Optimizing 3D Gaussian Splatting via Point Cloud Upsampling

Adrian Ramlal[1]

Yan Song Hu[1], John S. Zelek[1]

[1]Vision and Image Processing Group, Systems Design Engineering, University of Waterloo
{avdramla, y324hu, jzelek}@uwaterloo.ca

Abstract

3D Gaussian Splatting (3DGS) is a technique for creating and rendering 3D scenes, however its performance depends heavily on the quality of initial seed points. To improve 3DGS initialization, this study presents and evaluates several point cloud upsampling approaches: linear interpolation, triangular interpolation, spline-based surface reconstruction, moving least squares surface fitting, and Voronoi-based point generation. Additionally, this research introduces a depth-guided point lifting method that leverages depth maps to maintain geometric consistency with Structure-from-Motion (SfM) reconstructions. Through extensive experiments on the Mip-NeRF360 and Replica datasets, the proposed methods demonstrate improvements in reconstruction quality across diverse scene types. Results indicate that different upsampling strategies excel in different scenarios: surface reconstruction methods perform better with organic, detailed scenes, while simpler interpolation approaches are more suited for scenes dominated by piecewise-smooth geometries. In comparison, the depth-guided approach shows promise for adding geometry-aware points across the entire scene, importantly in texture-less regions. These findings, which provide preliminary practical guidelines for selecting appropriate upsampling methods based on scene characteristics and computational constraints, advances the understanding of how point cloud initialization affects 3DGS quality.

1 Introduction

3D Gaussian Splatting (3DGS) [1] has emerged as a groundbreaking rendering technique in computer vision and graphics. 3DGS represents scenes as a collection of Gaussian primitives, which are 3D semi-transparent clouds that vary in color and density. By strategically positioning and blending these Gaussian primitives, 3DGS can generate photorealistic representations of complex environments and synthesize novel viewpoints. As data-hungry spatial AI techniques continue to evolve, 3DGS has gained prominence as a powerful tool for tasks such as VR/AR, autonomous navigation, and scene understanding. The ability of 3DGS to provide dense volumetric representations of environments and enable the synthesis of novel viewpoints make it a compelling solution for spatial computing applications.

The generation of 3DGS representations from images requires two additional inputs: the relative poses of the input images and 3D points that serve as seed locations for the initial Gaussians. While it is established that 3DGS performance has a strong correlation with pose accuracy, the optimal strategy for initializing seed points remains an open research question. Traditional Structure-from-Motion (SfM) pipelines generate sparse point clouds that, while sufficient for camera pose estimation, may not provide optimal initialization for 3DGS. This sparsity can lead to gaps in the reconstructed scene and require additional optimization iterations to achieve satisfactory results. Moreover, the distribution of these points often follows high-texture regions, leaving important but texture-less areas under-represented.

This paper presents an investigation of methods for augmenting the initial point cloud used for seeding 3DGS, with the goal of improving scene reconstruction quality. Our results demonstrate that thoughtful augmentation of the initial point cloud can lead to improvements in the final reconstruction quality of 3DGS. Our primary contributions include:

1. An evaluation of the impact of interpolation, surface reconstruction and point lifting upsampling methods on 3DGS reconstruction quality.

2. Empirically derived guidelines for selecting appropriate upsampling methods based on scene characteristics.

2 Related Works

Proper initialization of seed Gaussian positions is essential for the effectiveness of 3DGS. This is because the Gaussian cloning and pruning process in 3DGS is driven by heuristics calculated from existing Gaussians. Gaussian creation depends on cloning existing Gaussians, making it challenging to distribute them effectively if initial seeds are poorly placed. Furthermore, a recent work by Lu et al. [2] indicates that 3DGS does not impose penalties for excessive Gaussian numbers, so minimizing the initialization of unnecessary Gaussians can enhance computational efficiency and reduce memory usage. The original 3DGS paper by Kerbl et al. [1] underscores this point, with an ablation study showing a decrease in PSNR performance of around 6 dB when Gaussians are initialized randomly.

The original 3DGS paper used the point cloud output from the SfM system COLMAP [3] as seed points. However, COLMAP's point cloud is not optimized as an initialization for 3DGS; rather, it is a byproduct of the pose estimation process. One straightforward approach to obtaining better seed points is to enhance the images with data from an additional sensor, such as LiDAR [4] or depth sensors [5]. However, introducing an extra sensor is often not ideal.

Machine-learned monocular depth enables point cloud augmentation with additional points without requiring extra sensor data. This approach is utilized by InstantSplat by Fan et al. [6] to allow 3DGS training that finishes training in seconds. InstantSplat achieves this speed by leveraging Dust3R [7], a machine learning-based SfM system that calculates depth to generate dense output maps. The dense and detailed initial point cloud provided by Dust3R is a key factor enabling the speed of the InstantSplat process. Another way to use machine learned depth information is proposed by Chan et al. [8]. Instead of using a different SfM system, the work by Chan et al. augments and realigns COLMAP point clouds using a Dense Prediction Transformer.

While machine-learned augmentation is one approach to enhancing point cloud initialization, this paper focuses on improving seed point clouds using traditional computer vision and graphics methods. One method using traditional methods is by Seibt et al. [9], who propose improving the initial point cloud by detecting and tracking additional points found using a dense SfM system called DFM4SFM. A similar work is by Hu et al. [10] uses the Photometric SLAM system Direct Sparse Odometry instead of COLMAP to do more dense point tracking. In contrast, our approach augments the existing COLMAP-generated point clouds rather than replacing the SfM system altogether.

3 Methods

We present several approaches for upsampling sparse point clouds to generate augmented initialization seeds for 3DGS. Our methods encompass three distinct strategies: interpolation between existing points, local surface reconstruction for generating geometrically-consistent new points, and a depth-guided point lifting approach that leverages depth data. The input to our pipeline consists of sparse point clouds generated through COLMAP reconstruction [3]. Our interpolation and surface-fitting methods operate exclusively on the original reconstructed points to avoid potential cascading errors that could arise from iterative point generation. The final point cloud used for 3DGS initialization is the union of the original and generated points.

3.1 Point Interpolation

3.1.1 Linear Upsampling

Our first approach generates new points by implementing a weighted linear interpolation between existing points. For each new point, we randomly select a point P_1 and find its nearest neighbor P_2, using k-nearest neighbor search (kNN). Then, a randomly selected weight, α, is utilized to interpolate the spatial coordinates (XYZ) and color (RGB):

$$P_{new} = \alpha P_1 + (1 - \alpha)P_2 \qquad (1)$$

Where $\alpha \in [0, 1]$. This method maintains local density features while filling gaps between existing points.

3.1.2 Triangular Upsampling

Building upon the linear approach, triangular upsampling utilizes barycentric coordinates to generate points within the triangular regions formed by existing point triplets. For each new point, we select a random point P_1 and find its two k-nearest neighbors, P_2 and P_3, to form a triangle. Then we generate the point's coordinates and color using barycentric coordinates:

$$P_{new} = \alpha P_1 + \beta P_2 + \gamma P_3 \qquad (2)$$

Where $\alpha + \beta + \gamma = 1$ and $\alpha, \beta, \gamma \geq 0$. This technique improves upon linear upsampling by considering the local surface geometry implied by the increased spatial context of point triplets.

3.2 Surface Reconstruction

3.2.1 Spline Upsampling

Our third approach leverages local B-spline surface fitting to generate new points that better preserve the underlying geometric structure. For each new point, we

randomly select an existing point P_c and identify its nearest neighbors ($k = 25$) to form a local neighborhood $\mathcal{N}(P_c)$. Within this neighborhood, we fit a bi-variate B-spline surface $S(x, y)$ using:

$$S(x, y) = \sum_{i=0}^{n_x} \sum_{j=0}^{n_y} c_{ij} N_{i,p}(x) N_{j,p}(y) \qquad (3)$$

Where $N_{i,p}(x)$ and $N_{j,p}(y)$ are B-spline basis functions of degree $p = 3$, c_{ij} are control points determined through smoothing spline optimization ($s = 0.1k$) and $n_x = n_y = \min(\max(4, \lfloor k/4 \rfloor), 8)$ are the number of knots in each direction [11]. To generate a new point, P_{new}, we randomly select an x_i and y_i value within $\mathcal{N}(P_c)$ and evaluate $S(x_i, y_i)$. RGB values for new points are computed through distance-weighted interpolation of the neighborhood values. This method captures surface geometries and generates smoothly distributed points by incorporating higher-order geometric information from the local neighborhood.

3.2.2   Moving Least Squares (MLS) Upsampling

Our fourth approach employs MLS surface fitting to generate new points that accurately represent local surface geometry. For each new point, we randomly select an existing point P_c and identify its nearest neighbors ($k = 10$) to form a local neighborhood $\mathcal{N}(P_c)$. Within this neighborhood, we fit a local polynomial surface of degree $d = 2$ using:

$$f(x, y) = \sum_{i+j \leq d} a_{ij} x^i y^j \qquad (4)$$

Where a_{ij} are coefficients determined by solving the weighted least squares problem:

$$\min_{a_{ij}} \sum_{p \in \mathcal{N}(P_c)} w(p)(z_p - f(x_p, y_p))^2 \qquad (5)$$

Where weights $w(p)$ inversely proportional to the distance from P_c so that closer points contribute more heavily to the fit [12]. A new point, P_{new}, is generated by randomly sampling a position within the neighborhood region and projecting this position onto the fitted surface. The RGB value is interpolated from neighboring points using distance-weighted averaging. This method leverages a continuous surface model that adapts to local geometry to smooth point distributions and minimize distortions, especially in higher curvature regions, by capturing finer surface details.

3.2.3   Voronoi Upsampling

Our fifth approach utilizes Voronoi cell decomposition to generate new points that adapt to local point density [13]. For the original point cloud, we construct a Voronoi diagram $\mathcal{V}$ that partitions the space into cells, where each cell V_i contains all points closer to the generating point P_i than to any other point. To ensure bounded cells, we augment the point set with a bounding box extending 10% beyond the point cloud extent. For each Voronoi cell V_i, we compute its properties:

$$\begin{cases} c_i = \dfrac{1}{|V_i|} \sum_{v \in V_i} v & \text{(cell center)} \\[2ex] r_i = \max_{v \in V_i} |v - c_i| & \text{(maximum radius)} \\[2ex] vol_i = \mathbb{E}[|v - c_i|]^3 & \text{(approximate volume)} \end{cases} \qquad (6)$$

New points are generated preferentially in larger Voronoi cells, which represent sparsely sampled regions. For each new point, we select a cell V_i with probability proportional to its volume vol_i and generate a random point P_{new} within a sphere of radius $0.5r_i$ centered at c_i. The new point's attributes are interpolated from the $k = 5$ nearest neighbors. This method balances the point distribution by generating more points in sparse regions while preserving local geometric features through attribute interpolation.

3.3   Depth-Guided Point Lifting

Motivated by work from Chan et al. [14], we present a depth-guided point lifting approach that augments sparse SfM reconstructions while maintaining geometric accuracy. Our method synthesizes information from multiple sources: RGB images, their depth maps, and camera parameters estimated through COLMAP's SfM reconstruction [3]. For each image in the dataset, we transform 2D pixels into 3D points:

1. Scale Factor Computation: To ensure geometric consistency with original SfM points, we compute optimal scale factor (s) by minimizing the error between projected depth points and their corresponding SfM points:

$$s = \frac{\sum(p_{raw} \cdot (p_{orig} - c))}{\sum p_{raw}^2} \qquad (7)$$

Where p_{raw} are the raw projected points, p_{orig} are the original SfM points, and c is the camera center.

2. Point Generation: New 3D points are generated through the following steps:

   (a) Point Selection: We sample points from the central region of each image (50% of image dimensions) to ensure reliable depth measurements.

   (b) Coordinate Transformation: For each randomly selected pixel $p = (u, v)$, we convert

it to world coordinates:

$$p_w = t_{inv} + (R_{inv} \cdot K^{-1}[u, v, 1]^T) \quad (8)$$

Where K is the intrinsic camera matrix.

(c) Depth Integration: We compute the final 3D position by scaling the ray from camera center to projected point using the depth value and optimal scale factors:

$$P_{3D} = c_{world} + s \cdot (d \cdot \hat{v}) \quad (9)$$

Where c_{world} is the camera center in world coordinates, d is the depth value, and $\hat{v}$ is the unit vector from camera center to projected point.

3. Color Assignment: RGB values are sampled from the corresponding pixels in the source image, to maintain visual consistency with the input data.

This method combines geometric information from both SfM and depth maps, resulting in dense point clouds that maintain consistency with the original sparse reconstruction while adding detail in previously under-sampled regions. The use of a scale factor derived from existing SfM points ensures that new points are aligned with the global reconstruction, while the central region sampling strategy helps minimize the impact of depth estimation errors that are emphasized near image boundaries.

4   Results and Discussion

Following 3DGS [1], we utilize the Mip-NeRF360 dataset [15] to evaluate our interpolation and reconstruction upsampling methods and we report the novel view synthesis metric of PSNR. Additionally, we evaluate on 8 scenes of the Replica Dataset [16], which provides synthetic depth maps alongside RGB images.

4.1   Evaluation on Mip-NeRF360

We evaluate on Mip-NeRF360 [15] across different upsampling ratios (4x, 8x, and 16x). Points clouds produced by our methods are shown in Fig. 1. The results, presented in Table 1, reveal several key insights into the effectiveness of these methods for 3DGS initialization.

Across all scenes, our upsampling methods generally maintain or improve the PSNR compared to the baseline 3DGS results with an average increase of approximately 0.258 dB per scene. The MLS approach consistently demonstrates superior performance, achieving the highest PSNR in 5 out of 8 scenes at various upsampling ratios. For instance, in the *kitchen* scene, MLS
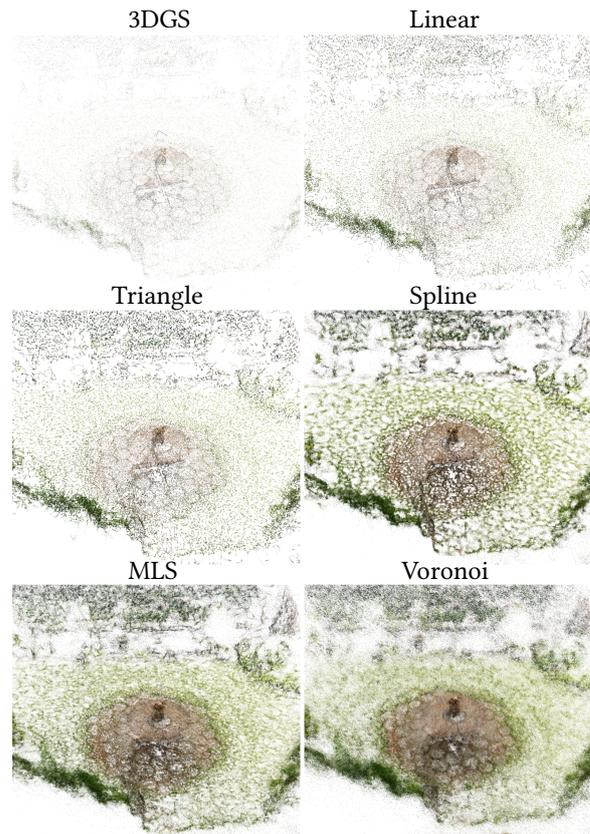
Figure 1: The 16x upscaled point clouds of *garden* from Mip-NeRF360 [15].

achieves a PSNR of 31.874 at 4x upsampling, representing a notable improvement over the baseline PSNR of 31.559. Interestingly, the simpler Linear and Triangle interpolation methods also show promise, particularly at lower upsampling ratios. For instance, the Linear method at 8x upsampling achieves a PSNR of 32.660, an improvement of 0.369 dB, on the *bonsai* scene.

4.1.1   Guidelines for Method Selection

Using the effectiveness of different upsampling methods across scenes, we derive guidelines for method selection based on scene characteristics, such as structural complexity and the presence of organic or planar forms. Firstly, in highly detailed scenes like *counter* and *room* surface reconstruction methods (e.g., Spline and MLS) tend to outperform simpler interpolation approaches, likely due to their ability to better capture subtle curvature and local detail. In scenes characterized by more organic content, like *flowers* and *bicycle*, MLS continues to show robust performance, suggesting its adaptability to irregular, natural shapes where preserving small-scale geometry is important. In contrast, for *treehill*, a more planar environment, the Voronoi method exhibits higher PSNR that all other methods, to a greater extent at lower upsampling ratios. This suggests that Voronoi's

Table 1: Upsampling Methods PSNR(↑) on Mip-NeRF360 Dataset [15].

Scene	3DGS	Linear			Triangle			Spline			MLS			Voronoi		
		4x	8x	16x	4x	8x	16x	4x	8x	16x	4x	8x	16x	4x	8x	16x
bicycle	25.106	25.270	25.261	25.225	25.248	25.262	25.236	25.163	25.160	25.188	25.227	**25.285**	25.281	25.143	25.120	25.182
bonsai	32.291	32.275	**32.660**	32.262	32.413	32.406	32.532	32.250	32.520	32.620	32.487	32.572	32.554	32.534	32.629	32.629
counter	29.112	29.228	29.280	29.285	29.292	**29.353**	29.301	29.199	29.284	29.334	29.226	29.310	29.322	29.194	29.259	29.227
flowers	21.382	21.455	21.507	21.426	21.490	21.496	21.445	21.469	21.462	21.537	21.520	21.608	**21.671**	21.437	21.514	21.547
garden	27.329	27.416	27.399	27.398	27.416	27.390	27.430	27.338	27.410	27.386	27.382	27.402	**27.485**	27.288	27.328	27.354
kitchen	31.559	31.719	31.525	31.777	31.796	31.812	31.534	31.442	31.755	31.294	**31.874**	31.626	31.852	31.625	31.596	31.171
room	31.620	31.452	31.599	31.634	31.740	31.596	31.789	31.879	31.733	**31.983**	31.865	31.873	31.915	31.772	31.878	31.951
stump	26.616	26.819	26.838	26.841	26.814	26.858	26.832	26.706	26.751	26.775	26.835	26.844	**26.864**	26.852	26.773	26.710
treehill	22.531	22.493	22.468	22.441	22.439	22.529	22.458	22.518	22.504	22.485	22.562	22.483	22.417	**22.689**	22.629	22.573

adaptive point distribution strategy is effective at handling flat, low-texture surfaces. The performance of the interpolation methods on the *bonsai* and *counter* scenes is a consequence of the prevalence of piecewise-smooth geometries, resulting from many object of regular geometries, tend to be over-smoothed by the surface reconstruction methods.

Thus, our findings demonstrate that thoughtful point cloud upsampling can enhance 3DGS reconstruction quality. The choice between interpolation and surface reconstruction methods should consider both scene characteristics and computational constraints. Simpler methods can sometimes achieve comparable results with lower computational overhead and lower upsampling ratios (4x or 8x) often provide a favorable trade-off between point cloud density and reconstruction quality.

4.2 Evaluation on Replica

To further validate our methods and evaluate the depth-guided point lifting approach, we conduct additional experiments on Replica [16]. A point cloud produced by the depth-guided approach is shown in Fig. 2. Replica's [16] scenes contains smaller initial points clouds (an average of 23000 points per scene compared to Mip-NeRF360's [15] 115000) so we investigate the effect of larger upsampling factors (16x and 32x). The results in Table 2 demonstrate that our methods generally improve the PSNR by an average or 0.646 db per scene.

Replica's *office* and *room* scenes contain similar objects and structure, resultantly we observe less variety in the methods with the highest PSNR. Most methods show improvement when scaling from 16x to 32x, suggesting more benefits from higher point densities. This contrasts with our findings on the Mip-NeRF360 [15] dataset, where benefits plateau earlier. The surface reconstruction methods demonstrate more stable performance across upsampling ratios, with consistent improvements at higher densities. This suggests their robustness in maintaining geometric fidelity even at higher upsampling rates.

3DGS   Depth-guided
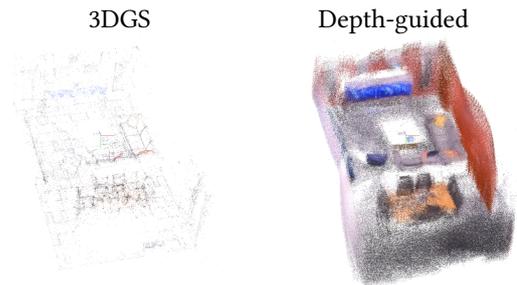
Figure 2: The 32x upscaled point clouds of *office3* from Replica [16].

4.2.1 Guidelines for Method Selection

Surface reconstruction methods (MLS and Voronoi) consistently outperform simple interpolation in all scenes. Again, as established with Mip-NeRF360 [15], the Voronoi method improves initialization for scenes containing planar, low-texture surfaces. MLS maintains higher PSNR likely due to its ability to handle the mixture of planar surfaces (walls, desks) and complex objects. Also, since these scenes contain much less objects than scenes like *counter*, so there are less opportunities for over-smoothing. The depth-guided point lifting approach shows continuous improvement at higher upsampling ratios and the most consistent performance across different scene types, suggesting that depth information provides valuable guidance for point placement. This method particularly excels in scenes with many low-texture objects at various depth levels, as evidenced by its performance in *office1* and *room2*. The performance of the depth-guided approach underscores the value of incorporating additional geometric information when available.

4.3 Training Time and Model Size

Analysis of computational results (Table 3) reveals efficiency trade-offs across our methods. Training times scale sub-linearly with upsampling ratio, with the baseline 3DGS requiring 1365s on Mip-NeRF360 [15] and geometric methods showing increases of up to 20% at

Table 2: Upsampling Methods PSNR(↑) on Replica [16].

Scene	3DGS	Linear		Triangle		Spline		MLS		Voronoi		Depth	
		16x	32x	16x	32x	16x	32x	16x	32x	16x	32x	16x	32x
office0	43.505	44.001	44.177	44.089	44.204	43.967	44.039	44.173	**44.373**	44.067	44.168	44.177	44.281
office1	41.126	41.117	41.205	41.046	41.301	41.284	41.377	41.538	41.689	41.280	41.437	41.812	**42.029**
office2	37.941	38.099	38.035	38.136	38.128	38.203	38.223	38.242	38.266	38.148	38.305	38.393	**38.406**
office3	37.495	37.852	37.881	37.892	37.837	37.970	38.087	38.041	**38.171**	37.899	38.002	37.728	37.888
office4	38.902	39.186	38.962	38.968	39.131	38.912	39.032	39.288	**39.534**	39.244	39.309	39.222	39.070
room0	37.812	38.037	38.059	38.057	38.063	38.032	38.104	38.095	**38.160**	38.027	38.030	37.891	37.948
room1	38.965	38.983	38.945	39.125	39.121	39.320	39.410	39.272	39.214	39.313	**39.519**	39.233	38.876
room2	39.642	39.846	39.981	39.964	40.009	39.918	39.928	39.972	40.040	39.991	40.053	40.143	**40.364**

Table 3: Average Training Time and Model Size for Upsampling Methods on Mip-NeRF360 [15] and Replica [16].

Metric	3DGS	4x						8x						16x					
		Lin.	Tri.	Spl.	MLS	Vor.	Dep.	Lin.	Tri.	Spl.	MLS	Vor.	Dep.	Lin.	Tri.	Spl.	MLS	Vor.	Dep.
Mip-NeRF360																			
• Time (s)	1365	1365	1370	1409	1433	1519	N/A	1430	1438	1475	1503	1541	N/A	1575	1582	1620	1635	1657	N/A
• Size (MB)	752	804	806	777	760	733	N/A	857	857	807	785	758	N/A	960	940	870	828	807	N/A
Replica																			
• Time (s)	487	502	506	523	540	569	714	549	536	559	568	605	830	607	606	627	634	683	930
• Size (MB)	211	234	234	217	219	219	219	261	256	232	233	233	252	319	307	268	264	263	319

16x upsampling. The depth-guided approach demonstrates the highest computational cost on Replica [16] (930s at 16x vs 487s baseline), reflecting the overhead of optimizing 3DGS with many promising points. Model size scales approximately linearly with upsampling ratio, though surface reconstruction methods show better efficiency than interpolation-based approaches. At 16x upsampling on Mip-NeRF360 [15], while Linear interpolation required 960 MB (compared to 752 MB baseline), MLS and Voronoi maintain smaller footprints of 828 MB and 807 MB respectively. Notably, these trends are consistent across both datasets, with Replica showing proportionally similar scaling at lower absolute values due to reduced scene complexity. These findings suggest that while our methods introduce computational overhead, they maintain efficiency even at higher upsampling ratios, with surface reconstruction approaches offering the most favorable balance between quality improvement and resource utilization.

4.4 Ablation Studies

We conduct an ablation study comparing our methods against random point sampling within the scene bounding box. We generate points by uniformly sampling XYZ coordinates within the scene bounds and randomly assigning RGB values from the initial cloud. The results show significantly degraded performance compared to all proposed upsampling methods. In Mip-NeRF360 [15], random sampling achieves PSNRs consistently 0.2 to 0.7 dB less than the original 3DGS initialization, especially at higher upsampling factors. This poor performance can be attributed to its failure to preserve local surface structure and color consistency, leading to noise in the initial Gaussian distribution that the optimization struggles to correct. This finding underscores that successful point cloud augmentation requires methods that respect the underlying scene geometry and maintain local color consistency, validating our approach of using geometry-aware upsampling strategies.

5 Conclusion

Our evaluations reveal that thoughtful augmentation of initial point clouds consistently improves reconstruction quality, with different methods showing distinct advantages based on scene characteristics. As a guideline, surface reconstruction methods excel in organic, detailed environments, while the Voronoi-based approach show strength in structured, planar scenes. Particularly, MLS demonstrates robust performance across varied environments, while simpler interpolation methods prove effective for scenes with many piecewise-smooth geometries. Our depth-guided approach shows particular promise to introduce scene wide details, particularly in low-texture regions that are missed by SfM. While computational overhead increases with upsampling ratio, the benefits to reconstruction quality often justify the additional resources. Future work might explore adaptive combinations of methods based on local scene geometry and traditional point cloud densification techniques could offer more performance gains. Learning-based approaches present an intriguing avenue, particularly given 3DGS's unique characteristic of concentrating Gaussians around high-frequency regions.

References

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, July 2023. [Online]. Available: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[2] J. Lu, Y. Zhang, Q. Shen, X. Wang, and S. Yan, "Poison-splat: Computation cost attack on 3d gaussian splatting," 2024. [Online]. Available: https://arxiv.org/abs/2410.08190

[3] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] H. Lim, H. Chang, J. B. Choi, and C. M. Yeum, "Lidar-3dgs: Lidar reinforced 3d gaussian splatting for multimodal radiance field rendering," 2024. [Online]. Available: https://arxiv.org/abs/2409.16296

[5] S. Ha, J. Yeon, and H. Yu, "Rgbd gs-icp slam," 2024. [Online]. Available: https://arxiv.org/abs/2403.12550

[6] Z. Fan, W. Cong, K. Wen, K. Wang, J. Zhang, X. Ding, D. Xu, B. Ivanovic, M. Pavone, G. Pavlakos, Z. Wang, and Y. Wang, "Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds," 2024.

[7] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," 2024. [Online]. Available: https://arxiv.org/abs/2312.14132

[8] K.-C. Chan, J. Xiao, H. L. Goshu, and K.-M. Lam, "Point cloud densification for 3d gaussian splatting from sparse input views," in *Proceedings of the 32nd ACM International Conference on Multimedia*, ser. MM '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 896–904. [Online]. Available: https://doi.org/10.1145/3664647.3681454

[9] S. Seibt, T. V. S.-L. Chang, B. von Rymon Lipinski, and M. E. Latoschik, "Dense 3D Gaussian Splatting Initialization for Sparse Image Data," in *Eurographics 2024 - Posters*, L. Liu and M. Averkiou, Eds. The Eurographics Association, 2024.

[10] Y. S. Hu, D. Mao, Y. Chen, and J. Zelek, "Towards real-time gaussian splatting: Accelerating 3dgs through photometric slam," 2024. [Online]. Available: https://arxiv.org/abs/2408.03825

[11] L. Piegl and W. Tiller, *The NURBS book (2nd ed.)*, ser. Monographs in Visual Communication. Berlin, Heidelberg: Springer-Verlag, 1997, p. 100–110.

[12] G. E. Fasshauer, "Approximate moving least-squares approximation: A fast and accurate multivariate approximation method," in *Curve and Surface Fitting*, A. Cohen, J.-L. Merrien, and L. L. Schumaker, Eds. Saint-Malo 2002: Nashboro Press, 2003, pp. 139–148. [Online]. Available: https://api.semanticscholar.org/CorpusID:10669315

[13] W. Y. Chang, "Surface reconstruction from points," UC San Diego, Department of Computer Science & Engineering, Technical Report CS2008-0922, may 2008, peer-Reviewed. [Online]. Available: https://escholarship.org/uc/item/9xv8x6n6

[14] K.-C. Chan, J. Xiao, H. L. Goshu, and K. man Lam, "Point cloud densification for 3d gaussian splatting from sparse input views," in *ACM Multimedia 2024*, 2024. [Online]. Available: https://openreview.net/forum?id=vW5070FoXi

[15] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," *CVPR*, 2022.

[16] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe, "The Replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.