Edge Consistency for 4D Gaussian Splatting in Dynamic Scene Rendering

Boya Shi[1,2], **Thomas N Guan**[2], **Xiaodong Yi**[2*]

[1]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[2]Intelligent Game and Decision Lab, National Innovation Institute of Defense Technology
boya.shi@sjtu.edu.cn, nyguan@sina.com, yixiaodong@nudt.edu.cn

Abstract

Existing dynamic scene rendering methods often struggle to preserve sharp edges and maintain temporal consistency. To address these challenges, we introduce Edge 4D Gaussian Splatting (Edge4DGS), a real-time rendering framework that renders fine-grained geometry from sparse monocular inputs in dynamic scenes. Edge4DGS proposes a hybrid geometric representation that augments Gaussian primitives with convex hulls, enabling accurate modeling of hard surfaces and complex boundaries. To enhance spatial precision, we introduce edge consistency regularization leveraging optical flow, guiding Gaussian distributions to align with true object contours. To enforce temporal coherence, we extend the regularization from discrete time steps to continuous unit intervals, enabling accurate motion modeling and reducing flickering artifacts. A two-stage coarse-to-fine optimization further improves geometric fidelity while preserving computational efficiency. Extensive experiments on monocular and multi-view motion datasets demonstrate that Edge4DGS achieves real-time, high-resolution rendering and consistently surpasses state-of-the-art methods, reducing LPIPS by 56.25%.

Introduction

High-fidelity reconstruction and real-time rendering of dynamic 3D scenes from sparse inputs remain a core challenge in computer graphics, crucial for realism and stability in VR, video editing, and visual effects (Knapitsch et al. 2017; Hedman et al. 2018; Barron et al. 2022; Pumarola et al. 2021; Park et al. 2021b; Li et al. 2022b). Existing methods often require dense input data or exhibit spatiotemporal artifacts in complex scenes. This problem is important in high-demand visual domains, such as film-grade VFX and AR/VR, where the human visual system is sensitive to edge discontinuities and temporal inconsistencies. Neural Radiance Fields (NeRFs) synthesize novel views via implicit volumetric rendering but require dense ray sampling, making them inefficient for real-time scenarios (Mildenhall et al. 2020; Park et al. 2021a; Tretschk et al. 2021; Pumarola et al. 2021; Song et al. 2023; Du et al. 2021; Gao et al. 2021). 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) enables real-time rendering with Gaussian distributions but struggles with fine and complex edges. This limitation is amplified in 4D extensions, where temporal dynamics introduce additional temporal inconsistencies.

Recent 4D extensions for dynamic scene modeling (Wu et al. 2023; Yang et al. 2023b; Duan et al. 2024) integrate temporal information into the rendering pipeline. However, these methods exhibit several key limitations. Due to the inherent smooth nature of Gaussian primitives, they struggle to preserve fine-grained spatial details, especially sharp structural boundaries, resulting in blurred edges. Furthermore, their temporal modeling often makes it hard to capture continuous motion trajectories, leading to temporal jitter and unstable edge representations in dynamic scenes. These methods are further constrained in handling complex motion and lack explicit mechanisms for modeling structural boundaries, neglecting geometric continuity and edge alignment. As a result, it is challenging to maintain temporally stable, sharp renderings under sparse input conditions. These limitations underscore the need for a robust, edge-aware framework that jointly models geometry and motion with enhanced spatial precision and temporal coherence.

High-fidelity rendering of dynamic scenes from sparse inputs poses several fundamental challenges. First, achieving a balance between smooth rendering and edge precision is nontrivial: naively reducing Gaussian scales may sharpen edges but introduces spatial discontinuities, underfitting, and instability during optimization. Rendering fine edges without substantially increasing the number of Gaussian primitives or computational overhead remains difficult. Second, maintaining spatiotemporal coherence requires temporally consistent representations; however, modeling time independently often leads to instability and disrupts spatial continuity, further degrading temporal consistency in rendered results. Third, developing a fully differentiable pipeline capable of multi-level rendering for structured scenes adds computational complexity to the system.

To address the challenges of accurate edge rendering and temporal consistency, we propose Edge 4D Gaussian Splatting (Edge4DGS), a novel dynamic scene rendering framework that integrates explicit geometric modeling with spatiotemporal regularization. Edge4DGS decomposes the rendering problem into four core components. First, to address the limitations of Gaussian primitives in capturing sharp edges, we propose a hybrid geometric representation that

*Corresponding author.

augments Gaussian primitives with differentiable convex hulls. Constructed from sparse point sets and tightly enclosing edge regions, these convex hulls enable precise modeling of complex object boundaries and hard surface transitions. Unlike existing methods that rely on Gaussian primitives, our hybrid representations exploit the complementary strengths of both representations: Gaussians provide smooth, compact volumetric coverage, while convex hulls preserve high-frequency structural detail. The convex hulls are initialized using a Fibonacci sphere sampling strategy and optimized end-to-end via differentiable projection and rasterization within the rendering pipeline. This design substantially improves geometric fidelity in sparse-input, dynamic scenarios.

Furthermore, to maintain consistent edge localization over time, we introduce a novel edge consistency regularization based on optical flow. By modeling pixel trajectories, this regularizer penalizes temporal misalignment in both geometry and appearance, enforcing photometric and geometric coherence across time. We minimize deviations in optical flow within edge regions to promote temporal alignment of dynamic structures, reducing artifacts caused by deformation or occlusion. Integrating this regularization into the optimization enhances both stability and realism, resulting in sharp and temporally coherent edge rendering under challenging dynamic conditions.

To capture both global structure and fine-grained detail in dynamic scenes, we design a two-stage optimization strategy that progressively refines the scene representation. In the coarse stage, Gaussian primitives are optimized to efficiently model large-scale geometry and motion, enabling a stable initialization. In the fine stage, differentiable convex hulls are jointly optimized with Gaussians to recover complex surface boundaries and sharp edges that Gaussians alone cannot represent. This joint optimization adaptively refines geometry in edge-dense regions while preserving temporal consistency. The coarse-to-fine strategy strikes a balance between accuracy and efficiency, enabling precise edge rendering from sparse inputs without incurring substantial computational overhead. This unified framework addresses both the spatial expressiveness and temporal stability limitations of existing dynamic scene rendering methods, enabling efficient and accurate reconstruction of complex, real-world environments.

To further enhance temporal coherence and mitigate motion inconsistencies, we extend edge consistency regularization from discrete frame pairs to continuous unit time intervals. Unlike existing methods that enforce consistency only at separate time steps, our formulation penalizes variations in optical flow across a temporal interval, allowing for robust modeling of motion dynamics. By the temporal regularization, Edge4DGS constrains the temporal motion of convex hulls to promote geometric continuity and appearance consistency. This facilitates the precise rendering of fine-grained geometric features in dynamic regions. Overall, by learning transferable edge-aware features, Edge4DGS effectively addresses the long-standing challenge of fine-grained edge rendering in dynamic scenes. Even under sparse input conditions, it accurately predicts edge appearance for moving objects, enabling high-fidelity, temporally consistent rendering. Overall, our contributions are as follows:

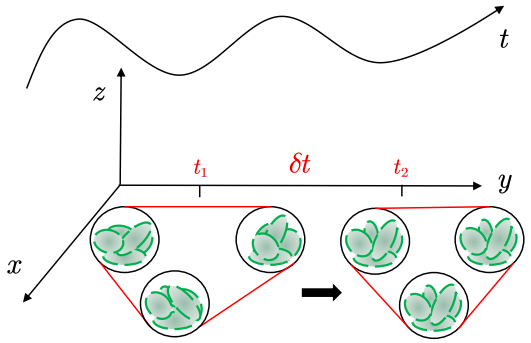

Figure 1: Simplified illustration. Our edge consistency regularization extends from separate time steps to continuous time intervals, ensuring temporal coherence. Red lines indicate convex hulls; circles represent point sets.

- A hybrid geometric representation that integrates Gaussian primitives with convex hulls, enabling an accurate rendering of sharp and complex surface boundaries.
- An edge consistency regularization leveraging optical flow to guide Gaussian distributions toward true object boundaries, improving the rendering fidelity of fine-grained edges in dynamic scenes.
- A temporal extension of edge consistency regularization from individual time steps to unit time intervals, enforcing motion coherence and enabling temporally consistent edge rendering.
- A two-stage coarse-to-fine optimization for edge-aware rendering, enhancing geometric fidelity while reducing point redundancy and memory overhead.

Related Work

In this section, we first review novel view synthesis. We then examine dynamic scene modeling, discussing limitations in rendering fine edges.

Novel View Synthesis. Novel view synthesis continues to be a central challenge in 3D reconstruction (Mildenhall et al. 2019; Srinivasan et al. 2019; Zhou et al. 2018; Broxton et al. 2020; Li et al. 2018; Su et al. 2020; Guo et al. 2019; Hu et al. 2022; Flynn et al. 2019; Kutulakos and Seitz 2000; Wood et al. 2023). Traditional techniques rely on interpolation or geometric models (Debevec, Taylor, and Malik 1996; Levoy and Hanrahan 1996; Gortler et al. 1996; Buehler et al. 2001; Penner and Zhang 2017; Seitz and Dyer 1999; Guo et al. 2015; Li et al. 2017). Neural Radiance Fields (NeRF) is an advanced volumetric rendering method that implicitly models the color and density of points using MLPs (Mildenhall et al. 2020; Riegler and Koltun 2020; Verbin et al. 2022; Kopanas et al. 2022; Bemana et al. 2022; Mildenhall et al. 2020, 2021; Barron et al. 2021; Zhang et al. 2020). Some extensions combine with voxel grids (Yi et al. 2023; Liu et al. 2023; Wang et al. 2023a) and meshes (Thies, Zollhöfer, and Nießner 2019; Waechter, Moehrle, and Goesele 2014),

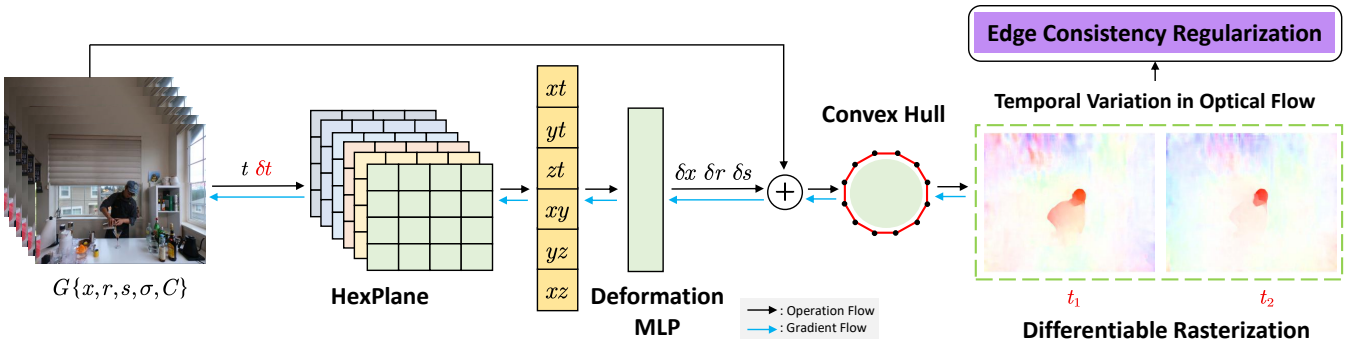

Figure 2: Framework Overview. After initialization, the sparse point cloud passes through six HexPlanes that decompose space-time into 2D planes. Features from these planes are fused via a lightweight MLP into a 4D Gaussian representation, which the deformation network uses to predict Gaussian motion. The deformed Gaussians are projected to 2D to form convex hulls, capturing complex geometries with flexibility and accuracy. Red lines show convex hulls; black dots indicate point sets.

which further enhance rendering quality. However, despite various optimizations (Müller et al. 2022; Fridovich-Keil et al. 2022; Chen et al. 2022; Yan, Li, and Lee 2023), real-time rendering is still constrained by the need to sample millions of rays. While 3DGS (Kerbl et al. 2023) enables real-time rendering, limited regularization reduces geometric precision. Mip-Splatting (Yu et al. 2024) and Multi-Scale 3DGS (Yu et al. 2024) mitigate aliasing, but lack strict geometric constraints, limiting image quality. Edge4DGS addresses this by introducing edge consistency regularization with a multi-level representation for rendering complex geometries.

Challenges in Dynamic Scene Modeling. Traditional rendering relies on explicit geometry, materials, and lighting models, reconstruct images via ray tracing or shading (Li et al. 2012; Collet et al. 2015; Kanade, Rander, and Narayanan 1997; Zitnick et al. 2004). These methods need extensive manual setup and are costly for complex scenes. Neural Radiance Fields (NeRFs) address this by implicitly modeling color and density with MLPs and differentiable volumetric rendering, enabling high-quality novel view synthesis. To improve rendering quality and adapt to dynamic scenes, there are several extensions, such as deformation-based framework (Park et al. 2021a; Du et al. 2021; Tretschk et al. 2021; Pumarola et al. 2021; Fang et al. 2022), scene decomposition (Gao et al. 2021; Song et al. 2023), keyframe extraction (Li et al. 2022b; Attal et al. 2023), streaming (Li et al. 2022a), flow field estimation (Li et al. 2021; Guo et al. 2023; Tian, Du, and Duan 2023) and explicit representation (Fridovich-Keil et al. 2023; Cao and Johnson 2023; Wang et al. 2023b; Fang et al. 2022; Gan et al. 2023; Shao et al. 2023). However, NeRFs' dense ray sampling leads to slow training and rendering. 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) addresses this by replacing implicit fields with explicit 3D Gaussians and has been extended to dynamic scenes (Luiten et al. 2023; Li et al. 2023; Yang et al. 2023a; Huang et al. 2024; Wu et al. 2023). Modeling dynamic scenes is challenging because temporal dynamics cause variations in Gaussian primitives, leading to unstable motion and blurred edges. Edge4DGS addresses these challenges with a hybrid representation that explicitly preserves fine-grained edges across time.

Method

As illustrated in Fig. 2, Edge4DGS reconstructs sharp edges and complex boundaries from sparse inputs using a hybrid representation of Gaussians and convex hulls. To capture temporal dynamics, Edge4DGS introduces a unit time interval to constrain motion variations of Gaussian primitives. Furthermore, edge consistency regularization extends optical flow supervision from discrete time steps to consecutive temporal intervals, reducing temporal artifacts and enabling precise edge rendering without increasing the point cloud size.

Edge 4D Gaussian Splatting

To achieve high-fidelity edge rendering, we propose a hybrid geometric representation that jointly leverages differentiable convex primitives and Gaussian primitives. The representation is defined as

$$Edge4DGS = \{\mathcal{G}, \mathcal{C}\}. \tag{1}$$

where $\mathcal{G} = \{G_m\}_{m=1}^{M}$ denotes the set of M 3D Gaussian primitives, and $\mathcal{C} = \{C_j\}_{j=1}^{J}$ denotes the set of J convex primitives. Each Gaussian primitive is parameterized as

$$G = (\mathbf{p},\ \mathbf{q},\ s,\ o,\ \mathbf{c}). \tag{2}$$

where $\mathbf{p}$ specifies its center position, $\mathbf{q}$ its orientation (represented as a quaternion), s its anisotropic scaling, and o its opacity. The color $\mathbf{c}$ is modeled via spherical harmonics (SH) coefficients to capture view-dependent appearance. Each convex primitive C_j is defined by a set of supporting half-spaces and associated rendering parameters:

$$C_j = \left(\{\mathbf{u}_{j,i},\ b_{j,i}\}_{i=1}^{K_j},\ \tau_j,\ \kappa_j,\ o_j,\ \mathbf{c}_j\right). \tag{3}$$

where $\mathbf{u}_{j,i} \in \mathbb{R}^3$ and $b_{j,i} \in \mathbb{R}$ denote the unit outward normal and offset of the i-th supporting plane, respectively; $\tau_j > 0$ controls the smoothness of the soft-max intersection;

$\kappa_j > 0$ modulates boundary sharpness; $o_j \in [0,1]$ specifies global opacity; and $\mathbf{c}_j \in \mathbb{R}^C$ encodes color or radiance coefficients.

To maintain spatiotemporal consistency in 4D, we introduce a deformation field that captures geometric variations, allowing 4D Gaussians to model dynamic scenes in a canonical space while preserving physical coherence. Extending 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023), originally designed for static scenes, to 4D is challenging: adding time-conditioned parameters makes Gaussian distributions evolve but introduces shape and position discontinuities across time (Yang et al. 2023b). This breaks the stability of differentiable rasterization and disrupts spatial consistency. Similarly, 4D-RotorGS (Duan et al. 2024), while decoupling space and time, fails to exploit spatial priors from previous time, causing motion incoherence in dynamic scene rendering. To preserve the physical meaning of Gaussians in dynamic scenes, we use a deformation field that learns 4D spatiotemporal features. Edge4DGS takes monocular images with sparse point clouds. The deformation network $\mathcal{D}$, concists of MLPs, predicts position ($\delta\mathbf{x}$), rotation ($\delta\mathbf{r}$), and scale (δs) changes for each Gaussian primitive, producing deformed Gaussian primitives $G(\mathbf{x}+\delta\mathbf{x}, \mathbf{r}+\delta\mathbf{r}, s+\delta s, \sigma, \mathbf{c})$, efficiently rendered via differentiable Gaussian rasterization.

After effectively modeling the temporal variaitons, we introduce convex hull modeling to effectively render scenes with complex geometries and hard edges. The convex hull constructs the smallest convex shape by processing the discrete Gaussian point cloud, tightly enclosing the points for a compact and accurate representation. A convex hull is the smallest convex polyhedron that contains a given set of points, forming a minimal enclosing shape, much like 'surrounding' the points with a thin wire. This approach allows for efficient representation and approximation of complex geometries' outer contours. Unlike Gaussian distributions, convex hulls provide a more precise representation of flat surfaces and hard edges, eliminating the blurring effects typically associated with Gaussian-based models.

To enable efficient convex hull rendering, we first define a point set that represents a 3D convex shape, rather than constructing the convex hull directly. For each convex primitive j, we model its geometry as the smooth intersection of K_j supporting half-spaces. Each half-space is described by a signed distance function

$$H_{j,i}(p) = \mathbf{u}_{j,i}^{\top} p + b_{j,i}. \tag{4}$$

where $\mathbf{u}_{j,i} \in \mathbb{R}^3$ denotes the unit outward normal of the i-th supporting plane and $b_{j,i} \in \mathbb{R}$ its offset. To ensure differentiability, we replace the conventional non-smooth maximum operator with a smooth approximation based on the log-sum-exp formulation:

$$S_j(p) = \frac{1}{\tau_j} \ln\!\left(\sum_{i=1}^{K_j} \exp\big(\tau_j \, H_{j,i}(p)\big) \right). \tag{5}$$

where the temperature parameter $\tau_j > 0$ controls the approximation tightness: as $\tau_j \to \infty$, $S_j(p)$ approaches $\max_i H_{j,i}(p)$, whereas smaller values yield a softer, more rounded intersection. Using this smooth maximum, we define a differentiable inside–outside indicator:

$$M_j(p) = \frac{1}{1 + \exp\big(\kappa_j \, S_j(p)\big)}. \tag{6}$$

where $\kappa_j > 0$ modulates the transition sharpness between interior and exterior regions. The implicit surface of the convex is given by the level set $S_j(p) = 0$, where $M_j(p) = 0.5$; increasing κ_j produces sharper boundaries. Finally, we assign each convex primitive a global opacity $o_j \in [0,1]$ and define its spatial contribution as

$$\alpha_j(p) = o_j \, M_j(p). \tag{7}$$

which can be interpreted as a smooth volumetric density or blending weight suitable for differentiable rendering pipelines.

During optimization, the point set is allowed to move freely, facilitating flexible shape deformation. The 3D points are then projected onto a 2D image plane using a pinhole camera model, from which the 2D convex hull is constructed. This projection is computed using both intrinsic and extrinsic camera parameters and is differentiable, allowing for backpropagation during optimization. The 2D convex hull is efficiently computed using the Graham Scan algorithm (Graham 1972), which sorts points by polar angle and checks the cross product to maintain convexity, ensuring an accurate projection of the convex shape's outline. Additionally, we initialize the point set of each convex shape using the Fibonacci sphere algorithm, distributing the points evenly on a spherical surface centered at the point cloud's center. The size of each convex shape is determined by the average distance to the three nearest smooth convex shapes. This approach generates smaller convex shapes in dense regions and larger ones in sparse regions, enabling adaptive scaling based on local geometry. The 3DGS-based rasterizer enables efficient backpropagation across multiple shapes, with all operations, such as point projection and convex hull computation, executed within custom CUDA kernels to ensure fast rendering.

Unlike previous method (Held et al. 2024) that depends on complex hyperparameter tuning for geometry learning, Edge4DGS utilizes strict edge consistency regularization to boost rendering efficiency in dynamic scenes. This approach achieves fine-grained, realistic rendering without sacrificing real-time performance. Edge consistency regularization within a unit time interval enhances the accuracy, realism, and consistency of dynamic scene rendering, especially in scenes with substantial motion and temporal variation.

Optimization Scheme

We now present a detailed explanation of our edge consistency regularization constraint. The initial set of smooth convex shapes is derived from a sparse point cloud obtained via structure-from-motion (SfM). However, the number of these initial shapes is insufficient for accurately representing complex scenes. Significantly increasing the number of Gaussians would result in excessive memory consumption. Additionally, these shapes struggle to represent flat surfaces effectively, as they are diffusely distributed in space. We use

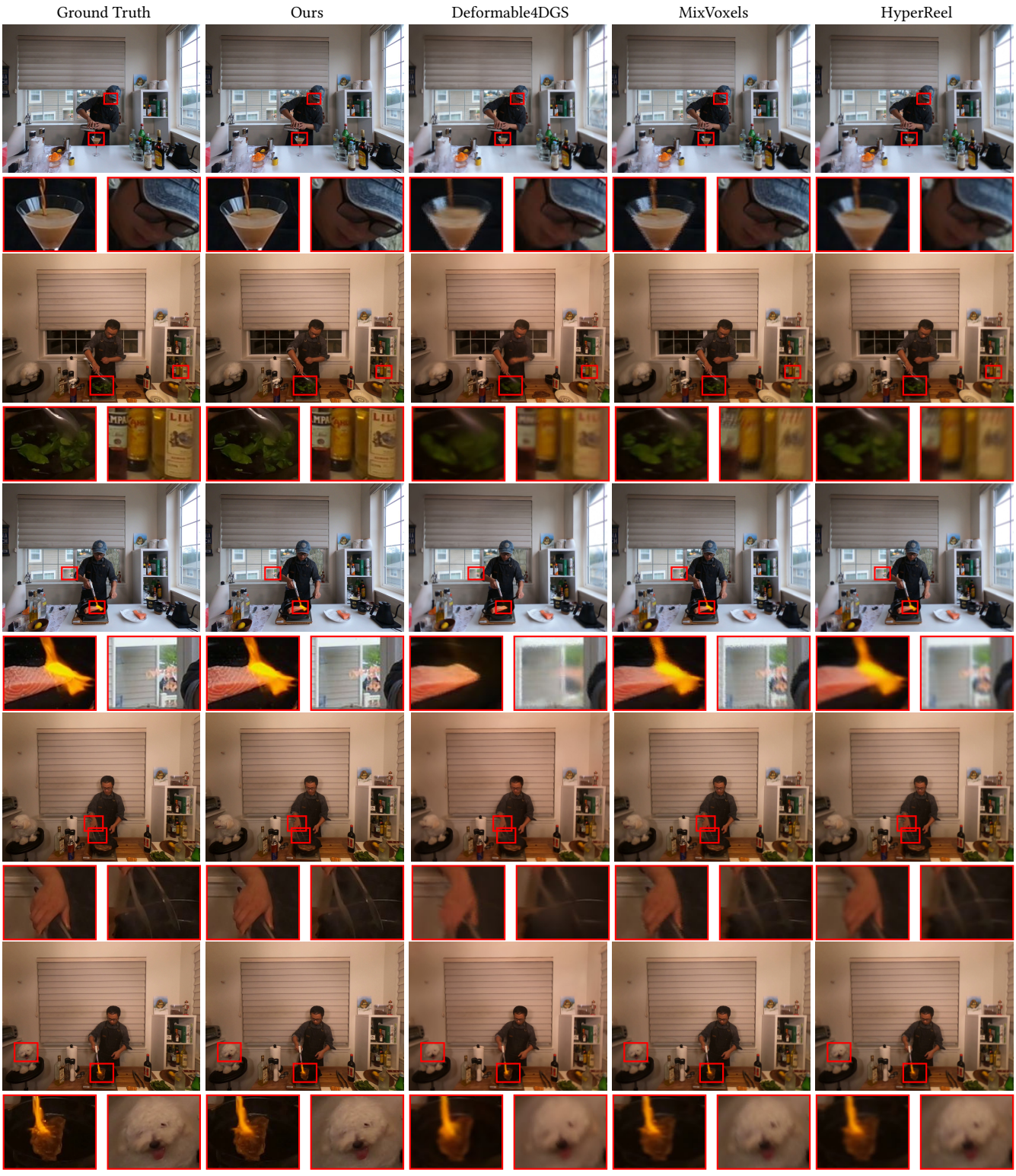

Figure 3: Qualitative Comparison on Plenoptic Video Dataset. Edge4DGS outperforms leading methods by rendering fine-grained edges in dynamic regions, *e.g.*, the rapidly moving head and the spiral pattern on the glass bottle cap. Furthermore, it renders sharper reflections, as seen in the light distributions on the glass bottle in row 2.

an adaptive control mechanism to dynamically add smooth convex shapes. Unlike previous methods that directly split convex shapes into K smaller ones to increase the number of Gaussian distributions, we adopt a two-stage training process to progressively capture complex edges. In the coarse stage, we use standard Gaussians to learn the basic geometric contours of the scene. In the fine stage, we construct the convex hull to efficiently represent and approximate more intricate geometries. Unlike Gaussian distributions, our hybrid representations offer a more precise representation of flat surfaces and hard edges, effectively avoiding the blurring issues often associated with Gaussian-based models.

Furthermore, to improve texture rendering accuracy in edge regions, we implement edge consistency regularization, encouraging the Gaussian point cloud to be uniformly distributed around actual surfaces. Optical flow tracks the motion of adjacent pixels, aiding in the recovery of the precise shape and details of object surfaces. This regularization effectively ensures accurate capture of motion in dynamic objects or scenes, enhancing the detail and precision of the final rendering and optimizing the distribution of the sparse point cloud. In dynamic scenes, Edge4DGS not only tracks the positional changes of objects but also helps identify and adjust the motion direction of convex shapes, leading to better spatial detail representation and accurate scene rendering.

To improve motion consistency in dynamic renderings, we introduce edge consistency regularization, which constrains optical flow variations within a unit time interval. Unlike traditional methods that infer pixel changes by using optical flow from previous frames, our approach directly penalizes optical flow variations within the given time interval. Optical flow captures detailed motion features, ensuring smooth and coherent transitions across frames. This reduce motion inconsistencies when rendering dynamic scenes. By constraining flow variations within a unit time interval, our approach accurately captures fine-grained motion details. Edge4DGS effectively represents moving objects and scene structure changes, resulting in more realistic edge renderings, especially in scenes with rapid motion and temporal variation. For a given pixel i, define $\boldsymbol{flow}_i(t)$ as the optical flow, δt denotes the unit time interval. Our edge consistency regularization over consecutive unit time intervals is formulated as follows:

$$\mathcal{L}_{4DEdge} = \sum_{\delta t} \sum_{i} \|\boldsymbol{flow}_i(t+\delta t) - \boldsymbol{flow}_i(t)\|^2. \quad (8)$$

We build upon the original 3DGS (Kerbl et al. 2023) by incorporating $\mathcal{L}_1$ and SSIM losses between the rendered and ground truth images. The final loss function is defined as follows:

$$\mathcal{L}_{Edge4DGS} = \mathcal{L}_{4DEdge} + \lambda_1\mathcal{L}_1 + \lambda_2\mathcal{L}_{\text{ssim}} + \lambda_3\mathcal{L}_{\text{tv}}. \quad (9)$$

where $\mathcal{L}_{\text{tv}}$ is the grid-based loss (Wu et al. 2023) and λ_o is the hyperparameter determined through validation.

Experiments

Datasets and Implementation Details

We evaluate our method on two dynamic scene datasets. The **Plenoptic Video Dataset** (Li et al. 2022b) captures real-world scenes with 17–20 GoPro views (1352×1014 resolution) featuring challenges like flames, shadows, and varied materials. The **D-NeRF Dataset** (Pumarola et al. 2021) provides monocular sequences with 50–200 training and 20 test images per scene, resized to 800×800. Our framework is implemented in PyTorch (Paszke 2019) and tested on a single RTX 3090 GPU; additional setup details are in the appendix.

Method	PSNR↑	SSIM↑	LPIPS↓	Train↓	FPS↑
DyNeRF (Li et al. 2022b)*	29.58	–	0.08	1344 h	0.015
StreamRF (Li et al. 2022a)	28.16	0.85	0.31	79 min	8.50
HyperReel (Attal et al. 2023)	30.36	0.92	0.17	9 h	2.00
NeRFPlayer (Song et al. 2023)	30.69	–	0.11	6 h	0.05
K-Planes (Fridovich-Keil et al. 2023)	30.73	0.93	0.07	190 min	0.10
MixVoxels (Wang et al. 2023b)	30.85	0.96	0.21	91 min	16.70
MSTH (Wang et al. 2023a)	29.46	0.92	0.17	36 min	2.66
DeformablEdge4DGS (Wu et al. 2023)	28.42	0.92	0.17	72 min	39.93
RealTimEdge4DGS (Yang et al. 2023b)	29.95	0.92	0.16	8 h	72.80
Ours	35.33	0.97	0.07	72 min	40.00

Table 1: Quantitative Comparison on Plenoptic Video Dataset. *: trained on 8 GPUs and tested only on the Flame Salmon scene.

Method	PSNR↑	SSIM↑	LPIPS↓	Train↓	FPS↑
D-NeRF (Pumarola et al. 2021)	29.17	0.95	0.07	24 h	0.13
TiNeuVox (Fang et al. 2022)	32.87	0.97	0.04	28 min	1.60
K-Planes (Fridovich-Keil et al. 2023)	31.07	0.97	0.02	54 min	1.20
FFDNeRF (Guo et al. 2023)	31.70	0.96	0.05	-	< 1.20
MSTH (Wang et al. 2023a)	30.40	0.97	0.05	9.80 min	-
V4D (Gan et al. 2023)	32.67	0.97	0.05	10.21 h	2.64
Deformable3DGS (Yang et al. 2023a)	39.31	0.99	0.01	26 min	85.45
DeformablEdge4DGS (Wu et al. 2023)	32.99	0.97	0.05	13 min	104.00
RealTimEdge4DGS (Yang et al. 2023b)	32.71	0.97	0.03	10 min	289.07
Ours	40.00	0.99	0.01	13 min	110.00

Table 2: Quantitative Comparison on D-NeRF Dataset.

Results

Evaluation on Plenoptic Video Dataset Table 1 compares our approach with leading NeRF- and Gaussian-based approaches on the Plenoptic Video dataset. Edge4DGS achieves 35.33 PSNR, about 4.5 dB higher than the best baseline, with the highest SSIM and lowest perceptual error. Unlike NeRF-based methods, which require hundreds of hours of training, Edge4DGS converges in 72 minutes and renders at 40 FPS in real time. Compared to Gaussian methods, it improves PSNR by over 6 dB with comparable efficiency, demonstrating superior fidelity, perceptual quality, and practical scalability for dynamic scene rendering. Figure 3 further demonstrates that Edge4DGS produces sharp, realistic renderings in challenging regions.

Evaluation on D-NeRF Dataset Table 2 reports results on the D-NeRF dataset. Edge4DGS achieves a PSNR of 40.00, SSIM of 0.99, and LPIPS of 0.01, substantially surpassing baseline methods. It exceeds the best NeRF-based method (TiNeuVox) by over 7 dB in PSNR with less than half the training time. Against Gaussian-based methods, it improves PSNR by over 7 dB, matches training efficiency, and achieves 110 FPS real-time rendering. With the lowest LPIPS score, Edge4DGS delivers state-of-the-art fidelity, fast convergence, and real-time inference for dynamic scene rendering. Figure 4 further confirms that Edge4DGS renders

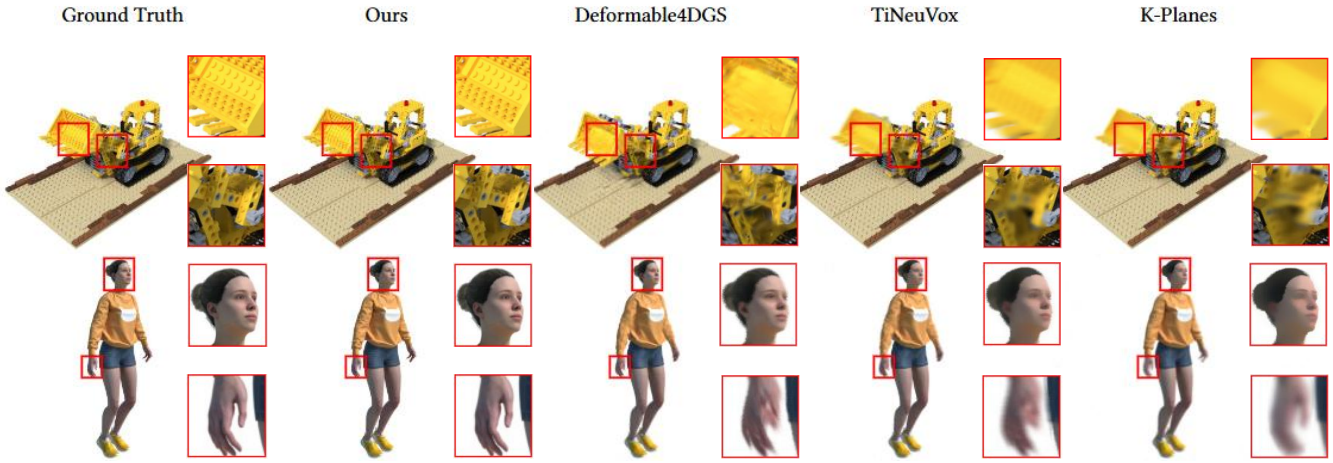

Figure 4: Qualitative Comparison on D-NeRF Dataset. Our approach demonstrates superior rendering of fine-grained edges, accurately rendering the complex contours of the bulldozer blade in *Lego* and the fingers in *Jumping jacks*.

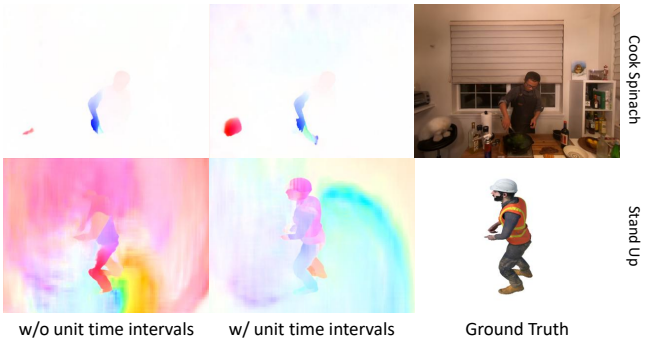

Figure 5: Optical Flow Visualization. Ground truth optical flows are extracted using Raft (Teed and Deng 2020).

ID	Convex	Edge	Time	PSNR↑	SSIM↑	LPIPS↓	Train↓
a				31.98	0.950	0.070	13 min
b	✓			35.68	0.971	0.052	14 min
c		✓		34.90	0.965	0.055	13 min
d		✓	✓	36.24	0.975	0.047	13 min
Full Resolution	✓	✓	✓	37.80	0.979	0.041	12 min
Full Joint	✓	✓	✓	38.80	0.981	0.018	14 min
Full Alternating	✓	✓	✓	39.50	0.989	0.012	14 min
Full Two-Stage	✓	✓	✓	**40.00**	**0.990**	**0.010**	**13 min**

Table 3: Ablation Study on D-NeRF Dataset. (*a*) Base, (*b*) Convex only, (*c*) Edge only (time-step), (*d*) Edge + interval, and (*Full*).

fine-grained edges, outperforming SOTA methods.

Ablation Studies

Convex Hull Representation. To evaluate the contribution of the convex primitive representation, we conduct an ablation in which only the convex module is enabled (Table **??**, *b*). Incorporating convex primitives alone yields a substantial PSNR improvement, accompanied by increases in SSIM and a reduction in perceptual error LPIPS. This gain demonstrates that the convex primitives effectively capture sharp surface boundaries and planar regions that cannot be well represented by Gaussian primitives alone. Experimental results confirm that convex primitives substantially enhance geometric fidelity and edge sharpness even.

Edge Regularization. We ablate edge consistency regularization by enabling it without convex primitives (Table **??**, *c*). Compared to the baseline, it raises PSNR from 31.98 to 34.90 without increasing point cloud size, showing that edge consistency regularization effectively constrains Gaussian distributions motion and reduces artifacts, enhancing fine edge rendering.

Temporal Edge Consistency. To assess temporal modeling, we evaluate a variant with temporal supervision (Table **??**, *d*). Adding the unit time interval constraint increases PSNR from 34.90 to 36.24, showing that enforcing optical flow consistency over continuous intervals provides stronger, more stable constraints than separate steps, better capturing motion trajectories and reducing flickering for coherent rendering. Figure 5 further illustrates that temporal consistency enables the accurate rendering of both the direction and gradient of object motion around sharp edges and fast-moving regions, while suppressing spurious background motion.

Two-Stage Optimization. We evaluate four training strategies for rendering quality and efficiency (Table **??**). Joint training updates Gaussians and convex hulls simultaneously but suffers from gradient conflicts, degrading rendering. Our two-stage decoupling avoids gradient interference and yields the highest fidelity. The alternating strategy switches between Gaussian and convex-edge objectives, improving over joint training but increasing runtime without matching two-stage quality. Progressive resolution starts at half resolution for faster convergence, then trains at full resolution, which enables faster speed. Experimental results confirm that the two-stage approach achieves the highest accuracy with balanced runtime.

Conclusion

In this study, we present the Edge 4D Gaussian Splatting (Edge4DGS) for high-quality, real-time dynamic scene modeling. To accurately reconstruct fine-grained edges, we integrate Gaussian distributions with convex hull representations, enabling a flexible, multi-level geometric modeling approach. Furthermore, we propose edge consistency regularization, extending geometric constraints from individual time steps to the entire unit time interval. This formulation enhances the temporal coherence of edge appearance while reducing point cloud complexity. Extensive experiments demonstrate that Edge4DGS surpasses existing methods in rendering quality, especially in preserving consistent, fine-grained edges. Moreover, Edge4DGS effectively handles complex temporal variations through transferable, edge-aware representations.

References

Attal, B.; Huang, J.-B.; Richardt, C.; Zollhoefer, M.; Kopf, J.; O'Toole, M.; and Kim, C. 2023. HyperReel: High-fidelity 6-DoF video with ray-conditioned sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16610–16620.

Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; and Srinivasan, P. P. 2021. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5855–5864.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2022. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5470–5479.

Bemana, M.; Myszkowski, K.; Revall Frisvad, J.; Seidel, H.-P.; and Ritschel, T. 2022. Eikonal fields for refractive novel-view synthesis. In *ACM SIGGRAPH 2022 Conference Proceedings*, 1–9.

Broxton, M.; Flynn, J.; Overbeck, R.; Erickson, D.; Hedman, P.; Duvall, M.; Dourgarian, J.; Busch, J.; Whalen, M.; and Debevec, P. 2020. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics (TOG)*, 39(4): 86–1.

Buehler, C.; Bosse, M.; McMillan, L.; Gortler, S.; and Cohen, M. 2001. Unstructured lumigraph rendering. *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

Cao, A.; and Johnson, J. 2023. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 130–141.

Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, 333–350. Springer.

Collet, A.; Chuang, M.; Sweeney, P.; Gillett, D.; Evseev, D.; Calabrese, D.; Hoppe, H.; Kirk, A.; and Sullivan, S. 2015. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (TOG)*, 34(4): 1–13.

Debevec, P. E.; Taylor, C. J.; and Malik, J. 1996. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

Du, Y.; Zhang, Y.; Yu, H.-X.; Tenenbaum, J. B.; and Wu, J. 2021. Neural radiance flow for 4D view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 14304–14314. IEEE Computer Society.

Duan, Y.; Wei, F.; Dai, Q.; He, Y.; Chen, W.; and Chen, B. 2024. 4D-Rotor Gaussian Splatting: Towards Efficient Novel View Synthesis for Dynamic Scenes. *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

Fang, J.; Yi, T.; Wang, X.; Xie, L.; Zhang, X.; Liu, W.; Nießner, M.; and Tian, Q. 2022. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*, 1–9.

Flynn, J.; Broxton, M.; Debevec, P.; DuVall, M.; Fyffe, G.; Overbeck, R.; Snavely, N.; and Tucker, R. 2019. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2367–2376.

Fridovich-Keil, S.; Meanti, G.; Warburg, F. R.; Recht, B.; and Kanazawa, A. 2023. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12479–12488.

Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q.; Recht, B.; and Kanazawa, A. 2022. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5501–5510.

Gan, W.; Xu, H.; Huang, Y.; Chen, S.; and Yokoya, N. 2023. V4D: Voxel for 4D novel view synthesis. *IEEE Transactions on Visualization and Computer Graphics*.

Gao, C.; Saraf, A.; Kopf, J.; and Huang, J.-B. 2021. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5712–5721.

Gortler, S. J.; Grzeszczuk, R.; Szeliski, R.; and Cohen, M. F. 1996. The lumigraph. *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

Graham, R. L. 1972. An efficient algorithm for determining the convex hull of a finite planar set. *nformation Processing Letters*, 1: 132–133.

Guo, K.; Lincoln, P.; Davidson, P.; Busch, J.; Yu, X.; Whalen, M.; Harvey, G.; Orts-Escolano, S.; Pandey, R.; Dourgarian, J.; et al. 2019. The relightables: Volumetric performance capture of humans with realistic relighting. *ACM Transactions on Graphics (TOG)*, 38(6): 1–19.

Guo, K.; Xu, F.; Wang, Y.; Liu, Y.; and Dai, Q. 2015. Robust non-rigid motion tracking and surface reconstruction using l0 regularization. In *Proceedings of the IEEE International Conference on Computer Vision*, 3083–3091.

Guo, X.; Sun, J.; Dai, Y.; Chen, G.; Ye, X.; Tan, X.; Ding, E.; Zhang, Y.; and Wang, J. 2023. Forward flow for novel

view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16022–16033.

Hedman, P.; Philip, J.; Price, T.; Frahm, J.-M.; Drettakis, G.; and Brostow, G. 2018. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (TOG)*, 37(6): 1–15.

Held, J.; Vandeghen, R.; Hamdi, A.; Deliege, A.; Cioppa, A.; Giancola, S.; Vedaldi, A.; Ghanem, B.; and Van Droogenbroeck, M. 2024. 3D Convex Splatting: Radiance Field Rendering with 3D Smooth Convexes. *arXiv preprint arXiv:2411.14974*.

Hu, T.; Yu, T.; Zheng, Z.; Zhang, H.; Liu, Y.; and Zwicker, M. 2022. Hvtr: Hybrid volumetric-textural rendering for human avatars. In *2022 International Conference on 3D Vision (3DV)*, 197–208. IEEE.

Huang, Y.-H.; Sun, Y.-T.; Yang, Z.; Lyu, X.; Cao, Y.-P.; and Qi, X. 2024. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4220–4230.

Kanade, T.; Rander, P.; and Narayanan, P. 1997. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE multimedia*, 4(1): 34–47.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4): 1–14.

Knapitsch, A.; Park, J.; Zhou, Q.-Y.; and Koltun, V. 2017. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (TOG)*, 36(4): 1–13.

Kopanas, G.; Leimkühler, T.; Rainer, G.; Jambon, C.; and Drettakis, G. 2022. Neural point catacaustics for novel-view synthesis of reflections. *ACM Transactions on Graphics (TOG)*, 41(6): 1–15.

Kutulakos, K. N.; and Seitz, S. M. 2000. A theory of shape by space carving. *International journal of computer vision*, 38: 199–218.

Levoy, M.; and Hanrahan, P. 1996. Light field rendering. *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

Li, H.; Luo, L.; Vlasic, D.; Peers, P.; Popović, J.; Pauly, M.; and Rusinkiewicz, S. 2012. Temporally coherent completion of dynamic shapes. *ACM Transactions on Graphics (TOG)*, 31(1): 1–11.

Li, L.; Shen, Z.; Wang, Z.; Shen, L.; and Tan, P. 2022a. Streaming radiance fields for 3D video synthesis. In *Advances in Neural Information Processing Systems*, 13485–13498.

Li, T.; Slavcheva, M.; Zollhoefer, M.; Green, S.; Lassner, C.; Kim, C.; Schmidt, T.; Lovegrove, S.; Goesele, M.; Newcombe, R.; et al. 2022b. Neural 3D video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5521–5531.

Li, Z.; Chen, Z.; Li, Z.; and Xu, Y. 2023. Spacetime gaussian feature splatting for real-time dynamic view synthesis. *arXiv preprint arXiv:2312.16812*.

Li, Z.; Ji, Y.; Yang, W.; Ye, J.; and Yu, J. 2017. Robust 3D human motion reconstruction via dynamic template construction. In *2017 International Conference on 3D Vision (3DV)*, 496–505. IEEE.

Li, Z.; Niklaus, S.; Snavely, N.; and Wang, O. 2021. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6498–6508.

Li, Z.; Wu, M.; Zhou, W.; and Yu, J. 2018. 4D Human Body Correspondences from Panoramic Depth Maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2877–2886.

Liu, Y.-L.; Gao, C.; Meuleman, A.; Tseng, H.-Y.; Saraf, A.; Kim, C.; Chuang, Y.-Y.; Kopf, J.; and Huang, J.-B. 2023. Robust dynamic radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13–23.

Luiten, J.; Kopanas, G.; Leibe, B.; and Ramanan, D. 2023. Dynamic 3D Gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*.

Mildenhall, B.; Srinivasan, P.; Tancik, M.; Barron, J.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*.

Mildenhall, B.; Srinivasan, P. P.; Ortiz-Cayon, R.; Kalantari, N. K.; Ramamoorthi, R.; Ng, R.; and Kar, A. 2019. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4): 1–14.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41(4): 1–15.

Park, K.; Sinha, U.; Barron, J. T.; Bouaziz, S.; Goldman, D. B.; Seitz, S. M.; and Martin-Brualla, R. 2021a. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5865–5874.

Park, K.; Sinha, U.; Hedman, P.; Barron, J. T.; Bouaziz, S.; Goldman, D. B.; Martin-Brualla, R.; and Seitz, S. M. 2021b. HyperNeRF: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics (TOG)*, 40(6): 1–12.

Paszke, A. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Penner, E.; and Zhang, L. 2017. Soft 3D reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)*, 36(6): 1–11.

Pumarola, A.; Corona, E.; Pons-Moll, G.; and Moreno-Noguer, F. 2021. D-NeRF: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10318–10327.

Riegler, G.; and Koltun, V. 2020. Free view synthesis. In *European Conference on Computer Vision*, 623–640.

Seitz, S. M.; and Dyer, C. R. 1999. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35: 151–173.

Shao, R.; Zheng, Z.; Tu, H.; Liu, B.; Zhang, H.; and Liu, Y. 2023. Tensor4D: Efficient neural 4D decomposition for high-fidelity dynamic reconstruction and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16632–16642.

Song, L.; Chen, A.; Li, Z.; Chen, Z.; Chen, L.; Yuan, J.; Xu, Y.; and Geiger, A. 2023. NeRFplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5): 2732–2742.

Srinivasan, P. P.; Tucker, R.; Barron, J. T.; Ramamoorthi, R.; Ng, R.; and Snavely, N. 2019. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 175–184.

Su, Z.; Xu, L.; Zheng, Z.; Yu, T.; Liu, Y.; and Fang, L. 2020. Robustfusion: Human volumetric capture with data-driven visual cues using a rgbd camera. In *The European Conference on Computer Vision*, 246–264.

Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *The European Conference on Computer Vision*, 402–419.

Thies, J.; Zollhöfer, M.; and Nießner, M. 2019. Deferred neural rendering: Image synthesis using neural textures. *Acm Transactions on Graphics (TOG)*, 38(4): 1–12.

Tian, F.; Du, S.; and Duan, Y. 2023. MonoNeRF: Learning a generalizable dynamic radiance field from monocular videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 17903–17913.

Tretschk, E.; Tewari, A.; Golyanik, V.; Zollhöfer, M.; Lassner, C.; and Theobalt, C. 2021. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 12959–12970.

Verbin, D.; Hedman, P.; Mildenhall, B.; Zickler, T.; Barron, J. T.; and Srinivasan, P. P. 2022. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5481–5490.

Waechter, M.; Moehrle, N.; and Goesele, M. 2014. Let there be color! Large-scale texturing of 3D reconstructions. In *European Conference on Computer Vision*, 836–850. Springer.

Wang, F.; Chen, Z.; Wang, G.; Song, Y.; and Liu, H. 2023a. Masked Space-Time Hash Encoding for Efficient Dynamic Scene Reconstruction. In *Advances in Neural Information Processing Systems*, 70497–70510.

Wang, F.; Tan, S.; Li, X.; Tian, Z.; Song, Y.; and Liu, H. 2023b. Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19706–19716.

Wood, D. N.; Azuma, D. I.; Aldinger, K.; Curless, B.; Duchamp, T.; Salesin, D. H.; and Stuetzle, W. 2023. Surface light fields for 3D photography. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 487–496.

Wu, G.; Yi, T.; Fang, J.; Xie, L.; Zhang, X.; Wei, W.; Liu, W.; Tian, Q.; and Wang, X. 2023. 4D Gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*.

Yan, Z.; Li, C.; and Lee, G. H. 2023. NeRF-DS: Neural radiance fields for dynamic specular objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8285–8295.

Yang, Z.; Gao, X.; Zhou, W.; Jiao, S.; Zhang, Y.; and Jin, X. 2023a. Deformable 3D Gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*.

Yang, Z.; Yang, H.; Pan, Z.; and Zhang, L. 2023b. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*.

Yi, T.; Fang, J.; Wang, X.; and Liu, W. 2023. Generalizable Neural Voxels for Fast Human Radiance Fields. *arXiv preprint arXiv:2303.15387*.

Yu, Z.; Chen, A.; Huang, B.; Sattler, T.; and Geiger, A. 2024. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19447–19456.

Zhang, K.; Riegler, G.; Snavely, N.; and Koltun, V. 2020. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*.

Zhou, T.; Tucker, R.; Flynn, J.; Fyffe, G.; and Snavely, N. 2018. Stereo magnification. *ACM Transactions on Graphics*, 37(4): 1–12.

Zitnick, C. L.; Kang, S. B.; Uyttendaele, M.; Winder, S.; and Szeliski, R. 2004. High-quality video view interpolation using a layered representation. *ACM transactions on graphics (TOG)*, 23(3): 600–608.