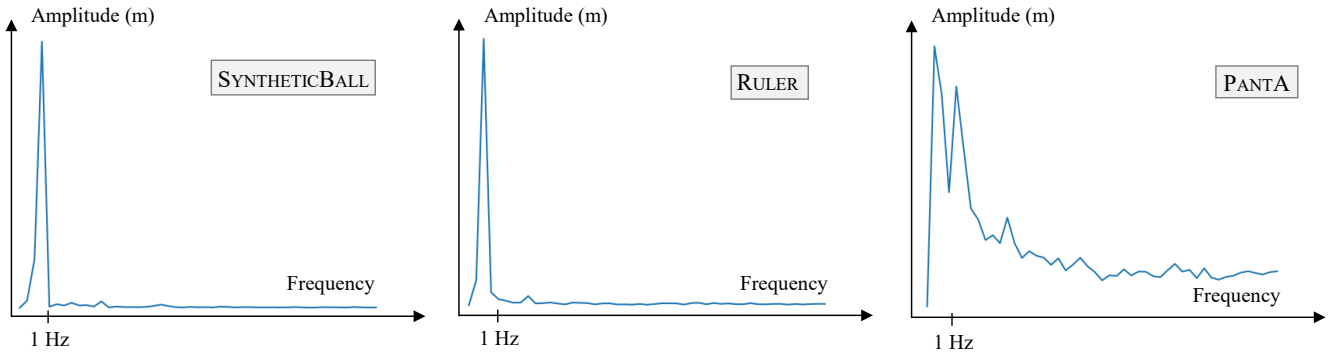


**Figure 4:** *Modal analysis results for the* SYNTHETIC BALL, RULER *and* PLANTA *scenes. Note how our method correctly identifies the single frequency of the first two motions. The third has a more complex profile.*

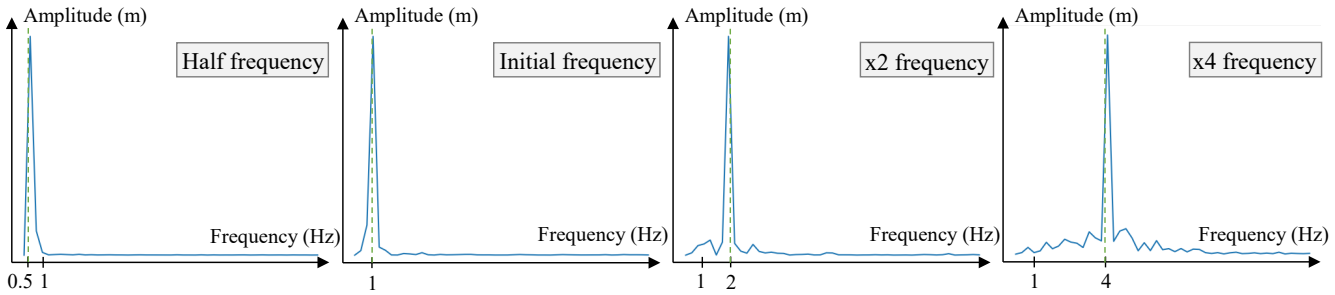


**Figure 5:** *Extracting different motion frequencies for modified versions of* SYNTHETIC BALL *scenes. We see that our method can accurately extract the physically correct frequency of the ball motion.*

and manipulating the modes, we are able to significantly change the velocity of the different moving objects in the scenes, creating convincing exaggerated motion effects. The pre-trained canonical radiance field allows us to interact with the scene from any new viewpoint, consistently handling the 3D geometry and occlusions in the scene. Our method is the first to allow both camera motion and physics-based synthesis from a simple monocular video.

### 5.2. Evaluation and Comparisons

We provide two quantitative evaluations of the quality of motion reconstruction using PSNR. Our first evaluation uses image error with respect to the input frames to evaluate the quality of the motion.

We ran our method, NR-NeRF [TTG*21] and also two very recent methods [GCD*22], [FYW*22]. Previous methods essentially fail when interpolating to unseen frames despite best effort parameter adjustment; we thus report PSNR only on the training set. As we can see, our method is on par or in some cases better than previous methods for motion capture, and allows physics-based motion synthesis.

| | BALL (PSNR↑) | PLANTA (PSNR↑) |
|---|---|---|
| Ours | 37.85 | 33.54 |
| NR-NeRF [TTG*21] | 23.93 | 32.34 |
| Guo et al. [GCD*22] | 25.02 | 30.62 |
| Fang et al. [FYW*22] | 28.01 | 34.10 |

**Table 1:** *Quantitative results on motion capture computed on the* BALL *and* PLANTA *scenes.*

In our second quantitative evaluation, we compare our reconstruction to NR-NeRF [TTG*21] on the synthetic scene. We use NR-NeRF as a representative method for other similar dynamic reconstruction methods such as [PSB*21,PCPMMN20]. Specifically, we place 6 points on the surface of the moving sphere in the synthetic scene, and export their ground truth positions in each frame. For each method, we evaluate the position of these in the canonical space at every frame; in a perfect reconstruction, their position in the canonical space would be identical. For our method, the average variation of the position of the points from their mean canonical position is 0.05, while for NR-NeRF, the displacement is 0.61. For reference, the sphere has a radius of 0.864. Unlike the task of novel view synthesis where inaccurate motion and geometry can still lead to plausible results, our task requires accurate motion reconstruction for modal analysis. This evaluation confirms that our method can accurately capture 3D motion. In Fig. 7, we show the motion estimated by NR-NeRF, which can be compared to the first example in Fig. 3. We note that NR-NeRF does not work well for specular scenes in the monocular setting.

Finally, we have performed several ablations to determine the effect of the parameters of our method. In most cases, the choice of parameters have a small effect on metrics. We report PSNR for various ablations on the Plant A scene in Tab. 2.

### 5.3. Limitations

The modal analysis relies on the assumption that the observed motion is an harmonic vibration. This means that our method can be only used on specific scenes including plants in the wind, pendulums or pieces of clothing. It does not work on non-perdiodic motions such as humans, falling objects or liquids for example.

We disable view dependent effects wherever motion is expected. This prevents us from handling moving specular objects. The ambiguity between motion and view-dependent effect is a common issue with non-rigid NeRFs and we hope to solve this in future work by leveraging physical priors during training. Non-rigid NeRFs also tend to fail when the frame to frame motion is to large. This is why we limit ourselves to scenes with small motion or to captures with a high framerate. For examples of these issues, please refer to the Appendix.

Another limitation is that the synthesized motion is projected on the entire scene without any knowledge of the geometry. Pulling on a leaf will make all the leaves move immediately. In cases where there are two independent objects that vibrate at a similar frequency, applying a force to one object will make the other object move as well. This can be solved by the user manually separating independent objects of the scene.

Due to the interpolation, extreme deformations can lead to a discontinuous field that creates visual artifacts. Fortunately, they are prevented by the spatial regularizer. However, this puts a limit to the forces a user can apply before breaking the geometry of the scene.


**Figure 6:** *Our four test scenes:* PLANTA, RULER, PLANTB *and* SYNTHETICBALL. *For each scene the top row shows frames of the input video and below are renderings of the motion manipulated with our method.*

The particle bounding box has to be manually placed. This can lead to shearing artefacts if the box is not carefully placed.

## 6. Conclusion

We present a new Lagrangian radiance-field-based representation to capture motion in a scene captured with a single handle-held video camera, focusing on mainly oscillating motion. We introduce a particle-based representation that creates a continuous representation of the motion in the scene over the video sequence. This particle-based representation allows us to perform modal analysis of the reconstructed motion, and finally we can manipulate the extracted modes to modify the motion in a physically-based manner. Ours is the first solution that allows novel-view synthesis with *physically-based motion manipulation* in a scene captured by a single hand-held camera.

Directions for future work include expanding the types of motion we can represent with our approach, and using the physics-based representation to improve the quality and speed of the motion reconstruction.


**Figure 7:** *Motion reconstruction for the same sequence as Fig. 3 using NR-NeRF [TTG*21]. First row is ground truth, second row is our method and third row is the output of NR-NeRF. Our method can capture higher-quality motion and geometry.*

| | PlantA | | | PlantB | | | Pendulum | | | Ball | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Density of particles | $\times 1/4$ | $\times 1$ | $\times 2$ | $\times 1/4$ | $\times 1$ | $\times 2$ | $\times 1/4$ | $\times 1$ | $\times 2$ | $\times 1/4$ | $\times 1$ | $\times 2$ |
| PSNR↑ | 31.41 | 32.08 | 32.13 | 33.66 | 33.87 | 33.74 | 30.47 | 30.23 | 29.0 | 37.7 | 27.5 | 29.42 |
| Time Reg. | 0.01 | 1.0 | 10.0 | 0.01 | 1.0 | 10.0 | 0.01 | 1.0 | 10.0 | 0.01 | 1.0 | 10.0 |
| PSNR↑ | 32.12 | 32.0 | 32.13 | 33.83 | 33.84 | 33.87 | 29.96 | 29.99 | 29.98 | 29.89 | 29.9 | 29.85 |
| Deformation Reg. | 1.0 | 100.0 | 1000.0 | 1.0 | 100.0 | 1000.0 | 1.0 | 100.0 | 1000.0 | 1.0 | 100.0 | 1000.0 |
| PSNR↑ | 32.93 | 31.40 | 31.03 | 34.04 | 33.89 | 33.85 | 30.43 | 30.16 | 29.97 | 31.58 | 30.92 | 29.88 |

**Table 2:** *Quantitative comparisons of parameters ablation on* PLANTA, PLANTB, PENDULUM *and* BALL *scenes. For each scene, the middle collumn shows the default parameters.*

### Acknowledgements

This research was funded by the ERC Advanced Grant FUNGRAPH (No.788065, http://fungraph.inria.fr). The authors are grateful to Adobe for generous donations and the OPAL infrastructure from Université Côte d'Azur for providing resources and support. The authors would also like to thank the anonymous reviewers for their valuable feedback.

## Appendix A: Examples of failure cases and limitations

### View dependent effects

As in previous works, we choose to disable the view direction input to the MLP for all points within the bounding box of the motion. We found that this was necessary for a good geometric reconstruction, as shown in Fig. 8: when the view direction is included for moving objects, the network learns to simulate the motion by creating false geometry and adapting its color to the viewpoint. Excluding the view direction for moving objects fixes this problem. Despite this correction, our method can still recover the motion in scenes with specular highlights (for instance plants where the leaves are highly specular). In Fig. 9, we test an extreme case, where the pendulum is a given a mirrored finish. We observe more artifacts, but the motion is still recovered.

### Large frame to frame motions

In cases where the observed motion has a large amplitude, the difference between succiessive images in the input increases, and the quaility of the reconstruction decreases. As we can see in Fig. 10 the method reconstructs multiple spheres along the trajectory of the pendulum. The deformation field then compresses and expands these spheres in turn to give the illusion of movement. In the figure we can observe the artifacts left by the compressed spheres.

### Synthesis limitation

The modal synthesis strictly restricts motions in the degrees of freedom of the ones observed. For example, if the input video shows a pendulum oscillating from left to right, then in all synthesized motion the pendulum will oscillate along the same axis. Although this might be a limitation in some cases, for example, if this prevents an interaction that would seem natural, this constraint can also be considered as a strong regularization that prevents highly non-physical states.

### References

[ACDS22] ABOU-CHAKRA J., DAYOUB F., SÜNDERHAUF N.: Particlenerf: Particle based encoding for online neural radiance fields in dynamic scenes. *arXiv preprint arXiv:2211.04041* (2022). 3

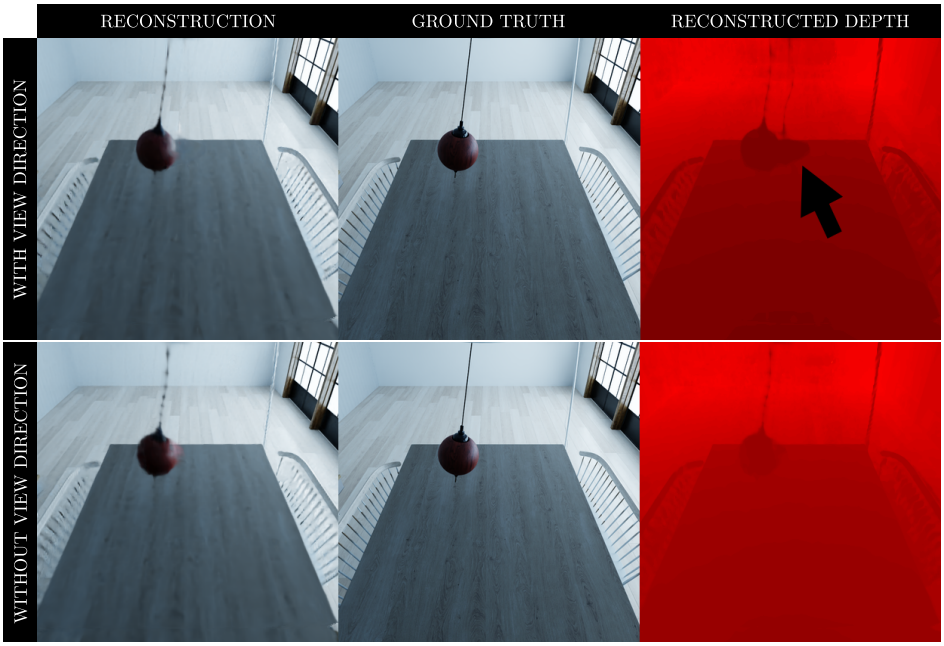


**Figure 8:** *View direction enabled for the whole scene (top row). While the reconstruction (left) is visually acceptable compared to the ground truth (middle), we observe in the depth map (right) that with view direction enabled, the model interpreted the motion as a change of color in a hallucinated extra piece of geometry. Without view-dependent effects (bottom), the reconstruction slightly degrades but the geometry is more consistent which is necessary for the modal analysis and synthesis.*

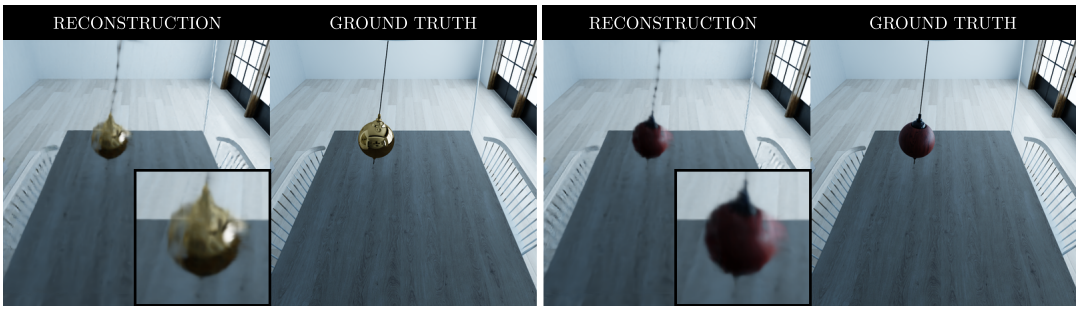


**Figure 9:** *Even without view-dependent effects, the pendulum shaded as a mirror (left) is reconstructed at the cost of some artifacts. For reference, we compare it to a less specular case (right).*

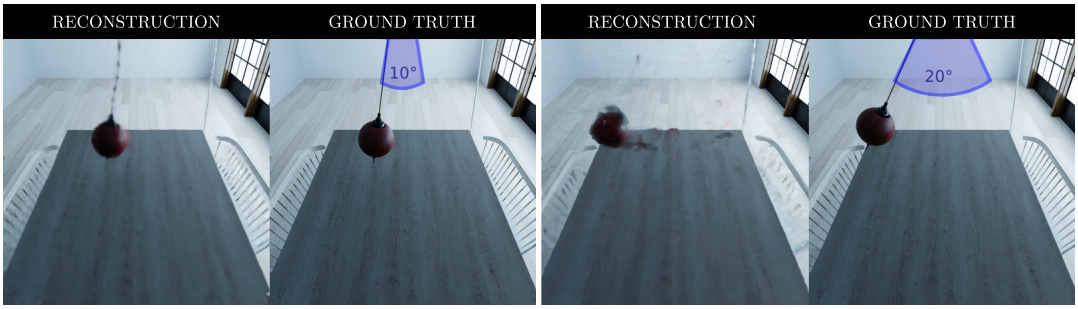


**Figure 10:** *Small motions (left) are accurately reconstructed by our method. Larger motions (right) increase the frame-to-frame differences, which degrades the reconstruction.*

[AXS*22] ATHAR S., XU Z., SUNKAVALLI K., SHECHTMAN E., SHU Z.: Rignerf: Fully controllable neural 3d portraits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 20364–20373. 3

[Bat06] BATHE K.: *Finite Element Procedures*. Prentice Hall, 2006. URL: https://books.google.fr/books?id=rWvefGICfO8C. 5

[CXG*22] CHEN A., XU Z., GEIGER A., YU J., SU H.: Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)* (2022). 2

[DBC*15] DAVIS A., BOUMAN K. L., CHEN J. G., RUBINSTEIN M., DURAND F., FREEMAN W. T.: Visual vibrometry: Estimating material properties from small motion in video. In *Proceedings of the ieee conference on computer vision and pattern recognition* (2015), pp. 5335–5343. 2, 3

[DCD15] DAVIS A., CHEN J. G., DURAND F.: Image-space modal bases for plausible manipulation of objects in video. *ACM Transactions on Graphics (TOG) 34*, 6 (2015), 1–7. 2, 3, 5, 6

[DRW*14] DAVIS A., RUBINSTEIN M., WADHWA N., MYSORE G. J., DURAND F., FREEMAN W. T.: The visual microphone: Passive recovery of sound from video. 3

[Far03] FARNEBÄCK G.: Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis* (2003), Springer, pp. 363–370. 3

[FKYT*22] FRIDOVICH-KEIL S., YU A., TANCIK M., CHEN Q., RECHT B., KANAZAWA A.: Plenoxels: Radiance fields without neural networks. 2

[FYW*22] FANG J., YI T., WANG X., XIE L., ZHANG X., LIU W., NIESSNER M., TIAN Q.: Fast dynamic radiance fields with time-aware neural voxels. *arXiv preprint arXiv:2205.15285* (2022). 2, 8

[GCD*22] GUO X., CHEN G., DAI Y., YE X., SUN J., TAN X., DING E.: Neural deformable voxel grid for fast optimization of dynamic view synthesis. In *Proceedings of the Asian Conference on Computer Vision* (2022), pp. 3757–3775. 2, 8

[GSKH21] GAO C., SARAF A., KOPF J., HUANG J.-B.: Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5712–5721. 2

[GXH*22] GAN W., XU H., HUANG Y., CHEN S., YOKOYA N.: V4d: Voxel for 4d novel view synthesis. *arXiv preprint arXiv:2205.14332* (2022). 2

[HLX*21] HABERMANN M., LIU L., XU W., ZOLLHOEFER M., PONS-MOLL G., THEOBALT C.: Real-time deep dynamic characters. *ACM Transactions on Graphics (ToG) 40*, 4 (2021), 1–16. 3

[HSO03] HAUSER K. K., SHEN C., O'BRIEN J. F.: Interactive deformation using modal analysis with constraints. In *Graphics Interface* (2003), vol. 3, pp. 16–17. 2, 3

[JLQ*20] JIN X., LI S., QU T., MANOCHA D., WANG G.: Deep-modal: real-time impact sound synthesis for arbitrary shapes. In *Proceedings of the 28th ACM International Conference on Multimedia* (2020), pp. 1171–1179. 3

[JP02] JAMES D. L., PAI D. K.: Dyrt: Dynamic response textures for real time deformation simulation with graphics hardware. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (2002), pp. 582–585. 3

[JYS*22] JIANG W., YI K. M., SAMEI G., TUZEL O., RANJAN A.: Neuman: Neural human radiance field from a single video. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII* (2022), Springer, pp. 402–418. 3

[KPLD21] KOPANAS G., PHILIP J., LEIMKÜHLER T., DRETTAKIS G.: Point-based neural rendering with per-view optimization. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 29–43. 3

[KYK*21] KANIA K., YI K. M., KOWALSKI M., TRZCINSKI T., TAGLIASACCHI A.: Conerf: Controllable neural radiance fields. *CoRR abs/2112.01983* (2021). URL: https://arxiv.org/abs/2112.01983, arXiv:2112.01983. 3

[LHR*21] LIU L., HABERMANN M., RUDNEV V., SARKAR K., GU J., THEOBALT C.: Neural actor: Neural free-view synthesis of human actors with pose control. *ACM Transactions on Graphics (TOG) 40*, 6 (2021), 1–16. 3

[LNSW21] LI Z., NIKLAUS S., SNAVELY N., WANG O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 6498–6508. 1

[LQC*23] LI X., QIAO Y.-L., CHEN P. Y., JATAVALLABHULA K. M., LIN M., JIANG C., GAN C.: PAC-neRF: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. In *The Eleventh International Conference on Learning Representations* (2023). URL: https://openreview.net/forum?id=tVkrbkz42vc. 3

[LSS*21] LOMBARDI S., SIMON T., SCHWARTZ G., ZOLLHOEFER M., SHEIKH Y., SARAGIH J.: Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (ToG) 40*, 4 (2021), 1–13. 3

[LZ21] LASSNER C., ZOLLHÖFER M.: Pulsar: Efficient sphere-based neural rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2021). 3

[MESK22] MÜLLER T., EVANS A., SCHIED C., KELLER A.: Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989* (2022). 2

[MST*20] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision* (2020), Springer, pp. 405–421. 2

[NMK*06] NEALEN A., MÜLLER M., KEISER R., BOXERMAN E., CARLSON M.: Physically based deformable models in computer graphics. In *Computer graphics forum* (2006), vol. 25, Wiley Online Library, pp. 809–836. 3

[PCPMMN20] PUMAROLA A., CORONA E., PONS-MOLL G., MORENO-NOGUER F.: D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020). 1, 2, 8

[PSB*21] PARK K., SINHA U., BARRON J. T., BOUAZIZ S., GOLDMAN D. B., SEITZ S. M., MARTIN-BRUALLA R.: Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5865–5874. 1, 2, 4, 8

[PSH*21] PARK K., SINHA U., HEDMAN P., BARRON J. T., BOUAZIZ S., GOLDMAN D. B., MARTIN-BRUALLA R., SEITZ S. M.: Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph. 40*, 6 (dec 2021). 2, 3, 4

[PW89] PENTLAND A., WILLIAMS J.: Good vibrations: Modal dynamics for graphics and animation. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques* (1989), pp. 215–222. 3

[PZX*21] PENG S., ZHANG Y., XU Y., WANG Q., SHUAI Q., BAO H., ZHOU X.: Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 9054–9063. 3

[SF16] SCHONBERGER J. L., FRAHM J.-M.: Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016). 3

[Sha91] SHABANA A. A.: *Theory of vibration*, vol. 2. Springer, 1991. 5

[TTG*21] TRETSCHK E., TEWARI A., GOLYANIK V., ZOLLHÖFER M., LASSNER C., THEOBALT C.: Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular

video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 12959–12970. 1, 2, 3, 4, 5, 8, 10

[TTM*22] TEWARI A., THIES J., MILDENHALL B., SRINIVASAN P., TRETSCHK E., YIFAN W., LASSNER C., SITZMANN V., MARTIN-BRUALLA R., LOMBARDI S., ET AL.: Advances in neural rendering. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 703–735. 2

[WCS*22] WENG C.-Y., CURLESS B., SRINIVASAN P. P., BARRON J. T., KEMELMACHER-SHLIZERMAN I.: Humannerf: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 16210–16220. 3

[XAS21] XU H., ALLDIECK T., SMINCHISESCU C.: H-nerf: Neural radiance fields for rendering and temporal reconstruction of humans in motion. *Advances in Neural Information Processing Systems 34* (2021), 14955–14966. 3

[XHKK21] XIAN W., HUANG J.-B., KOPF J., KIM C.: Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 9421–9431. 2

[XXP*22] XU Q., XU Z., PHILIP J., BI S., SHU Z., SUNKAVALLI K., NEUMANN U.: Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5438–5448. 3

[YSL*22] YUAN Y.-J., SUN Y.-T., LAI Y.-K., MA Y., JIA R., GAO L.: Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 18353–18364. 3

[YVN*22] YANG G., VO M., NEVEROVA N., RAMANAN D., VEDALDI A., JOO H.: Banmo: Building animatable 3d neural models from many casual videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 2863–2873. 3

[ZLY*21] ZHANG J., LIU X., YE X., ZHAO F., ZHANG Y., WU M., ZHANG Y., XU L., YU J.: Editable free-viewpoint video using a layered neural representation. *ACM Transactions on Graphics (TOG) 40*, 4 (2021), 1–18. 3