Fast and Robust 3D Gaussian Splatting for Virtual Reality

Xuechang Tu
Carnegie Mellon & Peking University
USA / China

Bernhard Kerbl
Carnegie Mellon University
USA

Fernando de la Torre
Carnegie Mellon University
USA

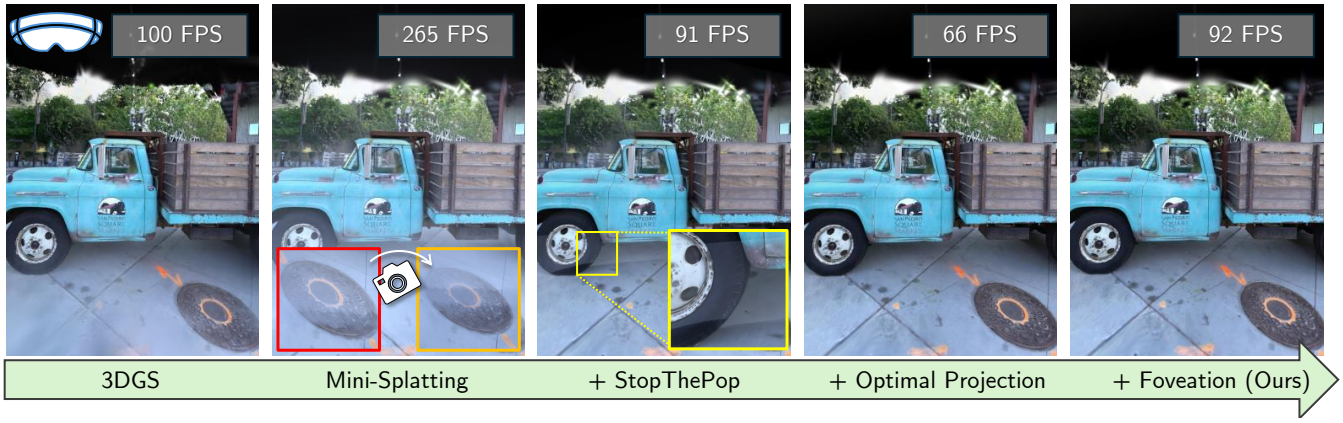

Figure 1: Modifications of our VR rendering pipeline to 3DGS. By reducing model size, Mini-Splatting speeds up rendering (top labels), but keeps 3DGS' popping artifacts (insets). StopThePop resolves popping, but has ghosting of large splats (closeup). Optimal Projection avoids this, but hurts performance. Adding foveated rendering, we get a robust, fast 3DGS pipeline for VR.

ACM Reference Format:
Xuechang Tu, Bernhard Kerbl, and Fernando de la Torre. 2024. Fast and Robust 3D Gaussian Splatting for Virtual Reality. In *SIGGRAPH Asia 2024 Posters (SA Posters '24), December 03-06, 2024.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3681756.3697947

1 INTRODUCTION

3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] has quickly risen to become a highly influential method for novel-view synthesis. The high rendering performance of tile-based rasterization opens the door for many real-time applications, enabling realistic scene rendering on weaker and mobile devices. However, the original 3DGS solution faces several challenges for virtual reality (VR) applications: the 3DGS codebase readily integrates with extended reality (XR) libraries (OpenXR), but suffers from temporal artifacts (popping). While tolerable for desktop viewing, the high resolution and immersion of VR applications for head-mounted displays (HMDs) amplify the visual artifacts, impacting the user experience.

In this work, we propose to synergize a variety of recent extensions to 3DGS and improve the VR experience. These extensions have been developed and considered in isolation; in this work, we show how they can be meaningfully combined to yield a high-quality, high-performance VR pipeline. Fig. 1 outlines our approach: we first use Mini-Splatting [Fang and Wang 2024] to raise rendering frame rates. Next, we employ StopThePop (STP) [Radl et al. 2024], to address temporal popping artifacts. The off-axis projection and field-of-view found in HMDs give rise to unsightly ghosting; we address it via Optimal Projection (OP) for 3DGS [Huang et al. 2024]. Finally, to ensure the recommended framerate of >72 FPS at HMD-native resolution, we use a simple foveated rendering mechanism.

SA Posters '24, December 03-06, 2024, Tokyo, Japan

ACM ISBN 979-8-4007-1138-1/24/12.
https://doi.org/10.1145/3681756.3697947

2 METHOD

In the following, we discuss the individual steps to arrive at our robust and fast pipeline: model size reduction, popping mitigation, projection error minimization, and foveation.

2.1 Reducing Model Size

Mini-Splatting [Fang and Wang 2024] is a cutting-edge method that first oversamples the scene and places Gaussians at likely surface locations using depth cues. This results in a detailed scene reconstruction, which can then be made sparse by removing Gaussians with low contribution. Mini-Splatting achieves up to 10× model size reduction, much higher frame rates and equal or superior quality metrics compared to 3DGS. However, modelling the same scene with fewer Gaussians leads to *larger* Gaussians on average, increasing the prevalence of noticeable popping artifacts [Radl et al. 2024].

2.2 Eliminating Popping Artifacts

To mitigate visible popping artifacts, we employ the STP [Radl et al. 2024] culling and blending solution. Their hierarchical depth computation eliminates virtually all temporal inconsistencies. However, their image formation is slightly different from Mini-Splatting: directly rendering a Mini-Splatting model with STP yields different results. Hence, we opt for fine-tuning the output of Mini-Splatting using STP training for 5k iterations.

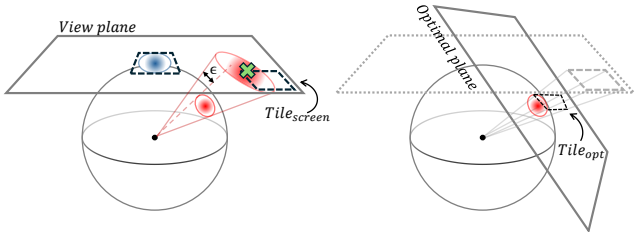

Figure 2: Consolidating STP [Radl et al. 2024] and OP. [Huang et al. 2024]. If a splat overlaps with a screen-space tile and its mean is outside (red splat), STP finds the highest density on the tile border (green cross) for culling. The affine approximation leads to a projection error ϵ towards the boundaries of the view frustum. OP finds a suitable projection plane to minimize this error. To enable STP, we project the tiles touched by each Gaussian onto its optimal plane, yielding non-rectangular quadrilaterals on whose border we compute the Gaussian's location of maximum density.

2.3 Minimizing Projection Error

The hierarchical depth computation of STP can fail to resolve large-scale Gaussian splats with extreme orientations. Such splats occur as a result of projection inaccuracy due to 3DGS using an affine approximation for projecting onto the view plane. This error has been analyzed in [Huang et al. 2024], along with a proposed solution: for each Gaussian, an optimal projection is chosen to compute its splat shape and rasterize it to the image. Unfortunately, this solution is not directly compatible with STP: to combine the benefits of both methods, we must adapt STP's sophisticated culling for custom projection planes. This is outlined in Fig. 2, which shows that projection error is low for splats near the center (blue) and high off-center (red). STP's splat culling requires finding their point of highest density on overlapping screen-space tile boundaries. For a splat on its optimal plane with mean $\mu \in \mathbb{R}^2$, tiles form general quadrilaterals. For a tile edge that can be "seen" at μ, denote it as $\mathbf{p} + t\mathbf{d}$. Taking the derivative of Gaussian density with respect to t and letting it equal zero, we find the point of highest density as $\mathbf{p} + ((\mathbf{d}^\top \Sigma^{-1}(\mu - \mathbf{p}))/(\mathbf{d}^\top \Sigma^{-1}\mathbf{d}))\mathbf{d}$, where Σ is the 2D covariance matrix. Note that at most two tile edges are visible at μ.

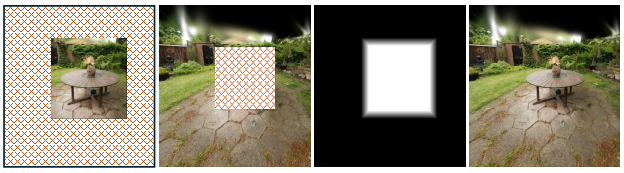

Figure 3: Simple foveated rendering pipeline. Peripheral areas can be sampled at lower resolution without hurting user experience. The overlapping regions of the two images are blended using the weights in the third image.

2.4 Foveated Rendering

Both STP and OP introduce a noticeable rendering overhead that cannot be mitigated by Mini-Splatting alone. To reach the target frame rate of 72+ FPS for the Meta Quest 3 on a single commodity GPU, we propose simple foveated rendering. We use a two-pass solution that renders the image center at full resolution, and the periphery at half resolution. To maximize culling efficiency, we use a tight projection frustum for the high-res center region. We avoid noticeable discontinuities via a transition band where we gradually adjust blending weights for high-res and low-res pixel values. Upscaling and compositing is done using NVIDIA Performance Primitives (NPP) and a custom CUDA kernel (Fig. 3).

3 RESULTS

We evaluated achieved frame rates and qualitative experiments on a Meta Quest 3 at native resolution (2064×2208 pixel) tethered to a desktop machine with an NVIDIA RTX 4090. We use the original SIBR framework with OpenXR support and our custom rasterizer. We conducted performance and user experience experiments on the scenes from the MipNeRF360 and Tanks&Temples datasets.

3.1 Performance

Our solution achieves the target frame rate of >72 FPS (i.e., <14 ms stereo) in the scenes from MipNeRF360 and Tanks&Temples. Table 1 shows how performance changes as we introduce the suggested features. High geometry load may require geometric level-of-detail solutions to ensure their processing in the target time frame.

Table 1: Stereo rendering times (ms) at Meta Quest 3 native resolution, averaging 250 frames per scene and added feature.

	3DGS	Mini-Splatting	+STP	+OP	+Fov. Rend.
Truck	9.8	3.7	15.1	15.1	10.8
Garden	14.4	4.0	13.6	13.6	11.5
Room	10.6	4.7	19.9	19.9	13.0

3.2 User Survey

We conducted a small user survey on 3 individuals familiar with visual computing and asked them to rate the VR experience of the 5 variants w.r.t. quality and responsiveness from 1–10. All participants ranked our quality highest, equal to a pipeline without foveation (avg. 10.0), but better for responsiveness (8.8 vs 8.1). Only Mini-Splatting had a better responsiveness rating (10.0), but much lower quality (4.2), suggesting that our method provides an ideal tradeoff.

REFERENCES

Guangchi Fang and Bing Wang. 2024. Mini-Splatting: Representing Scenes with a Constrained Number of Gaussians. In *Proceedings of the European Conference on Computer Vision (ECCV)*. arXiv:2403.14166 [cs.CV]

Letian Huang, Jiayang Bai, Jie Guo, Yuanqi Li, and Yanwen Guo. 2024. On the Error Analysis of 3D Gaussian Splatting and an Optimal Projection Strategy. arXiv:2402.00752 [cs.CV] https://arxiv.org/abs/2402.00752

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (July 2023).

Lukas Radl, Michael Steiner, Mathias Parger, Alexander Weinrauch, Bernhard Kerbl, and Markus Steinberger. 2024. StopThePop: Sorted Gaussian Splatting for View-Consistent Real-time Rendering. *ACM Transactions on Graphics* 4, 43, Article 64 (2024).