

2D Gaussian Splatting for Geometrically Accurate Radiance Fields

Revis@r
Institution1
`firstauthor@i1.org`

Arqueólog@
Institution2
`secondauthor@i2.org`

Hacker
Leonardo Mendonça
IMPA
`leonardo.mendonca@impa.br`

PhD Student
Leonardo Mendonça
IMPA
`leonardo.mendonca@impa.br`

1. Revisão

Sugestão para o Revis@r: ler as [orientações do CVPR](#). Tentem manter o relatório dentro de 8 páginas.

1.1. Resumo

Descreva brevemente o método e sua contribuição para *visão computacional* ou *computação gráfica*. Forneça sua avaliação sobre o escopo/magnitude da contribuição do artigo. Segue uma sugestão de estrutura para o resumo.

- Problema abordado;
- Motivação;
- Resumo do método;
- Lista de contribuições.

Descreva o método e faça uma análise justificando cada fórmula utilizada. A exposição está clara? Como poderia ser melhorada?

1.2. Pontos positivos

- Ideias interessantes validadas através de experimentalmente e de forma teórica, novas ferramentas, resultados impressionantes, ...
- O que alguém da área aprenderia lendo o paper?

1.3. Pontos negativos

- Falta de experimentos (quais?)
- Alegações enganosas e erros
- Difícil de reproduzir (Participação do Hacker)

1.4. Avaliação

Dê uma classificação geral do trabalho e do artigo em uma escala contínua de 1 a 5, onde 1 é o pior e 5 é o melhor. Especificamente: 1 = Rejeitar, 2 = Possivelmente rejeitar, 3 = Duvidoso, 4 = Possivelmente aceitar, 5 = Aceitar.

Deve ficar claro quais dos pontos positivos e negativos foram mais considerados.

2. Arqueólog@

Determinar onde este artigo se encaixa no contexto de trabalhos anteriores e posteriores. Você encontrou esse artigo e deve apresentar a ordem cronológica que o trabalho se encaixa. Sugestão: leia a seção de trabalhos relacionados.

Encontrar e relatar sobre um artigo mais antigo citado pelo artigo atual e um artigo mais recente que cita o artigo atual. Claro, explicar como eles se relacionam.

Além disso avalie se as referências estão adequadas? Liste referências que estão faltando.

Note que quase todos os trabalhos da nossa lista cita o 3D Gaussian Splatting (3DGS) [4] e EWA volume splatting [7]. Logo, o Arqueólog@ precisa ir além dessas referências.

3. Code and experiments

We have executed the 2DGS code, made available by the authors in <https://github.com/hbb1/2d-gaussian-splatting>. Despite the installation in-

structions present in the *README.md*, there were some difficulties:

1. The *environment.yml* file was not complete: we had to add the python modules *setuptools* and *matplotlib* for the installation and code to work without errors.
2. The 2DGS documentation fails to mention CUDA, whose installation is necessary for training and inference to work properly. We also went through a process of trial-and-error in order to install the right version of CUDA and the correct C++ kernels.

In spite of the problems with the installation, the execution of the code is made relatively straightforward by the command line interface implemented by the authors. Most of the hyperparameters can be set through command line arguments when calling *train.py*, as well as input and output names, image resolution and so on.

3.1. Code analysis

The most performance-heavy parts of the training process, notably the gaussian rendering, are done inside a CUDA-optimized C++ script, which employs the user's GPU for fast computing. The python files, however, are responsible for training, pruning and often calling the CUDA C++ files required to do so.

There were some differences between the methodology presented in the paper and the actual code implementation.

The densification strategy, mentioned only in passing in the text, is taken directly from 3DGS [4], and uses mostly the same hyperparameters. Between the training iterations *densify_from_iter* (default 500) and *densify_until_iter* (15000), the model will periodically split or clone certain gaussians, depending on their scale. After *opacity_reset_interval* (3000) epochs, the gaussians with opacity lower than *opacity_cull* (0,05) are removed, while the remaining ones have opacity reset to 0,01 (hardcoded). These hyperparameters are given without justification in [4] and not mentioned at all in [3].

The number of spherical harmonics for anisotropic coloring in each gaussian *sh_degree* (set to 3) is another "hidden" hyperparameter. The number of iterations before applying depth and normal regularization, respectively 3000 and 7000, are not only unmentioned in the text, but also hardcoded, which hinders future experiments to study the effect of these parameters.

3.2. Experiments

3.2.1 Depth distortion regularization

This regularization loss term seeks to minimize the distance between the depth of different gaussians intersected by the same ray, therefore grouping the gaussians near the physical surface of the object. The paper [3] suggests using the weight parameter $\alpha = 1000$ for bounded and $\alpha = 100$ for unbounded scenes. However, the code sets the default value of this parameter to zero, and, in the authors' own experiments, the value of α used changes from dataset to dataset. In the MipNeRF360 [1] dataset, this regularization is not used at all, and in Tanks and Temples [5], α is set to either 10 or 100, depending on the scene, and regardless of whether it is bounded or unbounded.

As such, in order to investigate the importance of this weight parameter, we have trained the model on the Bonsai scene of the MipNeRF360 dataset with resolution 390×260 , setting alternatively $\alpha = 0$ (as used in the authors' tests) and $\alpha = 1000$ (textually recommended in the paper for bounded scenes such as this one). Results are shown in fig. 1. One may observe in these tests that there are artifacts near the upper edge of frame 0, which is likely not covered by the training views, and a significant distortion in the glass windows of view 12, which illustrates 2DGS's inadequacy for modeling semi-transparent materials. There is no apparent difference, however, between the rendered images using $\alpha = 0$ and $\alpha = 1000$. This suggests that perhaps the depth distortion regularization does not have as strong an impact as suggested by the authors.

3.2.2 Training and rendering resolution

After training in 390×260 resolution, we rendered the optimized scene in a higher resolution of 1559×1039 . The goal of this experiment was to investigate the method's performance for novel-view synthesis in higher resolutions than the one it was trained on. Since the results were very similar for the reconstructions with and without the depth distortion, we use here $\alpha = 0$, the same value used by the authors in their evaluation of the MipNeRF360 dataset. Results are shown in fig. 2.

We conclude that reconstruction quality is directly connected to the rendering resolution: in the high-resolution render, the image is characterized by very thin gaussians, inconsistent with the scene geometry, which are not visible in low resolution and were therefore not penalized during training.

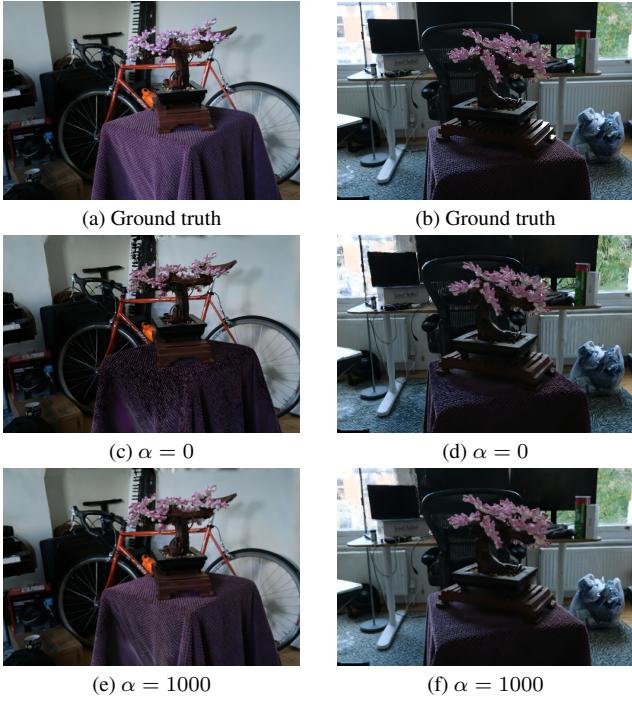


Figure 1. MipNeRF360 Bonsai scene, frame 0 (left) and 12 (right), after 30000 training iterations, rendered after training with different values of depth distortion weight α .

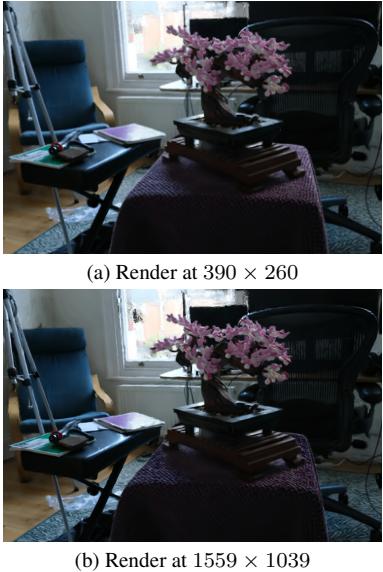


Figure 2. Comparison between images rendered with different resolutions, after being trained on 390×260 resolution. MipNeRF360 Bonsai scene, frame 15

4. Projeto de doutorado

4.1. 2D Half-Gaussian Splatting

The paper showed very positive results for surface reconstruction and novel-view synthesis using planar gaussians, which can be used to accurately model a great variety of smooth surfaces.

However, there are many human-made objects and structures that possess sharp edges, or that are better represented by manifolds with boundary. We propose therefore a modification of the method by replacing the 2D gaussians used in the paper with half-gaussians, defined by the distribution:

$$G(u, v) = \begin{cases} 2e^{-\frac{u^2+v^2}{2}}, & \text{if } v \geq 0 \\ 0, & \text{if } v < 0 \end{cases} \quad (1)$$

This modified shape allows one to economically represent surfaces with a common edge or surfaces whose boundary is a 1D manifold. An expected difficulty, however, is the fact that eq. 1 has a discontinuity at $v = 0$. In order for the function to be differentiable, we will need to define a smoothing function between the two half-planes $v < 0$ and $v > 0$, in order to allow for backpropagation. The "sharpness" of this smoothing function can be either set as a hyperparameter or as a trainable parameter for each half-gaussian, but systematic ablation studies are necessary to decide which strategy is more performant.

4.2. Mixed 2-3D Gaussian Splatting

Despite the good results for opaque surfaces, 2DGS reports poor reconstruction quality for translucent materials, which we can also observe in fig. 1. However, these materials can be accurately represented by volumetric (3D) gaussians, as seen in 3DGS [4] and EWA Splatting [7]. Therefore, we propose a mixed approach, where 2D and 3D gaussians coexist and can be jointly optimized to accurately capture both opaque surfaces and translucent volumes.

A natural objection is the matter of initialization: each point in the initial point cloud, produced by COLMAP, should initialize to either a 2D or 3D gaussian, but it is no evident that there is a way to determine beforehand what dimension each gaussian should have. One possible approach is to initialize every gaussian as 3D, and convert each of them to 2D if the ratio between its 3 dimensional scales reaches a certain threshold (i.e. if the gaussian becomes "flat enough").

We may also cite the drawback that, once we mix the 2D

and 3D gaussians, we lose 2DGS’s ability to obtain surface normals directly, which was useful for normal regularization and for mesh extraction. On the other hand, one can use of the approaches defined in GOF [6] or SuGaR [2], both of which obtained good results in characterizing the normal of a 3D gaussian.

Once these obstacles have been overcome, we should have a robust representation for novel-view synthesis in scenes that mix opaque and semi-transparent objects.

5. Conclusões

Apresente as conclusões, sugestões de título e um resultado ausente que o artigo poderia ter incluído.

References

- [1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022. [2](#)
- [2] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5354–5363, 2024. [4](#)
- [3] 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. [2](#)
- [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. [1](#), [2](#), [3](#)
- [5] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. [2](#)
- [6] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 2024. [4](#)
- [7] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa volume splatting. In *Proceedings Visualization, 2001. VIS’01.*, pages 29–538. IEEE, 2001. [1](#), [3](#)