Structure from Collision

Takuhiro Kaneko
NTT Corporation

Abstract

Recent advancements in neural 3D representations, such as neural radiance fields (NeRF) and 3D Gaussian splatting (3DGS), have made accurate estimation of the 3D structure from multiview images possible. However, this capability is limited to estimating the visible external structure, and it is still difficult to identify the invisible internal structure hidden behind the surface. To overcome this limitation, we address a new task called structure from collision (SfC), which aims to estimate the structure (including the invisible internal one) of an object from the appearance changes at collision. To solve this task, we propose a novel model called SfC-NeRF, which optimizes the invisible internal structure of the object through a video sequence under physical, appearance (i.e., visible external structure)-preserving, and keyframe constraints. In particular, to avoid falling into undesirable local optima owing to its ill-posed nature, we propose volume annealing, i.e., searching for the global optima by repeatedly reducing and expanding the volume. Extensive experiments on 115 objects involving diverse structures (i.e., various cavity shapes, locations, and sizes) and various material properties reveal the properties of SfC and demonstrate the effectiveness of the proposed SfC-NeRF.[1]

1. Introduction

Learning 3D representations from multiview images is a fundamental problem in computer vision and graphics, with applications across various domains, including augmented and virtual reality, gaming, robotics, and autonomous driving. Recent advancements in neural 3D representations, such as neural radiance fields (NeRF) [46] and 3D Gaussian splatting (3DGS) [32], have enabled accurate estimation of 3D structures from multiview images and yielded impressive results in novel view synthesis.

However, this benefit is limited to the estimation of the *visible external* structure, and it is still difficult to estimate the *invisible internal* structure hidden behind the surface.[2] For example, in Figure 1, the two objects have different internal structures, as shown in Figure 1(3)(b)(d). However, they are identical in static images, as shown in Figure 1(1)(a)(c). Consequently, a standard static neural 3D representation learning model (e.g., the voxel-based NeRF [62] used in this example) learns the same internal

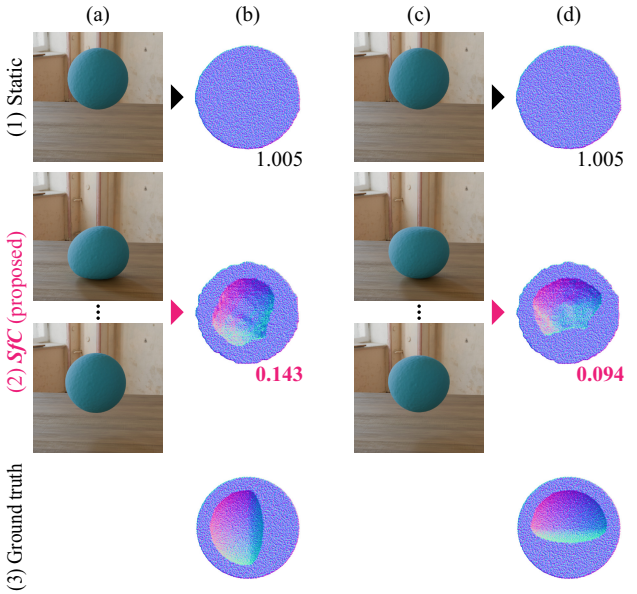

Figure 1. Concept of structure from collision (*SfC*). (a) and (c) Examples of training images taken from a certain viewpoint. (b) and (d) Cross-sectional views of the internal structures cut perpendicular to the viewpoint. The score indicates the chamfer distance ($\times 10^3\downarrow$) between the ground truth and estimated particles (the smaller, the better). Here, two objects appear to be identical in static images (1) but actually have different internal structures (3). (1) A static 3D representation learning model cannot distinguish the difference in internal structures (b)(d) because there is no difference in appearance in static images (a)(c). (2) To overcome this limitation, we address *SfC*. As shown in (a) and (c), changes in shape and appearance during collision are influenced by the internal structure. We utilize this property to identify the internal structure of the object. Although it is still difficult to identify perfectly owing to its ill-posed nature, our method has succeeded in capturing the bias in the location of the holes (b)(d).

[1] The project page is available at https://www.kecl.ntt.co.jp/people/kaneko.takuhiro/projects/sfc/.

[2] More strictly, when an object is transparent or translucent, it is possible to estimate the internal structure hidden behind the surface using a volume rendering-based 3D representation learning model (e.g., NeRF [46]) because it represents appearance on the basis of cumulative volume densities. However, this effect is limited when an object is nontransparent. This study aims to identify the internal structure even in the latter case.

structures for all objects (Figure 1(1)(b)(d)) and ignores the difference in internal structures. This misestimation of the internal structure can cause issues in practical applications, such as reproducing and simulating objects in virtual and augmented reality, as well as controlling forces during interactions with objects in robotics.

To overcome this limitation, we address a novel task called *structure from collision (SfC)*, the objective of which is to identify the structure (including the invisible internal one) of an object based on the observations at collision. This is motivated by the fact that the changes in appearance and shape at collision are influenced by the internal structures. For example, as shown in Figure 1(2), when there is a hole inside the sphere on the left side (Figure 1(3)(b)) or on the upper side (Figure 1(3)(d)), the sphere becomes crumpled in shape when it hits the ground. We consider using this property to identify the internal structure of the object.

We formulate *SfC* as the task of optimizing the invisible internal structures of an object under *physical*, *appearance (i.e., visible external structure)-preserving*, and *keyframe constraints*. Specifically, we implement this approach using *SfC-NeRF*, which consists of four components.

(1) Physical constraints. SfC is ill-posed because the observable data represent just one of many possible solutions. To address this, we narrow the solution space by incorporating *physical constraints*, specifically through the use of physics-augmented continuum NeRF (PAC-NeRF) [35].

(2) Appearance-preserving constraints. Owing to recent advancements in neural 3D representations, learning *visible external* structures is easier than learning *invisible internal ones*. Based on this, we first learn external structures using a standard static neural 3D representation learning model (voxel-based NeRF [62] in practice) using the first frame (e.g., Figure 1(1)). Then, we optimize the internal structures using a video sequence (e.g., Figure 1(2)). In the second step, to avoid damaging to the external structures learned in the first step when fitting to the entire video, we introduce *appearance-preserving constraints*, which optimize the internal structures while preserving the external ones.

(3) Keyframe constraints. In a collision video, a specific frame (for example, immediately after the collision) is effective for explaining the shape change caused by the collision. Based on this, we incorporate *keyframe constraints* to strengthen shape learning at the keyframe.

(4) Volume annealing. To prevent becoming stuck in undesirable local optima owing to the existence of multiple solutions, we develop *volume annealing*, in which the global optimum is searched for through an annealing process that repeatedly reduces and expands the volume.

We comprehensively evaluated our method on a dataset containing 115 objects with diverse structures (i.e., various cavity shapes, locations, and sizes) and various material properties. Our results reveal the properties of *SfC* and demonstrate the effectiveness of *SfC-NeRF*. Figure 1(2)(b)(d) shows examples of the results obtained using *SfC-NeRF*. Although it is challenging to perfectly match the internal structures to the ground truth because of the high degree of freedom in the solution, *SfC-NeRF* successfully identifies the deviation of the hole inside the sphere.

Our contributions can be summarized as follows:

- We address a novel task called *SfC*, whose aim is to identify structures (including the *internal* ones) from the appearance changes at collision.
- To solve *SfC*, we propose *SfC-NeRF*, which consists of four components: *physical*, *appearance-preserving*, and *keyframe constraints*, and *volume annealing*.
- Through extensive experiments on 115 objects, we demonstrate the effectiveness of *SfC-NeRF* while clarifying the properties of *SfC*. We also provide detailed results and implementation details in the Supplementary Material. Video samples are available at the project page.[1]

2. Related work

Neural 3D representations. 3D representation learning is a fundamental problem in computer vision and graphics. Recent advancements in neural 3D representations, such as NeRF [46] and 3DGS [32], have lead to significant breakthroughs, with various derivative models being proposed. These models can be roughly divided into three categories on the basis of their objectives. *(1) Improvement of quality* of rendered images or reconstructed 3D data [4–6, 24, 27, 36, 38, 42, 47, 66, 73, 78, 79], *(2) improvement of efficiency*, i.e., speeding up and reducing memory usage in training or inference [3, 10, 12, 16, 19, 22, 23, 30, 33, 34, 41, 43, 48–50, 56, 57, 59, 61, 62, 67, 70, 77], and *(3) incorporation of other modules or functionalities*, such as generative models [7–9, 11, 14, 18, 20, 29, 39, 51, 53, 58, 60, 63, 64, 69, 72, 76, 80] and physics/dynamics [1, 2, 13, 15, 17, 21, 28, 31, 35, 37, 44, 45, 52, 54, 55, 65, 71, 74, 75]. This study focuses on the third category, aiming to discover internal structures on the basis of dynamic observations under physical constraints. As these models are mutually developed, applying our approach to other models presents an interesting direction for future research.

Dynamic neural 3D representations. Dynamic neural 3D representations can be classified into two categories based on whether they incorporate physics. *(1) Non- (or weak) physics-informed models* [17, 37, 44, 45, 52, 54, 65, 74, 75] and *(2) physics-informed models* [1, 2, 13, 15, 21, 28, 31, 35, 55, 71]. The first category offers flexibility and can be applied to scenes or objects that are difficult to describe physically. However, it requires a large amount of training data and lacks interpretability owing to its fully data-driven and black-box natures. The second category, by introducing physics, provides better interpretability and narrows the solution space. However, it loses flexibility and is difficult to apply to scenes or objects that cannot be explained by physics. This study adopts a physics-informed model (the second category strategy) because *SfC* is an ill-posed problem, and physics plays an important role in narrowing the

solution space. However, in the future, it would be interesting to explore how the first category strategy could be used by expanding data and developing new theories.

Physics-informed neural 3D representations. Physics-informed neural 3D representations can be divided into two categories based on the problem setting. *(1) Forward engineering* [15, 28, 55, 71], where a physics-informed model is optimized to fit *static* scenes or objects, and then physics-informed dynamic simulations or interactive manipulations are performed. In most cases, the inside of the object is assumed to be *filled*, and internal factors, such as physical properties, are *manually adjusted* to achieve visually plausible results. *(2) Reverse engineering* [1, 2, 13, 21, 31, 35], which focuses on system identification—identifying internal factors (e.g., physical properties) from *dynamic* observations (i.e., video sequences). This study falls into the second category because it aims to reverse-engineer the *internal structure*, which is hidden but essential for describing the system, from collision videos.

Reverse engineering is generally ill-posed because the observable data represent just one of many possible solutions. To address this issue, methods in this category typically impose assumptions on internal factors that are not optimized. Previous studies have made various assumptions about the internal structure, which is the main focus of this study. For example, [13] assumes that the object is *translucent*, like smoke, allowing *part of the internal structure to be visible*. Other studies [1, 2, 21, 31, 35] consider *nontransparent* objects but assume that the interior is *filled*. Consequently, *nontransparent and unfilled objects* have not been sufficiently explored. Therefore, this study focuses on such objects. It is important to note that, as with conventional problems, solving *SfC* is challenging without making any assumptions. In this study, we assume that certain internal factors, such as physical properties, are known in advance. Even with this assumption, as shown in Figure 1 (where physical properties, such as mass, Young's modulus, and density, are identical), multiple solutions still exist, making *SfC* a challenging problem. The details of the problem settings are discussed in Section 3.1.

3. Method

3.1. Problem statement

We begin by defining the problem of *SfC*. Given a set of multiview videos in which objects collide (e.g., Figure 1(2)(a)(c)), the objective of *SfC* is to identify the structure of the object, including its *invisible internal* one, based on the appearance changes before and after the collision. Formally, the training data, i.e., a set of multiview videos, are defined as a collection of ground truth color observations $\hat{\mathbf{C}}(\mathbf{r}, t)$. Here, $\mathbf{r} \in \mathbb{R}^3$ 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 the distance from $\mathbf{o}$. During training, $\mathbf{r}$ is sampled from $\hat{\mathcal{R}}$, a collection of camera rays in the training data. $t \in \{t_0, \ldots, t_{N-1}\}$ represents the time, where N is the total number of frames. Given these data, we aim to estimate the 3D structure (both external and internal ones) of the object $\mathcal{P}^P(t_0)$, which corresponds to the ground truth $\hat{\mathcal{P}}^P(t_0)$. Here, we represent the 3D structures as particle sets, $\mathcal{P}^P(t_0)$ and $\hat{\mathcal{P}}^P(t_0)$, as shown in Figure 1(b)(d). Note that, during training, only the external appearance $\hat{\mathbf{C}}(\mathbf{r}, t)$ can be observed, while $\hat{\mathcal{P}}^P(t_0)$, which includes the internal structure, is not observable.

As discussed in Sections 1 and 2, *SfC* is an ill-posed problem with multiple solutions. Internal structures and physical properties, such as Young's modulus, have a mutually dependent relationship because both can explain the relationship between strain and stress. For example, a highly elastic object can be created either by making the object hollow or by using soft materials. To address this issue, PAC-NeRF [35] optimizes the *physical properties* on the assumption that *the inside of the object is filled*. In contrast, we address a complementary problem, namely optimizing the *internal structure* on the assumption that the *physical properties are known*. Specifically, we assume that the physical properties related to the material (e.g., Young's modulus $\hat{E}$, Poisson's ratio $\hat{\nu}$, and density $\hat{\rho}$) and mass $\hat{m}$ are known. Notably, even with this assumption, *SfC* remains a challenging problem because multiple internal structures can satisfy the same set of physical properties, as shown in Figure 1.

3.2. Preliminary: PAC-NeRF

As explained in the previous subsection, the problem settings differ between the PAC-NeRF study [35] and this study. However, since our model uses PAC-NeRF to describe the physics, we briefly review PAC-NeRF here. PAC-NeRF is a variant of NeRF that bridges the Eulerian grid-based scene representation [62] with Lagrangian particle-based differentiable physical simulation [26] for continuum materials, such as elastic materials, plasticine, sand, and fluids. PAC-NeRF obtains this functionality using three components: a continuum NeRF, a particle–grid interconverter, and a Lagrangian field.

Continuum NeRF. Continuum NeRF is built upon dynamic NeRF (NeRF for a dynamic scene) [54]. In the dynamic NeRF, the volume density field and color field for position $\mathbf{x}$, view direction $\mathbf{d}$, and time t are defined as $\sigma(\mathbf{x}, t)$ and $\mathbf{c}(\mathbf{x}, \mathbf{d}, t)$, respectively. On this basis, the color of each pixel $\mathbf{C}(\mathbf{r}, t)$ is rendered using volume rendering [46],

$$\mathbf{C}(\mathbf{r}, t) = \int_{s_n}^{s_f} T_{\mathbf{r}}(s, t)\sigma(\mathbf{r}(s), t)\mathbf{c}(\mathbf{r}(s), \mathbf{d}, t)ds, \quad (1)$$

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

This model can be trained using a pixel loss.

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

Dynamic NeRF is extended to continuum NeRF to describe the dynamics of continuum materials. This is achieved by applying the conservation laws to $\sigma(\mathbf{x}, t)$ and $\mathbf{c}(\mathbf{x}, \mathbf{d}, t)$:

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

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

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

where ρ indicates the physical density field, $\mathbf{T}$ is the internal Cauchy stress tensor, and $\mathbf{g}$ is gravitational acceleration. This equation can be solved in a differentiable manner via a differentiable material point method (DiffMPM) [26].

Particle–grid interconverter. DiffMPM is a particle-based method that conducts simulations in a Lagrangian space. However, these particles do not necessarily lie on the ray, making rendering difficult. Considering this, PAC-NeRF performs rendering in an Eulerian grid space with a voxel-based NeRF [62] and bridges these two spaces using grid-to-particle (G2P) and particle-to-grid (P2G) conversions:

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

where $\mathcal{F}^X = \{\sigma^X(\mathbf{x}, t), \mathbf{c}^X(\mathbf{x}, \mathbf{d}, t)\}$ for $X \in \{G, P\}$, where G and P represent Eulerian and Lagrangian views, respectively. When $\mathcal{F}^X$ is used with a subscript, i.e., $\mathcal{F}_x^X$ ($x \in \{i, p\}$), subscripts i and p indicate the grid node and the particle index, respectively. w_{ip} is the weight of the trilinear shape function defined at i and evaluated at p.

Lagrangian field. The physical simulation and rendering pipeline in PAC-NeRF proceeds as follows: (1) Volume densities and colors are initialized over the first frame of the video sequence in an Eulerian grid field $\mathcal{F}^{G'}(t_0)$. Here, we use superscript G' to distinguish $\mathcal{F}^{G'}$ from $\mathcal{F}^G$ used in Step (4). (2) Using the G2P process, $\mathcal{F}^{G'}(t_0)$ is converted to a Lagrangian particle field $\mathcal{F}^P(t_0)$. In this step, particles $\mathcal{P}^P(t_0)$ are sampled at intervals of half of the grid, i.e., $\frac{\Delta x}{2}$ (where Δx is the grid size), with random fluctuations. The alpha value (or amount of opacity) α_p^P is calculated for each particle by $\alpha_p^P = 1 - \exp(-\text{softplus}(\sigma_p^P))$, and a particle is removed if $\alpha_p^P < \epsilon$ ($\epsilon = 10^{-3}$ in practice). (3) The particle field in the next step, $\mathcal{F}^P(t_1)$, is calculated from $\mathcal{F}^P(t_0)$ via DiffMPM [26], where $t_1 = t_0 + \delta t$, and δt is the duration of the simulation time step. Similarly, the particle field in t, $\mathcal{F}^P(t)$, is calculated for $t \in \{t_0, \ldots, t_{N-1}\}$. (4) Using the P2G process, $\mathcal{F}^P(t)$ is converted to an Eulerian grid field $\mathcal{F}^G(t)$. (5) $\mathbf{C}(\mathbf{r}, t)$ is rendered based on $\mathcal{F}^G(t)$ using voxel-based volume rendering [62].

During training, two-step optimization is conducted. (i) $\mathcal{F}^{G'}(t_0)$ is initially optimized using the first frame of the video sequence by conducting the above process (1)–(5) for $t = t_0$. (ii) Physical properties, such as Young's modulus E and Poisson's ratio ν, are optimized for the entire video sequence by conducting the above process (1)–(5) for $t \in \{t_0, \ldots, t_{N-1}\}$. In both optimizations, $\mathcal{L}_{\text{pixel}}$ (Equation 3) is used as the objective function.

3.3. Proposal: SfC-NeRF

Similar to PAC-NeRF, *SfC-NeRF* performs two-step optimization, as shown in Figure 2. The first-step optimization (Figure 2(i)) is the same as that in PAC-NeRF; that is, $\mathcal{F}^{G'}(t_0)$ is initially optimized using the first frame of the video sequence. In this step, the filled object is learned, as shown in Figure 1(1). In contrast, the second step of optimization (Figure 2(ii)) differs owing to the difference in the optimization target. In PAC-NeRF, *physical properties* are optimized in this step, whereas in *SfC-NeRF*, the *internal structure* is optimized. Specifically, as explained in the previous section, we obtain particles $\mathcal{P}^P(t_0)$ based on $\sigma^P(t_0)$, which is calculated from $\sigma^{G'}(t_0)$ (Steps (1) and (2)); therefore, we select $\sigma^{G'}(t_0)$ as an optimization target.[3] In particular, we formulate *SfC* as a problem of optimizing $\sigma^{G'}(t_0)$ under *physical*, *appearance (i.e., external structure)-preserving*, and *keyframe constraints* along with *volume annealing*.

Physical constraints. As discussed in Section 3.1, we assume that the physical properties related to the material (e.g., Young's modulus $\hat{E}$, Poisson's ratio $\hat{\nu}$, and density $\hat{\rho}$) and mass $\hat{m}$ are known. We utilize them to narrow the solution space of *SfC*.

Physical constraints on material properties. We can reflect material-specific physical properties (e.g., Young's modulus $\hat{E}$, Poisson's ratio $\hat{\nu}$, and density $\hat{\rho}$) explicitly when constructing DiffMPM [26]. Motivated by this fact, we optimize $\sigma^{G'}(t_0)$ under *explicit material-specific physical constraints imposed by DiffMPM*.

Physical constraints on mass. Unlike physical material properties, mass is not determined only by the material and varies depending on the individual objects. Therefore, instead of representing the mass explicitly in DiffMPM, we constrain the mass using a *mass loss*.

$$\mathcal{L}_{\text{mass}} = \| \log_{10}(m) - \log_{10}(\hat{m}) \|_2^2, \tag{7}$$

$$m = \sum_{p \in \mathcal{P}^P(t_0)} \hat{\rho} \cdot \left(\frac{\Delta x}{2}\right)^3 \cdot \alpha_p^P, \tag{8}$$

[3] Note that Lagrangian particle optimization (LPO) [31] also considers a similar optimization (i.e., optimizing $\mathcal{F}^P(t_0)$ or $\mathcal{F}^{G'}(t_0)$ through a video sequence) for few-shot (sparse-view) learning. However, it aims to compensate for the *external* structure where the viewpoint is missing, and has *not* sufficiently considered the components necessary for estimating the internal structures, which are discussed in the next paragraphs. We demonstrate the limitations of LPO in the experiments (Section 4).

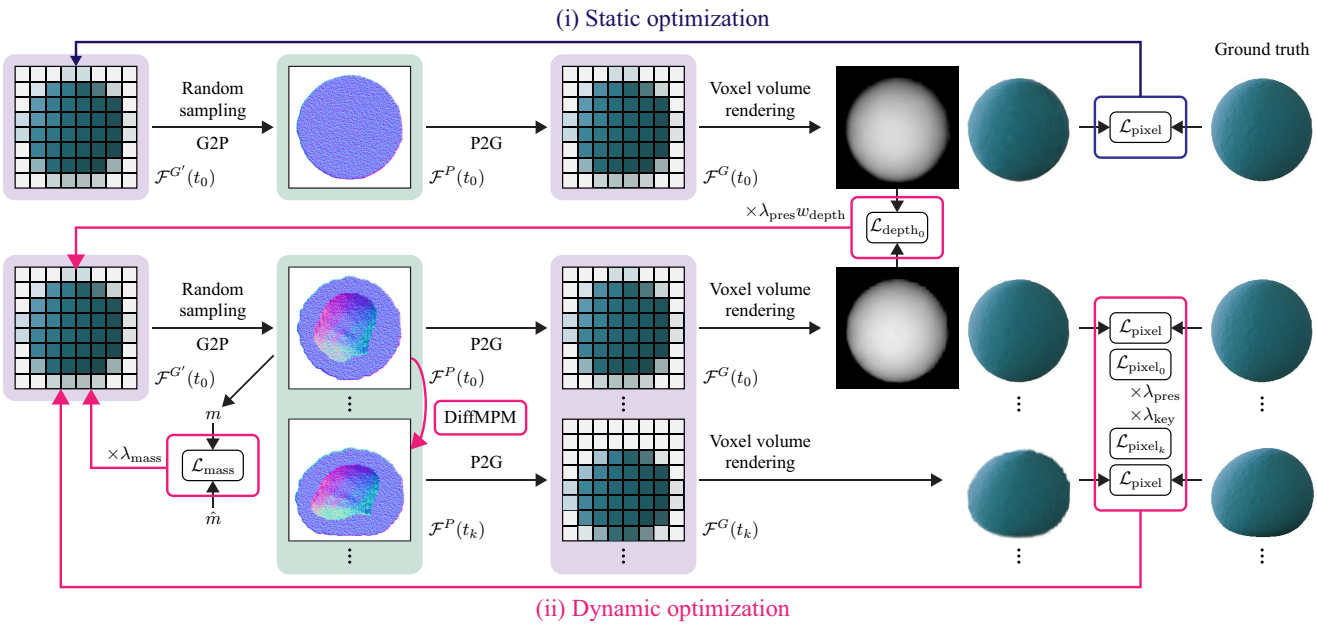

Figure 2. Optimization pipelines of SfC-NeRF. (i) The grid field $\mathcal{F}^{G'}(t_0)$ is initially optimized using the first frame of the video sequence. (ii) Subsequently, the structure (i.e., volume density $\sigma^{G'}(t_0) \in \mathcal{F}^{G'}(t_0)$) of the object is optimized through the entire video sequence with physical constraints ($\mathcal{L}_{\text{mass}}$ and DiffMPM), appearance-preserving constraints (i.e., $\mathcal{L}_{\text{pixel}_0}$ and $\mathcal{L}_{\text{depth}_0}$), and keyframe constraints ($\mathcal{L}_{\text{pixel}_k}$) along with standard pixel loss ($\mathcal{L}_{\text{pixel}}$).

where m and $\hat{m}$ are the estimated and ground truth masses, respectively. m is calculated by summarizing the mass of each particle indexed by $p \in \mathcal{P}^P(t_0)$. The mass of each particle is calculated by multiplying the physical density $\hat{\rho}$, the unit volume of particles $\left(\frac{\Delta x}{2}\right)^3$, and the alpha value α_p^P. In Equation 7, we employ a logarithmic scale to prioritize the matching of the scale.

Appearance-preserving constraints. As mentioned above, we use two-step optimization, i.e., (i) $\mathcal{F}^{G'}$ is initially optimized using the first frame of the video sequence (Figure 2(i)), and then (ii) $\sigma^{G'}$ is optimized through a video sequence (Figure 2(ii)). In Step (ii), the external structure (or surface) learned in Step (i) does not need to be changed, considering that learning the external structure is easier than learning the internal structure; however, the physical constraints discussed above are not sufficient to satisfy this requirement. Hence, we introduce appearance-preserving constraints at both the loss and training scheme levels.

Appearance-preserving loss. The standard pixel loss (Equation 3) treats the loss for each frame equally. It is insufficient to prevent the external structure, which is well learned in Step (i), from changing as a result of fitting the entire video sequence. Considering this, we employ a *pixel-preserving loss*, which encourages preservation of the appearance of the initial frame.

$$\mathcal{L}_{\text{pixel}_0} = \frac{1}{|\hat{\mathcal{R}}|} \sum_{\mathbf{r} \in \hat{\mathcal{R}}} \|\mathbf{C}(\mathbf{r}, t_0) - \hat{\mathbf{C}}(\mathbf{r}, t_0)\|_2^2. \tag{9}$$

This is the variant of the pixel loss (Equation 3) when $N = 1$. Because the constraints on the 2D projection plane alone are insufficient to preserve the 3D structure (e.g., objects with reversed concavity may be learned), we also incorporate a *depth-preserving loss* to encourage the preservation of the depth of the initial frame.

$$\mathcal{L}_{\text{depth}_0} = \frac{1}{|\hat{\mathcal{R}}|} \sum_{\mathbf{r} \in \hat{\mathcal{R}}} (\|\Delta_h Z(\mathbf{r}, t_0) - \Delta_h \tilde{Z}(\mathbf{r}, t_0)\|_2^2, + \|\Delta_v Z(\mathbf{r}, t_0) - \Delta_v \tilde{Z}(\mathbf{r}, t_0)\|_2^2), \tag{10}$$

where $Z(\mathbf{r}, t_0)$ and $\tilde{Z}(\mathbf{r}, t_0)$ are the depths predicted by the current model and the model before performing Step (ii), respectively. We use $\tilde{Z}(\mathbf{r}, t_0)$ because the ground truth depth is not observable. $Z(\mathbf{r}, t_0)$ is calculated by $Z(\mathbf{r}, t_0) = \int_{s_n}^{s_f} T_{\mathbf{r}}(s, t)\sigma(\mathbf{r}(s), t)sds$, and $\tilde{Z}(\mathbf{r}, t_0)$ is calculated in a similar manner. Δ_h and Δ_v are the operations that calculate the difference from the horizontally or vertically adjacent pixels, respectively. We compare differences rather than raw data to mitigate the negative effect of depth estimation errors caused by the change in volume densities.

Appearance-preserving training. Ideally, when an object is nontransparent, its appearance is not expected to change even if the internal volume density is changed. However, in the preliminary experiments, we found that it is difficult to retain the appearance learned in Step (i) by simple adaptation of the appearance-preserving losses. This motivates us to employ *appearance-preserving training*, i.e., reoptimizing $\mathcal{F}^{G'}(t_0)$ using the first frames of the video sequence every time after optimizing $\sigma_{G'}(t_0)$ for the entire sequence.

Keyframe constraints. As mentioned in the explanation of the appearance-preserving loss, the standard pixel loss treats the loss for each frame equally. However, in the preliminary experiments, we found that certain frames, particularly the frame immediately after the collision, are especially useful for explaining shape changes due to the internal structures. On the basis of this observation, we impose a *keyframe pixel loss* defined as follows:

$$\mathcal{L}_{\text{pixel}_k} = \frac{1}{|\hat{\mathcal{R}}|} \sum_{\mathbf{r} \in \hat{\mathcal{R}}} \|\mathbf{C}(\mathbf{r}, t_k) - \hat{\mathbf{C}}(\mathbf{r}, t_k)\|_2^2, \tag{11}$$

where k is the index of the keyframe (the frame immediately after the collision is used in practice).

Volume annealing. As previously discussed, we begin the optimization from the state in which the inside of the object is filled (Figure 2(i)). The internal structure is then optimized by reducing the volume using the above-mentioned techniques. Owing to these learning dynamics, if the volume reduction goes in the wrong direction and leads to a local optimum, it becomes challenging to find the global optimum. To address this issue, we introduce *volume annealing*, which involves alternating between the volume reduction mentioned above and the volume expansion. This strategy facilitates the search for the global optimum. Specifically, we implement the volume expansion by performing the G2P and P2G processes successively and replacing the obtained $\mathcal{F}^G(t_0)$ with $\mathcal{F}^{G'}(t_0)$.

Full objective. The full objective used in Step (ii) is expressed as follows:

$$\begin{aligned}\mathcal{L}_{\text{full}} &= \mathcal{L}_{\text{pixel}} + \lambda_{\text{mass}}\mathcal{L}_{\text{mass}} \\ &+ \lambda_{\text{pres}}(\mathcal{L}_{\text{pixel}_0} + w_{\text{depth}}\mathcal{L}_{\text{depth}_0}) + \lambda_{\text{key}}\mathcal{L}_{\text{pixel}_k}\end{aligned} \tag{12}$$

where λ_{mass}, λ_{pres}, w_{depth}, and λ_{key} are weighting hyperparameters. The effect of each loss is analyzed through an ablation study presented in Section 4.

4. Experiments

4.1. Experimental setup

We conducted three experiments to evaluate *SfC-NeRF* and explore the properties of *SfC*. First, we examined the impact of changes in internal structure, focusing on *cavity sizes* (Experiment I in Section 4.2) and *locations* (Experiment II in Section 4.3). Additionally, we explored the effect of *material properties* in Experiment III (Section 4.4). The main results are summarized here, with detailed results and implementation details provided in the Supplementary Material. Video samples are available on the project page.[1]

Dataset. Since *SfC* was a new task and there was no established dataset, we created a new dataset called *SfC dataset* based on the protocol of the PAC-NeRF study [35]. In particular, we prepared a total of 115 objects by changing the external shape, internal structure, and material of the objects. Figure 3 shows examples of data in this dataset. First, we prepared five external shapes: *sphere*, *cube*, *bicone*, *cylinder*, and *diamond*. Regarding the internal structure and material, we set the default values as follows: the cavity size rate for the filled object, s_c, was set to $(\frac{2}{3})^3$, the cavity location, l_c, was set to the center, and the material was defined as an elastic material with Young's modulus $\hat{E} = 10^6$ and Poisson's ratio $\hat{\nu} = 0.3$. For this default properties, one of them was changed as follows. (a) Three different sized cavities: $s_c \in \{0, (\frac{1}{2})^3, (\frac{3}{4})^3\}$. (b) Four different locations of cavities: the center l_c is moved in $\{\text{up}, \text{down}, \text{left}, \text{right}\}$. (c) Eight different elastic materials: those with four different Young's moduli $\hat{E} \in \{2.5 \times 10^5, 5 \times 10^5, 2 \times 10^6, 4 \times 10^6\}$ and those with four different Poisson's ratios $\hat{\nu} \in \{0.2, 0.25, 0.35, 0.4\}$. Additionally, seven different materials: two Newtonian fluids, two non-Newtonian fluids, two plasticines, and one sand. The physical properties of these seven materials were based on the PAC-NeRF dataset [35]. Thus, we created 5 external shapes $\times$ (1 default + 3 sizes + 4 locations + $(8+7)$ materials) $= 115$ objects.

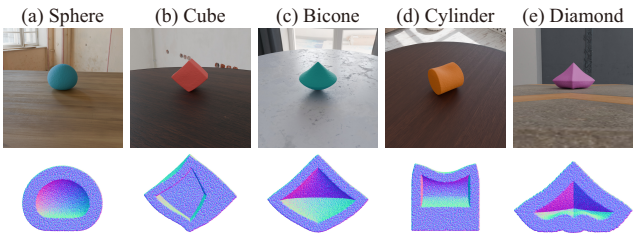

Figure 3. Examples of data in SfC dataset.

Following the PAC-NeRF study [35], the ground-truth data were generated using the MLS-MPM simulator [25], where each object falls freely under the influence of gravity and collides with the ground plane. The images were rendered under various environmental lighting conditions and ground textures using a photorealistic renderer. Each scene was captured from 11 viewpoints using cameras spaced on the upper hemisphere including the object.

Preprocessing. Following the PAC-NeRF study [35], we made two assumptions and preprocessing to focus on solving *SfC*. (1) The intrinsic and extrinsic parameters of cameras were known, and (2) the collision objects, such as the ground plane, were known. As mentioned in [35], the latter can be easily estimated from observed images. For preprocessing, we applied video matting [40] to exclude static background objects and concentrate the computation on the object of interest. This process also provides a background segmentation mask $\hat{B}(\mathbf{r}, t)$. NeRF can estimate a background segmentation mask $B(\mathbf{r}, t)$ by $B(\mathbf{r}, t) = 1 - T_{\mathbf{r}}(s_f, t)$. Taking advantage of this property, we also used a background loss $\mathcal{L}_{\text{bg}} = \|B(\mathbf{r}, t) - \hat{B}(\mathbf{r}, t)\|_2^2$ when calculating the pixel-related losses ($\mathcal{L}_{\text{pixel}}$, $\mathcal{L}_{\text{pixel}_0}$, and $\mathcal{L}_{\text{pixel}_k}$) with a weighting parameter of w_{bg}. In the experiments, this technique was applied to all models.

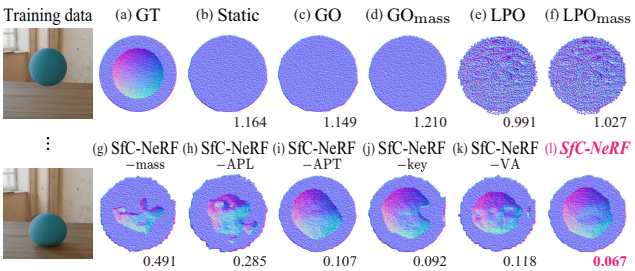

Figure 4. Comparison of learned structures for sphere objects with $s_c = (\frac{2}{3})^3$. The score under paticles indicates the CD ($\times 10^3\downarrow$). (c)–(f) GO/LPO failed to find optimal learning directions. (g)–(k) The ablated models failed to avoid improper solutions. (l) The full model overcomes these issues and achieves the best CD.

Comparison models. Since there is no established method for *SfC*, we adapted previous methods to make suitable for *SfC*. Specifically, we used grid optimization (*GO*) and Lagrangian particle optimization (*LPO*) [31] as baselines. GO and LPO are improved variants of PAC-NeRF, optimizing $\mathcal{F}^{G'}(t_0)$ and $\mathcal{F}^{P}(t_0)$, respectively, by utilizing $\mathcal{L}_{\text{pixel}}$ across a video sequence for few-shot (sparse-view) learning. For a fair comparison with *SfC-NeRF*, GO and LPO were trained with the ground-truth physical properties. Although the original GO and LPO do not use the mass information for training, it might not be fair to apply it solely to the proposed method. Therefore, we also examined *GO$_{mass}$* and *LPO$_{mass}$*, extensions of GO and LPO that incorporate $\mathcal{L}_{\text{mass}}$. Furthermore, as an ablation study, we compared *SfC-NeRF* with various variants: *SfC-NeRF$_{-mass}$*, *SfC-NeRF$_{-APL}$*, *SfC-NeRF$_{-APT}$*, *SfC-NeRF$_{-key}$*, and *SfC-NeRF$_{-VA}$*, in which the mass loss ($\mathcal{L}_{\text{mass}}$),[4] appearance-preserving losses ($\mathcal{L}_{\text{pixel}_0}$ and $\mathcal{L}_{\text{depth}_0}$), appearance-preserving training, keyframe loss ($\mathcal{L}_{\text{pixel}_k}$), and volume annealing were ablated, respectively. We also examined *Stacic*, a model trained only using the first frame of the video sequence, to assess the impact of optimization across videos.

Evaluation metric. As mentioned in Section 3.1, we use particles $\mathcal{P}^P(t_0)$ to represent the structure (including the internal structure) of an object, and aim to estimate $\mathcal{P}^P(t_0)$ that matches the ground truth $\hat{\mathcal{P}}^P(t_0)$. Therefore, we evaluated the model by measuring the distance between $\mathcal{P}^P(t_0)$ and $\hat{\mathcal{P}}^P(t_0)$ using a *chamfer distance (CD)*. The smaller the value, the higher the degree of matching.

4.2. Experiment I: Influence of cavity size

We first investigated the influence of the *cavity size* inside the object. Table 1 summarizes the quantitative results, while the qualitative results are provided in Figure 4, Appendix B.1, and on the project page.[1] Our findings are threefold. *(1) Limitations of GO and LPO [31].* GO, simple voxel grid optimization using $\mathcal{L}_{\text{pixel}}$, failed to find an appropriate optimization direction and led to the deterioration of $\mathcal{P}^P(t_0)$ as it fits the video. LPO showed slight improvement by moving particles within physical constraints via DiffMPM, but its effectiveness was limited because significant particle movement could alter the unit volume density, making it difficult to find the optimal internal structure. Furthermore, in both GO and LPO, using the mass knowledge with $\mathcal{L}_{\text{mass}}$ did not improve performance possibly because they lack appearance-preserving mechanisms, and forcing m close to $\hat{m}$ can damage the overall structure. *(2) Effectiveness of each component.* The ablation study confirms the importance of each model component. *(3) Increased difficulty with increased cavity size.* Because the optimization begins from a filled state, large cavity sizes require significant volume changes. We believe that this is a key reason for performance deteriorates as the cavity size increases.

[4] As explained in Appendix C.3, the mass information is not only used in the loss but also in adjusting the learning rate. In this experiment, we also ablated the latter to simulate a case where the mass is unknown.

s_c	0	$(\frac{1}{2})^3$	$(\frac{2}{3})^3$	$(\frac{3}{4})^3$	Avg.
Static	0.093	0.294	0.920	1.574	0.720
GO	0.091	0.301	0.941	1.586	0.730
GO$_{\text{mass}}$	0.081	0.319	1.244	2.291	0.984
LPO	0.092	0.284	0.841	1.406	0.656
LPO$_{\text{mass}}$	0.087	0.284	0.876	1.477	0.681
SfC-NeRF$_{-\text{mass}}$	0.089	0.226	0.550	1.148	0.503
SfC-NeRF$_{-\text{APL}}$	0.106	0.423	0.898	1.326	0.688
SfC-NeRF$_{-\text{APT}}$	0.085	0.261	0.332	0.661	0.335
SfC-NeRF$_{-\text{key}}$	0.082	0.127	0.211	0.325	0.186
SfC-NeRF$_{-\text{VA}}$	0.146	0.293	0.370	0.456	0.316
SfC-NeRF	0.081	0.122	0.195	0.262	0.165

Table 1. Comparison of CD ($\times 10^3\downarrow$) when varying the cavity size s_c. The scores were averaged over five external shapes.

l_c	left	right	up	down	Avg.
Static	0.841	0.842	0.815	0.813	0.828
GO	0.874	0.853	0.878	0.870	0.869
GO$_{\text{mass}}$	1.349	1.334	1.104	1.001	1.197
LPO	0.791	0.787	0.796	0.743	0.779
LPO$_{\text{mass}}$	0.824	0.817	0.828	0.775	0.811
SfC-NeRF$_{-\text{mass}}$	0.513	0.485	0.705	0.479	0.545
SfC-NeRF$_{-\text{APL}}$	0.845	0.783	0.805	0.583	0.754
SfC-NeRF$_{-\text{APT}}$	0.624	0.428	0.384	0.464	0.475
SfC-NeRF$_{-\text{key}}$	0.308	0.296	0.307	0.313	0.306
SfC-NeRF$_{-\text{VA}}$	0.542	0.596	0.333	0.385	0.464
SfC-NeRF	0.303 (0.367)	0.258 (0.431)	0.274 (0.448)	0.291 (0.417)	0.281 (0.416)

Table 2. Comparison of CD ($\times 10^3\downarrow$) when varying the cavity location l_c. The gray score in parenthesis indicates ACD ($\times 10^3$).

4.3. Experiment II: Influence of cavity location

We next examined the influence of *cavity location*. Table 2 summarizes the quantitative results, while qualitative results are available in Appendix B.1 and on the project page.[1] Similar to Experiment I, we observed two main findings: *(1) limitations of GO and LPO* and *(2) effectiveness of each component*. In addition, we discuss *(3) how well SfC-NeRF captures the cavity location*. A simple CD is insufficient for this evaluation because it does not account for the de-

$\hat{E}$	2.5×10^5	5.0×10^5	1.0×10^6	2.0×10^6	4.0×10^6
Static	0.920	0.921	0.920	0.920	0.920
SfC-NeRF	0.289	0.254	0.195	0.314	0.374
$\hat{\nu}$	0.2	0.25	0.3	0.35	0.4
Static	0.920	0.919	0.920	0.920	0.921
SfC-NeRF	0.196	0.198	0.195	0.207	0.224

Table 3. Comparison of CD ($\times10^3\downarrow$) when varying Young's moduls $\hat{E}$ and Poisson's ratio $\hat{\nu}$.

	Newtonian	Non-Newtonian	Plasticine	Sand
Static	0.921	0.919	0.920	0.920
SfC-NeRF	0.196	0.218	0.230	0.222

Table 4. Comparison of CD ($\times10^3\downarrow$) for various materials.

viation. Therefore, we calculated the *anti-chamfer distance (ACD)*, which measures the chamfer distance between the predicted particles $\mathcal{P}^P(t_0)$ and the ground truth particles $\tilde{\mathcal{P}}^P(t_0)$, where the cavity is placed on the opposite side. It is expected that this distance is larger than the original CD. The results confirmed that the original CD was smaller than the ACD. These findings indicate that *SfC-NeRF* can capture the positional deviation of the cavity.

4.4. Experiment III: Influence of material

Finally, we investigated the influence of *material properties*. Table 3 summarize the quantitative results for elastic materials when $\hat{E}$ and $\hat{\nu}$ were varied. Table 4 summarizes the quantitative results for other materials. Qualitative results are available in Appendix B.2 and on the project page.[1] These results demonstrate that *SfC-NeRF* improves structure estimation compared to the initial state, regardless of the material. However, the improvement rate depends on the material. For instance, when the object is soft, its shape changes significantly, making it difficult to capture the dynamic changes. In contrast, when the object is hard, there are fewer shape changes, which provide limited cues for estimating the internal structure, making learning more difficult. Thus, the proposed method is most effective when the object is moderately soft or hard. As an initial approach to address *SfC*, we proposed a general-purpose method in this study. However, it would be interesting to develop methods specifically tailored to individual materials in future work.

4.5. Application to future prediction

To demonstrate the practical importance of *SfC*, we investigated the effectiveness of *SfC-NeRF* for future prediction. Specifically, the first 14 frames were used for training, and the subsequent 14 frames were used for evaluation. We compared *SfC-NeRF*, which *optimizes internal structures* with *fixed physical properties*, with PAC-NeRF [35], which *optimizes physical properties* with *fixed (filled) internal structures*. Table 5 summarizes the results. SfC-NeRF outperforms PAC-NeRF in terms of peak-to-signal ratio (PSNR) and structural similarity index measure (SSIM) [68]. These results indicate that optimizing the internal structure is also crucial in practical scenarios.

	Internal structure	PSNR↑	SSIM↑
PAC-NeRF	Fixed (filled)	23.44	0.975
SfC-NeRF	Optimized	26.60	0.981

Table 5. Results of future prediction. The scores were averaged over all cavity sizes and locations for the 40 objects examined in Experiments I and II.

Error rate	-30%	-20%	-10%	0%	10%	20%	30%
Young's modulus $\hat{E}$	0.363	0.242	0.216	0.195	0.213	0.231	0.244
Poisson's ratio $\hat{\nu}$	0.240	0.231	0.208	0.195	0.200	0.214	0.236
Density $\hat{\rho}$	0.798	0.533	0.289	0.195	0.207	0.259	0.308

Table 6. Comparison of CD ($\times10^3\downarrow$) for inaccurate physical properties. In the 0% case, an elastic material with default settings ($s_c = \left(\frac{2}{3}\right)^3$, $l_c =$ center, $\hat{E} = 10^6$, and $\hat{\nu} = 0.3$) was used.

5. Discussion

Based on the above experiments, we observed promising results for *SfC*. However, our method has some limitations. (1) Our approach assumes that objects deform during collisions. Therefore, its performance depends on the material. For example, it may be difficult to apply to metal objects that do not deform. However, detecting small changes might help overcome this issue. (2) Since *SfC* is a novel task, this study focused on evaluating its fundamental performance using simulation data, leaving the validation with real data as a challenge for future research. Alternatively, to explore its potential with real data, we assessed its robustness to inaccurate physical properties. Table 6 presents the results when errors of up to 30% are introduced in the physical properties. A significant error (e.g., -30%) in $\hat{\rho}$ causes a notable degradation owing to its negative impact on volume estimation in $\mathcal{L}_{\text{mass}}$. However, in other cases, the degradation is moderate. All scores exceed those of the baselines listed in Table 1 (e.g., 0.841 by LPO). These results indicate that the proposed method is somewhat robust to inaccurate physical properties. Additional challenges related to real data are discussed in Appendix A.4.

6. Conclusion

We approached *SfC* to identify the invisible internal structure of an object, a task that remains challenging even with the latest neural 3D representations. We proposed *SfC-NeRF* as an initial model for addressing this challenge. *SfC-NeRF* solves *SfC* by optimizing the internal structures under physical, appearance-preserving, and keyframe constraints, along with volume annealing. As discussed in Section 5, our method has certain limitations. Nonetheless, this study suggests a new direction in the development of neural 3D representations, and we believe that future developments in this field will overcome these limitations.

References

[1] Jad Abou-Chakra, Feras Dayoub, and Niko Sünderhauf. ParticleNeRF: A particle-based encoding for online neural radiance fields. In *WACV*, 2024. 2, 3

[2] Jad Abou-Chakra, Krishan Rana, Feras Dayoub, and Niko Sünderhauf. Physically embodied Gaussian splatting: Embedding physical priors into a visual 3D world model for robotics. In *CoRL*, 2024. 2, 3

[3] Benjamin Attal, Jia-Bin Huang, Michael Zollhoefer, Johannes Kopf, and Changil Kim. Learning neural light fields with ray-space embedding networks. In *CVPR*, 2022. 2

[4] 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

[5] 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.

[6] 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

[7] 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

[8] 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.

[9] 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

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

[11] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3D: Disentangling geometry and appearance for high-quality text-to-3D content creation. In *ICCV*, 2023. 2

[12] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. HAC: Hash-grid assisted context for 3D Gaussian splatting compression. In *ECCV*, 2024. 2

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

[15] Yutao Feng, Yintong Shang, Xuan Li, Tianjia Shao, Chenfanfu Jiang, and Yin Yang. PIE-NeRF: Physics-based interactive elastodynamics with NeRF. In *CVPR*, 2023. 2, 3

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

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

[18] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole. CAT3D: Create anything in 3D with multi-view diffusion models. In *NeurIPS*, 2024. 2

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

[20] 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

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

[22] 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

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

[24] 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

[25] 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

[26] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. DiffTaichi: Differentiable programming for physical simulation. In *ICLR*, 2020. 3, 4

[27] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. GaussianShader: 3D Gaussian splatting with shading functions for reflective surfaces. In *CVPR*, 2024. 2

[28] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, and Chenfanfu Jiang. VR-GS: A physical dynamics-aware interactive Gaussian splatting system in virtual reality. *ACM Trans. Graph.*, 78, 2024. 2, 3

[29] Takuhiro Kaneko. AR-NeRF: Unsupervised learning of depth and defocus effects from natural images with aperture rendering neural radiance fields. In *CVPR*, 2022. 2

[30] Takuhiro Kaneko. MIMO-NeRF: Fast neural rendering with multi-input multi-output neural radiance fields. In *ICCV*, 2023. 2

[31] Takuhiro Kaneko. Improving physics-augmented continuum neural radiance field-based geometry-agnostic system identification with Lagrangian particle optimization. In *CVPR*, 2024. 2, 3, 4, 7

[32] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4), 2023. 1, 2

[33] 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

[34] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3D Gaussian representation for radiance field. In *CVPR*, 2024. 2

[35] 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. 2, 3, 6, 8

[36] Yanyan Li, Chenyu Lyu, Yan Di, Guangyao Zhai, Gim Hee Lee, and Federico Tombari. GeoGaussian: Geometry-aware Gaussian splatting for scene rendering. In *ECCV*, 2024. 2

[37] 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

[38] Zhihao Liang, Qi Zhang, Wenbo Hu, Ying Feng, Lei Zhu, and Kui Jia. Analytic-Splatting: Anti-aliased 3D Gaussian splatting via analytic integration. In *ECCV*, 2024. 2

[39] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-resolution text-to-3D content creation. In *CVPR*, 2023. 2

[40] 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

[41] David B. Lindell, Julien N. P. Martel, and Gordon Wetzstein. AutoInt: Automatic integration for fast neural volume rendering. In *CVPR*, 2021. 2

[42] Jiayue Liu, Xiao Tang, Freeman Cheng, Roy Yang, Zhihao Li, Jianzhuang Liu, Yi Huang, Jiaqi Lin, Shiyong Liu, Xiaofei Wu, Songcen Xu, and Chun Yuan. MirrorGaussian: Reflecting 3D Gaussians for reconstructing mirror reflections. In *ECCV*, 2024. 2

[43] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *NeurIPS*, 2020. 2

[44] Zhicheng Lu, Xiang Guo, Le Hui, Tianrui Chen, Min Yang, Xiao Tang, Feng Zhu, and Yuchao Dai. 3D geometry-aware deformable Gaussian splatting for dynamic view synthesis. In *CVPR*, 2024. 2

[45] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3D Gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024. 2

[46] 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. 1, 2, 3

[47] 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

[48] 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

[49] 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.

[50] Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3D Gaussian splatting for accelerated novel view synthesis. In *CVPR*, 2024. 2

[51] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 2

[52] 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

[53] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *ICLR*, 2023. 2

[54] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *CVPR*, 2021. 2, 3

[55] Ri-Zhao Qiu, Ge Yang, Weijia Zeng, and Xiaolong Wang. Feature Splatting: Language-driven physics-based scene synthesis and editing. In *ECCV*, 2024. 2, 3

[56] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. DeRF: Decomposed radiance fields. In *CVPR*, 2021. 2

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

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

[59] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *NeurIPS*, 2021. 2

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

[61] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *CVPR*, 2022. 2

[62] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 1, 2, 3, 4

[63] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. LGM: Large multi-view Gaussian model for high-resolution 3D content creation. In *ECCV*, 2024. 2

[64] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. DreamGaussian: Generative Gaussian splatting for efficient 3D content creation. In *ICLR*, 2024. 2

[65] 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

[66] 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

[67] Huan Wang, Jian Ren, Zeng Huang, Kyle Olszewski, Menglei Chai, Yun Fu, and Sergey Tulyakov. R2L: Distilling neural radiance field to neural light field for efficient novel view synthesis. In *ECCV*, 2022. 2

[68] 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. 8

[69] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. ProlificDreamer: High-fidelity and diverse text-to-3D generation with variational score distillation. In *NeurIPS*, 2023. 2

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

[71] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. PhysGaussian: Physics-integrated 3D Gaussians for generative dynamics. In *CVPR*, 2024. 2, 3

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

[73] Zhiwen Yan, Weng Fei Low, Yu Chen, and Gim Hee Lee. Multi-scale 3D Gaussian splatting for anti-aliased rendering. In *CVPR*, 2024. 2

[74] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3D Gaussians for high-fidelity monocular dynamic scene reconstruction. In *CVPR*, 2024. 2

[75] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4D Gaussian splatting. In *ICLR*, 2024. 2

[76] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. GaussianDreamer: Fast generation from text to 3D Gaussians by bridging 2D and 3D diffusion models. In *CVPR*, 2024. 2

[77] 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

[78] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-Splatting: Alias-free 3D Gaussian splatting. In *CVPR*, 2024. 2

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

[80] Shijie Zhou, Zhiwen Fan, Dejia Xu, Haoran Chang, Pradyumna Chari, Tejas Bharadwaj, Suya You, Zhangyang Wang, and Achuta Kadambi. DreamScene360: Unconstrained text-to-3D scene generation with panoramic Gaussian splatting. In *ECCV*, 2024. 2