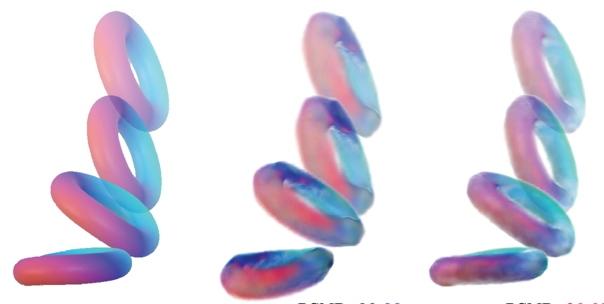
Improving Physics-Augmented Continuum Neural Radiance Field-Based Geometry-Agnostic System Identification with Lagrangian Particle Optimization

Takuhiro Kaneko
NTT Corporation

Abstract

Geometry-agnostic system identification is a technique for identifying the geometry and physical properties of an object from video sequences without any geometric assumptions. Recently, physics-augmented continuum neural radiance fields (PAC-NeRF) has demonstrated promising results for this technique by utilizing a hybrid Eulerian–Lagrangian representation, in which the geometry is represented by the Eulerian grid representations of NeRF, the physics is described by a material point method (MPM), and they are connected via Lagrangian particles. However, a notable limitation of PAC-NeRF is that its performance is sensitive to the learning of the geometry from the first frames owing to its two-step optimization. First, the grid representations are optimized with the first frames of video sequences, and then the physical properties are optimized through video sequences utilizing the fixed first-frame grid representations. This limitation can be critical when learning of the geometric structure is difficult, for example, in a few-shot (sparse view) setting. To overcome this limitation, we propose Lagrangian particle optimization (LPO), in which the positions and features of particles are optimized through video sequences in Lagrangian space. This method allows for the optimization of the geometric structure across the entire video sequence within the physical constraints imposed by the MPM. The experimental results demonstrate that the LPO is useful for geometric correction and physical identification in sparse-view settings.[1]



PSNR: 23.99 PSNR: 30.07

(a) Ground truth | (b) PAC-NeRF | (c) +LPO (proposed)

$E = 1.00 \times 10^6$, $\nu = 0.300$ | $E = 1.89 \times 10^6$, $\nu = 0.215$ | $E = 1.15 \times 10^6$, $\nu = 0.299$

Figure 1. Impact of the proposed Lagrangian particle optimization (LPO) in sparse-view geometric-agnostic system identification. We aim to identify the geometry and physical properties of an object from visual observations *without any geometric assumptions* in *severe* (e.g., sparse-view) settings. As shown in (b), a standard PAC-NeRF [41] has difficulty learning the geometry in a sparse-view setting (particularly, when the number of viewpoints is three). Consequently, it also fails to estimate the physical properties (Young's modulus E and Poisson's ratio μ). As shown in (c), LPO is useful for correcting this failure and succeeds in improving the identification of both geometry and physical properties.

1. Introduction

System identification is a technique used to identify the geometry and physical properties of an object based on visual observations. It has been actively studied in computer vision and physics because of its wide range of applications, including 3D reproduction, environmental understanding, and content creation. To solve this problem in a realistic environment, it is important to adequately address both the *extrinsic geometry* structure and the *intrinsic physical* properties. For example, to identify the geometry and physical properties of an object from the observation that it collides with the ground, it is necessary to understand not only the appearance and shape change in the *geometry* but also other internal factors in *physics* which are necessary to explain this phenomenon without any discrepancies over time. Generally, this problem is ill-posed and challenging to resolve. However, the emergence of powerful geometric representations by neural fields [82], 2D–3D connections by neural rendering [74], and trainable physical simulations using differentiable physical simulators [1] provide clues for solving this problem. For instance, previous studies [34, 35, 47] succeeded in estimating physical properties such as velocity, mass, friction, inertia, and elasticity directly from video sequences by combining differentiable physical simulators with differentiable renderers.

Although these studies have yielded promising results in terms of physical property estimation, their applicability is restricted by the assumption that the geometric struc-

[1] Video samples are available at https://www.kecl.ntt.co.jp/people/kaneko.takuhiro/projects/lpo/.

ture of a scene is *completely known*, which makes it difficult to apply them in practical scenarios. To eliminate this assumption, *geometry-agnostic system identification*, which is a technique for identifying the geometry and physical properties of an object from visual observations *without any geometric assumptions*, has been studied. Specifically, a pioneering model involves physics-augmented continuum neural radiance fields (PAC-NeRF) [41], which is an extension of NeRF [50] that is enforced to follow the conservation laws of continuum mechanics. PAC-NeRF obtains this functionality utilizing a hybrid Eulerian–Lagrangian representation, that is, the geometry (volume density and color fields) is represented with the Eulerian grid representations of NeRF [72], which is transformed into a material space via Lagrangian particles, and physical simulation is conducted on the particles using a material point method (MPM) (particularly, differentiable MPM (DiffMPM) [32]). Because all modules are differentiable, PAC-NeRF can be trained directly with multi-view video sequences and can optimize the geometry and physics without explicit supervision.

Although PAC-NeRF has enabled the tackling of new tasks, its notable limitation is that its performance is sensitive to the learning of the geometry from the first frames of the video sequences owing to its two-step optimization. First, the grid representations were trained with the first frames of the video sequences, and then the physical properties were optimized across the video sequences by utilizing the frozen first-frame grid representations. This limitation makes it difficult to apply it to cases in which geometry learning is difficult, for example, in a few-shot (sparse-view) setting, as shown in Figure 1(b).

One possible solution is to train the Eulerian grid representations using all video sequences and not just the first frame. However, a critical problem with this approach is that Eulerian grid representations cannot be trained with explicit physical constraints on the particles because they are optimized in Eulerian (i.e., world) space and cannot reflect all physical phenomena occurring in Lagrangian (i.e., material or particle) space; for example, they cannot propagate gradients calculated for particle positions.

To solve this problem, we propose *Lagrangian particle optimization (LPO)*, in which the positions and features (i.e., volume density and color) of the particles are optimized in *Lagrangian* space and *not* in Eulerian space. Contrary to the abovementioned possible solution, this method allows for the optimization of the geometric structure across video sequences with *explicit* physical constraints on the particles imposed by the MPM. Thus, the gradients calculated for the particle positions are reflected.

Furthermore, LPO is useful not only for geometry correction, but also for corrections to the physical identification because this task is closely related, that is, an accurate geometry is useful for accurately identifying the physical properties, and vice versa. This motivated us to utilize the geometry corrected by LPO to re-identify the physical properties and propose *iterative geometry–physics optimization* for gradually seeking the optimized states. Figure 1(c) shows the effectiveness of this method.

We evaluated the effectiveness of LPO in sparse-view settings. In particular, we first applied LPO to a pretrained PAC-NeRF and demonstrated that LPO is useful for correcting geometric structures through video sequences. Subsequently, we employed LPO in iterative geometry–physics optimization and demonstrated that LPO contributes to improving the performance of physical identification. Ablation and comparative studies were conducted to determine the importance of each component.

Our contributions can be summarized as follows:

- To improve the performance of *geometry-agnostic system identification*, we propose *LPO*, which optimizes the position and features of the particles in Lagrangian space to optimize the geometrical structures across video sequences within the physical constraints of an MPM.
- We propose *iterative geometry–physics optimization* to utilize the geometry corrected by LPO for re-identifying physical properties in an iterative manner.
- We experimentally demonstrated that LPO is useful for geometric correction and physical identification in sparse-view settings. We also provide detailed analyses and extended results in the Supplementary Material. Video samples are available at the link indicated on the first page of this manuscript.[1]

2. Related work

Neural radiance fields (NeRF). Novel view synthesis is a fundamental problem in computer vision and graphics and has been addressed in a large body of research. Recently, a substantial breakthrough has been achieved with the emergence of neural fields that represent a scene utilizing a coordinate-based network (e.g., a survey paper [82]). NeRF [50] is a representative variant of such neural fields and has attracted significant attention because of its geometrical consistency and high-fidelity novel view synthesis. Various studies have been conducted since the emergence of NeRF. These can be categorized into three topics. (1) Improvements to the image quality and expanding of the applicable settings, such as wild or few-shot settings (e.g., [2–4, 8, 11, 12, 14, 30, 33, 48, 51, 56, 65, 76, 79, 85, 87, 89, 91]), (2) speeding up the training or inference speed (e.g., [4, 6, 9, 18, 20, 26, 29, 30, 38, 40, 45, 46, 52, 53, 58, 63, 64, 72, 80, 81, 86]), and (3) incorporating other models or functionalities, such as generative models [22, 28, 71] (e.g., [5–7, 15, 23, 37, 54, 55, 59, 67, 70, 84]) and dynamics/physics (e.g., [13, 19, 24, 25, 41, 43, 57, 60, 68, 75, 90]). Among these, this study relates to the first in terms of its application to few-shot settings. This study relates to the third in terms of seeking physics-informed models. As studies on NeRF are benefiting from developments in adjacent categories, applying our ideas to other categories and models will be an interesting future research direction.

NeRF with dynamics/physics. As mentioned above, NeRFs with dynamics/physics have been actively studied (e.g., [13, 19, 24, 25, 41, 43, 57, 60, 68, 75, 90]). These studies have one characteristic in common: they learned time-varying neural fields from video sequences. These can be roughly categorized into two models: (1) non- (or weak) physics-informed models (e.g., [19, 25, 43, 57, 60, 68, 75, 90]) and (2) physics-informed models (e.g., [13, 24, 41]). The advantage of the first is that it can be applied to scenes or objects that are difficult to describe physically; however, its disadvantage is that it requires a large amount of training data to learn dynamics from scratch, and the learned representations are not necessarily interpretable because they are not based on physics. The advantage of the second is that it can obtain interpretable representations based on physics; however, the disadvantage is that its application is limited to physically describable objects or scenes. In particular, PI-NeRF [13] targets smoke scenes and does not address the boundary conditions; therefore, it cannot handle solid or contact materials. NeuroFluid [24] focuses on fluid dynamics grounding and solves it using an intuitive physics-based approach in which formal instruction in physics is not explicitly defined. In contrast, PAC-NeRF [41] is based on a principled and interpretable physical simulator and can be applied to various materials, including Newtonian/non-Newtonian fluids, granular media, deformable solids, and plasticine. This study focused on the PAC-NeRF and attempted to widen its applicability while prioritizing its interpretability. However, a Lagrangian particle representation is commonly utilized in physics (e.g., NeuroFluid [24][2]); therefore, applying our ideas, that is, LPO, to other models is an interesting research topic.

NeRF for few-shot (sparse-view) settings. In practice, it is often laborious or impractical to collect multi-view images. Recent studies [8, 12–14, 33, 56, 65, 68, 79, 85, 87, 89] addressed this problem by refining NeRF such that it could be applied to few-shot (sparse-view) settings. These methods can be roughly divided into three approaches. (1) Regularization using external models. For example, normalizing flow [16]-based [56], VGG [69]-based [89], depth-based [14, 65, 79], and CLIP [62]-based [33] regularizations have been proposed. (2) Utilization of general and transferable models [8, 12, 68, 87]. They are trained using external datasets, and in several studies, they are finetuned for each scene. (3) Introduction of advanced training methods, such as gradual training to prevent overfitting to sparse views, including frequency regularization [85] and layer-by-layer growing strategies [13]. Among these categories, the proposed LPO is categorized as the third one, that is, optimization is conducted for each scene, and external models/datasets are not required. The main difference between the LPO and previous methods is that LPO attempts to optimize geometric structures through video sequences with the *physical constraints* of an MPM. However, previous studies have not sufficiently considered these constraints. Owing to this difference in applications, the proposed method is complementary to, rather than competitive with, other methods.

[2]Note that NeuroFluid optimizes the *transition probability of particles* under the assumption that the initial particle positions are known, that is, fixed. In contrast, LPO optimizes the initial *particle positions* to correct the geometry estimation of the first frames. Therefore, NeuroFluid and LPO are complementary.

System identification of soft bodies. Soft bodies are not only high-dimensional, but also allow large deformations; therefore, it is challenging to identify their geometry and physical properties. Typical methods can be categorized into three types. (1) Gradient-free methods [73, 77], (2) neural network-based methods [42, 66, 83], and (3) differentiable physics-simulation-based methods [10, 17, 21, 27, 35, 41, 47, 61]. The strength of the methods in the first and second categories is that they are flexible; however, they have difficulty achieving a high accuracy owing to their black-box modeling. Methods in the third category require sophisticated modeling to fill the gap between simulations and the real world; however, they have recently demonstrated promising results owing to advancements in differentiable physical simulators. The third category can be further divided into two subcategories. (i) Methods that assume that watertight geometric mesh representations of objects are available [17, 21, 27, 35, 47, 61] and (2) methods that do not assume this [10, 41]. Among these methods, we focused on the final category, PAC-NeRF [41], which prioritizes general and fast characteristics. Specifically, PAC-NeRF adopts an MPM that can be applied to various materials, including elastic objects, sand [39], fluids [36], and foam [88]. In particular, PAC-NeRF utilizes a differentiable MPM (DiffMPM) [32] to construct a fully differentiable simulation rendering system. This study is based on this advancement. In particular, sensitivity to geometry learning from the first frames is one of the bottlenecks of PAC-NeRF. Therefore, this study focused on alleviating this bottleneck and attempted to widen its applicability to severe (e.g., sparse-view) settings.

3. Method

In this section, we first define the problem statement (Section 3.1) and then briefly review PAC-NeRF [41], upon which our method is built (Section 3.2). Subsequently, we explain the main idea of the proposed method, that is, geometric correction using LPO (Section 3.3), and then introduce its application to physical identification (Section 3.4).

3.1. Problem statement

Given a set of multi-view video sequences, our objectives are as follows: (1) to recover geometric representations[3] of the target dynamic object (particularly continuum materials) through video sequences and (2) identify its physical properties. More formally, the training data comprise

[3]Here, both appearance and shape are treated as part of the geometric representation. We attempted to recover both.

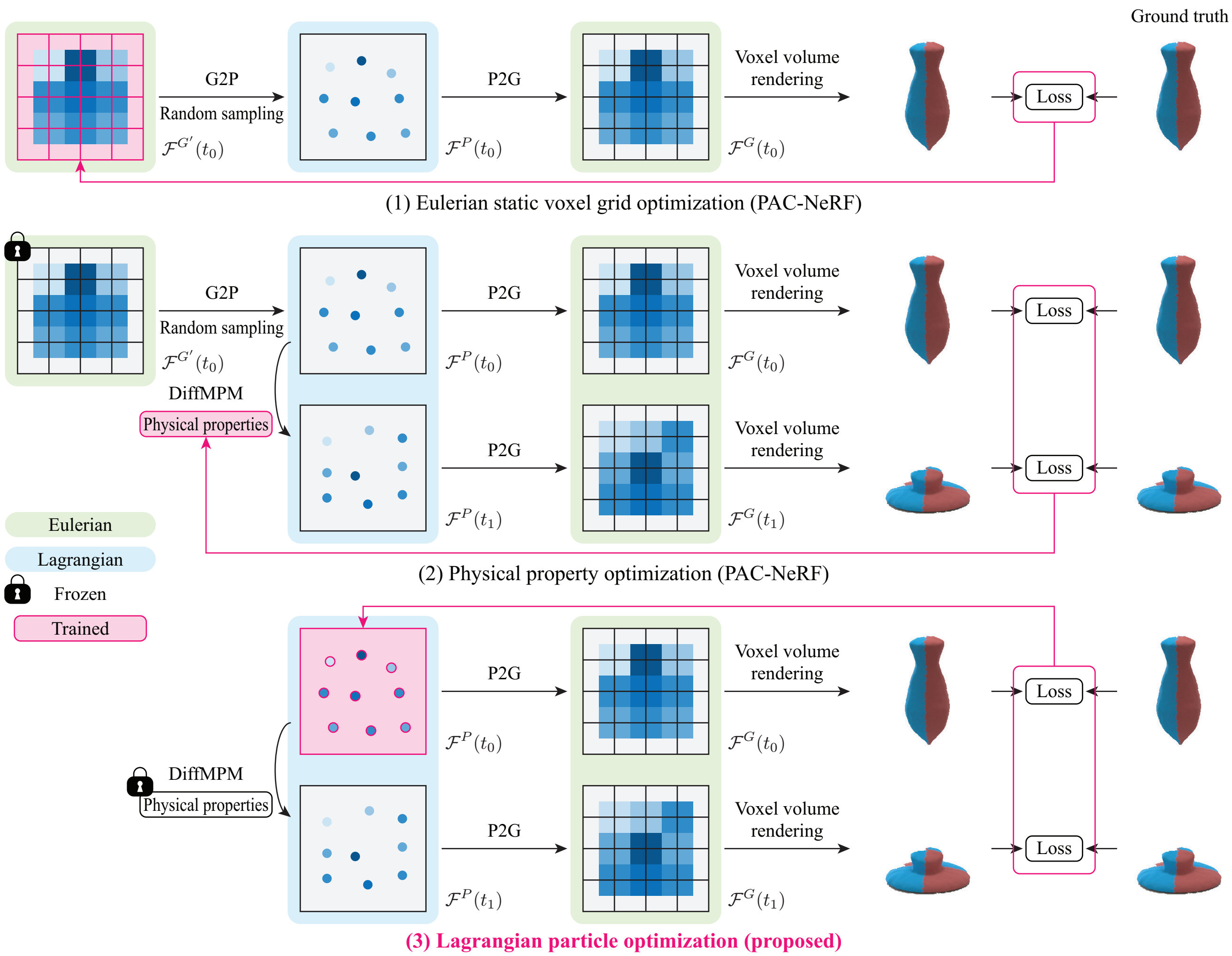


Figure 2. Training pipelines of PAC-NeRF (1)(2) and the proposed LPO (3).

ground truth color observations, that is, $\hat{\mathbf{C}}(\mathbf{r}, t)$. Here, $\mathbf{r}(s) \in \mathbb{R}^3$ is a camera ray defined as $\mathbf{r}(s) = \mathbf{o} + s\mathbf{d}$, where $\mathbf{o} \in \mathbb{R}^3$ is the camera origin, $\mathbf{d} \in \mathbb{S}^2$ is the view direction, and $s \in [s_n, s_f]$ is a distance from the origin. $\mathbf{r}$ is sampled from a set of ray collections in the training data, that is, $\hat{\mathcal{R}}$. $t \in \mathbb{R}^+$ is a time variable, and during training, it is sampled from $\{t_0, \dots, t_{N-1}\}$, where N is the number of frames. Given these training data, we attempt to (1) predict the color observation $\mathbf{C}(\mathbf{r}, t)$ that can recover $\hat{\mathbf{C}}(\mathbf{r}, t)$ from the training data and those for novel views, and (2) identify the physical properties. In particular, to widen its applicability, we addressed a scenario in which input views are sparse, that is, $\hat{\mathcal{R}}$ is limited.

3.2. Preliminary: PAC-NeRF

To solve the abovementioned problem, PAC-NeRF employs three core components: a continuum NeRF, particle-grid interconverters, and a Lagrangian field.

Continuum NeRF. PAC-NeRF extends the standard (static) NeRF to a continuum NeRF to address the dynamics of continuum materials. To achieve this, a standard NeRF is first extended to a dynamic NeRF [60] for dynamic scenarios. In a dynamic NeRF, the volume density and color fields are defined as $\sigma(\mathbf{r}, t)$ and $\mathbf{c}(\mathbf{r}, \mathbf{d}, t)$, respectively, where the time variable t is introduced to represent the dynamics. The color of each pixel $\mathbf{C}(\mathbf{r}, t)$ is calculated using volume rendering [49]

$$\mathbf{C}(\mathbf{r}, t) = \int_{s_n}^{s_f} T(s, t)\sigma(\mathbf{r}(s), t)\mathbf{c}(\mathbf{r}, \mathbf{d}, t)ds + \mathbf{c}_{bg}T(s_f, t), \tag{1}$$

$$T(s, t) = \exp\left(-\int_{s_n}^{s} \sigma(\mathbf{r}(u), t)du\right), \tag{2}$$

where $\mathbf{c}_{bg}$ denotes the background color. The model is trained with pixel-wise loss.

$$\mathcal{L}_{pixel} = \frac{1}{N}\sum_{i=0}^{N-1}\frac{1}{|\hat{\mathcal{R}}|}\sum_{\mathbf{r}\in\hat{\mathcal{R}}}\|\mathbf{C}(\mathbf{r}, t_i) - \hat{\mathbf{C}}(\mathbf{r}, t_i)\|_2^2. \tag{3}$$

Furthermore, dynamic NeRF is extended to a continuum NeRF to represent continuum materials. To achieve this, conservation laws for the velocity field are imposed on the volume density and color fields.

$$\frac{D\sigma}{Dt} = 0, \ \frac{D\mathbf{c}}{Dt} = \mathbf{0}, \tag{4}$$

whereas the material derivative of an arbitrary time-dependent field $\phi(\mathbf{x}, t)$ is defined as $\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi$. Here, $\mathbf{v}$ is the velocity field and follows momentum conservation for continuum materials:

$$\rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \boldsymbol{T} + \rho \mathbf{g}, \tag{5}$$

where ρ is the physical density field, $\boldsymbol{T}$ is the internal Cauchy stress tensor, and $\mathbf{g}$ is the acceleration owing to gravity. This equation is solved using DiffMPM [32].

Particle-grid interconverters. As shown in Figure 2(1)(2), in PAC-NeRF, a physical simulation is conducted in Lagrangian particle space using DiffMPM [32], whereas neural rendering is performed in Eulerian grid space with a discretized voxel-based NeRF [72]. To bridge these two spaces, grid-to-particle (G2P) and particle-to-grid (P2G) conversions are conducted.

$$\mathcal{F}_p^P \approx \sum_i w_{ip} \mathcal{F}_i^G, \ \mathcal{F}_i^G \approx \frac{\sum_p w_{ip} \mathcal{F}_p^P}{\sum_p w_{ip}}, \tag{6}$$

where $\mathcal{F}_*^*$ is a field (e.g., volume density or color field); P and G denote particle and grid views, respectively; i and p indicate the index of the grid node and particle, respectively; and w_{ip} is the weight of the trilinear shape function defined on i and evaluated at p.

Lagrangian field. As shown in Figure 2(1), during training, the Eulerian voxel field $\mathcal{F}^{G'}(t_0)$ is optimized for the first frames of the video sequences. Subsequently, this field is converted into a Lagrangian particle field $\mathcal{F}^P(t_0)$ using G2P, in which the positions of the particles are determined by random sampling around voxel grids. The geometric shape is represented in Lagrangian space by removing the particles whose alpha values are small, that is, the remaining particles represent the shape of the object. As shown in Figure 2(2), the particle field in the next step, that is, $\mathcal{F}^P(t_1)$, where $t_1 = t_0 + \delta t$ and δt is the duration of the simulation time step, is calculated based on $\mathcal{F}^P(t_0)$ using DiffMPM [32]. After this process, $\mathcal{F}^P(t_1)$ is converted to the Eulerian voxel field $\mathcal{F}^G(t_1)$ using P2G, and pixels are rendered based on this field using voxel-based volume rendering [72], in which the volume density and color of a sample on a ray are rendered as follows:

$$\sigma(\mathbf{x}, t) = \text{softplus}(\text{interp}(\mathbf{x}, \sigma^G)), \tag{7}$$

$$\mathbf{c}(\mathbf{x}, \mathbf{d}, t) = \text{MLP}(\text{interp}(\mathbf{x}, \mathbf{c}^G), \mathbf{d}), \tag{8}$$

where σ^G and $\mathbf{c}^G$ denote the volume density and color fields, respectively, discretized on fixed voxel grids.

3.3. Geometric correction with LPO

As explained above, in PAC-NeRF, the Eulerian voxel fields $\mathcal{F}^{G'}(t_0)$ are optimized for the first frames of the video sequences (Figure 2(1)). These are fixed while optimizing the physical properties using video sequences (Figure 2(2)). This approach is problematic when the learning of the Eulerian voxel field from the first frames is difficult, for example, in sparse-view settings, because failure of this learning cannot be corrected after the process. This also means that in dynamic scenes, information over time is useful for covering a lack of information owing to sparse views; however, this approach cannot utilize this information to optimize the volume density and color.

One possible solution is to train $\mathcal{F}^{G'}(t_0)$ with the entire video sequence and not just the first frame. However, in this approach, the optimization targets are limited to the volume density and color of the grids; therefore, it is difficult to reflect the physics-based optimization that occurs in Lagrangian particle space.[4] This can cause excessive modification of the geometry beyond physical constraints.

Alternatively, we developed *Lagrangian particle optimization (LPO)*, in which the geometric structure is optimized *not* in Eulerian grid space but in *Lagrangian particle* space, as shown in Figure 2(3). More formally, in the Eulerian voxel grid optimization, σ^G (Equation (7)) and $\mathbf{c}^G$ (Equation (8)) are optimized for the *fixed* voxel grid $\mathbf{x}^G$. In contrast, in LPO, not only the volume density and color fields of the particles, that is, σ^P and $\mathbf{c}^P$, but also the particle position, that is $\mathbf{x}^P$, are *optimized*.[5] Because $\mathbf{x}^P$ is defined in Lagrangian space, it can reflect the physical constraints of the MPM. Consequently, we can optimize the geometric structures within the physical allowance.

3.4. Physical identification with LPO

Identifying both the geometry and physical properties from limited observations is ill-posed and challenging because there is a chicken-and-egg relationship between the geometric structure and the physical properties, that is, accurate geometry estimation is necessary for accurate physical identification and vice versa. Considering that particle-based geometric correction is highly challenging when there is a large gap between the ground truth and the predicted pixels owing to the high-dimensional nature (e.g., in extreme cases, particles can diverge), in practice, we apply LPO after physical property optimization, as shown in Figure 2.

However, an interesting question is *whether the corrected geometric structures can be utilized to improve the*

[4] More precisely, as explained in Section 3.2, particles are randomly sampled around equally spaced voxel grids, and the Eulerian voxel fields (precisely, alpha values calculated from them) are only utilized to mask them (with a non-differentiable way). Consequently, gradients calculated for the particle positions cannot propagate to $\mathcal{F}^{G'}(t_0)$.

[5] Note that the values of σ^P and $\mathbf{c}^P$ are changed in training but they are time-invariant, i.e., have the same values across sequences. Therefore, the conservation laws (Equation (4)) are preserved.

Algorithm 1 Iterative geometry–physics optimization

Input: Ground-truth color observations $\hat{\mathbf{C}}(\hat{\mathbf{r}}, t)$
  $\hat{\mathbf{r}} \in \hat{\mathcal{R}}$: Rays from observed viewpoints
  $t \in \{t_0, \ldots, t_{N-1}\}$: Time
Output: Physical properties
for $i = 1$ to R **do**
  // (1) Eulerian static voxel grid optimization
  if $i = 1$ **then**
    Optimize voxel grids with $\hat{\mathbf{C}}(\hat{\mathbf{r}}, t_0)$
  else
    Optimize voxel grids with $\hat{\mathbf{C}}(\hat{\mathbf{r}}, t_0)$ and $\check{\mathbf{C}}(\check{\mathbf{r}}, t_0)$
  end if
  // (2) Physical property optimization
  Optimize physical properties with $\hat{\mathbf{C}}(\hat{\mathbf{r}}, t)$
  // (3) Lagrangian particle optimization
  Optimize particles with $\hat{\mathbf{C}}(\hat{\mathbf{r}}, t)$
  // (4) Color prediction for unobserved views $\check{\mathbf{r}} \in \check{\mathcal{R}}$
  Predict $\check{\mathbf{C}}(\check{\mathbf{r}}, t_0)$
end for

identification of the physical properties. We developed *iterative geometry–physics optimization* (Algorithm 1) to answer this question. As described above, we first conduct PAC-NeRF optimization ((1) and (2)) and then apply LPO (3). Subsequently, we render the images in the first frame based on the updated σ^P, $\mathbf{c}^P$, and $\mathbf{x}^P$(4). We denote them as $\check{\mathbf{C}}(\check{\mathbf{r}}, t_0)$, where $\check{\mathbf{r}} \in \check{\mathcal{R}}$ and $\check{\mathcal{R}}$ is designed such that it covers the missing area in $\hat{\mathcal{R}}$ in the training data. We retrain PAC-NeRF from scratch using the combination of the original $\hat{\mathbf{C}}(\hat{\mathbf{r}}, t_0)$ and synthesized $\check{\mathbf{C}}(\check{\mathbf{r}}, t_0)$ as training data. Specifically, to alleviate the negative effect caused by incomplete geometry correction, which can occur at the beginning of the iterative calculation, we only utilize $\check{\mathbf{C}}(\check{\mathbf{r}}, t_0)$ for the Eulerian static voxel grid optimization and do not utilize it to optimize the physical properties and particles.

Another possible solution is to update the voxel grids by applying P2G to the updated σ^P, $\mathbf{c}^P$, and $\mathbf{x}^P$ instead of (4) and (1). However, we found that the repeated utilization of G2P and P2G tends to produce artifacts (such as those typically occurring in the erosion process) owing to their approximate nature (Equation (6)).[6] Therefore, the full recalculation approach described above was adopted.

4. Experiments

4.1. Experimental setup

Two experiments were conducted to investigate the effectiveness of the proposed LPO. First, we evaluated the effectiveness of LPO for geometric correction (Section 4.2) and then we investigated the utility of LPO for physical identification (Section 4.3). The main results of these experiments are presented here, and the detailed analyses, extended results, and implementation details are provided in the Supplementary Material. Video samples are available at the link indicated on the first page of this manuscript.[1] In this section, we provide the experimental setup common to both experiments and other setups in subsequent sections.

Dataset. We investigated the benchmark performance on the dataset provided by the original study on PAC-NeRF [41].[7] This dataset comprised nine scenes and various continuum materials, including Newtonian fluids (*Droplet* and *Letter* with fluid viscosity μ and bulk modulus κ), non-Newtonian fluids (*Cream* and *Toothpaste* with shear modulus ν, bulk modulus κ, yield stress τ_Y, and plastic viscosity η), granular media (*Trophy* with friction angle θ_{fric}), deformable solid (*Torus* and *Bird* with Young's modulus E and Poisson's ratio ν), and plasticine (*Cat* and *Playdoh* with E, ν, and τ_Y). In each scene, the objects fall freely under the influence of gravity and undergo collisions. The ground-truth simulation data were generated using the MLS-MPM framework [31]. A photorealistic simulation engine rendered objects under diverse environmental lighting conditions and ground textures. Each scene was captured from 11 viewpoints with cameras evenly spaced on the upper hemisphere, including the object. To evaluate our method under sparse-view settings, three views were used for training and the remaining eight views were used for testing. To show the robustness to the view settings, we provide the results for the other view settings in Appendix C.

Assumptions and preprocessing. For a fair comparison, we follow the assumptions and preprocessing used in PAC-NeRF [41]. It was assumed that the intrinsic and extrinsic parameters of the set of static cameras were previously known. Moreover, it was assumed that collision objects, such as the ground plane, were previously known. As mentioned in [41], this is not difficult to estimate from the observed images. For preprocessing, a video matting framework [44] was applied to remove static background objects and focus the rendering computation on the target object.

4.2. Evaluation of geometric correction

Compared models. In the first experiment, we investigated the effectiveness of LPO on the geometric correction (i.e., the method described in Section 3.3). Three models were used as the baseline. (I) *PAC-NeRF* was the same as that in the original [41] and was trained with all views, including the eight views that were used for testing. We examined this model to determine the upper bound of its performance. (II) *PAC-NeRF-3v* was trained using three views. The training settings were the same as those for (I) except that the

[6] A similar phenomenon was observed in the original study on PAC-NeRF [41]. Based on this observation, they rendered an image in the first frame by utilizing the voxel grids obtained after the P2G and G2P conversions, that is, $\mathcal{F}^G(t_0)$, instead of the original voxel grids $\mathcal{F}^{G'}(t_0)$ to make the rendering conditions the same as those in other frames.

[7] Note that although only one dataset was used, this dataset is useful for assessing the versatility because it includes a wide variety of materials, surpassing the scope of previous studies (e.g., elastic objects only in [10]).

	PAC-NeRF	PAC-NeRF-3v	+LPO	+LPO-F	+LPO-P	+GO	PAC-NeRF-3v†	+LPO	+LPO-F	+LPO-P	+GO
PSNR↑	35.99	27.39	29.22	28.22	28.89	28.33	28.47	30.11	29.31	29.87	29.39
SSIM↑	0.989	0.978	0.980	0.979	0.980	0.979	0.980	0.982	0.981	0.981	0.981
LPIPS↓	0.020	0.034	0.032	0.033	0.032	0.033	0.033	0.031	0.032	0.031	0.031

Table 1. Comparison of PSNR↑, SSIM↑, and LPIPS↓ on **geometric correction**. The scores for each scene and qualitative results are provided in Table 5 and Figures 3–7, respectively, in the Supplementary Material.

number of views varied. (III) *PAC-NeRF-3v*† was an improved variant of (II) for sparse-view settings. An interesting question is whether LPO, which is a few-shot learning method for *dynamic* scenes, can be used with other few-shot learning methods, such as those for *static* scenes. To answer this question, we adjusted the Eulerian static voxel grid optimization (Figure 2(1)) of PAC-NeRF-3v, such that it became robust to sparse views. Specifically, we scheduled a surface regularizer to reduce unexpected artifacts, introduced a view-invariant pixel-wise loss to compensate for the lack of views, and adjusted the training length to prevent overfitting. Details are provided in Appendix A.[8] We applied LPO to both (II) and (III) to investigate the effect of the initial Eulerian static voxel grid optimization. Hereafter, we denote these variants by *+LPO*. Furthermore, three comparative and ablation models were evaluated to determine the importance of each component. (i) *+LPO-F* and (ii) *+LPO-P* are variants of LPO, in which only the features and positions of the particles are optimized. (iii) *+GO* optimized Eulerian voxel grids through the entire video sequence instead of using Lagrangian particles. (i)–(iii) are used as alternatives to *+LPO*. These variants were applied to both (II) and (III).

Evaluation metrics. We evaluated the performance of the geometric correction using metrics commonly used to assess the performance of novel view synthesis in NeRF studies: the peak signal-to-noise ratio (*PSNR*), structural similarity index (*SSIM*) [78], and learned perceptual image patch similarity (*LPIPS*) [92]. In particular, we calculated the scores using all test data over time.

Results. The results are summarized in Table 1. Our findings are threefold. *(1) PAC-NeRF-3v/3v*† *vs. +LPO.* +LPO improved both baselines in terms of all metrics. In particular, the utility of +LPO for PAC-NeRF-3v† indicates an improvement in the Eulerian static voxel grid optimization and that of LPO are complementary. *(2) +LPO vs. +LPO-F/P.* We found that +LPO was superior or comparable to +LPO-F/P. This is because the feature and position optimizations are complementary. Intuitively, feature optimization is useful for correcting features of the *appearance* that are invisible in the first frame but visible after the object has moved. Similarly, position optimization is effective for correcting features of the *shape* that are invisible in the first frame but visible after the object has moved. In severe (e.g., sparse-view) settings, the utilization of both is important. *(3) +LPO vs. +GO.* +LPO outperforms +GO, possibly because +LPO can optimize the geometry adequately within the physical constraints of the MPM, whereas +GO cannot because of the absence of an explicit physical constraint.

[8] In the preliminary experiments, we examined the previous representative few-shot learning methods (e.g., DietNeRF [33] and FreeNeRF [85]). However, we found that they were less stable than PAC-NeRF-3v, possibly because in our experimental settings, the number of views was small (three) despite the wide range of the views (upper hemisphere), and explicit voxel representations were more useful than the fully implicit representation in [33, 85]. However, this study and previous studies are complementary. Further investigations will be important in future work.

4.3. Evaluation of physical identification

Compared models. In the second experiment, we verified the usefulness of LPO for physical identification (i.e., the method described in Section 3.4). In addition to the models described in Section 4.2, we examined *+None*, for which iterative training (Algorithm 1) was conducted without geometric correction (i.e., step (3) was skipped). This model was used to investigate the importance of geometric correction. For a fair comparison, we set the number of iterations (i.e., R in Algorithm 1) to four for all models. We use the superscript 4 (e.g., +LPO^4) to specify this.

Evaluation metrics. To evaluate the performance of the physical identification, we measured the absolute distance between the ground truth and the estimated physical properties. For an easy comparison, we calculated these distances after adjusting the scale (i.e., either a logarithmic scale or a linear scale) following the study of PAC-NeRF [41]. The smaller the values, the better was the performance.

Results. The results are summarized in Table 2. Our findings are fourfold. *(1) PAC-NeRF-3v/3v*† *vs. +LPO*^{4}. +LPO^4 improved the physical identification of PAC-NeRF-3v/3v† in most cases. In particular, the effectiveness of +LPO^4 for PAC-NeRF-3v† indicates that the improvement in the Eulerian static voxel grid optimization and that of LPO are complementary for physical identification. *(2) +LPO*^{4} *vs. +LPO-F*^{4}*/P*^{4}. In some cases, superiority depends on the physical properties because the physical properties interact with each other, and finding the optimal balance is difficult. However, we found that +LPO-F^4/P^4 sometimes had obvious difficulties (e.g., PAC-NeRF-3v+LPO-F^4 on Bird and PAC-NeRF-3v†+LPO-P^4 on Playdoh), whereas +LPO^4 exhibited good stability. We consider that the joint optimization of the features and positions is useful for tackling difficult situations. *(3) +LPO*^{4} *vs. +GO*^{4}. +GO^4 also experienced apparent difficulties in some cases (e.g., on Bird). +LPO^4 exhibited a more stable performance. *(4) +LPO*^{4} *vs. +None*^{4}. +None^4 sometimes worsened the performance. The results indicated that simple iterations were insufficient and that geometric correction was essential.

		PAC-NeRF	PAC-NeRF-3v	+LPO[4]	+LPO-F[4]	+LPO-P[4]	+GO[4]	+None[4]	PAC-NeRF-3v†	+LPO[4]	+LPO-F[4]	+LPO-P[4]	+GO[4]	+None[4]
Droplet	$\log_{10}(\mu)$	0.039	0.140	0.112	0.112	0.119	0.111	0.131	0.136	0.082	0.068	0.067	0.082	0.090
	$\log_{10}(\kappa)$	0.017	1.285	1.628	1.638	1.466	1.675	1.684	1.263	0.106	1.139	0.520	1.520	1.656
Letter	$\log_{10}(\mu)$	0.041	0.674	0.015	0.026	0.079	0.048	0.530	0.379	0.010	0.087	0.007	0.118	0.436
	$\log_{10}(\kappa)$	0.039	6.772	0.174	0.411	1.040	0.865	6.083	5.229	0.060	0.027	0.127	0.138	5.060
Cream	$\log_{10}(\mu)$	0.090	0.311	0.178	0.163	0.234	0.183	0.251	0.179	0.100	0.073	0.065	0.092	0.076
	$\log_{10}(\kappa)$	0.132	0.215	0.158	0.249	0.027	0.028	0.244	0.336	0.121	0.197	0.142	0.193	0.339
	$\log_{10}(\tau_Y)$	0.007	0.014	0.004	0.030	0.005	0.006	0.025	0.009	0.006	0.001	0.004	0.010	0.008
	$\log_{10}(\eta)$	0.015	0.281	0.183	0.256	0.083	0.061	0.252	0.195	0.033	0.059	0.079	0.096	0.051
Toothpaste	$\log_{10}(\mu)$	0.026	1.891	0.109	0.156	0.119	0.164	0.215	0.252	0.031	0.201	0.005	0.252	0.138
	$\log_{10}(\kappa)$	0.247	1.580	0.601	1.356	0.597	0.648	0.729	1.436	0.673	0.630	0.899	0.590	0.596
	$\log_{10}(\tau_Y)$	0.066	0.201	0.114	0.191	0.075	0.250	0.061	0.199	0.093	0.064	0.117	0.149	0.054
	$\log_{10}(\eta)$	0.013	0.373	0.003	0.267	0.013	0.007	0.047	0.212	0.009	0.016	0.005	0.007	0.042
Torus	$\log_{10}(E)$	0.019	0.277	0.061	0.008	0.083	0.010	0.054	0.074	0.036	0.026	0.039	0.015	0.074
	ν	0.023	0.085	0.001	0.120	0.031	0.074	0.316	0.131	0.007	0.040	0.033	0.050	0.129
Bird	$\log_{10}(E)$	0.013	0.449	0.067	0.240	0.074	0.313	0.246	0.123	0.027	0.186	0.039	0.231	0.296
	ν	0.029	0.102	0.001	0.372	0.075	0.365	0.581	0.141	0.047	0.072	0.008	0.132	0.145
Playdoh	$\log_{10}(E)$	0.286	0.290	0.116	0.580	0.120	0.304	0.170	0.521	0.133	0.474	2.121	0.232	2.148
	$\log_{10}(\tau_Y)$	0.038	0.283	0.165	0.027	0.214	0.196	0.237	0.110	0.173	0.079	1.179	0.197	0.967
	ν	0.076	0.495	0.111	0.173	0.109	0.248	0.128	0.212	0.063	0.133	0.110	0.127	0.106
Cat	$\log_{10}(E)$	0.855	1.301	0.973	1.068	1.071	1.127	1.066	1.192	0.706	0.534	0.712	0.783	0.793
	$\log_{10}(\tau_Y)$	0.026	0.120	0.107	0.117	0.113	0.086	0.112	0.084	0.067	0.014	0.076	0.069	0.072
	ν	0.027	0.044	0.004	0.036	0.069	0.190	0.079	0.118	0.003	0.027	0.007	0.028	0.063
Trophy	θ_{fric} [rad]	0.048	0.053	0.055	0.051	0.056	0.057	0.054	0.030	0.039	0.041	0.043	0.044	0.039

Table 2. Comparison of the absolute differences between the ground-truth and the estimated physical properties on **physical identification**. The smaller the values, the better was the performance. The values of the physical properties (i.e., not the absolute differences) are provided in Tables 6 and 7 in the Supplementary Material. The qualitative comparisons are presented in Figures 3–9 in the Supplementary Material.

	PAC-NeRF-3v	+LPO	+LPO[4]	PAC-NeRF-3v†	+LPO	+LPO[4]
PSNR↑	27.39	29.22	29.65	28.47	30.11	30.34
SSIM↑	0.978	0.980	0.982	0.980	0.982	0.983
LPIPS↓	0.034	0.032	0.030	0.033	0.031	0.029

Table 3. Comparison of PSNR↑, SSIM↑, and LPIPS↓ on **geometric recorrection**. The scores for each scene and the qualitative comparisons are provided in Table 8 and Figures 3–9, respectively, in the Supplementary Material.

Applications. As mentioned above, there is a chicken-and-egg relationship between the geometric structure and physical properties. An interesting question is whether the corrected physical properties can be used to re-estimate the geometry (the opposite problem). To answer this question, we investigated the geometry identification performance after physical re-identification. The results are summarized in Table 3. These results indicate that accurate physical identification is useful for improving the geometry identification performance.

5. Discussion

Based on the above experiments, we demonstrated promising results for geometric correction and physical identification. However, our methods have some limitations. (1) Our methods require a longer training time due to the introduction of LPO (Section 3.3)[9] and the iterative algorithm (Section 3.4).[10] However, it is not trivial to obtain robustness to sparse views only by increasing the training time because overfitting is a typical factor that causes learning to fail. (2) LPO is sensitive to the state before applying LPO (e.g., either PAC-NeRF-3v or -3v†) because solving geometry-agnostic system identification in sparse-view settings is ill-posed and challenging. We believe that the advancements in previous few-shot learning methods (Section 2) and the newly introduced few-shot learning method with physical constraints will solve this problem.

6. Conclusion

We proposed LPO to improve PAC-NeRF-based geometric-agnostic system identification. Our core idea is to optimize the geometry not in Eulerian space but in Lagrangian space, utilizing the particles to directly reflect the physical constraints of an MPM. The results demonstrate that LPO is useful for both geometric correction and physical identification. Although we focused on PAC-NeRF while prioritizing its high generality, Lagrangian particles are commonly employed in physics-informed models (e.g., several studies discussed in Section 2). We expect that our ideas can be utilized in other models or tasks.

[9]The calculation time of LPO (Figure 2(3)) is almost identical to that of the main process of physical property optimization (Figure 2(2)) because the forward and backward processes are identical with different optimization targets. Similarly, the calculation times for +LPO-F, +LPO-P, and +GO were almost identical to those for +LPO, indicating that the improvement was attributable to the ingenuity of the algorithm and not to an increase in the calculation cost. The computation times are discussed in detail in Appendix B.2.

[10]The total computation time increases linearly when running Algorithm 1 repeatedly but is adjustable under a quality-and-time trade-off as discussed in Appendix B.3.

References

[1] Chayan Banerjee, Kien Nguyen, Clinton Fookes, and George Karniadakis. Physics-informed computer vision: A review and perspectives. *arXiv preprint arXiv:2305.18035*, 2023. 1

[2] 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. In *ICCV*, 2021. 2

[3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022.

[4] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-NeRF: Anti-aliased grid-based neural radiance fields. In *ICCV*, 2023. 2

[5] Eric R. Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In *CVPR*, 2021. 2

[6] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022. 2

[7] Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. In *ICCV*, 2023. 2

[8] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *ICCV*, 2021. 2, 3

[9] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial radiance fields. In *ECCV*, 2022. 2

[10] Hsiao-yu Chen, Edith Tretschk, Tuur Stuyck, Petr Kadlecek, Ladislav Kavan, Etienne Vouga, and Christoph Lassner. Virtual elastic objects. In *CVPR*, 2022. 3, 6

[11] Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. In *CVPR*, 2022. 2

[12] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (SRF): Learning view synthesis for sparse views of novel scenes. In *CVPR*, 2021. 2, 3

[13] Mengyu Chu, Lingjie Liu, Quan Zheng, Erik Franz, Hans-Peter Seidel, Christian Theobalt, and Rhaleb Zayer. Physics informed neural fields for smoke reconstruction with sparse data. *ACM Trans. Graph.*, 41(4), 2022. 2, 3

[14] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *CVPR*, 2022. 2, 3

[15] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. GRAM: Generative radiance manifolds for 3D-aware image generation. In *CVPR*, 2022. 2

[16] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *ICLR*, 2017. 3

[17] Tao Du, Kui Wu, Pingchuan Ma, Sebastien Wah, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. DiffPD: Differentiable projective dynamics. *ACM Trans. Graph.*, 41(2), 2021. 3

[18] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2

[19] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4D facial avatar reconstruction. In *CVPR*, 2021. 2, 3

[20] Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. FastNeRF: High-fidelity neural rendering at 200FPS. In *ICCV*, 2021. 2

[21] Moritz Geilinger, David Hahn, Jonas Zehnder, Moritz Bächer, Bernhard Thomaszewski, and Stelian Coros. ADD: Analytically differentiable dynamics for multi-body systems with frictional contact. *ACM Trans. Graph.*, 39(6), 2020. 3

[22] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 2

[23] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleNeRF: A style-based 3D-aware generator for high-resolution image synthesis. In *ICLR*, 2022. 2

[24] Shanyan Guan, Huayu Deng, Yunbo Wang, and Xiaokang Yang. NeuroFluid: Fluid dynamics grounding with particle-driven neural radiance fields. In *ICML*, 2022. 2, 3

[25] Xiang Guo, Jiadai Sun, Yuchao Dai, Guanying Chen, Xiaoqing Ye, Xiao Tan, Errui Ding, Yumeng Zhang, and Jingdong Wang. Forward flow for novel view synthesis of dynamic scenes. In *ICCV*, 2023. 2, 3

[26] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *ICCV*, 2021. 2

[27] Eric Heiden, Miles Macklin, Yashraj Narang, Dieter Fox, Animesh Garg, and Fabio Ramos. DiSECt: A differentiable simulation engine for autonomous robotic cutting. In *RSS*, 2021. 3

[28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2

[29] Tao Hu, Shu Liu, Yilun Chen, Tiancheng Shen, and Jiaya Jia. EfficientNeRF: Efficient neural radiance fields. In *CVPR*, 2022. 2

[30] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-MipRF: Tri-Mip representation for efficient anti-aliasing neural radiance fields. In *ICCV*, 2023. 2

[31] Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Trans. Graph.*, 37(4), 2018. 6

[32] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. Diff-

Taichi: Differentiable programming for physical simulation. In *ICLR*, 2020. 2, 3, 5
[33] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting NeRF on a diet: Semantically consistent few-shot view synthesis. In *ICCV*, 2021. 2, 3, 7
[34] Miguel Jaques, Michael Burke, and Timothy Hospedales. Physics-as-inverse-graphics: Unsupervised physical parameter estimation from video. In *ICLR*, 2020. 1
[35] Krishna Murthy Jatavallabhula, Miles Macklin, Florian Golemo, Vikram Voleti, Linda Petrini, Martin Weiss, Breandan Considine, Jérôme Parent-Lévesque, Kevin Xie, Kenny Erleben, Liam Paull, Florian Shkurti, Derek Nowrouzezahrai, and Sanja Fidler. gradSim: Differentiable simulation for system identification and visuomotor control. In *ICLR*, 2021. 1, 3
[36] Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. The affine particle-in-cell method. *ACM Trans. Graph.*, 34(4), 2015. 3
[37] Takuhiro Kaneko. AR-NeRF: Unsupervised learning of depth and defocus effects from natural images with aperture rendering neural radiance fields. In *CVPR*, 2022. 2
[38] Takuhiro Kaneko. MIMO-NeRF: Fast neural rendering with multi-input multi-output neural radiance fields. In *ICCV*, 2023. 2
[39] Gergely Klár, Theodore Gast, Andre Pradhana, Chuyuan Fu, Craig Schroeder, Chenfanfu Jiang, and Joseph Teran. Drucker-prager elastoplasticity for sand animation. *ACM Trans. Graph.*, 35(4), 2016. 3
[40] Andreas Kurz, Thomas Neff, Zhaoyang Lv, Michael Zollhöfer, and Markus Steinberger. AdaNeRF: Adaptive sampling for real-time rendering of neural radiance fields. In *ECCV*, 2022. 2
[41] Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chenfanfu Jiang, and Chuang Gan. PAC-NeRF: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. In *ICLR*, 2023. 1, 2, 3, 6, 7
[42] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *ICLR*, 2019. 3
[43] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021. 2, 3
[44] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L. Curless, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. In *CVPR*, 2021. 6
[45] David B. Lindell, Julien N. P. Martel, and Gordon Wetzstein. AutoInt: Automatic integration for fast neural volume rendering. In *CVPR*, 2021. 2
[46] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *NeurIPS*, 2020. 2
[47] Pingchuan Ma, Tao Du, Joshua B. Tenenbaum, Wojciech Matusik, and Chuang Gan. RISP: Rendering-invariant state predictor with differentiable simulation and rendering for cross-domain parameter estimation. In *ICLR*, 2022. 1, 3
[48] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021. 2
[49] Nelson Max. Optical models for direct volume rendering. *IEEE Trans. Vis. Comput. Graph.*, 1(2), 1995. 4
[50] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2
[51] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P. Srinivasan, and Jonathan T. Barron. NeRF in the dark: High dynamic range view synthesis from noisy raw images. In *CVPR*, 2022. 2
[52] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4), 2022. 2
[53] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H. Mueller, Chakravarty R. Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. DONeRF: Towards real-time rendering of compact neural radiance fields using depth oracle networks. *Comput. Graph. Forum*, 40(4), 2021. 2
[54] Michael Niemeyer and Andreas Geiger. CAMPARI: Camera-aware decomposed generative neural radiance fields. In *3DV*, 2021. 2
[55] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 2
[56] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, 2022. 2, 3
[57] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, 2021. 2, 3
[58] Martin Piala and Ronald Clark. TermiNeRF: Ray termination prediction for efficient neural rendering. In *3DV*, 2021. 2
[59] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *ICLR*, 2023. 2
[60] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *CVPR*, 2021. 2, 3, 4
[61] Yiling Qiao, Junbang Liang, Vladlen Koltun, and Ming Lin. Differentiable simulation of soft multi-body systems. In *NeurIPS*, 2021. 3
[62] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3
[63] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. DeRF: Decomposed radiance fields. In *CVPR*, 2021. 2

[64] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. KiloNeRF: Speeding up neural radiance fields with thousands of tiny MLPs. In *ICCV*, 2021. 2

[65] Barbara Roessle, Jonathan T. Barron, Ben Mildenhall, Pratul P. Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *CVPR*, 2022. 2, 3

[66] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *ICML*, 2020. 3

[67] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative radiance fields for 3D-aware image synthesis. In *NeurIPS*, 2020. 2

[68] Shuai Shen, Wanhua Li, Zheng Zhu, Yueqi Duan, Jie Zhou, and Jiwen Lu. Learning dynamic facial radiance fields for few-shot talking head synthesis. In *ECCV*, 2022. 2, 3

[69] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3

[70] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. EpiGRAF: Rethinking training of 3D GANs. In *NeurIPS*, 2022. 2

[71] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019. 2

[72] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 2, 5

[73] Tetsuya Takahashi and Ming C. Lin. Video-guided real-to-virtual parameter transfer for viscous fluids. *ACM Trans. Graph.*, 38(6), 2019. 3

[74] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul P. Srinivasan, Edgar Tretschk, Yifan Wang, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, Tomas Simon, Christian Theobalt, Matthias Nießner, Jonathan T. Barron, Gordon Wetzstein, Michael Zollhöefer, and Vladislav Golyanik. Advances in neural rendering. *Comput. Graph. Forum*, 41, 2022. 1

[75] 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. In *ICCV*, 2021. 2, 3

[76] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In *CVPR*, 2022. 2

[77] Bin Wang, Longhua Wu, KangKang Yin, Uri Ascher, Libin Liu, and Hui Huang. Deformation capture and modeling of soft objects. *ACM Trans. Graph.*, 94, 2015. 3

[78] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4), 2004. 7

[79] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. NerfingMVS: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*, 2021. 2, 3

[80] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. NeX: Real-time view synthesis with neural basis expansion. In *CVPR*, 2021. 2

[81] Liwen Wu, Jae Yong Lee, Anand Bhattad, Yu-Xiong Wang, and David Forsyth. DIVeR: Real-time and accurate neural radiance fields with deterministic integration for volume rendering. In *CVPR*, 2022. 2

[82] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Comput. Graph. Forum*, 41, 2022. 1, 2

[83] Zhenjia Xu, Jiajun Wu, Andy Zeng, Joshua. B Tenenbaum, and Shuran Song. DensePhysNet: Learning dense physical object representations via multi-step dynamic interactions. In *RSS*, 2019. 3

[84] Yang Xue, Yuheng Li, Krishna Kumar Singh, and Yong Jae Lee. GIRAFFE HD: A high-resolution 3D-aware generative model. In *CVPR*, 2022. 2

[85] Jiawei Yang, Marco Pavone, and Yue Wang. FreeNeRF: Improving few-shot neural rendering with free frequency regularization. In *CVPR*, 2023. 2, 3, 7

[86] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021. 2

[87] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 2, 3

[88] Yonghao Yue, Breannan Smith, Christopher Batty, Changxi Zheng, and Eitan Grinspun. Continuum foam: A material point method for shear-dependent flows. *ACM Trans. Graph.*, 34(5), 2015. 3

[89] Jason Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. NeRS: Neural reflectance surfaces for sparse-view 3D reconstruction in the wild. In *NIPS*, 2021. 2, 3

[90] Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao. FDNeRF: Few-shot dynamic neural radiance fields for face reconstruction and expression editing. In *SIGGRAPH Asia*, 2022. 2, 3

[91] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. NeRF++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[92] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7