Image Based Level-of-Detail Optimization for Large-Scale 3D Reconstruction

Ole Siemers*
Supervised by: Diana Marin†

TU Wien

Abstract

This work introduces a coarse-to-fine optimisation approach for 3D Gaussian Splatting (3DGS) that constructs a Level of Detail (LoD) hierarchy during optimisation. By progressively adjusting resolution, the method reduces computational costs and accelerates optimisation. Based on the sampling distance, defined as the ratio between the resolution at which the model was optimised and the resolution at which it is viewed, a selective rendering approach further reduces the number of processed primitives, mitigating aliasing errors, at the cost of increased GPU memory usage due to the storage of multiple LoD levels. The approach is evaluated against 3DGS and EWA-Filtering baselines on widely used 360° and aerial datasets, focusing on low-resolution reconstructions and zoom-out trajectories. Results show that the method accelerates optimisation and reduces the number of rendered primitives – particularly in distant or low-resolution views – while maintaining visual quality comparable to the baselines. Although the technique requires additional GPU memory during rendering, it offers a practical approach towards large-scale scene rendering applications, such as remote sensing.

Keywords: Gaussian Splatting, Large-Scale, Level of Detail

1 Introduction

3D Gaussian Splatting (3DGS) has emerged as a strong representation for novel view synthesis due to its combination of high visual quality and real-time rasterisation-based rendering [6]. However, the efficiency and fidelity of 3DGS degrade as scenes become larger and viewpoints move away from the distribution of the training images. In such cases, a model optimised for one sampling regime may be rendered under another, leading to oversampling, aliasing artefacts, and unnecessary processing of primitives. At the same time, large-scale reconstructions increase optimisation time, memory footprint, and rendering cost.

These issues motivate optimisation strategies that explicitly account for scale. Coarse-to-fine supervision has recently been shown to improve optimisation efficiency and, in several settings, to reduce the number of primitives needed to represent a scene. Likewise, anti-aliasing methods for Gaussian splatting have highlighted the importance of matching reconstruction frequency to the sampling conditions of both optimisation and rendering. In parallel, large-scale radiance-field systems increasingly rely on level-of-detail (LoD) mechanisms to keep rendering tractable on limited hardware.

This paper studies whether coarse-to-fine optimisation can serve not only as a training schedule, but also as a direct mechanism for constructing an LoD hierarchy during a single optimisation run. We progressively optimise 3DGS on an image pyramid, save the intermediate models, and merge them into a multi-resolution representation. We then propose a selective rendering strategy that chooses the appropriate LoD according to the ratio between the sampling distance at which a primitive was optimised and the sampling distance at which it is currently viewed. This avoids a separate post-processing stage for constructing LoDs from a full-resolution model.

The resulting method builds on the standard 3DGS pipeline, adds only two scalar attributes per Gaussian, and leaves the underlying densification pipeline unchanged, which adaptively splits and clones Gaussians based on gradient magnitude. It offers clear benefits: progressive optimisation reduces optimisation time and model size substantially, while selective rendering improves behaviour in low-resolution and zoom-out regimes by reducing both aliasing and the number of rendered primitives.

Our contributions are:

- A coarse-to-fine optimisation schedule for 3DGS that saves intermediate models and turns them into an LoD hierarchy within a single training run.
- A sampling-distance-based selective rendering algorithm that chooses the appropriate LoD at render time using per-Gaussian metadata.
- An evaluation on standard novel view synthesis benchmarks and larger aerial scenes, showing reduced optimisation cost, improved low-resolution behaviour, and fewer rendered primitives for distant

*ole.siemers@tuwien.ac.at
†diana.marin@tuwien.ac.at

views.

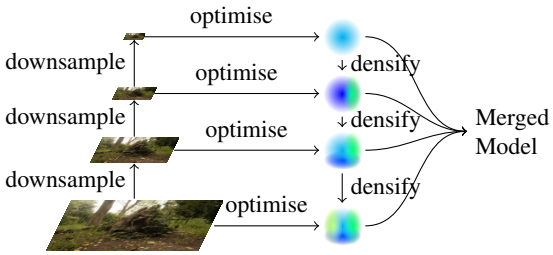


Figure 1: The reconstruction is progressively constructed using low- to high-frequency images as a supervision signal for the optimisation. Starting from low resolution, the supervision signal is refined as the optimisation progresses. Before upsampling, the model that fits the lower resolution is saved. Finally, the optimisation is supervised at full resolution, and the models are merged.

2 Related Work

3D Gaussian Splatting and anti-aliasing. 3DGS represents a scene as anisotropic Gaussians that are projected and alpha-blended in screen space, enabling differentiable optimisation and real-time rendering [6]. Its efficiency has made it a widely used baseline for novel view synthesis. However, because standard 3DGS evaluates projected splats at pixel centres, rendering a model outside the sampling regime in which it was optimised can produce visible aliasing artefacts. Mip-Splatting addresses this issue through world-space and screen-space filtering, including an EWA-style screen-space filter that enforces a minimum projected footprint [16, 17]. Analytic integration approaches anti-aliasing by evaluating splats over the full pixel area, albeit at a higher computational cost [9].

Coarse-to-fine optimisation in Gaussian splatting. Several recent methods exploit the observation that Gaussian splatting tends to fit low frequencies first. Coarse-to-fine strategies, therefore, begin with a reduced or smoothed supervision signal and progressively increase detail. Prior work has explored image-frequency modulation, frequency-aware regularisation, and flexible LoD construction during optimisation, showing that progressive schedules can reduce model size and speed up convergence while retaining competitive quality [3, 13, 2, 5]. Our approach follows this line of work, but focuses specifically on using the saved intermediate models as a practical LoD hierarchy for rendering, rather than treating the schedule only as an optimisation aid.

Large-scale reconstruction and LoD rendering. Scaling radiance fields to large environments remains challenging due to optimisation and memory constraints. Existing systems address this either by partitioning scenes into chunks or by building hierarchical structures that can be rendered at different levels of detail. Chunk-based methods can be effective, but require explicit post-processing to assemble the final hierarchy [7, 11, 12, 15]. In contrast, our method constructs the LoD hierarchy directly from intermediate optimisation states and uses a sampling-based selection rule at render time. The goal is not to replace full hierarchical streaming systems, but to provide a lightweight first step that already captures substantial efficiency gains for low-resolution and distant views.

3 Method

We build on the standard 3DGS pipeline and introduce two additions: a progressive resolution schedule during optimisation, and a sampling-distance-based rule for selecting the appropriate LoD during rendering.

3.1 Progressive LoD Optimisation

Let the full-resolution supervision images define the finest level of an image pyramid. Instead of optimising 3DGS directly at full resolution, we start from a lower-resolution version of each image and progressively increase the supervision resolution over the course of optimisation. In the experiments of this paper, we use a five-level pyramid with scales $\left(\frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1\right)$ and upsample the supervision signal at regular iteration intervals.

Our optimisation proceeds exactly as in 3DGS, except that at iteration t the loss is computed against the supervision image at the current pyramid level. Before each upsampling step, the current Gaussian model is saved. After the final full-resolution stage, the saved models are merged into a single point cloud that contains all levels of detail. This yields a LoD hierarchy "for free" from one optimisation run.

The intuition behind our model is the following: Low-resolution images contain lower spatial frequencies and can be matched with fewer primitives. Early optimisation, therefore, propagates gradients through fewer pixels and typically produces more compact models. Later stages refine the representation only where finer detail is needed. The optimisation process is illustrated in Figure 1.

For each Gaussian, we store two additional scalar attributes:

1. The sampling distance at which the primitive was observed during optimisation.

2. The relative supervision scale (downsampling factor) of the model in which the primitive was saved.

These attributes are sufficient to determine which LoD should be used during rendering.

3.2 Sampling-Distance-Based Selective Rendering

Selective rendering is based on the observation that the sampling distance of a surface patch depends on the image resolution, focal length, and distance to the camera. Under the pinhole camera model, the sensor sampling distance is

$$s = \frac{1}{r}, \tag{1}$$

where r denotes image resolution. The world-space sampling distance is then

$$S = s\frac{d}{f}, \tag{2}$$

where d is the Euclidean distance from the camera to the primitive and f is the focal length.

At render time, we compare the current sampling distance to the one under which a primitive was optimised. With hats denoting optimisation-time quantities, the relative sampling distance becomes

$$S_r = \frac{S}{\hat{S}} = \frac{d\,\hat{f}_y\,\hat{r}_y}{\hat{d}\,f_y\,r_y}. \tag{3}$$

Intuitively, $S_r = 1$ means the primitive is rendered under the same sampling conditions as during optimisation. If $S_r > 1$, the current view corresponds to a coarser sampling regime, and a lower-resolution LoD may be more appropriate.

Each saved model corresponds to a discrete supervision scale S_o. Given a merged LoD hierarchy, we clamp the current sampling distance to the available range and keep a primitive if

$$|S_o - \mathrm{clamp}(S_r, s_{\min}, s_{\max})| \le 0.5. \tag{4}$$

To blend between two consecutive LoD, a blending ratio $b \in [0,1]$ is used. When $b > 0$, Gaussians of two LoD can overlap.

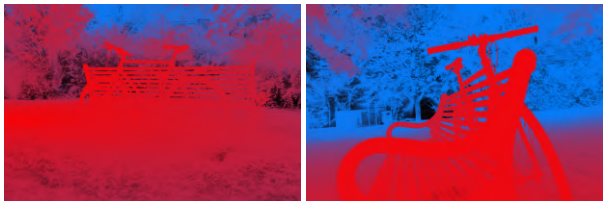
$$|S_o - \mathrm{clamp}(S_r, s_{\min}, s_{\max})| \le 0.5 + 0.5b \tag{5}$$

The opacity of the Gaussians is adjusted by the factor o.

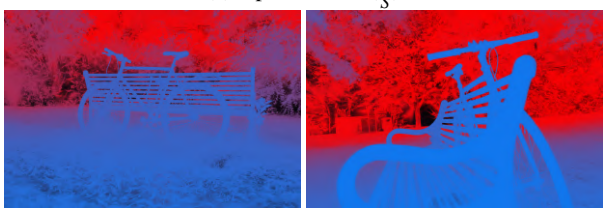
$$o = -\frac{|S_o - \mathrm{clamp}(S_r, s_{\min}, s_{\max})|}{b} + \frac{0.5f}{b} + 0.5 \tag{6}$$

Primitives outside the selected level are culled before rasterisation. In practice, this means that detailed Gaussians are used only where the current viewpoint justifies them, while coarser regions fall back to lower-resolution models, as exemplified in Figure 2.

This strategy differs from screen-space anti-aliasing filters. EWA-style filtering smooths the projected footprint of a Gaussian during rasterisation; our method instead switches to a model that was actually optimised for the appropriate sampling regime. The two are therefore orthogonal and can also be combined.



(a) Optimisation ($\frac{1}{\hat{S}}$)



(b) Rendering (S)

(c) Reconstruction $S_r = \frac{S}{\hat{S}}$

Figure 2: Visualisation of three sampling-distance notions (top to bottom) for two viewpoints (left, right). The optimisation sampling distance is approximated by the distance to the closest optimisation view from which a Gaussian was optimised (Subfigure a). The rendering sampling distance depends on the distance to the current view camera, with nearby points seen at higher resolution (Subfigure b). The reconstruction sampling distance is the ratio between both quantities (Subfigure c). For views outside the optimisation distribution (left), parts of the scene are reconstructed at lower resolution; for views close to it (right), the sampling distance is more uniform, and the scene is reconstructed at a single resolution.

4 Results

We evaluate the method against the GraphDeco-Inria 3DGS baseline [6] and against EWA-Filtering as used in Mip-Splatting [16]. For progressive optimisation, experiments are conducted on all Mip-NeRF 360 scenes [1], two Tanks & Temples scenes (*Truck* and*Train*) [8], and two Deep Blending scenes (*Dr. Johnson* and *Playroom*) [4]. Following common practice, every 8th image is used for testing. Unless stated otherwise, models are optimised for 30k iterations and densified until iteration 25k. Renderings of the test sets are produced without blending. We report PSNR (higher is better), SSIM (higher is better), and LPIPS (lower is better) to assess perceptual quality and structural similarity.

To analyse selective rendering under out-of-distribution viewpoints, we additionally render zoom-out trajectories on one Mip-NeRF 360 scene (*Bicycle*) and three larger aerial scenes (*Building* [14], *ArtSci* [10], and *Wolf an der Au* - internally captured). These experiments focus on two questions: how reconstruction quality changes when rendering at lower resolution, and how the number of rendered Gaussians and frame time evolve as the camera moves away from the scene.

4.1 Progressive Resolution Scheduling

Progressive resolution scheduling can be adjusted to balance faster optimisation of smaller models with higher-quality models. As shown in Table 1, the schedule leading to the highest quality in our study uses a five-level linear pyramid and early upsampling intervals. The smallest model is obtained fastest with quadratic pyramid scaling and late upsampling. Compromising between model size and reconstruction quality, the linear pyramid scaling with constant upsampling intervals is used for further comparison in this work.

A central question is whether progressive resolution scheduling improves the optimisation trade-off between quality, model size, and runtime. Across all benchmark datasets, the answer is yes for efficiency, but not for final full-resolution quality. These findings are summarised in Table 2.

Relative to standard 3DGS, progressive optimisation reduces the average model size from 2.26×10^6 to 1.31×10^6 Gaussians and shortens total optimisation time from 191.6 to 136.7 minutes. This corresponds to roughly half the number of primitives and about two-thirds of the optimisation time. The cost is a small drop in reconstruction quality: average PSNR decreases from 27.63 to 27.28 dB, SSIM from 0.843 to 0.826, and LPIPS increases from 0.183 to 0.217. These results summarised in Table 3 indicate that using progressive resolution is an effective model reduction and optimisation speedup strategy, even though in our experiments it does not improve the full resolution image metrics over the baseline. The mean iteration time is reduced because fewer gradients are backpropagated from smaller images to fewer primitives. The progress of the model size is shown in Figure 6. Our method should be understood primarily as an efficiency-oriented optimisation schedule that preserves comparable quality, rather than as a universal quality-improvement method.

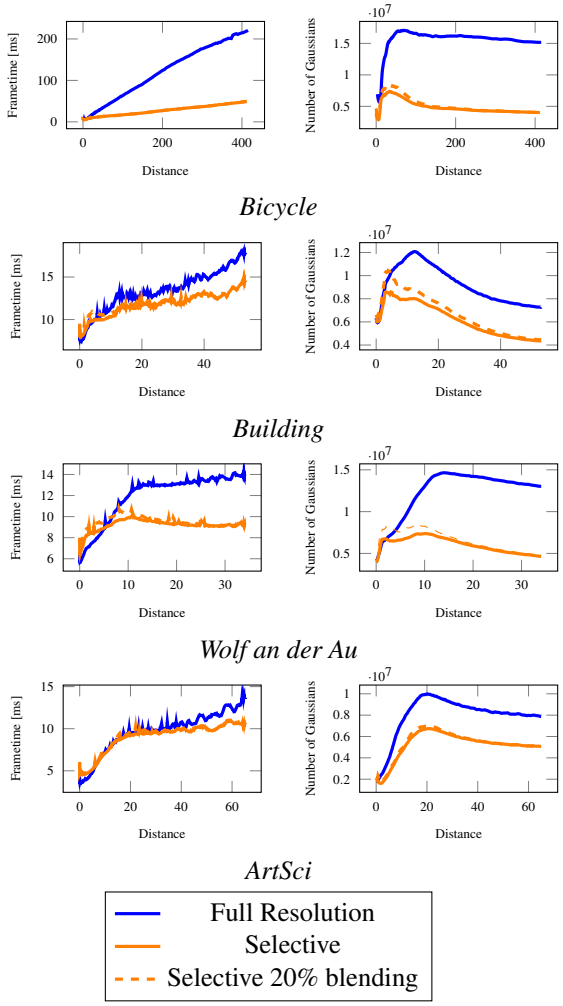


Figure 3: Number of rendered Gaussians and the time a frame takes to render in milliseconds versus the camera movement distance in a sequence of images when zooming out. Selective rendering reduces the number of rendered Gaussians and takes less time to render a frame when the scene is viewed from far away compared to the 3DGS implementation.

4.2 Selective Rendering

The second question is whether the saved intermediate models can serve as an effective LoD hierarchy for rendering. Here, the answer is clearly positive for low-resolution rendering and zoom-out trajectories.

When a full-resolution 3DGS model is rendered at reduced output resolution, characteristic oversampling artefacts appear: thin structures become too thick and overly bright, as can be seen in Figure 8. Selective rendering mit-

EWA-Filtering	Upsampling	Resolution	PSNR ↑	SSIM ↑	LPIPS ↓	Size ↓	Duration ↓
✗	constant	linear	27.04	0.82	0.22	$1.41 \cdot 10^6$	137.29
✗	constant	quadratic	26.73	0.81	0.24	$1.32 \cdot 10^6$	129.23
✗	early	linear	**27.25**	**0.83**	**0.20**	$1.70 \cdot 10^6$	149.45
✗	early	quadratic	27.13	0.83	0.21	$1.62 \cdot 10^6$	143.18
✗	late	linear	26.68	0.80	0.25	$1.02 \cdot 10^6$	128.14
✗	late	quadratic	25.91	0.77	0.29	$7.56 \cdot 10^5$	**117.83**
✓	constant	linear	26.84	0.81	0.23	$1.72 \cdot 10^6$	148.64
✓	constant	quadratic	26.62	0.80	0.25	$1.44 \cdot 10^6$	128.45
✓	early	linear	27.05	0.83	0.22	$2.17 \cdot 10^6$	161.47
✓	early	quadratic	26.92	0.82	0.22	$2.04 \cdot 10^6$	151.10
✓	late	linear	26.39	0.79	0.26	$2.25 \cdot 10^6$	154.57
✓	late	quadratic	25.74	0.76	0.29	$1.37 \cdot 10^6$	123.23

Table 1: Comparison of the reconstruction quality for different 5-layer pyramid resolutions and upsampling intervals with and without EWA-Splatting. Upsampling intervals are either constant (6k, 12k, 18k, 24k), early (2k, 5k, 11k, 19k), or late (11k, 19k, 25k, 28k). Downsampling factors in the image pyramid are linear (1, 2, 3, 4, 5) or quadratic (1, 2, 4, 8, 16).

Dataset	**Progressive**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Model Size ↓**	**Duration [min] ↓**
Deep Blending	✗	29.7	0.91	0.24	$1.96 \cdot 10^6$	29.93
Deep Blending	✓	29.77	0.91	0.24	$1.22 \cdot 10^6$	23.16
MipNeRF 360	✗	27.95	0.83	0.17	$2.53 \cdot 10^6$	138.43
MipNeRF 360	✓	27.48	0.81	0.21	$1.47 \cdot 10^6$	96.73
Tanks and Temples	✗	24.15	0.86	0.17	$1.36 \cdot 10^6$	23.26
Tanks and Temples	✓	23.88	0.84	0.2	$7.22 \cdot 10^5$	16.76

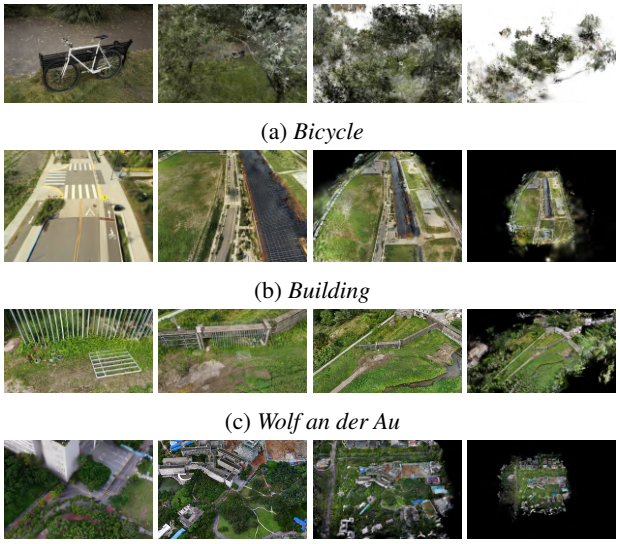
Table 2: Average error metrics, model size and total optimisation time for each dataset. Reconstruction quality is slightly lower when optimising with progressive resolution, while the model size is smaller and optimisation is faster.

Method	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Model Size ↓**	**Duration [min] ↓**
3DGS	27.634	0.843	0.183	2 260 313	191.6
Prog. Schedule	27.275	0.826	0.217	1 314 356	136.7
EWA-Filtering	27.429	0.837	0.198	2 933 159	198.7
EWA-Filtering + Prog.	27.159	0.820	0.232	1 763 174	150.2

Table 3: Average error metrics, model size and total optimisation time aggregated over all datasets. Progressive resolution scheduling slightly decreases qualitative metrics while significantly reducing model size and optimisation time.

Scene	Selective	SSIM	PSNR	LPIPS	Duration	Model Size	Visible
Bicycle	✗	0.75	24.56	0.21	44.20	$1.13 \cdot 10^7$	$1.73 \cdot 10^6$
Bicycle	✓	0.66	22.77	0.31	23.88	$2.25 \cdot 10^7$	$5.23 \cdot 10^5$
Building	✗	0.67	21.23	0.41	57.61	$4.25 \cdot 10^6$	$4.34 \cdot 10^5$
Building	✓	0.64	20.93	0.43	50.47	$1.62 \cdot 10^7$	$3.58 \cdot 10^5$
ArtSci	✗	0.56	20.95	0.51	44.52	$5.21 \cdot 10^6$	$1.24 \cdot 10^5$
ArtSci	✓	0.54	20.89	0.52	45.02	$2.01 \cdot 10^7$	$1.02 \cdot 10^5$
Wolf an der Au	✗	0.39	18.56	0.44	45.31	$8.02 \cdot 10^6$	$8.87 \cdot 10^5$
Wolf an der Au	✓	0.36	18.30	0.47	39.03	$2.29 \cdot 10^7$	$5.85 \cdot 10^5$

Table 4: Comparison of the reconstruction quality metrics, optimisation duration in minutes, and model size in number of primitives of the large-scale scenes with and without using selective rendering.



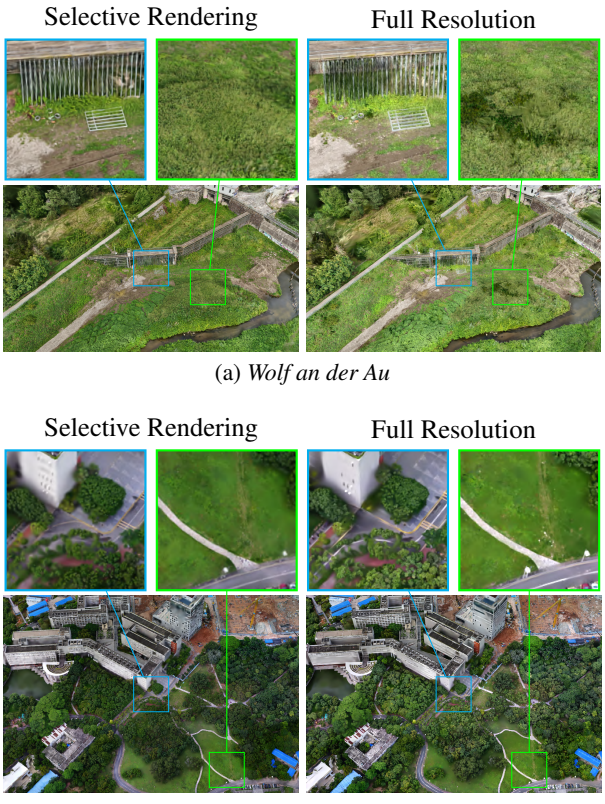
(a) *Bicycle*

(b) *Building*

(c) *Wolf an der Au*

(d) *ArtSci*

Figure 4: Renderings using selective rendering of two scenes on a zoom-out trajectory at four distances. The four images, from left to right, are taken at 0%, 16%, 33%, and 66% of the trajectory.



(a) *Wolf an der Au*

(b) *ArtSci*

Figure 5: Side-by-side comparison of selective rendering (left) against the full resolution 3DGS baseline (right) showing a far view of the *Wolf an der Au* (a) and *ArtSci* b scenes. Enlarged regions highlight areas of interest.

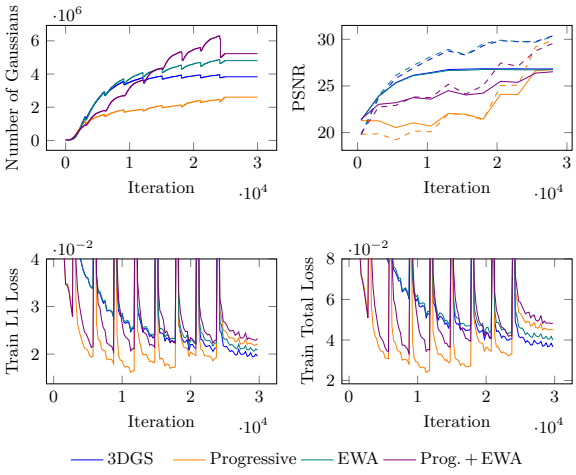


Figure 6: Progress of the number of Gaussians and PSNR measured against full resolution ground-truth during the course of optimisation of the *Stump* scene. The PSNR on the test set is drawn as a dotted line.

igates these artefacts by switching to a model optimised at a more appropriate scale.

In the quantitative comparison across output resolutions, the baseline quality decreases steadily as the render resolution is reduced, while selective rendering maintains markedly better low-resolution behaviour. The combined variant (EWA-Filtering + selective rendering) performs best overall at low resolutions, showing that the two approaches are complementary.

At the same time, selective rendering reduces the number of processed primitives. The standard 3DGS and EWA-filtered baselines keep approximately the same number of visible Gaussians across resolutions, whereas selective rendering decreases the count as resolution drops. In our experiments, selective rendering already uses only about 40% of the Gaussians of the baseline at full resolution, with an even larger advantage at lower resolutions, as shown in Figure 7.

The same effect becomes evident along zoom-out trajectories. As the camera moves farther from the scene, larger parts become visible, and the raw number of visible Gaussians initially increases. However, once parts of the view are better represented by coarser levels, selective rendering switches away from the finest model, reducing both the number of rendered primitives and the average frame time, as shown in Figure 3. This behaviour is visible for the *Bicycle*, *Building*, *Wolf an der Au* and *Art Sci* scenes and supports the claim that the method is particularly useful for large scenes viewed from afar. Images along the camera trajectories are shown in Figure 4. Qualitatively the aliasing error is reduced and the appereance of the scene whene viewed from far away is more natural, as shown in Figure 5.

These benefits come with some trade-offs. Hard switch-

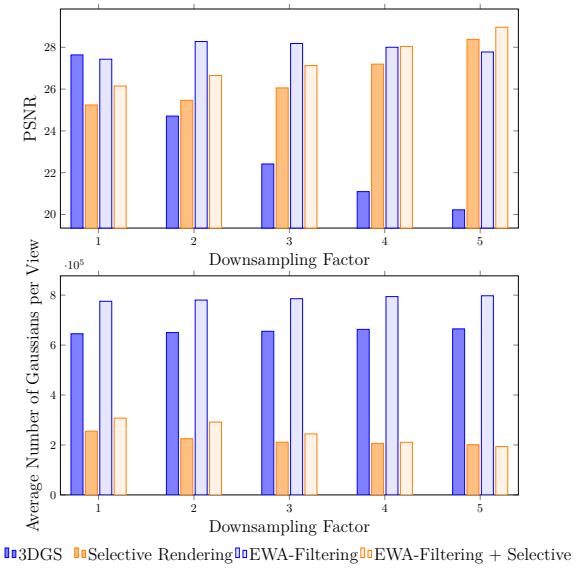


Figure 7: Comparison of the average PSNR and number of Gaussians per view at different fractions of the original resolution. The quality of the baseline decreases at lower resolutions, while it increases with selective rendering. For selective rendering, the number of primitives decreases with resolution and is generally lower than with 3DGS and EWA-Filtering, where the number of primitives remains constant across resolutions.

ing between LoDs can produce visible seams at LoD boundaries when the camera is moving. Blending between levels reduces these seams, but also increases the number of rendered primitives, especially at high-resolution viewpoints, thereby weakening the efficiency benefit. In addition, because all saved levels are merged into a single representation, the full hierarchy occupies more GPU memory than a single-resolution level model.

5 Conclusion

We presented a simple way to turn coarse-to-fine 3DGS optimisation into a practical LoD construction mechanism. By progressively optimising on an image pyramid and saving intermediate models, we obtain a multi-resolution representation in a single run. By storing per-Gaussian sampling metadata, we can then render the merged model selectively according to the current sampling regime.

Empirically, the method provides a useful trade-off between efficiency and quality. Progressive optimisation significantly reduces optimisation time and model size while maintaining comparable reconstruction quality. Selective rendering further improves behaviour in low-resolution and zoom-out settings by reducing aliasing and lowering the number of rendered primitives. The strongest gains, therefore, appear not at the original training resolution, but in scenarios where the viewing conditions differ from the optimisation distribution.

The main limitations are the increased memory required by the merged hierarchy and the potential for seams at LoD boundaries. Future work could address both issues through more compact storage, streaming of LoD levels, and smoother cross-level transitions. More broadly, the results suggest that sampling-aware optimisation and rendering are a promising direction for scaling Gaussian splatting to larger scenes and more diverse target devices.

Acknowledgements

This work was partly funded by the Austrian security research program KPASS of the Federal Ministry of Finance, as part of the PostDisaster project, and supported by computational resources provided by the Austrian Science Fund (FWF) [10.55776/F77].

References

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

[2] Youyu Chen, Junjun Jiang, Kui Jiang, Xiao Tang, Zhihao Li, Xianming Liu, and Yinyu Nie. Dash-Gaussian: Optimizing 3D Gaussian Splatting in 200 Seconds. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11146–11155. IEEE, June 2025.

[3] Umar Farooq, Jean-Yves Guillemaut, Adrian Hilton, and Marco Volino. Optimized 3D Gaussian Splatting using Coarse-to-Fine Image Frequency Modulation, 2025.

[4] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep Blending for Free-Viewpoint Image-Based Rendering. *ACM Transactions on Graphics*, 37(6):1–15, 2018.

[5] Jaewoo Jung, Jisang Han, Honggyu An, Jiwon Kang, Seonghoon Park, and Seungryong Kim. Relaxing Accurate Initialization Constraint for 3D Gaussian Splatting, March 2024.

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

[7] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and

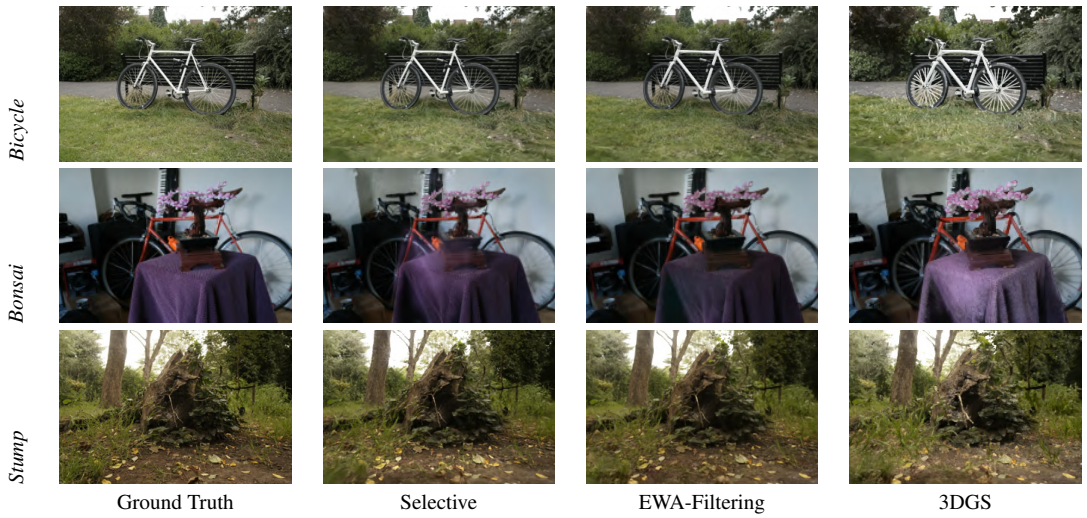


Figure 8: Qualitative comparison of the ground truth images and the renderings at one-quarter resolution from the test set of different scenes. Selective rendering (2nd column) effectively reduces aliasing errors, similar to EWA-Filtering (3rd column), which are present in renderings from 3DGS (4th column), where thin structures are rendered too thick and too bright.

George Drettakis. A Hierarchical 3D Gaussian Representation for Real-Time Rendering of Very Large Datasets. *ACM Transactions on Graphics*, 43(4), July 2024.

[8] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM Transactions on Graphics*, 36(4):1–13, July 2017.

[9] Zhihao Liang, Qi Zhang, Wenbo Hu, Lei Zhu, Ying Feng, and Kui Jia. *Analytic-Splatting: Anti-Aliased 3D Gaussian Splatting via Analytic Integration*, pages 281–297. Number 15075 in Lecture notes in computer science. Springer Nature Switzerland, Berlin, Heidelberg, November 2024.

[10] Liqiang Lin, Yilin Liu, Yue Hu, Xingguang Yan, Ke Xie, and Hui Huang. Capturing, Reconstructing, and Simulating: the UrbanScene3D Dataset. In *ECCV*, pages 93–109, 2022.

[11] Yang Liu, He Guan, Chuanchen Luo, Lue Fan, Junran Peng, and Zhaoxiang Zhang. CityGaussian: Real-time High-quality Large-Scale Scene Rendering with Gaussians, 2024.

[12] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-GS: Towards Consistent Real-time Rendering with LOD-Structured 3D Gaussians. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–15, 2025.

[13] Yunji Seo, Young Choi, Hyunseung Son, and Youngjung Uh. FLoD: Integrating Flexible Level of Detail into 3D Gaussian Splatting for Customizable Rendering. *ACM Transactions on Graphics*, 44(4):1–16, 7 2025.

[14] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-NeRF: Scalable Construction of Large-Scale NeRFs for Virtual Fly- Throughs. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12912–12921. IEEE, June 2022.

[15] Felix Windisch, Lukas Radl, Thomas Kohler, Michael Steiner, Dieter Schmalstieg, and Markus Steinberger. A LoD of Gaussians: Unified Training and Rendering for Ultra-Large Scale Reconstruction with External Memory, 2025.

[16] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-Splatting: Alias-free 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19447–19456, June 2024.

[17] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. EWA Splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, July 2002.