Sugestão de relatório

Revisor
Marcelo de Sousa
Universidade de São Paulo
marcelo.sousa@usp.br

Arqueólogo
Marcelo de Sousa
Universidade de São Paulo
marcelo.sousa@usp.br

Hacker
Fabrício Assunção
IME
fabricio.asfora.001@gmail.com

PhD Student
Esteban Wirth
esteban.wirth.97@gmail.com

1. Review

The paper ”4D Gaussian Splatting for Real-Time Dynamic Scene Rendering” presents an innovative framework that combines spatial and temporal modeling for rendering dynamic scenes. It builds on foundational Gaussian splatting techniques, introducing key advancements that improve performance, accuracy, and efficiency.

1.1. Summary

- **Problem Addressed:** The challenge of rendering dynamic scenes with high visual fidelity, low latency, and efficient memory usage in real-time.
- **Motivation:** Existing methods, including 3D Gaussian Splatting, struggle to address temporal dynamics and adapt to high-motion or deformable environments.
- **Method Summary:**
  - **HexPlane Architecture:** Introduces multi-resolution voxel planes that encode spatial and temporal data compactly, balancing computational performance and detail fidelity.
  - **Deformation Decoders:** Multi-head decoders adjust Gaussian attributes dynamically to accommodate motion and deformations.
  - **Differentiable Splatting:** Implements splatting techniques that enhance rendering quality while reducing artifacts.
  - **Initialization via SfM:** Employs Structure from Motion techniques to generate consistent initial representations for subsequent dynamic modeling.
- **List of Contributions:**
  - Unified spatio-temporal modeling framework.
  - Improved memory efficiency and adaptability to dynamic environments.
  - Extensive validation on datasets like D-NeRF and HyperNeRF, achieving state-of-the-art results.
  - Introduction of HexPlane as a core innovation for real-time Gaussian splatting.

The methodology is thorough, though some areas, such as the mathematical rationale for deformation adjustments and the selection of loss functions, could benefit from further elaboration. Visual comparisons and extended experimental results would also enhance clarity.

1.2. Positive Points

- Demonstrates innovative integration of spatio-temporal modeling using HexPlane.
- Achieves superior rendering quality with reduced latency and memory usage.
- Provides practical applications for VR, simulations, and generative modeling.
- Extends foundational Gaussian splatting approaches, creating new research opportunities.

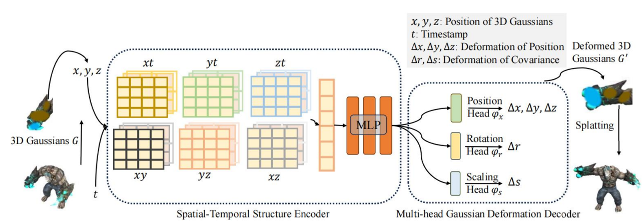

Figure 1. Overview of the Spatial-Temporal Structure Encoder and Multi-head Gaussian Deformation Decoder pipeline. The figure shows how 3D Gaussians are adjusted dynamically to handle position, rotation, and scaling across spatial and temporal dimensions.

1.3. Negative Points

- Limited experimental evaluation on large-scale and diverse datasets.
- Lack of detailed explanation for certain architectural design choices.
- Potential reproducibility challenges due to limited documentation of parameters and dependencies.

1.4. Evaluation

Score: **4.7/5** Recommendation: Accept with minor revisions. The paper makes transformative contributions but requires additional experiments and elaboration on specific methodological details to maximize its impact.

2. Archaeological Context

2.1. Historical Context

The paper "4D Gaussian Splatting for Real-Time Dynamic Scene Rendering" builds on the foundation of 3D Gaussian Splatting, addressing its limitations by integrating temporal dynamics. Central to this work is the HexPlane architecture, which employs multi-resolution voxel planes to encode Gaussian attributes such as position, size, and orientation across spatial and temporal dimensions (x, y, z, t). HexPlane's design enables efficient representation and real-time adaptability, addressing challenges like memory overhead and motion complexity. [1] [Figure 2]

When compared to K-Planes, HexPlane provides significant advancements. K-Planes utilize bilinear interpolation across multiple 2D planes to approximate 3D structures, as depicted in the attached image. However, K-Planes struggle with temporal dynamics and finer-grained spatial resolution due to their reliance on lower-dimensional feature interpolation. HexPlane improves on this by integrating multi-resolution voxel grids that account for both spatial and temporal variations, allowing precise deformation handling and smoother transitions over time.

The table of related works highlights these advancements by quantitatively comparing HexPlane's performance in terms of memory efficiency, computational cost, and visual fidelity. Unlike earlier methods that focus on static or quasi-static scenes, HexPlane achieves real-time adaptability and superior reconstruction quality for dynamic scenarios.

2.2. Current Works and Future Connections

The innovations in this paper have inspired various applications and future directions:

- **Medical Applications:** Extensions of the framework support surgical simulations, leveraging HexPlane for real-time organ deformation modeling. This improves the accuracy of training environments for medical professionals. [4]
- **Text-to-4D Synthesis:** Combines HexPlane with generative text-based models to create dynamic virtual environments, advancing interactive storytelling and immersive VR. [3]

2.3. Reference Adequacy

The paper cites key foundational works, such as 3D Gaussian Splatting and differentiable splatting techniques, but could benefit from additional references:

- Research on neural representations for deformable objects.
- Studies on distributed 4D rendering techniques for collaborative systems.
- Comparative analyses of voxel-based encoding strategies for dynamic scenes.

The table of related works could be expanded to provide quantitative comparisons, such as rendering times and memory usage, further illustrating HexPlane's advantages over alternative architectures.

2.4. Conclusion

The paper represents a significant advancement in Gaussian splatting techniques, addressing temporal dynamics through the innovative HexPlane architecture. Its applications span medical simulations, VR, and generative modeling, emphasizing its transformative potential. With extended validation on diverse datasets and detailed exploration of HexPlane's design choices, the framework could set a new standard in dynamic scene rendering.

3. Hacker

3.1. Source Code Evaluation

While **4D Gaussians** source code implements some theoretical notions, it suffers from poor organization and is hardly useful. While working, this code is tough to modify

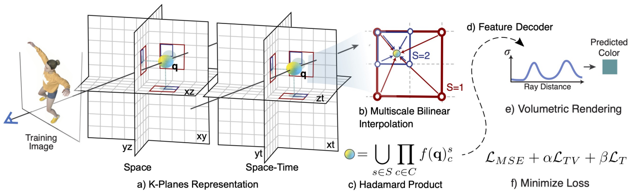

Figure 2. Overview of K-Planes architecture. The figure illustrates multiscale bilinear interpolation across 2D spatial and space-time planes, combined with feature decoding and volumetric rendering for ray-based color prediction.

for any other experiment or dataset due to the lack of comments and documentation and because most of the hyperparameters are hardcoded. Even the README itself does not give sufficient information to set up the datasets for Hypernerf and hence limits reproducibility in Colab notebooks and presents a major barrier to end researchers. The code connects theory with practice by means of the following elements. It makes use of spherical harmonics for color space mapping, not discussed in the paper, for smooth directional lighting. Time is included in the framework of 3D Gaussian Splatting to realize temporal pre-filtering for motion coherence with reduced aliasing. Secondly, hexagonal plane sampling replaces a full 4D grid, reducing spatial complexity from $O(n^4)$ to $O(n^2)$ for increased scalability and robustness during upscale.

With all those strengths, the code also has important design weaknesses: dead code and commented-out code in the repository, not modularizing encoders and decoders into their respective classes as per object-oriented programming, and a messy repository that makes it difficult to navigate. In the process of cleaning up the code base, the code also needs reorganization to be more user-friendly and reproducible.

3.2. Experiments and Reproducibility Testing

Authors performed experiments on DNerf's Trex dataset consisting of 2 NVIDIA T4 GPUs with 16GB GDDR6 each. The default hyperparameters taken from the paper represent grid learning rate (0.0016 initial, 0.000016 final), deformation learning rate (0.00016 initial, 0.000016 final), total number of iterations - 20,000, and the period to prune 8,000 iterations. The reason behind choosing DNerf is there is little detail on how Hypernerf datasets are collected, without which they are impracticable for use. This is further reduced by this limitation and the necessity of manual edits in data loaders for reproducibility in new datasets.

The method with default hyperparameters balances temporal stability, minimal aliasing, and smooth transitions best. Reducing iterations degraded the temporal coherence and increased aliasing, while doubling the grid learning rate accelerated convergence but caused flickering in high-frequency regions. Halving the learning rates of the deformations or pruning intervals led to sharper details with the introduction of some artifacts, while doubling the deformation rates resulted in improved temporal coherence at the cost of lost fine details. Very aggressive changes, such as a 5× deformation learning rate and a doubled grid learning rate, caused instability and poor visual quality. Reducing iterations and grid learning rates by half resulted in balanced coherence at the expense of increased aliasing in textures. These results indicate that the method is sensitive to its hyperparameters and therefore careful tuning is required in order to achieve the best performance.

3.3. Conclusion

While **4D Gaussians** bridges theory and implementation effectively, poor code design, insufficient documentation, and hardcoded configurations hinder reproducibility and adaptability. The experiments confirmed strong performance with default settings but revealed high sensitivity to parameter changes. Refactoring the codebase, improving documentation, and externalizing configurations are necessary to make the method more accessible for researchers.

4. PhD project

Even though the current model is a steep improvement over previous models and has very good results there are a few areas that could be improved. Particularly, there is a problem for identifying, and therefore modeling, static gaussians vs dynamic gaussians. Very much related to this limitation there is also a lacking of background points in teh development of the scene. In order to address this problem we could try to use 2 dimensional gaussians.

By using 2DGS [2] we can recreate a mesh and have background points being developed which tackles the lack of background points problem we just mentioned. Furthermore by introducing a reasonable assumption of background points being more likely to remain static while object points being more likely to be dynamic we can also improve our capability to detect which gaussians should be static and which should be dynamic.

Another improvement that is provided by using 2DGS is that we have a better assessment of depth as discussed in [2]. More over, we can create a depth mesh that in essence identifies in what section of the scene there are objects. By having this mesh vary over time it creates a better method to identify where in the scene there is movement and where in the scene there is static behavior. This improves the problems that we identified at the beginning of the section.

In order to be able to do this we would have to include depth and normal consistency loss functions as are introduced in [2]. This would also mean that rather than tracking the variance of the gaussians through time; we would instead track the two vectors that correspond to highest and least variation of the gaussian. In particular we need to create time dependent rotation and translation matrices that keep a consistent representation of the scene from different view points.

5. Conclusions

The paper "4D Gaussian Splatting for Real-Time Dynamic Scene Rendering" represents a significant advancement in Gaussian Splatting techniques, particularly through the integration of spatio-temporal modeling via the innovative HexPlane architecture. The primary contributions, such as the unified spatio-temporal framework, enhanced memory efficiency, and adaptability to dynamic environments, position this work at the forefront of real-time dynamic scene rendering.

The presented results demonstrate superior rendering quality with reduced latency and memory usage, validating the method's effectiveness on datasets like D-NeRF and HyperNeRF. Furthermore, practical applications in areas such as virtual reality, medical simulations, and generative modeling highlight the transformative potential of this approach.

However, the study has certain limitations that could be addressed in future work. The experimental evaluation was confined to specific datasets, and a more comprehensive analysis across diverse scenarios would reinforce the robustness of the conclusions. Additionally, the lack of detailed explanations for certain architectural aspects and the sensitivity to hyperparameters indicate a need for more thorough documentation and the development of more robust parameterization strategies to facilitate reproducibility and adaptability of the method.

Based on the analyses conducted, the following alternative titles are suggested for the paper:

- "HexPlane: A Spatio-Temporal Approach for Real-Time Dynamic Scene Rendering with 4D Gaussian Splatting"
- "Real-Time Dynamic Scene Rendering Using 4D Gaussian Splatting: Integrating Spatio-Temporal Modeling with HexPlane Architecture"
- "Advancements in Dynamic Scene Rendering: 4D Gaussian Splatting and the Innovative HexPlane Architecture"

Additionally, a missing result that could enrich the paper is the inclusion of a comparative evaluation on a more diverse and larger-scale dataset. This would allow for assessing the generalizability of the method and its applicability across different contexts, thereby strengthening the argument for HexPlane's superiority over existing approaches. Incorporating additional metrics, such as performance in highly dynamic environments or under significant lighting variations, would also provide a more comprehensive understanding of the methodology's capabilities and limitations.

In summary, the work offers valuable contributions to the field of dynamic scene rendering, providing a solid foundation for future research and practical applications. With improvements in documentation, expanded experimental evaluations, and exploration of new directions—such as distinguishing between static and dynamic Gaussians—the proposed framework has the potential to set new standards in the area.

References

[1] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Angjoo Kanazawa, Jonathan T Barron, and Ren Ng. K-planes: Explicit radiance fields in space, time, and appearance. *arXiv preprint arXiv:2301.10241*, 2023. 2

[2] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 3, 4

[3] Huan Ling, Seung Wook Kim, Antonio Torralba, Sanja Fidler, and Karsten Kreis. Align your gaussians: Text-to-4d with dynamic 3d gaussians and composed diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[4] Yifan Liu, Chenxin Li, Chen Yang, and Yixuan Yuan. Endogaussian: Gaussian splatting for deformable surgical scene reconstruction, 2024. 2