Mip-Splatting

Revisor
Leonardo Nanci
leonardonanci@id.uff.br

Arqueólogo
Fabricio Assunção
fabricio.asfora.001@gmail.com

Hacker
Fernanda Viviani
fernandaviviani.professional@gmail.com

Doutorando
Davi Guimarães
davi.guima.castro@outlook.com

1. Review

The volumetric scene representation using 3D Gaussian Splatting (3DGS) presented by Kerbl et. al in 2023 demonstrated impressive novel view synthesis (NVS) results, achieving high fidelity and rendering eficiency. Following this work, different applications, extensions, and improvements have been developed. Based on signal processing theory, Mip-Splatting proposes the use of convolutions to reduce aliasing. First, by limiting the maximum frequency of the Gaussians in 3D space; then by using a 2D filter to reduce mip-like alias.

1.1. Abstract

In the context of 3D Gaussian Splatting (3DGS), Mip-Splatting addresses the aliasing problem encountered when applying solutions based on 3DGS. Specifically, the paper tackles artifacts that arise when testing a scene at sampling frequencies different from those used during the training phase. These effects become particularly evident when performing zoom-in or zoom-out operations, either by changing the focal length or by altering the camera position in the scene.

The proposed solution is rooted in sampling theory, primarily the Nyquist-Shannon sampling theorem, which states that the sampling frequency must be at least twice the maximum frequency of the continuous signal to be accurately represented. The authors offer solutions for two types of problem: zoom-in and zoom-out.

They observed that performing a zoom-in, i.e., increasing the sampling frequency, results in the appearance of several high-frequency Gaussians. To prevent the creation of such high-frequency components, the authors propose limiting the maximum frequency of the Gaussians created based on the frequency and distance of the training images. The camera configuration that enables the best representation of each Gaussian, i.e. that best samples the primitive, determines this limit. By applying a low-pass Gaussian convolution filter to each object in the 3D space, the high frequencies are removed according to the requirements of each Gaussian. Since the convolution of two Gaussians results in a new Gaussian, the only effect of this filtering is the reduction of frequency. In addition, such operations create negligible computational overhead.

On the other hand, zoom-out introduces artifacts mainly due to the dilation operation, which was not discussed in the original 3DGS paper. This procedure alters the distribution of Gaussians in the screen space, making them appear larger in the case of zoom-out and smaller when zooming in. The authors propose replacing this manipulation with the application of a 2D convolution that approximates the light integration in physical image capture. To achieve this, they used a low-pass Gaussian filter, with the scale of the filter set to cover the area of a single pixel in screen space.

Experiments conducted under various conditions demonstrate the effectiveness of the proposed changes compared to the original 3DGS implementation and other antialiasing techniques, as well as rendering methods based on NeRFs. During the tests, different evaluation strategies were used, with training conducted at one or multiple resolutions (frequencies), depending on the type of artifact under analysis. According to the results, the authors' approach outperforms the other techniques in most cases. Ablation studies in the supplementary material imply that the original 3DGS would not work without the (flawed) dilation operation.

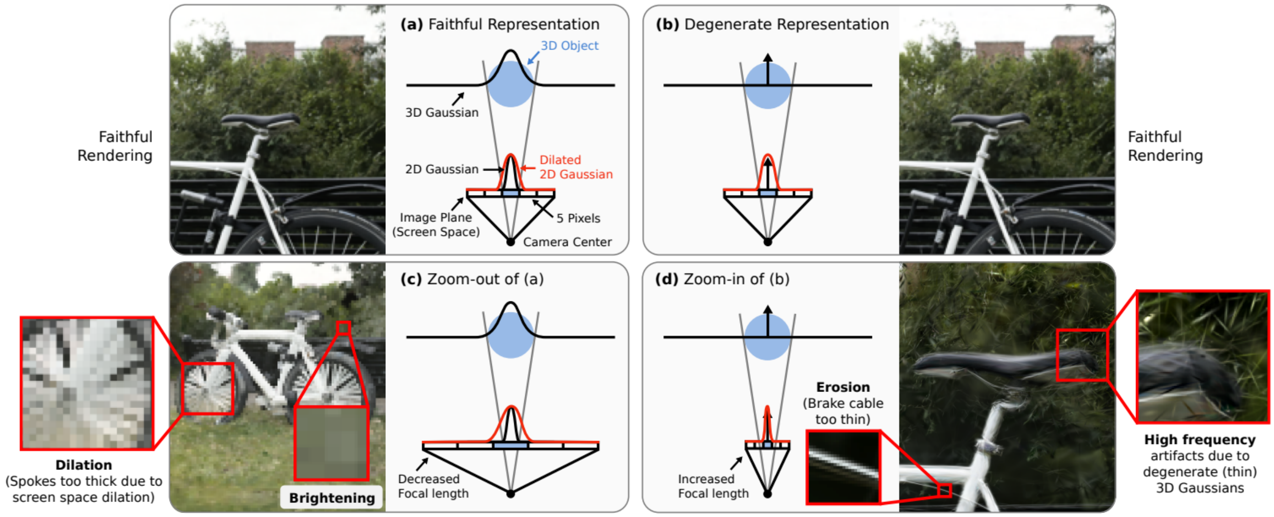

Figure 1. 3D Gaussian Splatting renders images by representing 3D Objects as 3D Gaussians which are projected onto the image plane followed by 2D Dilation in screen space as shown in (a). The method's intrinsic shrinkage bias leads to degenerate 3D Gaussians exceed sampling limit as illustrated by the δ function in (b) while rendering similarly to 2D due to the dilation operation. However, when changing the sampling rate (via the focal length or camera distance), we observe strong dilation effects (c) and high frequency artifacts (d).

1.2. Positive Aspects

- Proposals are grounded in signal processing theory; specifically, the Nyquist-Shannon theorem.
- Simple modifications that lead to significant improvements in results compared to the original method, with negligible processing overhead.
- Identification of the dilation operation, not mentioned in the original 3DGS paper, which introduces artifacts.
- Extensive experimentation covering various aspects of the original method and the proposed changes.

1.3. Negative Aspects

- Lack of explanation for the parameterization of the 3D filter.
- Little description of multi-scale training technique.
- Typos in the abstract and supplementary material.
- Introduction of focal length as an input parameter.

1.4. Evaluation

The authors propose a well-founded improvement to the original method that does not introduce significant rendering overhead. Their work is validated using standard benchmark datasets and show state-of-the-art results.

Overall rating: accept

2. Archeologist

2.1. EWA Splatting: The Foundations of Prefiltering

EWA Splatting method was one of the precursors to most modern volumetric rendering techniques, including Mip-Splatting. EWA Splatting used pre-filtering with Elliptical Gaussian Kernels to create smooth and continuous projections. It is very useful in the case of rendering of clouds and medical data, which do not have well-defined boundaries. The innovation of low-pass filtering before sampling greatly reduced aliasing by eliminating high-frequency artifacts. However, one big limitation with EWA Splatting was its smoothing, which was limited to just 2D. Indeed, this process eventually drives inaccuracies into the associated depth control mechanism and subsequently generates imprecise overlaps, or occlusions. On its part, the MIP-Splatting novel rendering technique transitions the associated operation into 3D in pursuit of increased precision as well as adaptability during depth changes.

2.2. Mip-NeRF: Bridging Neural Rendering and Mip-Splatting

Mip-NeRF introduced the concept of prefiltering into the domain of NeRFs and further influenced the multiscale design of Mip-Splatting. While traditional NeRF relies on one-dimensional Ray Marching, Mip-NeRF pioneered cone tracing to treat rays as cones that expand through 3D space. This increased the capability for handling high-frequency regions by adaptively smoothing textures and edges. Besides, the representation of points as continuous Gaussian distributions across multiple scales using the method known as Integrated Positional Encoding (IPE) influenced the direct multiscale smoothing approach of Mip-Splatting. Thus, adaptive sampling and representation across scales are the basic concepts that shaped Mip-Splatting in its development for dynamically balancing the preservation of fine details with efficient rendering across various zoom levels.

2.3. Mipmap-GS: Addressing Mip-Splatting's Early Limitations

Mipmap-GS resolved the most important key challenges in Mip-Splatting's initial implementations: fixed-size Gaussian splats. With the deformable splats, it could let the splats adapt dynamically according to zoom and resolution from the viewer in order not to lose detail after zoom-ins and avoid aliasing during zoom-outs. An adaptive mipmapping system integrated further improved efficiency and quality of Mip-Splatting where precomputed Gaussians at multiple scales have their choice based on distance from the camera. Moreover, the technique by Mipmap-GS enables it inherently to generate scale-independent pseudo-ground truths in real time without human edits. These works thus refined Mip-Splatting to offer more flexibility with increased precision and artifact-free visuals.

2.4. Analytic Splatting: Pushing the Boundaries of Accuracy

Analytic Splatting introduced a new dimension in accuracy to volumetric rendering, replacing discrete sampling with exact analytical computation. This is accomplished by computing the complete Gaussian integral for every pixel without approximating the contribution of each splat by sampled points. Since this technique does not have the possible sources of sampling error, it manages to preserve a level of detail without artifacts, especially accentuated on highly complex textures or small objects. Very computationally expensive, although a set of proposed optimizations made this technique viable for interactive rendering. These advances overcame most of the important limitations of early Mip-Splatting methods: they provided high-fidelity results, while still enabling real-time performance. Thereby, Analytic Splatting became one of the milestones for the extension of capabilities of Mip-Splatting toward sharper and more accurate renderings.

3. Code and experiments

- Evaluate the reproducibility of the method

An example of improvement would be to optimize the installation of packages with pip. When we implemented the code we had problems with !pip install -r requirements.txt The list of libraries contained there could already be present at the beginning of the code for automatic installation when starting to run it.

Another problem was with the management of files and folders, one of the main reasons is that the two datasets together were more than 12GB, so we had to choose just one image to generate the 3D visualization, since if we used both datasets it would consume almost the entire memory of our Google Drive. The organization of the dataset folder was very important because we had initial difficulties in the code with an error in the directory path. We suggest adding checks to ensure the dataset directory is configured correctly before extracting, removing data, or running the test environment.

The results obtained by viewing the .PLY file with the image generated within the viewer provided by the authors were not very expressive. The generated image is not clear and full of aliasing artifacts, presenting blurs and shadows. In the report's attachment we include two images to clarify the result, one generated by the authors of the paper and the other generated by the hacker.

Think about other experiments

To add something new to the provided code, we can implement a 3D visualization of the processed scenes, such as generating a .PLY (Polygonal Mesh) file and viewing it in a library such as Open 3D. If the code is running locally, Open 3D provides an interactive window to explore the data. While in Google Colab it is possible to generate the 3D visualization image itself for analysis. In both cases, it would not be necessary to use the viewer provided by the authors of the paper.

Could the work be reproduced by one or more graduate students?

It is clear that the work cannot be reproduced by one or more graduate students. Firstly because the code and its datasets are very heavy, requiring all of them on a GPU with more than 20GB and having plenty of storage space locally or in the cloud because the two datasets together are more than 12GB. Furthermore, the code has many errors that need to be corrected and its test environment takes more than 3 hours to run and generate the .PLY file with the image that will be viewed in 3D.

Compare the formulas implemented in the code with the equations in the paper

In general, the important details of algorithms and systems are discussed adequately. Below we list two formulas implemented in the code and their related equations in the paper with the aim of showing how the mathematics behind the final result obtained works.

- 3D Gaussian Splatting: The formula below demonstrates how the geometry of each scaled 3D Gaussian Gk is parametrized by an opacity (scale) $\alpha k \in [0, 1]$, center pk $\in R^{3x1}$ and covariance matrix $\sum \kappa \in R^{3x3}$

$$\mathcal{G}_k(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mathbf{p}_k)^T \mathbf{\Sigma}_k^{-1}(\mathbf{x}-\mathbf{p}_k)}$$

- 3D Smoothing: By employing 3D Gaussian smoothing, they ensure that the highest frequency component of any Gaussian does not exceed half of its maximal sampling rate for at least one camera. Given the maximal sampling rate $\hat{\upsilon}$k for a primitive, they aim to constrain the maximal frequency of the 3D representation. This is achieved by applying a Gaussian low-pass filter $\mathcal{G}_{\text{low}}$ to each 3D Gaussian primitive $\mathcal{G}\kappa$ before projecting it onto screen space.

$$\mathcal{G}_k(\mathbf{x})_{\text{reg}} = (\mathcal{G}_k \otimes \mathcal{G}_{\text{low}})(\mathbf{x})$$

This operation is efficient as convolving two Gaussians with covariance matrices $\sum 1$ and $\sum 2$ results in another Gaussian with variance $\sum 1 + \sum 2$. Hence,

$$\mathcal{G}_k(\mathbf{x})_{\text{reg}} = \sqrt{\frac{|\mathbf{\Sigma}_k|}{|\mathbf{\Sigma}_k + \frac{s}{\hat{\nu}_k} \cdot \mathbf{I}|}} \, e^{-\frac{1}{2}(\mathbf{x}-\mathbf{p}_k)^T (\mathbf{\Sigma}_k + \frac{s}{\hat{\nu}_k} \cdot \mathbf{I})^{-1} (\mathbf{x}-\mathbf{p}_k)}$$

4. Doctorate Project

The Mip-Splatting method uses a 2D Mip filter, inspired by mipmap, to reduce dilation and erosion in images when zoomed out and zoomed in, respectively.

The proposed PhD student project would be about trying to apply the idea behind this pyramidal structure to 3D Gaussians. In other words, instead of constructing the scene's Gaussians once and using that construction to render the pixels, we construct the scene's Gaussians several times, each representing a different image quality.

The reason for doing this would be that when rendering the image, we would render the parts closest to the camera with the scene with high quality (which would have more Gaussians) and the parts furthest from the camera with the scene with less quality (which would have fewer Gaussians). In this way, we could generate images of a scene with higher quality, without using a large number of Gaussians on the entire scene at once.

5. Conclusions

The 3D Gaussian Splatting method, while impressive in novel view synthesis, suffers from artifacts when changing the sampling rate. This is due to a lack of 3D frequency constraints and the use of a 2D dilation filter.

To address this, the paper introduces a 3D smoothing filter that constrains the size of 3D Gaussian primitives based on the maximum sampling frequency, eliminating high-frequency artifacts during zooming. Additionally, the authors replace 2D dilation with a 2D Mip filter, which simulates a 2D box filter, mitigating aliasing and dilation issues.

Their evaluation, including training on single-scale images and testing on multiple scales, confirms the effectiveness of our approach.

6. Attachments (Hacker)

Figure 2. Our .PLY generated image viewer

Figure 3. 3D Image generated by the authors