3D Dynamic Scene Interpolation with Gaussian Splatting

Sanghyun Hahn[1]*, Wonjae Ho[2]*, Jungwoo Park[2]*
[1]Department of Mechanical and Aerospace Engineering, SNU
[2]Department of Electrical and Computer Engineering, SNU
{steve0221, teddy_bear, lawjwpark}@snu.ac.kr

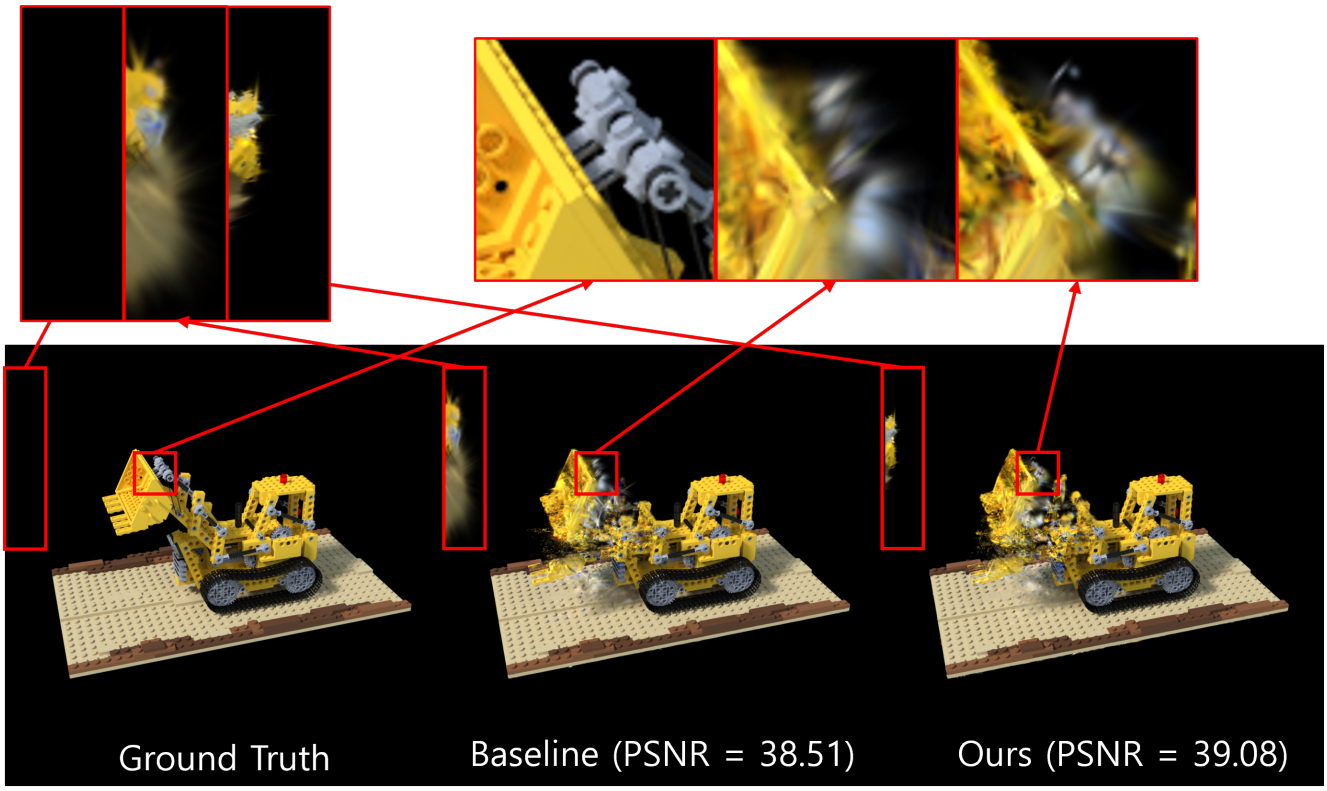


Figure 1. Novel view renders and the ground truth of the D-NeRF Lego dataset. [8] Our method reduces anomalies in the interpolated scene by incorporating an image loss generated from blended Gaussians, and outperforms Deformable 3D Gaussians. [13]

Abstract

Novel view synthesis for dynamic scenes is a important task in computer vision and graphics. Neural Radiance Fields (NeRF) have been an effective approach for static images, and recent extensions to dynamic scenes have been developed. However, NeRF-based methods suffer from high computational costs and slow rendering speeds. 4D Gaussian Splatting (4D-GS), which was inspired from 3D Gaussian Splatting (3D-GS), provides real-time rendering but still fails to interpolate across time effectively due to overfitting. To address the overfitting issue, this study introduces novel methods to enhance the interpolation of 3D dynamic scenes over time. We propose two key regularization terms: ***Single Gaussian Loss****, which employs linear interpolation on 3D Gaussian parameters, and* ***Multiple Gaussian Loss****, which improves training by penalizing significant deviations of multiple interpolated scenes from the original scene. These methods show improvements in generating realistic interpolations, advancing the novel view synthesis for dynamic scene.*

*Equal contributions

1. Introduction

Novel view synthesis for dynamic scenes is a crucial task in computer vision and graphics, since it plays a critical role in various ranges such as augmented reality (AR) and virtual reality (VR). NERF [6] is a powerful method to achieve the novel view synthesis task for static images. Inspired by NeRF, researchers have expanded the novel view synthesis task to dynamic scenes [1, 4, 7, 8] and challenged this task based on the NeRF framework. These metrics focus on the deformation between each frame which offers interpolated images for novel views.

However, NeRF based approaches have limitations regarding their computational cost and rendering speed. To solve this problem, 3D Gaussian Splatting (3D-GS) [2] was introduced, offering a boosted rendering speed by representing each scene with a sum of 3D Gaussians. This method supports real-time rendering and is also differentiable, allowing gradient based approaches for optimization. Recent works [5] applied 3D-GS to dynamic scenes by generate a set of 3D Gaussians for each frame. However, this approach is data-heavy and not robust since it does not utilize correspondences between frames.

In order to overcome the drawbacks, recent works have introduced 4D Gaussian splatting (4D-GS) [11, 14]. The 4D Gaussians represent calculates the Gaussian parameters of each timestep with a Gaussian Deformation Field, which takes the initial Gaussian and the desired timestep as the input. This method represents 3D scene along with their timewise differences, which greatly reduces the number of parameters while maintaining the ability to understand the dynamic motion through time.

However, 4D-GS still has limitations that it can not effectively interpolate between the time domain as visualized in Fig. 2. In other words, the neural network is strongly overfitted to time and does not understand the continuity of events.

In response, this study introduces novel methods aimed at enhancing the interpolation of 3D dynamic scenes over time. Two novel loss terms are proposed to address these limitations:

- **Single Gaussian Loss:** We propose a linear interpolation method operated directly on the 3D Gaussian parameters. By applying a time-based ratio, this method facilitates smoother interpolation between adjacent frames, resulting in more realistic and continuous outputs.
- **Multiple Gaussian Loss:** To further improve the training process of the deformation network, we suggest Multiple Gaussian loss. The loss function renders the multiple interpolated Gaussians and penalizes significant deviations from the original scene. This regularization helps reduce overfitting and encourages the network to create more realistic interpolations with less anomalies.

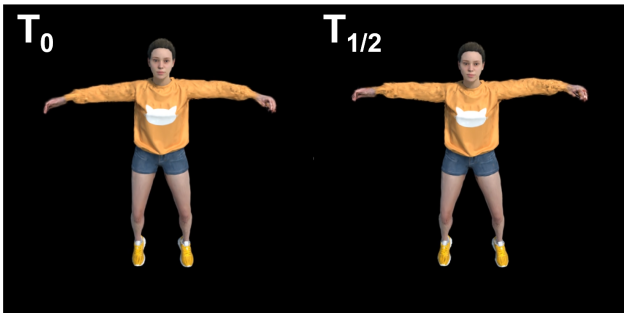


Figure 2. Visualization of the overfitting phenomenon of Deformable 3D Gaussians [13]. The baseline model fails to create natural novel view images in interpolated timesteps ($T_{1/2}$) compared to timesteps that were given ground truth (T_0).

2. Related works

2.1. NeRF for Dynamic Scenes

Neural Radiance Fields (NeRF) is one of the most popular works to challenge the novel view synthesis task through the use of Multi-Layer Perceptron (MLP). The MLP maps the 5D input of coordinate $\mathbf{x} = (x, y, z)$ and view direction $\mathbf{d} = (\theta, \phi)$ into a 4D output of color $\mathbf{c} = (r, g, b)$ and density σ. From the outputs, NeRF exploits volumetric rendering to generate images captured from any viewpoint.

NeRF for dynamic scenes share the same idea of radiance fields and volumetric rendering. One branch of dynamic NeRF [1, 8] uses a 6D input of $(\mathbf{x}, \mathbf{d}, t)$ to learn the radiance field. The MLPs in these methods learn the deformation of coordinates through time, including the affect of motion though the dynamic scenes to the model. Another approach merges the point-based approach into NeRF. [12] This method first generates a neural point cloud from CNN, then utilizes the point cloud to calculate the radiance field for volumetric rendering. However, the existing NeRF based methods fail to achieve real-time rendering, even though they partially produce high-quality outputs.

2.2. Dynamic Gaussian Splatting

3D-GS [2] represents the 3D scene as a point cloud with a sum of 3D Gaussians. Each 3D Gaussian is expressed as follows, where x is the position, Σ is the covariance matrix, and μ is the center of the Gaussian :

$$p(x|\mu, \Sigma) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{1}$$

The optimization process is performed by decomposing the covariance matrix Σ into scaling matrix $\mathbf{S}$ and rotation matrix $\mathbf{R}$:

$$\Sigma = RSS^T R^T \tag{2}$$

For the rendering process, the visible Gaussians are projected onto the camera plane, and each pixel is calculated by the combination of Gaussians.

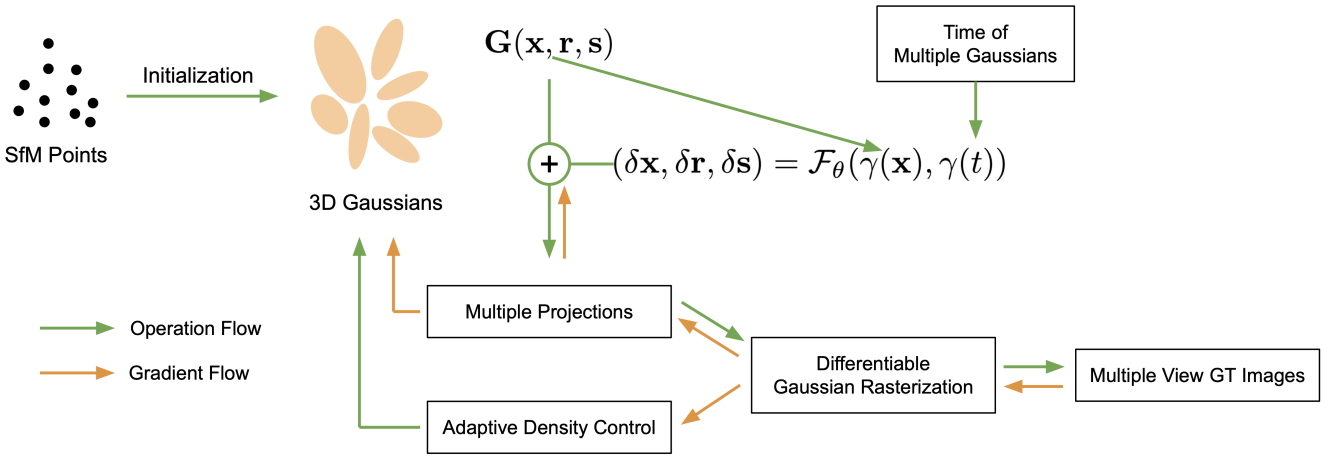
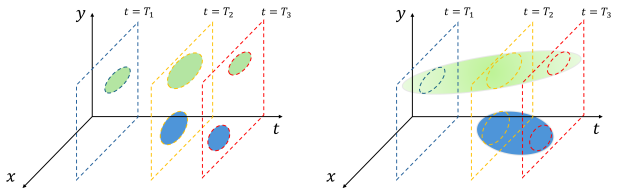


Figure 3. The overview of our pipeline. Structure from Motion (SfM) [10] initializes the 3D Gaussians for the initial frame. The initial Gaussian set is deformed by the MLP to obtain the Gaussian set at the desired timestep.



(a) First Image (b) Second Image

Figure 4. Comparison between 3D Gaussian Splatting (a) and 4D Gaussian Splatting (b). 3D-GS requires a set of Gaussians for every frame, while 4D-GS simply renders the sections from the 4D Gaussian for each frame.

The intuitive method for applying 3D-GS to dynamic scenes is generating a set of Gaussians for every frame. [3] However, this method requires a total number of Nt parameters where N is the number of Gaussian parameters in a single scene, and t is the number of frames. Since N is already a large value, this method demands an excessive data space. Recent studies[9, 11, 13, 14] suggest using a neural network for obtaining the deformation of the Gaussian parameters between frames. These methods are called Deformable 3D Gaussians or 4D Gaussian splatting, where they reduced the number of required parameters to $N + F$ where F is the number of parameters for the neural network. The comparison between 3D-GS and 4D-GS are visualized in Fig. 4

3. Method

The input to the the model is a set of monocular images with the corresponding camera parameters and timestamps. To generate a novel view image from the given input, the 3D Gaussian set at the desired timestep must be constructed. Yang et al. [13] created a deformation network that calculates the offset for the parameters of each Gaussian. This deformation network $\mathcal{F}_\theta$ takes the given time t and center position $\mathbf{x}$, and returns the deformation parameters of the particular Gaussian $\mathbf{G}_0$:

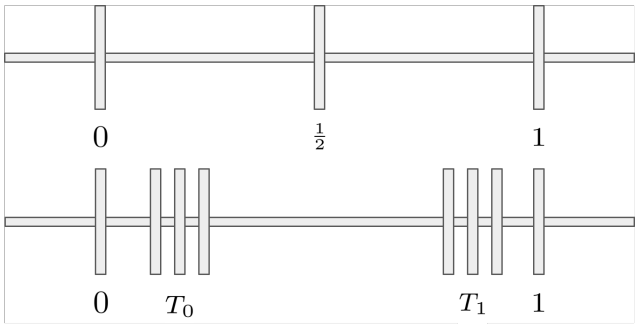


Figure 5. Visualization of Blended Gaussian Losses: Single Gaussian Loss (Top) and Multiple Gaussian Loss (Bottom). Gaussians are interpolated into blending gaussians of different timesteps. The blended gaussians are rendered into novel view images and are compared with ground truth to calculate Blended Gaussian Loss.

$$(\delta\mathbf{x}, \delta\mathbf{r}, \delta\mathbf{s}) = \mathcal{F}_\theta(\gamma(\mathbf{x}), \gamma(t)) \tag{3}$$

where γ represents positional encoding. The initial set of 3D Gaussians, $\mathbf{G_0}$ are obtained from a point cloud of a random cubic, or alternatively from a point cloud constructed by Structure from Motion (SfM) if multi-views are available.

However, this model alone struggles to interpolate scenes within time properly.

$$(\delta\hat{\mathbf{x}}_\tau, \delta\hat{\mathbf{r}}_\tau, \delta\hat{\mathbf{s}}_\tau) = \mathcal{F}_\theta(\gamma(\mathbf{x}), \gamma(\tau)) \tag{4}$$

In other words, $\mathbf{G_{t+\tau}}$, where τ is a value between 0 and 1, resembles $\mathbf{G_t}$ or $\mathbf{G_{t+1}}$ due to strong overfitting.

Scale	1/2			1/3			1/4			1/5		
Method	PSNR↑	SSIM↑	MS-SSIM↑	PSNR↑	SSIM↑	MS-SSIM↑	PSNR↑	SSIM↑	MS-SSIM↑	PSNR↑	SSIM↑	MS-SSIM↑
Original	39.0235	0.9912	0.9969	37.1688	0.9866	0.9943	35.7012	0.9831	0.9920	33.9837	0.9781	0.9872
Gaussian Interpolation	39.0416	0.9912	0.9969	37.1760	0.9866	0.9942	35.7217	0.9832	0.9920	34.0632	0.9784	0.9875
Original + L_{SG}	38.9907	0.9911	0.9968	37.3082	0.9870	0.9943	35.7440	0.9831	0.9919	33.7628	0.9777	0.9868
G.I + L_{SG}	39.0130	0.9912	0.9968	37.3077	0.9869	0.9943	35.7792	0.9832	0.9920	33.8522	0.9781	0.9872
Original + L_{MG}	39.0860	0.9911	0.9968	37.4173	0.9869	0.9944	35.7709	0.9830	0.9919	33.8948	0.9779	0.9868
G.I + L_{MG}	39.1106	0.9912	0.9968	37.4139	0.9869	0.9944	35.7926	0.9831	0.9919	33.9803	0.9783	0.9872

Table 1. **Quantitative comparison on synthetic dataset**. We compare the proposed loss terms with/without Gaussian interpolation in different downscaling of the inputs. The PSNR, SSIM, MS-SSIM values are reported, while the color for the PSNR cells denote the best and second best results.

From this baseline model, we experimented three different metrics to enhance the interpolation results.

3.1. Gaussian Interpolation

Gaussian interpolation is a method that directly applies linear interpolation to the gaussian parameters.

$$(\delta\hat{\mathbf{x}}_\tau, \delta\hat{\mathbf{r}}_\tau, \delta\hat{\mathbf{s}}_\tau) = (1-\tau)(\delta\mathbf{x}_0, \delta\mathbf{r}_0, \delta\mathbf{s}_0) + \tau(\delta\mathbf{x}_1, \delta\mathbf{r}_1, \delta\mathbf{s}_1) \quad (5)$$

Directly blending the gaussians can be a solution to overfitting issues.

3.2. Blended Gaussian Loss with Single Gaussian

We created an additional loss leveraging the interpolated Gaussian parameters. In Single Gaussian Loss, only the midpoint of consecutive Gaussians are considered. Subsequently, the novel view image is rendered from the obtained Gaussian parameters and the L1 loss between the rendered image and the ground truth is computed. This loss penalizes anomalies created during Gaussian interpolation.

$$\mathcal{L}_{SG} = \mathcal{L}_1(\mathbf{I}_0, \hat{\mathbf{I}}_{\frac{1}{2}}) + \mathcal{L}_1(\mathbf{I}_1, \hat{\mathbf{I}}'_{\frac{1}{2}}) \quad (6)$$

3.3. Blended Gaussian Loss with Multiple Gaussians

In order to enhance Single Gaussian Loss, the Multiple Gaussian Loss renders images from multiple Gaussians with different interpolation ratios. Subsequently,the average L1 loss is calculated compared to the ground truth images. In our experiments, 6 images are interpolated and compute the average loss is utilized as the Multiple Gaussian Loss. The Blended Gaussian Losses are visualized in Fig. 5.

$$\mathcal{L}_{MG} = \sum_{\tau \in T_0} \mathcal{L}_1(\mathbf{I}_0, \hat{\mathbf{I}}_\tau) + \sum_{\tau \in T_1} \mathcal{L}_1(\mathbf{I}_1, \hat{\mathbf{I}}'_\tau) \quad (7)$$

4. Experiment

The synthetic dataset [8] was used in the experiment in order to achieve fast training. To evaluate the interpolation quality, the dataset was downsampled with ratios from 2 to 5 and the remains were used as ground truth. The Gaussian Deformation Field was trained with six options: Original loss term, Gaussian interpolation, Original loss term with Single/Multiple Gaussian Loss, and G.I with Single/Multiple Gaussian Loss. The results were evaluated by rendering the novel view images for views that match the ground truth, and comparing them with the ground truth via Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). The quantitative results are described in Tab. 1.

When Gaussian interpolation was utilized, there was a consistent increase in the PSNR and SSIM metrics. Addition of Blended Gaussian Loss contributed to reducing irregularities in the interpolated gaussians as shown in Fig. 1. In particular, the use of the Multiple Gaussian Loss yielded the highest metric scores compared to other methods.

5. Conclusion

In this work, we proposed a Gaussian interpolation method to avoid undesired results from deformation field overfitting. Also, Blended Gaussian Loss is introduced which reduces severe distruptions of the interpolated scene. From the quantitative evaluation of experiment results, both Gaussian interpolation and Blended Gaussian Loss showed improvement in SSIM metrics. Comparing the two Blended Gaussian Losses, Multiple Gaussian Loss performed better across all cases.

However, despite the improvements shown by Blended Multiple Gaussian Loss, realistic interpolated scenes cannot be effectively generated from downsampled training images. We suggest that leveraging generative models such as diffusion models to create novel view images may potentially improve the overall quality of the deformation field.

References

[1] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023. 2

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4):1–14, 2023. 2

[3] Agelos Kratimenos, Jiahui Lei, and Kostas Daniilidis. Dynmf: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian splatting. *arXiv preprint arXiv:2312.00112*, 2023. 3

[4] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022. 2

[5] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023. 2

[6] 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. *Computer Vision–ECCV 2020*, pages 405–421, 2020. 2

[7] 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. 2

[8] 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. 1, 2, 4

[9] Jiawei Ren, Liang Pan, Jiaxiang Tang, Chi Zhang, Ang Cao, Gang Zeng, and Ziwei Liu. Dreamgaussian4d: Generative 4d gaussian splatting. *arXiv preprint arXiv:2312.17142*, 2023. 3

[10] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 3

[11] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023. 2, 3

[12] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5438–5448, 2022. 2

[13] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023. 1, 2, 3

[14] Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023. 2, 3