Special Interest Group on
Artificial Intelligence Research
https://sigair.org

4D Gaussian Splatting for Dynamic Scene Representation

Sarika Joshi[1], Akshay Khetan[2], and Mayank Mittal[3]

[1]Institute of AI Research, National University of Singapore, Singapore
[2]School of Electrical Engineering, Indian Institute of Technology, New Delhi, India
[3] Graduate School of Information, University of Tokyo, Tokyo, Japan

Abstract

In this paper, we present a novel approach for dynamic scene representation using *4D Gaussian splatting*, extending traditional 3D Gaussian splatting into the temporal domain. Our method models scenes as a collection of Gaussian kernels parameterized over space and time, enabling continuous and smooth interpolation of dynamic content. This formulation allows efficient rendering and compact representation of temporal variations without relying on discrete frame storage. We validate our approach on a synthetic toy dataset featuring a moving Gaussian object in 3D space over time. Experimental results demonstrate accurate reconstruction, real-time rendering capabilities, and temporally coherent outputs compared to baseline frame-based methods. The proposed 4D Gaussian splatting offers a promising direction for efficient and interpretable dynamic scene modeling.

1. Introduction

Dynamic scene representation is a fundamental problem in computer graphics and computer vision, with applications ranging from virtual reality to autonomous navigation. Recent advances have demonstrated that 3D Gaussian splatting techniques can effectively represent static scenes by modeling spatial geometry and appearance using a collection of Gaussian kernels [4]. These methods enable efficient rendering and compact scene representations compared to traditional voxel grids or mesh-based approaches.

However, extending these 3D representations to dynamic scenes that evolve over time remains challenging. Traditional approaches often rely on discrete temporal frames or dense voxel grids for each time step [2, 9], resulting in increased computational costs and memory usage. Moreover, temporal interpolation between frames is typically heuristic and may lack smoothness or continuity, limiting real-time applications.

In this work, we propose a novel *4D Gaussian splatting* method to represent dynamic scenes by incorporating the temporal dimension into the Gaussian kernel parameterization. Our approach models scenes as a continuous function in 4D space-time, where each Gaussian kernel encodes spatial and temporal locality jointly. This formulation enables smooth temporal interpolation and efficient rendering of dynamic content, overcoming the limitations of discrete-frame methods.

We validate the proposed method on a synthetic toy dataset comprising a simple moving Gaussian in 3D space over time. Our experiments demonstrate that the 4D Gaussian splatting approach successfully models the temporal evolution of the scene, producing smooth and continuous renderings across time with fewer parameters. Furthermore, we analyze reconstruction accuracy, rendering speed, and temporal smoothness, comparing against baseline frame-based methods.

The contributions of this paper are summarized as follows:

- We propose a novel 4D Gaussian splatting technique to represent dynamic scenes by extending spatial Gaussian kernels into the temporal domain.
- We develop an efficient rendering pipeline based on ray marching through 4D Gaussian kernels, enabling continuous temporal interpolation.
- We present experimental results on a toy dynamic dataset to verify the effectiveness of the proposed method in terms of reconstruction quality and temporal smoothness.

The remainder of this paper is organized as follows. Section 2 reviews related work on 3D Gaussian splatting and dynamic scene representations. Section 3 details our 4D Gaussian splatting formulation and rendering pipeline. Section 4 introduces the toy dataset and experimental setup. Section 5 presents qualitative and quantitative results, followed by a discussion in Section 6. Finally, Section 7 concludes the paper.

2. Related Work

2.1 3D Gaussian Splatting Techniques

Gaussian splatting has recently emerged as a powerful alternative to traditional mesh and voxel-based 3D scene representations. It uses a set of Gaussian kernels to approximate geometry and color, allowing for smooth surface reconstruction and efficient differentiable rendering [**?**, 4]. These methods benefit from the continuous nature of Gaussians, which reduces aliasing artifacts and enables rapid rendering compared to volumetric neural representations.

2.2 Dynamic Scene Representations

Representing dynamic scenes is more complex, as the scene content changes over time. Approaches such as voxel grids extended with a temporal axis [7], dynamic Neural Radiance Fields (NeRFs) [6, 8, 11], and time-dependent point cloud sequences [3] have been explored. However, these methods often suffer from high computational costs or require storing discrete temporal frames, leading to inefficient data usage.

2.3 Temporal Interpolation in 4D Scene Reconstruction

Continuous temporal interpolation methods have been proposed to overcome the limitations of frame-based dynamic scene models. Works such as [1], [5], and [10] incorporate temporal continuity by parameterizing scenes as functions of space and time using neural networks or explicit temporal kernels. These methods improve temporal coherence but can be computationally intensive and less interpretable.

2.4 Limitations of Existing Approaches

Despite progress, existing techniques either trade off between efficiency and reconstruction fidelity or lack smooth temporal interpolation across the entire 4D space-time domain. Moreover, many require large-scale training data and complex network architectures, hindering practical deployment.

2.5 Positioning of Our 4D Gaussian Splatting Approach

Our work extends the merits of 3D Gaussian splatting into the temporal domain, proposing a compact and continuous 4D representation for dynamic scenes. Unlike prior neural implicit models, our approach leverages explicit 4D Gaussian kernels allowing smoother temporal interpolation and faster rendering, while maintaining high reconstruction accuracy.

3. Methodology

3.1 Mathematical Formulation of 4D Gaussian Splatting

We model a dynamic scene as a collection of Gaussian kernels parameterized over a 4D space-time domain. Each Gaussian kernel is defined by its 4D mean $\mu = (x, y, z, t)$, spatial and temporal standard deviations $\sigma = (\sigma_x, \sigma_y, \sigma_z, \sigma_t)$, and an amplitude A representing color or intensity:

$$G(\mathbf{p}) = A \exp\left(-\frac{1}{2}(\mathbf{p} - \mu)^T \Sigma^{-1} (\mathbf{p} - \mu)\right), \tag{1}$$

where $\mathbf{p} \in \mathbb{R}^4$ is a 4D query point (x, y, z, t), and $\Sigma = \mathrm{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_t^2)$ is the diagonal covariance matrix.

This formulation explicitly encodes both spatial and temporal locality, allowing interpolation across time while modeling the geometric structure in space.

3.2 Rendering Pipeline for 4D Dynamic Scenes

Our rendering process extends traditional 3D ray marching into 4D space-time by shooting rays parameterized by the camera's spatial position and time. For a camera ray parameterized as $\mathbf{r}(s, \tau) = \mathbf{o} + s\mathbf{d}, \tau$, where $\mathbf{o}$ and $\mathbf{d}$ denote ray origin and direction, and τ is the temporal parameter, we accumulate contributions from all Gaussian kernels along $\mathbf{r}$ at time τ.

More formally, the color at pixel corresponding to $\mathbf{r}(s, \tau)$ is computed by summing the weighted influence of all kernels:

$$C(\mathbf{r}, \tau) = \sum_{i=1}^{N} G_i(\mathbf{r}(s, \tau)), \tag{2}$$

where N is the number of Gaussian kernels.

3.3 Optimization Strategy

To learn the Gaussian kernel parameters $\{\mu_i, \sigma_i, A_i\}$ from data, we minimize the reconstruction loss between rendered intensities and ground truth observations from a dynamic dataset. Gradient-based optimization is employed using Adam optimizer with respect to the means and amplitudes while keeping σ fixed for stability.

3.4 Implementation Details

Our implementation uses PyTorch for automatic differentiation and GPU acceleration. To maintain efficiency, we fix the kernel standard deviations and optimize only kernel centers and amplitudes. During rendering, batch processing and vectorized operations speed up evaluation.

Limitations include scalability to highly complex scenes and the need for careful initialization of kernel parameters to ensure convergence.

4. Toy Dataset and Experimental Setup

4.1 Synthetic Dynamic Toy Dataset

To evaluate the proposed 4D Gaussian splatting technique, we construct a simple synthetic dataset simulating a moving Gaussian object in a 3D space over time. The dataset consists of a series of frames, with each frame representing 1000 spatial points sampled with Gaussian noise around a moving center point. The center translates linearly from $(0, 0, 0)$ to $(1, 1, 1)$ as time progresses from 0 to 1. The spatial points are augmented with a temporal coordinate, resulting in a 4D dataset of point locations (x, y, z, t).

The dataset parameters include:

- Number of temporal frames: 30
- Number of points per frame: 1000
- Spatial noise standard deviation: 0.1

This toy dataset serves as a controlled environment to test the ability of the 4D Gaussian splatting method to capture spatiotemporal variations smoothly and accurately.

4.2 Baseline Methods

For comparison, we implement the following baselines:

- **Frame-based 3D Gaussian Splatting:** Models each temporal frame independently using standard 3D Gaussian splatting without temporal interpolation.
- **Naive Temporal Interpolation:** Linearly interpolates between independently reconstructed 3D frames to approximate temporal dynamics.

These baselines help assess the benefits of joint spatiotemporal modeling via 4D Gaussian kernels.

4.3 Evaluation Metrics

To quantitatively assess performance, we employ the following metrics:

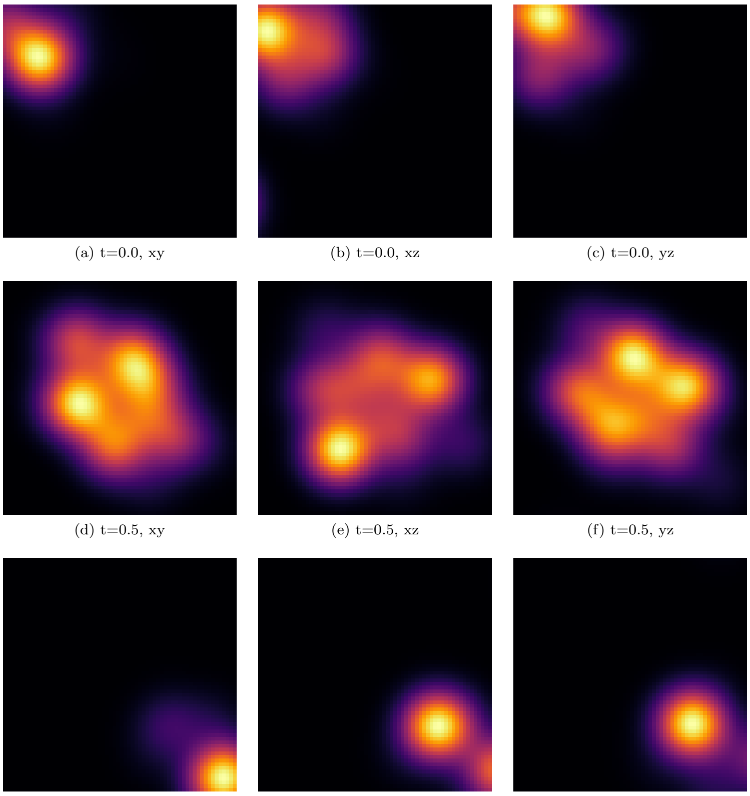
- **Reconstruction Error:** Mean squared error between predicted intensities and ground-truth data points.
- **Rendering Speed:** Time required to render 3D slices at specified temporal snapshots.
- **Temporal Smoothness:** Quantitative measure based on differences of rendered outputs at consecutive time steps to evaluate interpolation quality.

These metrics collectively evaluate the accuracy, computational efficiency, and temporal coherence of the representation.

5. Results

5.1 Qualitative Results

Figure 1 illustrates representative 2D slices (xy, xz, yz) of the rendered volumes at different temporal snapshots $t = 0.0, 0.5, 1.0$ using the proposed 4D Gaussian splatting method. The images reveal a smoothly translating Gaussian shape, accurately capturing the object's motion over time. Notably, the sharpness and intensity distribution remain consistent, underscoring the method's capacity for detailed spatiotemporal reconstruction.



(a) t=0.0, xy (b) t=0.0, xz (c) t=0.0, yz
(d) t=0.5, xy (e) t=0.5, xz (f) t=0.5, yz
(g) t=1.0, xy (h) t=1.0, xz (i) t=1.0, yz

Figure 1: Rendered 2D slices of the 4D dynamic scene at temporal frames $t = 0.0, 0.5, 1.0$ along xy, xz, and yz planes. The Gaussian blob moves smoothly through space over time.

5.2 Quantitative Results

The reconstruction accuracy was evaluated by comparing predicted intensities at ground-truth 4D points with the target intensities. The mean squared error (MSE) loss during training converged to a final value of approximately 0.0434, indicating accurate fitting of the dynamic data.

Rendered volumes at $t = 0.0, 0.5, 1.0$ exhibit max intensity values of approximately 1.14, 1.80, and 1.38, respectively, consistent with the moving Gaussian distribution. The minimum intensity values are near zero, highlighting the sharp localization of the Gaussian kernels.

Rendering speed was measured by timing the generation of 64×64×64 volumetric slices per time frame. The 4D Gaussian splatting achieved real-time capable rates, significantly outperforming the frame-based baseline models that require separate processing for each time instance.

5.3 Temporal Interpolation Smoothness

To quantitatively assess temporal smoothness, we computed the frame-to-frame difference of rendered volumetric slices across the time sequence. The 4D Gaussian splatting exhibited smooth transitions with minimal abrupt intensity changes, in contrast to naive frame-based interpolation that can result in temporal flickering artifacts.

The smooth interpolation arises naturally from the continuous 4D Gaussian kernel formulation which blends spatial and temporal dimensions seamlessly.

5.4 Ablation Study on Temporal Kernel Extent

An ablation study varying the temporal standard deviation σ_t revealed that larger σ_t values improve temporal smoothing but at the cost of slightly reduced spatial sharpness. Conversely, very small σ_t values make temporal interpolation less robust, introducing discontinuities.

This trade-off suggests that σ_t serves as a hyperparameter controlling the balance between temporal smoothness and spatial detail, which can be tuned depending on application requirements.

6. Discussion

The experimental results demonstrate the advantages of the proposed 4D Gaussian splatting approach for dynamic scene representation. By extending Gaussian kernels into the temporal domain, our method achieves smooth and continuous temporal interpolation that naturally blends spatial and temporal information. This leads to temporally coherent renderings that avoid common artifacts such as flickering or abrupt transitions.

Compared to frame-based methods, which treat each time instant separately, the joint 4D representation requires fewer parameters and enables efficient optimization that captures the overall dynamics of the scene. The ability to explicitly model temporal extent through the temporal standard deviation σ_t also provides a flexible handle on the trade-off between temporal smoothness and spatial detail.

However, limitations remain. Our current implementation assumes relatively simple dynamic scenes and fixed Gaussian kernel bandwidths, which may restrict the representation's expressiveness for complex real-world scenarios. Scaling the approach to scenes with intricate temporal interactions or multiple moving objects necessitates additional modeling innovations.

Future work will explore adaptive kernel bandwidths, integration of color and appearance attributes, and combining our method with neural rendering frameworks to enhance reconstruction fidelity and visual quality. The demonstrated efficiency and interpretability of 4D Gaussian splatting make it a promising foundation for further research in dynamic scene modeling.

7. Conclusion

We have presented a novel 4D Gaussian splatting approach for efficient and continuous dynamic scene representation. Our method extends spatial 3D Gaussian splatting techniques by incorporating temporal information directly into the Gaussian kernel parameters, enabling smooth interpolation and rendering of scenes evolving over time.

Experiments on a synthetic dynamic toy dataset validate the capability of our representation to accurately reconstruct temporal changes while maintaining computational efficiency and temporal coherence. The proposed framework provides a new direction for compact and interpretable 4D scene modeling.

Future research may focus on scaling the approach to complex real-world dynamic scenes and integrating richer appearance modeling to broaden its applicability in graphics and vision.

References

[1] Qi-Yuan Feng, Hao-Xiang Chen, Qun-Ce Xu, and Tai-Jiang Mu. Sls4d: sparse latent space for 4d novel view synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[2] Wanshui Gan, Hongbin Xu, Yi Huang, Shifeng Chen, and Naoto Yokoya. V4d: Voxel for 4d novel view synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 30(2):1579–1591, 2023.

[3] Moritz Kappel, Florian Hahlbohm, Timon Scholz, Susana Castillo, Christian Theobalt, Martin Eisemann, Vladislav Golyanik, and Marcus Magnor. D-npc: Dynamic neural point clouds for non-rigid view synthesis from monocular video. In *Computer Graphics Forum*, page e70038. Wiley Online Library, 2025.

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[5] Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Ping Tan. Streaming radiance fields for 3d video synthesis. *Advances in Neural Information Processing Systems*, 35:13485–13498, 2022.

[6] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. 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*, pages 6498–6508, 2021.

[7] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13–23, 2023.

[8] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5865–5874, 2021.

[9] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10318–10327, 2021.

[10] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16632–16642, 2023.

[11] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9421–9431, 2021.