Report of "CoherentGS: Sparse Novel View Synthesis with Coherent 3D Gaussians"

Diana Aldana
IMPA
diana.aldana@impa.br

Horácio Brescia Macêdo Henriques
Universidade Federal Fluminense
horaciomacedo@id.uff.br

Mohara de Oliveira Nascimento
Instituto Tecgraf/PUC-Rio
mohara.civil@gmail.com

Victor Ferrari Pinto Sassi
Universidade Federal Fluminense
victorferrari@id.uff.br

1. Reviewer

1.1. Summary

Recently, gaussian splatting has emerged as a powerful, compact 3D representation of 3D scenes from multiple views ([2]). This technique, despite useful, suffers from heavy overfitting when trained on sparse views. This problem has been tackled for other representations such as NeRFs ([8, 11]), but did not have any advancement for gaussian representations. In that sense, CoherentGS: Sparse Novel View Synthesis with Coherent 3D Gaussians (Coherent GS) aims to mitigate the artifacts generated by sparse views by introducing a coherence during training. Specifically, they initialize isometric gaussians using depth maps and enforce constraints on the movement and opacity of the gaussians considering the optical flow between images and correspondences across views. This mitigates the heavily anisotropic behavior of gaussians during training.

Based on the sparse views ($2 \sim 4$ images), CoherentGS uses an already trained FlowFormer [] and Depth Anything [] to obtain the flows and monocular depth maps of each view. This last one is used to construct segmentation maps M via depth quantization. Now, to initialize the gaussians determined by the positions $\mathbf{x}$, colors $\mathbf{c}$ and covariance matrix Σ, we consider the coherence of a single view and multi-views. For a single view i, an isometric gaussian is initialized for each pixel p at a depth from the camera determined by the monocular depth D_i^m. At first, they initialize its rotation matrix to identity, and consider the radius as $r = f \cdot D_i^{init}/H$ where f is the focal lenght of the ith image and H is the image's height; this guarantees the gaussian splat covers the pixel p.

For a multi-view, they consider the correspondences between the depth masks of different images $M_{i \leftrightarrow j}$. At this stage, closeness is enforced between gaussians associated to pixels p and q of different images i and j that project to the same 3D point. This is done via optimization using the loss term

$$s^*, o^* = \arg\min_{\mathbf{s},\mathbf{o}} \sum_{(i,j)} \sum_{p} \Big|\Big| M_{i \leftrightarrow j} \odot \Big(g(s_i \cdot D_i^m[p] + o_i, p) - g(s_j \cdot D_j^m[q] + o_j, q) \Big) \Big|\Big|_1$$

where $\mathbf{s}, \mathbf{o}$ are the scales and offsets of all (isometric) gaussians and the function $g(d, p)$ projects the pixel p to a depth d in the space. Then, they are mainly adjusting the scales and offsets to reduce the distance between gaussians coming from pixels projected from the same spatial point. This way, the initialization of the gaussians is given by $D^{init} = \mathbf{s} \cdot D^{mono} + \mathbf{o}$.

This gives an structured initialization of gaussians. Then, they train the gaussians to fit the scene maintaining coherence. First, the positions vary only along the ray via a residual depth ΔD_i that is computed using a decoder. The scale is also implicitly updated by the formula of the radius r. This means that the training of position and scale of the gaussians is done implicitly by optimizing the parameters of the decoder. The opacity is trained in a similar fashion. Additionally, to guarantee smoothness in the geometry, they use regularization terms based on total variation,

$$\mathcal{L}_{multi} = (1-\lambda_s) \left|\left| \nabla \left(\frac{1}{1+R} \right) \right|\right|_1 + \lambda_s \left|\left| \nabla \left(\mathbf{S} \odot \frac{1}{1+R} \right) \right|\right|_1$$

(where R refers to the rendered depth in a pixel p) and op-

tical flow,

$$\mathcal{L}_{flow} = \sum_{(i,j)} \sum_{p} \left|\left| M_{i \leftrightarrow j} \odot \Big(g(D_i[p], p) - g(D_j[q], q) \Big) \right|\right|_1 .$$

The loss term $\mathcal{L}_{multi}$ is confusing and simplifies the notation (such as suppressing the summations), which leads to confusion. Besides, it is not clear why the second summand is necessary since the notation **S** was not clarified. On the other hand, it is easy to see what is the term $\mathcal{L}_{flow}$ doing. It ensures that the depth of pixels projected to the space that are in the same segmentation mask is approximately the same.

1.2. Strenghts

- The paper is very well written, with good illustrative figures to explain their method.
- The experiments were complete.
- The idea of using generative models to inpaint the empty regions left by occlusion was an interesting.

1.3. Weaknesses

- The decoder structure could have been better described to understand the extra parameters used.
- Some notation was not clearly stated in the paper.
- A previous reference of the topic was ignored ([9]). However, since it was a master dissertation, it is understandable to have omitted it accidentally.

1.4. Evaluation

I consider the method to be clear and innovative, with mostly good explanations and organized structure. Additionally, the points to improve are mostly to further improve the work, and do not raise big concerns. Furthermore, using generative models to complete the scene is an interesting direction of work. Considering the previous points, I would accept this work with a rating of 5.

2. Archaeologist

The present work is a new chapter in the history of NeRF-related works that aim to reconstruct scenes with a very limited dataset and high quality. Being part of an already existing history proves this is a challenging problem with awe-inspiring solutions that breed new iterations. The lack of variety in its training dataset can lead models to overfit their results, becoming overly competent in displaying the scene on known positions and directions but inferring unusable data between the images, which is the most interesting part of such reconstructions.

As part of a whole, CoherentGS itself cites lots of other works in its Related Works section. Since that 3D Gaussian Splatting is at best one year old at the time of publishing, it is reasonable to expect that most previous work will not work with 3DGS, but focus on the challenge of reconstructing such radiance fields with NeRFs. Still, most works features at this section are mentioned only as a means to illustrate the problem better, and serve as an inspiration for better regularizing training conditions with little data. The section explicitly mentions:

- RegNeRF: Regularizing Neural Radiance Fields for View Synthesis from Sparse Inputs, which samples unobserved views (predicted using a trained normalizing flow model, maximizing predicted log-likelihood) and rendering unseen patches for better regularizing geometry during training. This work also proposes an annealing strategy for controlling ray density near the ray start, which can produce undesirable artifacts [5];
- Depth-supervised NeRF: Fewer Views and Faster Training for Free, or DS-NeRF for short, uses SfM sparse points used in training as a means to supervise depth training for each key point and color training for each pixel on image, producing better reconstructions out of few views [1];
- ViP-NeRF: Visibility Prior for Sparse Input Neural Radiance Fields, which extracts visibility priors from plane sweep volumes to use as dense supervision at no extra pre-training cost, differently from other NeRF-based solutions aiming at reconstructing scenes from sparse views [7];
- FreeNeRF: Improving Few-shot Neural Rendering with Free Frequency Regularization, which focuses efforts into regularizing frequency in NeRF's inputs and penalizing near-camera fields, since that high frequencies are a huge problem while reconstructing NeRF with sparse views [11];
- FlipNeRF: Flipped Reflection Rays for Few-shot Novel View Synthesis, which reflects sampled rays as a cheap way to generate extra views for better constraining the training process. Extra rays are reflected according to the normal vector whenever the angle between the original ray and the normal does not exceed ninety degrees [6];
- SparseNeRF: Distilling Depth Ranking for Few-shot Novel View Synthesis, which exploits depth priors from other sources, such as pre-trained depth models and coarse depth models from customer-level sensors, to serve as supervision for depth training, given constraints to account for inaccurate depth maps [8].

The aforementioned works propose different strategies for better regularization for NeRF optimization. The future 3D Gaussian Splatting works that came afterwards inherit this spirit. Since 3DGS is also interested in reconstructing real-life scenes, contemporary works leverage advantages to reconstruct sparsely sampled scenes better through some way to enhance splat regularization.

The first known 3DGS work to tackle this specific and challenging scenario is SparseGS, first published in late 2023, which uses a custom Loss function that leverages depth samples for supervising splat position together with a custom operator for pruning sparse points [9]. Still it is recent enough to be considered contemporary to other techniques published around the same time frame as CoherentGS, such is the case of FSGS and DNGaussian.

FSGS (or Few-Shot Gaussian Splatting), published on ECCV 2024 as "Real-Time Few-Shot View Synthesis using Gaussian Splatting", runs a method dubbed Gaussian Unpooling that iteratively redistributes Gaussians while using monocular depth priors in order to optimize splat position during training better while filling vacant areas [13]. DNGaussian, on the other hand, published on CVPR 2024 as "DNGaussian: Optimizing Sparse-View 3D Gaussian Radiance Fields", uses a two-step depth regularization process to constrain 3D geometry without compromising color details, plus a two-step depth normalization procedure to enhance reconstruction through normalizing depth patches and refocus small depth changes [3].

Since CoherentGS is very new, published only a couple of months ago, no published works have been developed based on its techniques. However, some works acknowledge CoherentGS as a valid work worthy of citation, albeit in the form of another work that tackles the problem of properly reconstructing a radiance field of a sparsely captured scene. At the time of writing, CoherentGS was cited at least thirteen times, according to Google Scholar. From those citations, at least three of them were published in peer-reviewed portals, such is the case of "GeoRGS: Geometric Regularization for Real-Time Novel View Synthesis from Sparse Inputs" on the IEEE Transactions on Circuits and Systems for Video Technology journal [4], "FewViewGS: Gaussian Splatting with Few View Matching and Multi-stage Training" accepted on NeurIPS 2024 [12], and last but not least the recently published "GaussianObject: High-Quality 3D Object Reconstruction from Four Views with Gaussian Splatting" on ACM Transactions on Graphics Volume 43, Issue 6 from December 2024 [10].

3. Hacker

The code structure is very similar to the 3DGS [2], the main difference is related to the Gaussians representation. Since CoherentGS introduces coherency based on depth, additional scripts are included in the *scene* folder, such as *depth-layering.py* and *decoder.py*. This latter is responsible for scaling the Gaussian shapes, in accordance to the initial depth, vertical focal length and the input image height.

3.1. Code reproducibility

Initially, the code was not published, so we had to contact the authors who provided it to us, with the condition of academic use. In this available version, a code manipulation was required inside the *decoder* file, in the function called *forward*, the necessary change was the insertion of a variable cast. Despite that, the code has easy reproducibility and the README is also simple and clear. Three demo tests were performed to attest to this, with the datasets: *flower*, *room* and *horns*. Each test was run in about 30 minutes and generates as output three videos with the normal map, depth map, and 3D reconstruction, as seen in the Figure 1. The flower demo results highlights the characteristics of the method of not reconstructing unseen spaces, avoiding artifacts, the authors suggest do an inpainting with a diffusion model to cover the blank regions. This inpainting was not tested here.

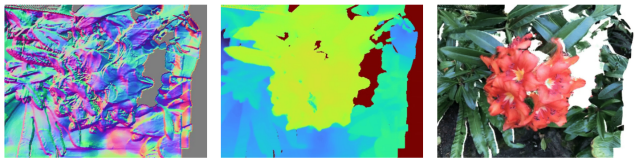

Figure 1. Flower demo (to left to right: normal map, depth map and 3D reconstruction).

3.2. Experiments

Since CoherentGS intends to perform a 3D reconstruction with sparse input images, around 3 or 4, the position and number of cameras is a decisive factor for a good result. Considering that, the first experiment proposed was to reconstruct the flower scene with 4 cameras instead 3 (default). As result, the new scene present less blank spaces, as shown in Figure 2. Nevertheless, the time spent in running step changed from 32 to 47 minutes.

The second experiment was to change the number of interactions from 20 thousand (default) to 10 thousand, both with 4 cameras. Although the 3D reconstruction with fewer

Figure 2. 3D reconstruction with: 3 cameras (on the left) and 4 cameras (on the right).

Figure 3. 3D reconstruction with: 20k interactions (on the left) and 10k interactions (on the right).

Figure 4. 3D reconstruction with: 3 cameras and 20k interactions (on the left) and 4 cameras and 10k interactions (on the right).

interactions has a blurred appearance (Figure 3), the time savings in execution are substantial, from 47 to 12 minutes.

Aiming to obtain an optimized result, a test was carried out changing the number of cameras and interactions. The parameters suggested for optimization were: 4 cameras and 10 thousand interactions. Comparing this with the default (3 cameras and 20 thousand interactions) is possible to note that the proposed set of parameters present a better result both in appearance and in time (from 32 to 15 minutes).

The last experiment was to add a scale factor to calculate the radius of the Gaussians, the factor tested was 4. The Figure 5 highlights the break in coherence generated by this change, resulting in a very blurred image.

In conclusion, these tests showed that the camera is a key parameter to obtain a great result, while interaction can be reduced so that a faster result can be produced.

Figure 5. 3D reconstruction with 4 cameras and 10k interactions and scale factor: 1 (on the left) and 4 (on the right).

4. PhD Student

After analyzing the CoherentGS papers and discussions, we propose four research directions for future work in neural scene reconstruction.

The first direction focuses on Naval and Submarine Engineering Applications. There is significant potential to adapt CoherentGS for underwater imaging scenarios where obtaining multiple views is inherently challenging. This research would develop specialized techniques for creating 3D models of submerged equipment and structures using limited imagery from ROVs and divers. A key challenge would be handling water turbidity, refraction, and scattering effects while maintaining coherent reconstruction. Success in this domain could enable automated monitoring of marine ecosystems and more efficient underwater infrastructure inspection.

The second area explores Historical Monument Reconstruction. This work would adapt CoherentGS for reconstruction from sparse historical photographs that often have varying quality and unknown camera parameters. The research would need to develop techniques to handle incomplete and degraded image data while preserving important architectural details. Methods for combining historical and modern imagery in a coherent reconstruction pipeline would be particularly valuable. This research topic could improve our ability to digitally preserve and restore cultural heritage sites.

The third direction investigates Adaptive Multi-Resolution Gaussian Distribution. This research would develop a dynamic approach to Gaussian placement based on scene complexity, using denser distributions in areas requiring great detail (like complex geometry and fine textures) while implementing sparse distributions in more straightforward regions (such as flat surfaces and uniform textures). The research would create metrics to automatically determine optimal Gaussian density based on local scene properties and design efficient optimization strategies for variable-density Gaussian fields. This approach could

reduce computation time while maintaining or improving quality of the CoherentGS.

The fourth area focuses on Enhanced Transparent Object Handling through specialized regularization. This research would extend CoherentGS to better handle transparent surfaces by developing methods to model both reflected and transmitted light paths. It would create physically based constraints for glass, water, and other transparent materials, enabling the reconstruction of scenes with multiple transparent layers. The work would involve designing loss functions that encourage physically plausible transparent object reconstruction while incorporating principles from light transport theory to handle complex transparent geometries.

These proposals strengthen CoherentGS by enhancing efficiency with adaptive resolution and improving real-world use with better transparency handling. The underwater and historical reconstruction applications leverage the method's ability to work with sparse views while pushing it into important practical domains. All four directions focus on coherent reconstruction while introducing novel technical contributions. They balance practical applications and fundamental improvements to the approach. Advances in these areas would significantly advance state-of-the-art neural scene reconstruction while opening new application domains for the technology.

5. Conclusion

CoherentGS is a method that seeks to reduce the impact of sparse views on gaussian representation of scenes, as done for other scene representations such as NeRFs. It introduces coherence during training, minimizing the overfitting of gaussians. Such work may offer a good option for few views, however, as commented previously, it slows down considerably when dense views are available, making it unappropriated for big scenes. In general, it is an useful method in the scope it is defined, and may serve as inspiration to develop further research in the line of 3D scene representation with sparse views.

References

[1] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 2

[2] 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. 1, 3

[3] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20775–20785, 2024. 3

[4] Zhaoliang Liu, Jinhe Su, Guorong Cai, Yidong Chen, Binghui Zeng, and Zongyue Wang. Georgs: Geometric regularization for real-time novel view synthesis from sparse inputs. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024. 3

[5] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 2

[6] Seunghyeon Seo, Yeonjin Chang, and Nojun Kwak. Flipnerf: Flipped reflection rays for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22883–22893, 2023. 2

[7] Nagabhushan Somraj and Rajiv Soundararajan. Vip-nerf: Visibility prior for sparse input neural radiance fields. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 2

[8] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9065–9076, 2023. 1, 2

[9] Haolin Xiong. Sparsegs: Real-time 360° sparse view synthesis using gaussian splatting. Master's thesis, University of California, Los Angeles, 2024. 2, 3

[10] Chen Yang, Sikuang Li, Jiemin Fang, Ruofan Liang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Gaussianobject: High-quality 3d object reconstruction from four views with gaussian splatting. *ACM Transactions on Graphics (TOG)*, 43(6):1–13, 2024. 3

[11] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8254–8263, 2023. 1, 2

[12] Ruihong Yin, Vladimir Yugay, Yue Li, Sezer Karaoglu, and Theo Gevers. Fewviewgs: Gaussian splatting with few view matching and multi-stage training. *arXiv preprint arXiv:2411.02229*, 2024. 3

[13] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *European Conference on Computer Vision*, pages 145–163. Springer, 2025. 3