Color-Aware Early Termination for Efficient 3D Gaussian Splatting in Real-Time AR/VR Execution

Handong Ji
Courant Institute
New York University
New York, USA

Haiyu Wang
Tandon School of Engineering
New York University
New York, USA

Sai Qian Zhang
Tandon School of Engineering
New York University
New York, USA

Abstract

Virtual Reality (VR) and Augmented Reality (AR) systems require high-fidelity, real-time rendering to maintain user immersion and presence. While 3D Gaussian Splatting (3DGS) has emerged as a leading technique for photorealistic synthesis, it imposes significant computational demands, particularly when rendering at the high resolutions and frame rates required by head-mounted displays (HMDs). Standard 3DGS implementations often expend substantial computation on primitives with negligible visual impact, due to their reliance on a fixed global transmittance threshold that does not reflect the non-uniform color sensitivity of the human visual system.

In this work, we introduce a perceptually guided early-termination mechanism that accelerates 3DGS execution by exploiting the limits of human visual perception. Rather than relying on fixed heuristics, we derive an adaptive, color-dependent scalar based on just-noticeable-difference (JND) models. To avoid additional runtime overhead, the complex geometric containment tests are precomputed and stored in a lightweight lookup table (LUT), enabling early termination of compositing through a constant-time comparison. Experimental results show that our method substantially improves rendering efficiency while maintaining high visual fidelity, making it well suited for resource-constrained VR/AR systems.

Keywords: 3D Gaussian Splatting, Perceptual Model, Color Science

1 Introduction

The rapid proliferation of Virtual Reality (VR) and Augmented Reality (AR) has fundamentally transformed how users interact with digital content, spanning applications from immersive gaming to professional training simulations. The success of these systems depends critically on the rendering engine's ability to generate photorealistic imagery at stereoscopic resolutions and high refresh rates, which are essential for maintaining user presence and preventing motion sickness.

Recently, 3D Gaussian Splatting (3DGS) [8] has emerged as a dominant rendering technique in this space, delivering higher visual fidelity and substantially faster training compared to implicit Neural Radiance Fields (NeRFs) [13]. Despite these advantages, 3DGS places considerable computational pressure on the GPUs available in head-mounted displays. In particular, the primary bottleneck arises during alpha compositing, where millions of depth-sorted Gaussian primitives must be processed for each frame.

To simplify the computation of 3DGS, prior optimization techniques have focused on exploiting characteristics of human perception, including leveraging gaze behavior to adaptively reduce rendering resolution (e.g., foveated rendering [4, 6, 17, 18]), as well as culling or sparsifying Gaussian primitives based on view frustum, opacity, or contribution thresholds.

In this work, we explore a novel dimension for accelerating 3DGS rendering by explicitly leveraging fundamental properties of human color perception. Human vision exhibits context-dependent color insensitivity, commonly characterized by Weber's Law [15] and formalized through Just-Noticeable-Difference (JND) models. The perceptibility of a new color contribution depends strongly on the background's luminance and chromaticity: a faint residual color may be discernible against a dark, uniform background, yet become imperceptible in brighter or more textured regions. By relying on rigid, global mathematical thresholds, conventional 3DGS implementations inevitably expend computation on contributions that fall below the limits of human perception, resulting in redundant rendering work. Motivated by this observation, we propose to simplify 3DGS execution by aligning its early-termination criterion with principles from color science. Our specific contributions are as follows:

- We derive a dynamic pruning threshold grounded in the Human Visual System (HVS). Unlike conventional fixed constants, our threshold adapts to the local luminance and chromaticity of the accumulated pixel color, ensuring that only perceptually indistinguishable contributions are discarded.
- We show that the complex color-space transformations required for JND analysis can be precomputed and encoded in a lightweight lookup table (LUT), enabling perceptual constraints to be enforced through an $O(1)$ check within the critical ray-marching loop.
- We validate our approach on standard benchmark scenes, demonstrating that it effectively prunes redundant Gaussian primitives. This leads to measurable improvements in frame rate while preserving high perceptual visual quality.

2 Related Work

2.1 3D Gaussian Splatting and Pruning

3D Gaussian Splatting (3DGS) [8] employs a point-based alpha-compositing formulation to synthesize high-fidelity images. For a given pixel, the final color C is obtained through

front-to-back volume rendering over N depth-sorted Gaussian primitives along the camera ray:

$$C = \sum_{k=1}^{N} c_k \alpha_k T_k, \quad \text{where } T_k = \prod_{j=1}^{k-1}(1-\alpha_j). \tag{1}$$

where c_k and α_k denote the color and opacity of the k-th Gaussian, respectively, and T_k represents the accumulated transmittance. While this formulation yields high visual fidelity, evaluating it over millions of Gaussian primitives per frame introduces substantial computational overhead. As a result, mitigating this cost has become a key challenge.

A widely adopted optimization strategy is pruning, which seeks to permanently remove redundant Gaussian primitives from the scene representation. Methods such as LightGaussian [5] and EAGLES [7] introduce significance metrics based on opacity, scale, or geometric contribution to identify and remove low-impact Gaussian primitives during training or post-processing. Similarly, Mini-Splatting [10] and Scaffold-GS [12] employ anchor-based representations or quantization techniques to compress the static model size. Crucially, these approaches operate on the *static* scene representation, reducing the global number of primitives independently of viewing context or human perception.

In contrast, our work targets the *dynamic* rendering phase by analyzing the visibility of Gaussian primitives during the ray-marching process. Rather than permanently retaining or discarding primitives, our method evaluates their contributions on the fly based on the accumulated pixel color. Conceptually, our approach is orthogonal to prior pruning techniques, as it removes redundancy within the rasterization pipeline itself rather than altering the underlying scene representation.

2.2 Color Science in Rendering

Standard rendering pipelines typically operate in linear RGB color space. However, RGB is not perceptually uniform: Euclidean distances in RGB do not correspond to perceptual differences as perceived by the Human Visual System (HVS). To more accurately model human vision, color science relies on physiologically motivated color spaces. The LMS color space captures the responses of the Long, Medium, and Short wavelength cones in the retina. LMS can be further transformed into the DKL color space [3], which is grounded in opponent-process theory. DKL decomposes color into three cardinal axes: luminance ($L + M$), red–green ($L - M$), and blue–yellow ($S - (L + M)$).

A key advantage of DKL for rendering applications is the axis-alignment assumption [16]. Prior work suggests that detection mechanisms along these three axes are largely independent, allowing Just-Noticeable-Difference (JND) thresholds to be modeled as axis-aligned ellipsoids rather than complex, coupled perceptual regions. Furthermore, visual sensitivity varies significantly across the visual field. As shown by Ujjainkar et al. [16], color discrimination thresholds increase with retinal eccentricity, causing JND ellipsoids to expand as stimuli move from the fovea toward the periphery. Building on these physiological insights, our work formulates a perceptually grounded termination criterion that dynamically adapts to color discrimination limits.

2.3 Foveated Rendering in AR/VR

Foveated Rendering (FR) exploits the reduced visual acuity of peripheral vision to lower rendering workloads in VR/AR headsets. Recent advances have begun integrating FR techniques directly into neural rendering pipelines.

Prior work such as MetaSapiens [11] demonstrates real-time neural rendering on edge devices by combining efficiency-aware pruning with accelerated foveated rendering schemes. Building on this line of work, A3FR [18] explicitly integrates eye tracking with 3D Gaussian Splatting, dynamically modulating the density or scale of Gaussian primitives based on the user's gaze. Other approaches leverage variable-rate shading or employ separate neural representations to render peripheral regions at reduced complexity.

Existing 3DGS-based foveated rendering methods primarily rely on spatial level-of-detail adjustments, reducing either the number of projected splats or the rendering resolution in peripheral regions. While our work does not implement a closed-loop, gaze-contingent system, we instead focus on the lower-level psychophysical limits governing color compositing. By explicitly modeling the relationship between chromaticity and the perceptual visibility of residual contributions, our sensitivity analysis provides a practical and theoretically grounded foundation for optimizing transmittance thresholds within foveated rendering pipelines.

3 Methodology

3.1 Bounding the Residual Color Contribution

To theoretically justify our early-termination condition during volume rendering, we derive an upper bound on the potential color contribution of the remaining Gaussian primitives along a ray. Recall the volume rendering formulation from Section 2.1. Assuming normalized color values $c \in [0,1]^3$, we isolate the residual color contribution R_i from the i-th Gaussian to the end of the ray. Factoring out the current accumulated transmittance T_i, we obtain:

$$R_i = \sum_{k=i}^{N} c_k \alpha_k T_k = T_i \sum_{k=i}^{N} c_k \underbrace{\left(\alpha_k \prod_{j=i}^{k-1}(1-\alpha_j)\right)}_{w_k} \tag{2}$$

Here, we define the term inside the summation as the weight w_k. We observe that the sum of these weights forms a telescoping series, which simplifies to:

$$\sum_{k=i}^{N} w_k = 1 - \prod_{k=i}^{N}(1-\alpha_k) \tag{3}$$

Since $\alpha_k \in [0,1]$, the product term lies in $[0,1]$, implying that the sum of weights satisfies $0 \le \sum w_k \le 1$.

Finally, given that the color components c_k are bounded within $[0,1]$, we arrive at the upper bound for the residual:

$$R_i \le T_i \cdot \left(\sum_{k=i}^{N} w_k\right) \le T_i \tag{4}$$

This result confirms that the remaining color contribution is strictly trapped within a cube of side length T_i, justifying

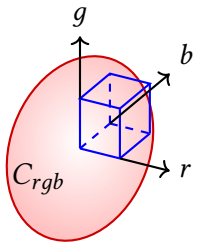


Figure 1. Geometric containment of the residual color cube within the JND ellipsoid in RGB color space.

the approximation of truncating the ray when T_i falls below our perceptually guided threshold.

3.2 Geometric Criterion of Early Termination

To apply the bound derived above, we determine the maximum allowable transmittance T_{limit} such that the residual color volume remains visually imperceptible. We cast this as a geometric containment problem: the residual cube of side length T_i must lie entirely within the local JND ellipsoid. Figure 1 illustrates this containment relationship.

Following recent work in perceptual graphics [16], we model color sensitivity in the DKL color space. We assume that the JND threshold can be approximated by an **axis-aligned ellipsoid** in DKL space, defined by a diagonal scaling matrix $\mathbf{K}$:

$$\mathcal{E}_{DKL} : (\mathbf{y} - \mathbf{C}_{dkl})^T \mathbf{K} (\mathbf{y} - \mathbf{C}_{dkl}) \leq 1 \tag{5}$$

where $\mathbf{y}$ represents a color in DKL space. However, 3DGS rendering operates in the linear RGB space. Let $\mathbf{M}$ be the linear transformation matrix from RGB to DKL such that $\mathbf{y} = \mathbf{M}\mathbf{x}$:

$$(\mathbf{M}\mathbf{x} - \mathbf{M}\mathbf{C}_{rgb})^T \mathbf{K} (\mathbf{M}\mathbf{x} - \mathbf{M}\mathbf{C}_{rgb}) \leq 1$$

$$\mathcal{E}_{RGB} : (\mathbf{x} - \mathbf{C}_{rgb})^T \underbrace{(\mathbf{M}^T \mathbf{K} \mathbf{M})}_{\mathbf{Q}} (\mathbf{x} - \mathbf{C}_{rgb}) \leq 1 \tag{6}$$

Here, $\mathbf{Q}$ is a symmetric positive-definite matrix describing the perceptual metric in RGB space. Recall from the previous section that the potential residual color contribution R_i is bounded by a cube of side length T_i. Geometrically, the residual vector relative to the current color $\mathbf{C}_{rgb}$ can be expressed as $\Delta\mathbf{x} = T_i \cdot \mathbf{v}$, where $\mathbf{v} \in 0, 1^3$ enumerates the vertices of the unit cube. > Owing to the convexity of the JND ellipsoid, the residual cube is entirely contained within the perceptual threshold if and only if all of its vertices satisfy the ellipsoid inequality. Substituting $\Delta\mathbf{x}$ into Eq. (6):

$$(T_i \mathbf{v})^T \mathbf{Q} (T_i \mathbf{v}) \leq 1 \implies T_i^2 \cdot (\mathbf{v}^T \mathbf{Q} \mathbf{v}) \leq 1$$

To ensure that all potential residual contributions remain imperceptible, the accumulated transmittance is required to satisfy the above condition for every vertex of the residual cube. This leads to a sufficient condition for early termination:

$$T_i \leq \min_{\mathbf{v} \in \{0,1\}^3, \mathbf{v} \neq \mathbf{0}} \frac{1}{\sqrt{\mathbf{v}^T \mathbf{Q} \mathbf{v}}} \tag{7}$$

This derivation enables us to precompute the minimal required transmittance T for a given local color context, effectively replacing the fixed ϵ with a rigorous, perceptually grounded threshold.

3.3 LUT & GPU Implementations

To enforce the perceptual bounds derived in Eq. (7) without incurring significant arithmetic overhead during rendering, we discretize the linear RGB color space $[0, 1]^3$ into a lightweight 8^3 grid. For each voxel center $\mathbf{c}$, we precompute the maximum admissible transmittance T_{limit} offline. Specifically, we transform the local metric tensor $\mathbf{G} * rgb$, obtained from the Hessian of the non Euclidean CIEDE2000 color difference, into the uncorrelated DKL color space using standard colorimetric projection pipelines [1, 3, 14]. The conservative termination threshold is determined by the shortest principal axis of the resulting JND ellipsoid: $T_{limit} = \min_i(\lambda_{ii})^{-1/2}$, where λ_{ii} denote the diagonal elements of the projected metric tensor $\mathbf{G}_{dkl}$. This precomputation produces a compact scalar field of perceptual thresholds, typically ranging from 10^{-3} to 10^{-2}, that maps local colors to their corresponding visibility limits.

Computation-Memory Trade-off. Directly evaluating the complex non Euclidean DKL metric within the inner ray marching loop would shift the performance bottleneck from memory bandwidth to computation. The heavy floating point operations required for per Gaussian evaluation would significantly increase register pressure per thread, thereby reducing active warp occupancy on the Streaming Multiprocessors. Through precomputation, we instead replace costly arithmetic with a single highly efficient memory access, preserving throughput in the performance critical rendering loop.

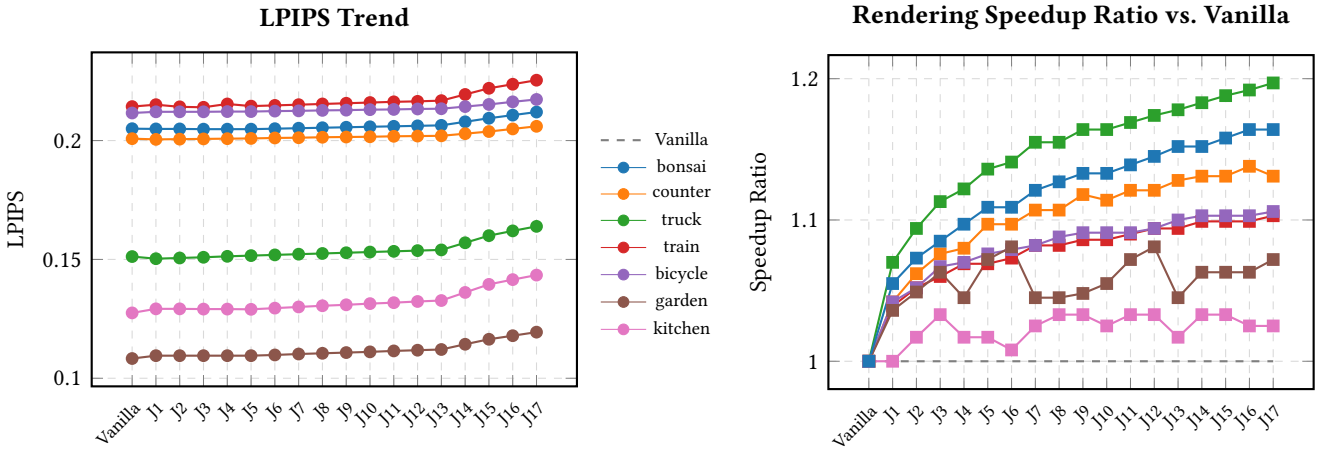
Cache Hierarchy and Texture Fetch. Our discrete 8^3 LUT is extremely lightweight and requires only 2 KB of storage. During CUDA tile based rasterization, this compact footprint ensures that the entire structure remains resident in cache. As a result, fetching T_{limit} incurs near zero latency and avoids global memory transactions. In addition, we retrieve the threshold using a strict nearest neighbor lookup. We intentionally avoid trilinear interpolation, in contrast to Plenoxels [19], to bypass hardware texture filtering units. This design both saves critical execution cycles and strictly preserves the conservative mathematical guarantees of the precomputed perceptual bounds.

4 Experiments

4.1 Settings

We evaluate our approach on seven representative scenes selected from the Mip NeRF 360 [2] and Tanks and Temples [9] datasets. These scenes cover a broad spectrum of visual characteristics, including unbounded environments and high frequency textures, offering a rigorous benchmark for perceptual rendering quality. All experiments are conducted on a single NVIDIA T4 GPU with 16 GB of VRAM and 12 GB of system memory. Our implementation builds upon the official 3D Gaussian Splatting framework and augments the core tile based CUDA rasterizer to support LUT

Figure 2. Qualitative degradation with threshold scaling.



(a) LPIPS Trend **(b) Speedup Ratio**

Figure 3. Quantitative analysis across different scenes.

based early termination, while leaving the training pipeline unchanged.

We compare our method with the official implementation in 3dGS [8], referred to as the vanilla baseline, which uses a fixed global transmittance threshold $\epsilon = 10^{-4}$. This baseline serves as the reference for both visual fidelity and performance.

4.2 Performance & Sensitivity Analysis

To evaluate the performance of our perceptual bound, we introduce a scaling factor γ to modulate the pruning aggressiveness: $T_{LUT} = \gamma \cdot T_i$. Throughout our experiments, we use J_k to denote the configuration where $\gamma = k$.

Visual Quality. As illustrated by the representative example in Fig. 2, there is almost no perceptible difference between Vanilla 3DGS and our JND method at the default setting (J_1). This visual consistency holds true across all evaluated scenes, which suggests that our conservative derivation aligns with the JND threshold, achieving nearly lossless rendering. In Fig. 3(a), we show the image quality changes with the varying scaling factor γ. This high fidelity is robustly maintained until γ exceeds 10, after which image quality noticeably deteriorates, with visible artifacts appearing in high-frequency regions.

Rendering Efficiency. The quantitative results in Fig. 3(b) exhibit a clear trend of diminishing returns in performance. While the transition from the vanilla baseline to J_1 yields a substantial speedup, further increases in γ lead to progressively smaller gains in FPS.

4.3 Ablation Study

In this section, we compare our JND method ($\gamma = 1.0$) against a baseline method, termed *Naive method*, to justify the necessity of local color-perceptual information.

The Naive Baseline. The Naive method mimics a simple global increase of the transmittance threshold. It uses a single scalar value calculated from the cubic mean of our entire LUT, effectively ignoring the local color variance (ΔE) and spatial perceptual importance.

Comparison Results. As shown in Table 1, our method consistently outperforms the naive baseline across all perceptual metrics. At comparable speedup levels, the naive approach exhibits a pronounced degradation in LPIPS, as it prunes Gaussian primitives without accounting for their color contributions. Even when the naive method is adjusted to achieve similar visual quality, our approach maintains a higher speedup. These results highlight the importance of leveraging local color information to preserve rendering fidelity.

5 Conclusions

We propose a perceptually guided early termination mechanism for 3D Gaussian Splatting that replaces fixed thresholds with a JND based lookup table. Our method improves rendering efficiency while preserving visual fidelity. The tunable γ parameter enables further gains in foveated rendering by exploiting reduced chromatic sensitivity in peripheral vision. Orthogonal to static pruning, our approach provides a lightweight solution for real time rendering.

Table 1. Quantitative Comparison. Bold indicates the best result between *Naive* and *JND*.

Method	SSIM ↑	PSNR ↑	LPIPS ↓	FPS ↑	Speedup
Vanilla	0.869	27.30	0.174	20.9	1.000×
Naive	0.818	25.85	0.247	**22.6**	**1.084**×
JND	**0.866**	**27.21**	**0.175**	21.8	1.043×

References

[1] 1999. Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB.

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5470–5479.

[3] Andrew M Derrington, John Krauskopf, and Peter Lennie. 1984. Chromatic mechanisms in lateral geniculate nucleus of macaque. *The Journal of Physiology* 357, 1 (1984), 241–265.

[4] Runze Fan, Jian Wu, Xuehuai Shi, Lizhi Zhao, Qixiang Ma, and Lili Wang. 2025. Fov-gs: Foveated 3d gaussian splatting for dynamic scenes. *IEEE Transactions on Visualization and Computer Graphics* (2025).

[5] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. 2024. LightGaussian: Unbounded 3D Gaussian Compression with 15x Reduction and 200+ FPS. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[6] Linus Franke, Laura Fink, and Marc Stamminger. 2025. Vr-splatting: Foveated radiance field rendering via 3d gaussian splatting and neural points. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 8, 1 (2025), 1–21.

[7] Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. 2024. EAGLES: Efficient Accelerated 3D Gaussians with Lightweight EncodingS. In *European Conference on Computer Vision (ECCV)*.

[8] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–14.

[9] Arno Knapitsch, Jaesik Park, Zhou-Yin Qian, and Vladlen Koltun. 2017. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. In *ACM Transactions on Graphics (TOG)*, Vol. 36. ACM New York, NY, USA, 1–13.

[10] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. 2024. Compact 3D Gaussian Splatting for Static and Dynamic Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1611–1620.

[11] Weikai Lin, Yu Feng, and Yuhao Zhu. 2025. MetaSapiens: Real-Time Neural Rendering with Efficiency-Aware Pruning and Accelerated Foveated Rendering. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Volume 1*. 669–682. doi:10.1145/3669940.3707227

[12] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. 2024. Scaffold-GS: Structured 3D Gaussians for View-Adaptive Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1016–1026.

[13] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision (ECCV)*.

[14] Nathan Moroney, Mark D Fairchild, Robert WG Hunt, Changjun Li, M Ronnier Luo, and Todd Newman. 2002. The CIECAM02 color appearance model. In *Color and imaging conference*, Vol. 2002. Society for Imaging Science and Technology, 23–27.

[15] Stanley Smith Stevens. 1975. *Psychophysics: Introduction to Its Perceptual, Neural, and Social Prospects*. Wiley.

[16] Nitish Ujjainkar, E Shahan, K Chen, B Duinkharjav, Q Sun, and Y Zhu. 2024. Exploiting Human Color Discrimination for Memory- and Energy-Efficient Image Encoding in Virtual Reality. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '24)*. ACM, La Jolla, CA, USA, 1–15.

[17] Haiyu Wang, Wenxuan Liu, Kenneth Chen, Qi Sun, and Sai Qian Zhang. 2025. Process only where you look: Hardware and algorithm co-optimization for efficient gaze-tracked foveated rendering in virtual reality. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture*. 344–358.

[18] Shuo Xin, Haiyu Wang, and Sai Qian Zhang. 2025. A3FR: Agile 3D Gaussian Splatting with Incremental Gaze Tracked Foveated Rendering in Virtual Reality. In *Proceedings of the 39th ACM International Conference on Supercomputing*. 279–292.

[19] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance Fields without Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5501–5510.

A Proof: Bounding the Residual Color Contribution in 3DGS

1. **Objective:** To prove that at any step i during volume rendering, the remaining total color contribution R_i is bounded by the accumulated current transmittance T_i assuming normalized color values $c \in [0,1]^3$.
2. **Definitions:** The final pixel color C is computed via front-to-back alpha blending of N Gaussians:

$$C = \sum_{k=1}^{N} c_k \alpha_k T_k$$

Where the accumulated transmittance T_k is defined as:

$$T_k = \prod_{j=1}^{k-1} (1 - \alpha_j), \quad \text{with } T_1 = 1$$

(Here, $T_1 = 1$ follows the empty product convention, representing zero initial occlusion.)

3. **Derivation of the Residual Formula:** The residual sum R_i from the i-th Gaussian to the end of the ray is:

$$R_i = \sum_{k=i}^{N} c_k \alpha_k T_k$$

For any $k \geq i$, we can factor T_k into two parts:

$$T_k = \underbrace{\left[\prod_{j=1}^{i-1} (1 - \alpha_j)\right]}_{T_i} \cdot \left[\prod_{j=i}^{k-1} (1 - \alpha_j)\right]$$

Substituting this back into the residual equation:

$$R_i = T_i \sum_{k=i}^{N} c_k \underbrace{\left(\alpha_k \prod_{j=i}^{k-1} (1 - \alpha_j)\right)}_{w_k}$$

4. **Bounding the Weights:** We observe that the sum of these weights is a telescoping series:

$$\begin{aligned} w_k &= (1 - (1 - \alpha_k)) \prod_{j=i}^{k-1} (1 - \alpha_j) \\ &= \prod_{j=i}^{k-1} (1 - \alpha_j) - \prod_{j=i}^{k} (1 - \alpha_j) \end{aligned}$$

For $k = i$, the empty product $\prod_{j=i}^{i-1}(1 - \alpha_j) = 1$. Similarly to $T_1 = 1$, this physically indicates zero preceding occlusion for this residual segment, yielding $w_i = \alpha_i$. Summing these weights from $k = i$ to N:

$$\sum_{k=i}^{N} w_k = 1 - \prod_{k=i}^{N}(1 - \alpha_k)$$

Since $\alpha_k \in [0, 1]$, the product $\prod(1 - \alpha_k)$ must lie in the range $[0, 1]$. Therefore:

$$0 \leq \sum_{k=i}^{N} w_k \leq 1$$

5. **Conclusion:** Given that the color components c_k are within the range $[0, 1]$, the maximum possible value for each channel in R_i is:
$$R_i^{(c)} \leq T_i \cdot (1) = T_i$$
6. **Result:** The remaining color contribution is strictly bounded by T_i. In the RGB color space, this means the future color sum is trapped within a cube of side length T_i in linear RGB space that shrinks toward zero as $T_i \to 0$.