Surf-LEAF: An Urban Environment Gaussian Splatting Mesh Extraction Pipeline

Jannick Schröer Rico-Marcel Benning Landson Guo Qinglin Chen
Department of Computer Science, ETH Zurich
{jschroeer, rbenning, languo, qinglchen}@ethz.ch

Abstract

Gaussian Splatting is a recent point-based scene representation that yields real-time, high-fidelity novel-view synthesis across wide scenes. The extraction of meshes from this representation would enable compatibility with a wide range of animation, simulation, and urban modeling tasks. Yet, in urban scenes, the resulting meshes suffer on sharp corners, low-contrast surfaces, and large-scale geometry, which often lead to artifacts such as floaters and over-smoothed meshes.

We present a custom mesh extraction pipeline designed specifically for urban settings. Our approach begins with a surface-level point cloud extraction step optimized to preserve structural detail, followed by an advancing front surface reconstruction algorithm that robustly handles nonuniform point densities and supports scalable processing. Finally, post-processing is applied to repair artifacts and ensure mesh consistency. We evaluated our method on synthetic (TartanAir) and real-world (Nerfstudio) datasets, showing that it achieves higher mesh fidelity and structural preservation compared to existing techniques. Furthermore, our method reduces the Chamfer distance by around 10 % over the strongest baseline (DN-Splatter) on TartanAir.

1. Introduction

Accurate 3D meshes are essential components of animation, simulation, autonomous navigation, and digital urban planning pipelines [7, 20, 21], where they serve as the geometric backbone for physical interactions, rendering, and environmental understanding in these systems. Mesh reconstruction provides a crucial role in these applications, providing the geometric foundation for rendering, physics-based interaction, and spatial reasoning [15]. As a result, generating clean, accurate, and scalable meshes is critical for downstream tasks that rely on structural fidelity and topological consistency.

Recent advances in 3D scene representation have seen the emergence of Gaussian Splatting, a point-based rendering technique that models scenes as collections of anisotropic gaussians rather than explicit geometry [11]. This method achieves high-quality novel view synthesis, enabling real-time displacement in a scene with even certain extensions allowing real-time scene manipulation [9]. However, despite its strengths in image generation, Gaussian Splatting lacks direct support for mesh extraction. This gap limits its use in applications that require geometric reasoning, physical simulation, or spatial planning [7, 20, 21].

The need for reliable mesh extraction is especially pronounced in urban environments, where traditional methods often fail to cope with key challenges. These include the presence of sharp corners, large flat surfaces, low-contrast textures (such as roads and building facades), non-uniform point sampling, and noise [21]. Moreover, urban-scale scenes are typically vast and complex, creating bottlenecks in memory and computation for many mesh reconstruction algorithms [17]. Existing techniques either over-smooth important features or break down at scale, producing low-resolution surfaces, floaters, or incomplete geometry [20].

In this report, we present **Surf-LEAF** (**Surf**ace-**L**evel **E**xtraction with **A**dvancing-**F**ront reconstruction), a novel mesh extraction pipeline designed specifically to overcome these challenges in the context of outdoor urban scenes. Our method starts by extracting a surface-level point cloud [7] from the Gaussian splats, focusing on filtering out noise. We then apply the Advancing Front Surface Reconstruction algorithm [5], which operates on a 3D Delaunay triangulation and incrementally constructs the mesh using a priority queue based on geometric criteria such as circumsphere radius and overlap. The algorithm supports a flexible triangle insertion strategy comprising of propagation, gluing, hole filling, and ear filling that adapts to complex local geometries while ensuring mesh continuity. It also naturally chunks large scenes in order to process them without running out of memory. A final post-processing step refines the mesh by filling holes, producing visually coherent surfaces.

This work aims to close the gap between modern point-

based rendering methods and traditional mesh-based applications, enabling the use of splatted scenes not only for visualization but also for geometry-aware simulation, planning, and interaction.

2. Related Works

Novel view synthesis methods. Neural Radiance Fields (NeRFs) [16] have revolutionized 3D scene reconstruction by enabling photorealistic novel view synthesis from sparse 2D images. However, extracting explicit mesh representations from NeRFs pose significant challenges due to their implicit volumetric nature, which lacks well-defined surfaces.

Traditional mesh extraction methods, such as Marching Cubes [15], are not directly applicable to NeRFs because they rely on explicit surface definitions. To address this, researchers have developed alternative approaches. For example, NeRFMeshing [20] introduces a Signed Surface Approximation Network (SSAN) that distills a NeRF into a geometrically accurate 3D mesh. This method calculates a Truncated Signed Distance Function (TSDF) [17] and a feature appearance field, allowing the extraction of 3D meshes that are compatible with real-time rendering on various devices.

Gaussian Splatting (3DGS) [11] has emerged as a real-time novel view synthesis technique and a compelling alternative to NeRF. Its original implementation, as well as the CUDA-accelerated variant gsplat [27], have gained significant traction in both research and industry due to their rendering speed and quality, often significantly outperforming their NeRF counterparts [8]. The integration of gsplat into the Nerfstudio framework via Splatfacto [22] has further improved accessibility, allowing rapid experimentation and deployment.

Recent works have proposed various extensions to the standard Gaussian Splatting pipeline to improve reconstruction quality, robustness, and generalization. DN-Splatter [23] and QED-Splatter enhance learning by incorporating depth supervision loss, which helps guide the optimization of 3D Gaussians towards geometrically consistent structures. These methods demonstrate improved performance, particularly in sparse or noisy input settings.

2DGS [8] proposes a flattening of the gaussians for better handling of normals and depth. WildGaussians [12] and Splatfacto-W [26] extend the applicability of splatting techniques by introducing a robust pipeline for uncontrolled, in-the-wild image collections, addressing issues like inconsistent lighting, human occlusions, and background gaussians that introduce inconsistent depth images.

Mesh extraction from Gaussian Splats. While Gaussian splatting excels at fast photorealistic rendering, extracting coherent mesh geometry from millions of unstructured gaussians remains challenging. SuGaR [7] introduces a surface alignment regularizer that encourages gaussians to lie on the surfaces of the scene. It then samples points from this refined distribution and uses Poisson reconstruction [10] to obtain detailed, watertight meshes.

DN-Splatter [23] (with its AGS-Mesh extension [21]) uses a similar approach. By guiding gaussians to align with the surface through depth and normal loss, the method also enables direct mesh extraction through Poisson reconstruction.

A commonly used strategy for extracting meshes from Gaussian splats involves rendering depth maps of the scene and fusing them using a Truncated Signed Distance Function (TSDF) [17]. This approach is used by 2DGS [8], producing accurate and view-consistent meshes with controllable resolution and masking. TSDF fusion is not limited to 2DGS and can also be broadly applied to other Gaussian splatting methods.

Mesh extraction for Urban LiDAR scans. Early urban LiDAR approaches typically employ global reconstruction techniques such as Poisson surface reconstruction [10]. Poisson methods are resilient to sensor noise and are well-suited to structured environments but can over-smooth fine details and require careful trimming to avoid artifacts in regions of low point density.

Delaunay-based reconstruction methods offer an alternative, constructing a 3D Delaunay tetrahedralization of the input points and labeling tetrahedra as interior or exterior based on visibility, graph cuts, or other global priors. These techniques can produce high-quality, topologically consistent surfaces, especially in urban environments where planar structures are common. Within this family, the Advancing Front Reconstruction algorithm [5] provides a more local and incremental strategy: rather than classifying tetrahedra globally, Advancing Front Reconstruction traverses the Delaunay complex by incrementally adding triangles to a growing surface front. This greedy, front-propagating behavior allows it to produce clean, manifold meshes and avoid overfitting to noise.

More recent work, such as ImMesh [14], has addressed these challenges by tightly integrating LiDAR SLAM and mesh reconstruction. ImMesh partitions the scene into a voxel grid and applies fast, local 2D triangulation within each cell using a pull-commit-push strategy, producing explicit triangle meshes in real time.

Hole filling. Polygon meshes reconstructed from real-world scans or generative pipelines often suffer from topological flaws such as holes, gaps, and fragmentation, caused by occlusions, tessellation artifacts, or geometric misalignments. Numerous techniques have been proposed to address these defects through hole filling and gap closing.

For simple holes bounded by well-defined edge loops, Liepa [13] proposed a fair triangulation scheme with smoothness priors, while Zhao et al. [28] introduced

an advancing-front meshing strategy followed by Poisson-based vertex optimization to produce visually coherent patches. Podolak and Rusinkiewicz [18] extended this approach to a volumetric setting by applying graph-cut optimization over tetrahedral partitions to generate watertight and self-intersection-free fillings.

To close narrow gaps between adjacent surface patches, proximity-based zippering methods have been developed. Barequet and Kumar [1] and Borodin et al. [4] aligned nearby boundary curves and progressively merged them, while Bischoff and Kobbelt [3] employed voxel-based detection of defective regions followed by local patch regeneration to improve robustness.

Although these approaches perform well on moderately clean inputs, they struggle with irregular hole topologies, ambiguous boundary configurations, and heavily fragmented geometry. Moreover, these methods lack robustness guarantees and often fail to produce valid, watertight outputs when operating on noisy or non-manifold meshes.

Alpha Wrapping [19] offers a compelling alternative for such challenging scenarios. As a reconstruction technique grounded in 3D alpha complex theory, it incrementally constructs a subcomplex from the Delaunay triangulation of the input, filtering out topological artifacts. It guarantees manifoldness and geometric validity of the output, making it particularly well-suited for aggressive and reliable hole and gap closure in complex mesh scenarios.

3. Method

The Surf-LEAF pipeline, shown in Figure 1, is compatible with any existing Gaussian Splatting algorithm for the splatting component. For our evaluations, we use QED-Splatter as the Gaussian Splatting method of choice due to its prior demonstrated accuracy in outdoor environments, using RGB-D images as input.

A level set of points lying on the surface of the scene is then extracted using the same methodology as in the SuGaR mesh extraction pipeline [7]. That is, depth maps are generated from the gaussians at each training viewpoint. For each pixel in each depth map, a set of points $p_i = p + t_i v$ are randomly sampled, where p is the reprojected pixel location of the point, v is the line of sight, and $t_i \in [-3\sigma_g(v), 3\sigma_g(v)]$ is a sample from the 1D Gaussian slice located along the direction v. Given a level set threshold λ, if there exists i, j such that $p_i < \lambda < p_j$, then the level set point is determined via linear interpolation between p_i and p_j.

Following the extraction of the level-set point cloud using the SuGaR-inspired approach, we generate a structured surface mesh using the Advancing Front Surface Reconstruction algorithm [5]. This algorithm incrementally constructs a triangulated manifold based on a surface-focused Delaunay triangulation of the input points. Unlike volumetric methods that rely on implicit representations or uniform sampling assumptions, this algorithm selects triangles sequentially in a greedy manner to build a coherent mesh.

The process begins by identifying the Delaunay triangle with the smallest circumsphere radius, which serves as the initial seed for the mesh. This triangle defines the starting boundary, called the advancing front. From this boundary, the candidate triangles incident to the current front are evaluated and added iteratively.

Candidate triangles are managed in a priority queue, ranked according to a plausibility score. This score balances geometric fidelity and mesh quality by considering three main factors: the circumsphere radius (favoring locally dense sampling), the dihedral angle between adjacent triangle normals (to encourage smooth surfaces), and internal angle constraints to avoid degenerate configurations.

To ensure the topological validity of the growing mesh, only triangles that meet the manifold constraints are selected. These fall into four categories based on their relation to the current front: extension, hole filling, ear filling, and gluing. Each operation modifies the front while preserving the orientability and manifoldness of the resulting surface. The algorithm continues until no valid candidate triangles remain.

The advancing front approach is particularly well suited for our use-case, as it keeps structural details, works with non-uniform sampling, and thrives when large planar surfaces are common. It produces clean, high-quality meshes that are geometrically faithful to the input point cloud, though it does not guarantee water-tightness in all cases. This results in a mesh that has holes in areas with insufficient point cloud density.

To tackle this issue, we apply post-processing to our output mesh. However, given the prevalence and irregular shapes of the holes resulting from the Advancing Front algorithm, conventional hole-filling techniques are insufficient for comprehensive detection and repair. Therefore, we adopt a more robust method: Alpha Wrapping [19]. This approach iteratively constructs a subcomplex derived from a 3D Delaunay triangulation, starting with an initial enclosing triangulation and progressively removing boundary tetrahedra. Alpha Wrapping eliminates internal structures, fine details, and cavities, consistently producing a valid, oriented, 2D manifold meshes free of self-intersections, even from highly fragmented input meshes.

Figure 2 illustrates the complete mesh-fixing pipeline. Initially, isolated mesh components are removed, followed by a two-step smoothing procedure [2] to generate a clean and uniform base mesh suitable for Alpha Wrapping. Subsequently, Alpha Wrapping effectively resolves remaining holes and gaps. Finally, HC Laplacian Smoothing [24] and mesh simplification via Quadric Edge Collapse Decimation [6] are conducted to achieve smoother surfaces and enhance storage efficiency.

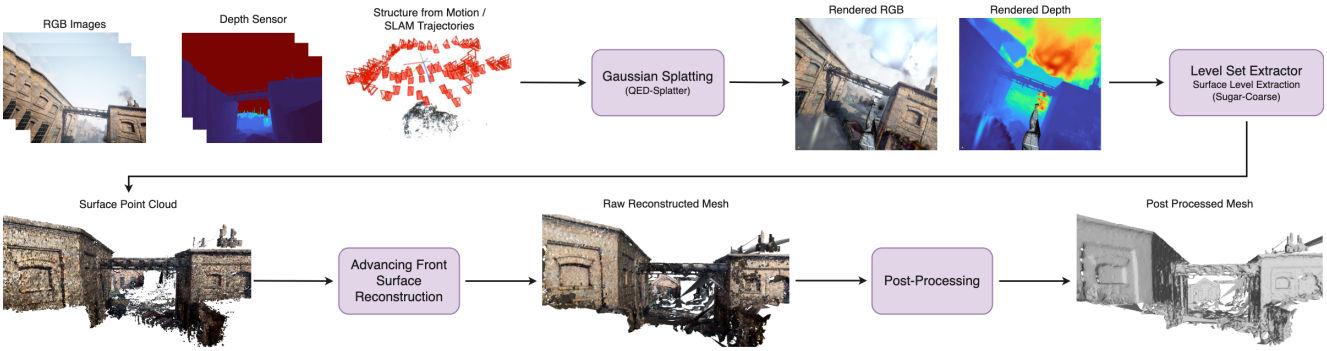

Figure 1. Pipeline Overview of Surf-LEAF

Method	Input	Meshing Algorithm Used
SuGaR [7]	RGB-D Images	Poisson (level-set)
DN-Splatter [23]	Gaussians	Poisson direct
2DGS [8]	RGB-D Images	TSDF
TSDF [17]	Depth	TSDF
Ours	RGB-D Images	Adv. Front + Post Processing

Table 1. Comparison of the meshing algorithms tested

Method	Chamfer Distance ↓	Vertices ↑	Meshing Duration ↓
SuGaR [7]	0.6189	241,941	1,761.702s
DN-Splatter [23]	0.5875	408,994	99.307s
2DGS [8]	0.7441	286,331	**72.053s**
TSDF [17]	0.9011	243,235	478.685s
Ours (Raw)	**0.5277**	**457,506**	1,816.771
Ours (Post Proc.)	0.5326	284,128	1,952.432

Table 2. Quantitative comparison of mesh extraction methods on the TartanAir Abandoned Factory P001 scene

In practice, the Alpha Wrapping-based pipeline successfully resolves most holes, producing clean, topologically sound meshes. Note that the method can be computationally intensive, requiring a trade-off between the effectiveness of hole repair and the preservation of sharp features — a detailed analysis of these trade-offs is provided in the appendix 6.3.

4. Results

4.1. Qualitative Result

Figure 3 presents a qualitative comparison of mesh extraction methods evaluated on the TartanAir [25] Abandoned Factory P001 scene and the Nerfstudio Floating Tree scene [22]. Each scene was converted into gaussians using QED-Splatter, and each mesh extraction algorithm was run using outputs extracted from the same Gaussian splats. In order to reuse the same results from the QED-Splatter scene, we employed the DN-Splatter [23] implementation for all meshing algorithms listed in Table 1.

It is important to note that RGB-D data is used as input for QED-Splatter in the TartanAir Abandoned Factory scene, with notably no depth noise due to it being a synthetic dataset. For the Nerfstudio Floating Tree scene, only RGB data was used as input for QED-Splatter due to no depth data being provided with the dataset.

4.2. Quantitative Result

Table 2 presents a quantitative comparison of mesh extraction methods evaluated on the TartanAir [25] Abandoned Factory P001 scene. Since TartanAir does not provide a public ground truth point cloud, a point cloud was obtained by sampling from the ground truth depth images. A scaling factor of 0.0417 was applied to the ground truth point cloud, matching the scaling factor applied by Nerfstudio [22] while training to normalize the output on the Abandoned Factory P001 scene. Chamfer distance was then computed using a 10, 000-point sample from each mesh, applying global registration using RANSAC followed by ICP to align the point clouds.

Meshing durations were measured by running each pipeline and mesh extraction algorithm with the following computer specifications: AMD Ryzen 7 3800X, 32 GB RAM, NVIDIA RTX 4060 Ti (16 GB VRAM). Note: the measurement does not include the time needed for Gaussian Splatting.

5. Discussion

5.1. Qualitative Result

As seen in 3, our method produces visually clean and well-structured meshes, with smooth surfaces and strong preservation of large planar regions such as building facades and ground roads. Fine details are retained even in cluttered areas, and sharp edges are reconstructed with minimal smoothing. Due to the inherent limitations of the Advancing Front algorithm [5], which does not guarantee water tightness, some holes remain in the resulting meshes. How-

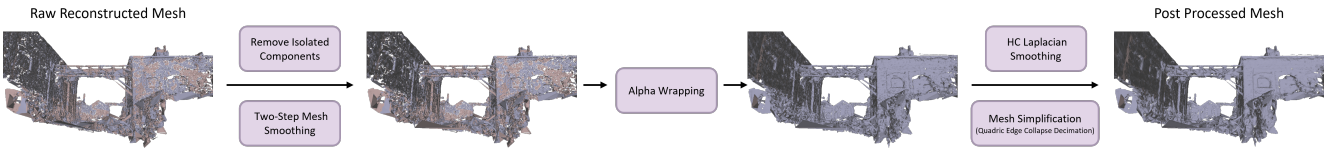

Figure 2. Pipeline overview of post-processing. The color of mesh triangular faces indicates the normal direction.

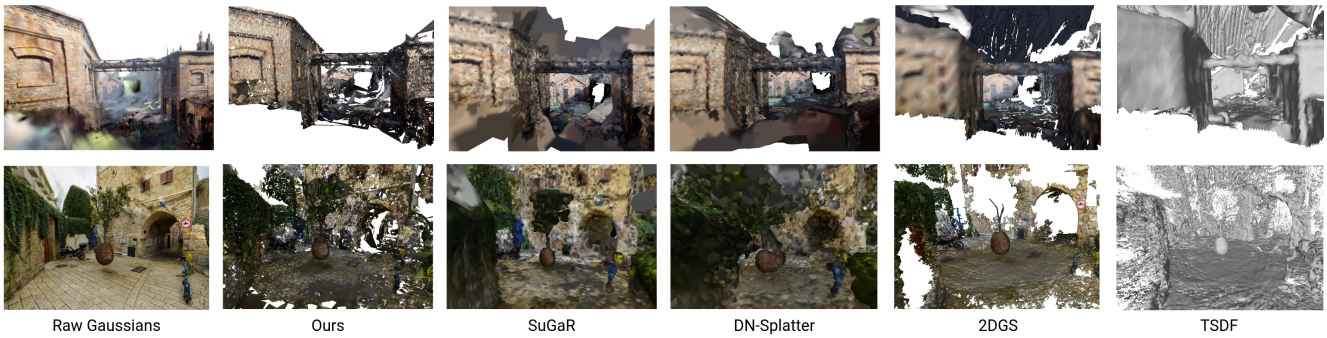

Figure 3. Qualitative results for the TartanAir AbandonedFactory P001 scene [25] (top) and the Nerfstudio Floating Tree scene (bottom) [22]

ever, our post-processing algorithm can effectively fill in these holes, removing the piecewise nature of the raw Advancing Front mesh while preserving detail, fine edges, and sharp corners.

SuGaR and DN-Splatter [7, 23] also produce relatively clean surfaces, but struggle with complex geometry, such as thin or disconnected structures. For example, the pipe running between buildings in the TartanAir [25] scene is incorrectly meshed as a solid structure. Additionally, their output surfaces tend to be bumpier, lacking the geometric regularity achieved by our method.

In contrast, TSDF methods [8, 17] exhibit significant artifacts. Both methods suffer from issues related to skybox inclusion, leading to large, unwanted geometry that distorts the mesh. Moreover, these voxel-based methods are highly sensitive to the voxel size: setting it too low causes loss of structural detail, while increasing it introduces memory bottlenecks and crashes in large-scale scenes like this one.

5.2. Quantitative Result

Our method outperforms all baselines in reconstruction accuracy, achieving the lowest Chamfer distance of 0.5277 with the raw mesh and 0.5326 with the processed mesh, indicating a closer geometric fit between the reconstructed mesh and the ground truth surface. In comparison, DN-Splatter [23], the next-best method, reports a higher Chamfer distance of 0.5875, while other baselines like SuGaR [7] and TSDF [8, 17] exhibit significantly larger errors. In particular, the TSDF-based algorithms suffer from significant artifacts resulting from the skybox, which end up greatly increasing the Chamfer Distance score.

In terms of mesh complexity, our pipeline generates the densest surface with 457,506 vertices. However, many of these appear to be redundant as after post-processing our final mesh has a vertex count of 284,128. Although DN-Splatter and 2DGS [8, 23] produce dense meshes, their higher Chamfer distances suggest that increased vertex count does not necessarily translate into better geometric fidelity without effective surface filtering and reconstruction strategies.

Curiously, our post-processing algorithm ends up increasing the Chamfer distance, albeit slightly. This is likely due to minor inaccuracies in the way the holes are filled that do not perfectly align with the underlying surface, resulting in a slight Chamfer distance increase (about a 0.01% increase). However, this is a reasonable trade-off as it significantly increases the visual quality of our mesh, and our approach still remains the best quantitatively out of the algorithms tested.

Regarding runtime, our method is the most computationally intensive, with a total meshing duration of 1,952.432 seconds and an unprocessed runtime of 1,816.771 seconds. This is largely due to our pipeline building on SuGaR's surface-level point cloud extraction [7], which we retain for its robustness. However, unlike SuGaR, which employs Poisson surface reconstruction [10], we apply the Advancing Front algorithm [5] to generate the final mesh. This choice introduces additional computational overhead due to the geometric reasoning involved in Delaunay triangulation and adaptive triangle insertion, but it results in improved mesh accuracy and structural continuity. While DN-Splatter and 2DGS [8, 23] are considerably faster (99.307s

and 72.053s, respectively), they compromise on reconstruction quality. SuGaR and TSDF [7, 17] also require substantial computation time (1,761.702s and 478.685s), yet they under-perform in both Chamfer distance and mesh resolution when compared to our approach.

Overall, the results demonstrate that our method delivers superior geometric accuracy and mesh detail at the cost of increased computational time. This tradeoff is particularly justified in applications where structural precision and topological integrity are critical, such as autonomous navigation, simulation, or digital twin construction in urban environments.

6. Conclusion

We have introduced **Surf-LEAF**, a dedicated mesh-extraction pipeline that bridges 3D Gaussian Splatting with the structural meshing demands of urban-scale applications. By addressing issues such as sharp edges, low-contrast facades, and uneven sampling, our method turns splatted scenes into reliable geometric models suitable for simulation and planning. Our pipeline mainly introduced the following advances: By combining Surface-Level Point Extraction [7] with Advancing-Front Surface Reconstruction [5], we are able to retain geometric detail such as planes and sharp corners, while still discarding floaters. Second, by adding a post-processing step, we are able to enforce consistency and substantially reduce visible cracks without incurring the heavy smoothing of volumetric methods.

6.1. Empirical Performance

Quantitative and qualitative evaluations on both synthetic (TartanAir) and real-world (Nerfstudio) datasets demonstrate that our method outperforms state-of-the-art baselines in terms of geometric accuracy, achieving the lowest Chamfer distance (≈ 0.533) among the various meshing algorithms tested while qualitatively performing much better at retaining sharp corners and fine details.

6.2. Trade-offs and Limitations

The geometric accuracy comes at the cost of a longer reconstruction time (≈ 2000 s), and the advancing-front strategy does not guarantee watertightness in extremely sparse regions. The method also assumes reasonably accurate depth during gaussian generation.

6.3. Future Work

Looking ahead, we identify two main areas for improvement.

1. A more parallelized implementation of the Surface-Level Extraction [7] could drastically speed up the pipeline.
2. Incorporating Gaussian splatting models with traits from Splatfacto-W and WildGaussians [12, 26] could improve the reconstruction accuracy in our intermediate step.

We hope the insights and open-source release of Surf-LEAF will catalyze faster, richer, and more versatile 3D reconstruction pipelines to enable broader research and next-generation applications.

References

[1] Gershon Barequet and Sariel Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. In *Proc. of the Tenth ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 82–91, 1997. 3

[2] Alexander Belyaev and Yutaka Ohtake. A comparison of mesh smoothing methods. In *Israel-Korea Bi-national conference on geometric modeling and computer graphics*. Citeseer, 2003. 3

[3] Sebastian Bischoff and Leif Kobbelt. Structure preserving cad model repair. In *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling*, pages 3–12. ACM, 2005. 3

[4] Pavel Borodin, Tatiana L. Kunii, and Kiyoshi Koyamada. Fast and robust mesh zippering for hole filling. *The Visual Computer*, 18(4):233–242, 2002. 3

[5] Tran Kai Frank Da and David Cohen-Steiner. Advancing front surface reconstruction. In *CGAL User and Reference Manual*. CGAL Editorial Board, 6.0.1 edition, 2024. Package: `Advancing front surface reconstruction`. 1, 2, 3, 4, 5, 6

[6] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, 1997. 3

[7] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. ArXiv preprint 2311.12775. 1, 2, 3, 4, 5, 6

[8] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH '24 Conference Papers*, pages 1–11. ACM, 2024. 2, 4, 5

[9] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, and Chenfanfu Jiang. Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality, 2024. 1

[10] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, page 61–70, Goslar, DEU, 2006. Eurographics Association. 2, 5

[11] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time

radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 1, 2

[12] Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. Wildgaussians: 3d gaussian splatting in the wild. *NeurIPS*, 2024. 2, 6

[13] Peter Liepa. Filling holes in meshes. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 200–205, 2003. 2

[14] Jiarong Lin, Chongjian Yuan, Yixi Cai, Haotian Li, Yunfan Ren, Yuying Zou, Xiaoping Hong, and Fu Zhang. Immesh: An immediate lidar localization and meshing framework. *IEEE Transactions on Robotics*, 39(6):4312–4331, 2023. 2

[15] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, page 163–169, New York, NY, USA, 1987. Association for Computing Machinery. 1, 2

[16] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[17] Helen Oleynikova, Zachary Taylor, Marius Fehr, Juan Nieto, and Roland Siegwart. Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017. 1, 2, 4, 5, 6

[18] Joshua Podolak and Szymon Rusinkiewicz. A planar-reflective symmetry transform for 3d shapes. In *ACM SIGGRAPH 2005 Papers*, pages 549–559. ACM, 2005. 3

[19] Cédric Portaneri, Mael Rouxel-Labbé, Michael Hemmer, David Cohen-Steiner, and Pierre Alliez. Alpha wrapping with an offset. *ACM Trans. Graph.*, 41(4), 2022. 3

[20] Marie-Julie Rakotosaona, Fabian Manhardt, Diego Martin Arroyo, Michael Niemeyer, Abhijit Kundu, and Federico Tombari. Nerfmeshing: Distilling neural radiance fields into geometrically-accurate 3d meshes, 2023. 1, 2

[21] Xuqian Ren, Matias Turkulainen, Jiepeng Wang, Otto Seiskari, Iaroslav Melekhov, Juho Kannala, and Esa Rahtu. Ags-mesh: Adaptive gaussian splatting and meshing with geometric priors for indoor room reconstruction using smartphones, 2024. 1, 2

[22] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 2, 4, 5

[23] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing, 2024. 2, 4, 5

[24] Jörg Vollmer, Robert Mencl, and Heinrich Mueller. Improved laplacian smoothing of noisy surface meshes. In *Computer graphics forum*, pages 131–138. Wiley Online Library, 1999. 3

[25] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer. Tartanair: A dataset to push the limits of visual slam. In *CoRL 2020*, 2020. 4, 5

[26] Congrong Xu, Justin Kerr, and Angjoo Kanazawa. Splatfacto-w: A nerfstudio implementation of gaussian splatting for unconstrained photo collections, 2024. 2, 6

[27] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025. 2

[28] Xing Zhao, Paul Wonka, and Niloy J. Mitra. Closing the loop: Hole filling by loop optimization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):890–901, 2007. 2

Surf-LEAF: An Urban Environment Gaussian Splatting Mesh Extraction Pipeline

Supplementary Material

7. Hyper Parameters in Alpha Wrapping

Alpha Wrapping operates by constructing a subcomplex of a 3D Delaunay triangulation, effectively forming a watertight surface approximation of an input mesh. Two crucial hyperparameters influence this algorithm: *Alpha*, which defines the maximal allowable radius of the circumsphere for triangles in the output mesh, and *Offset*, specifying the maximal allowable distance between the input surface and the generated mesh.

Figure 4 presents a detailed analysis of how variations in *Alpha* and *Offset* parameters impact the resulting mesh quality and hole filling performance. As illustrated, smaller *Alpha* and *Offset* values produce meshes that tightly conform to the original geometry, improving geometric fidelity. However, excessively small values can prevent the algorithm from bridging gaps or repairing holes, leading to incomplete surfaces. At the same time, they significantly increase mesh complexity (number of faces) and computational overhead (elapsed time). Conversely, excessively large parameter values lead to oversmoothing, compromising geometric fidelity and resulting in loss of detail.

To achieve an optimal balance between effective hole filling, computational efficiency, and preservation of geometric detail, we selected parameters $\mathit{Alpha} = \frac{1}{1{,}000}$ and $\mathit{Offset} = \frac{1}{2{,}500}$, as highlighted in Figure 4. These parameters demonstrated the most favorable trade-off, effectively filling holes while preserving structural intricacies without excessive smoothing or computational cost.

8. Source Code and Example Data

Our source code can be found in our Github Repository, together with a guide on generating your first mesh with our pipeline using the TartanAir Dataset.

9. Work Distribution

- **Jannick Schröer** Euler Nerfstudio Setup; Research into existing meshing pipelines; Testing and Eval of TSDF; SuGaR and DN-Splatter; Customization of QED-Splatter for Meshing; Modification of DN-Splatter Pipeline; Writing of Poster; Creation of Github Repository and Nerfstudio meshing pipeline script; Writing Report
- **Rico-Marcel Benning** Dataset Search; Research into existing meshing pipelines; Research into 3DGS for large scenes; Research into hole fixing and Alpha Wrapping post-processing; Testing and Eval of TSDF; Testing and customization of GOF; Printing and Design of the Poster; Writing Report
- **Landson Guo** Euler Inria Setup; Dataset Search; Research into existing meshing pipelines; Testing and customization of GOF; Testing of Tetrahedral Grids; Customization of QED-Splatter for Meshing; Writing of Poster; Writing Report; Creation of Github Repository and Nerfstudio meshing pipeline script; Chamfer Distance calculation and Quantitative Analysis scripts
- **Qinglin Chen** Dataset Search; Research into existing meshing pipelines; Research into 3DGS for large scenes; Research into Gaussian Splatting for outdoor scene; Add sky mask to 2DGS; Use pre-trained depth prediction method to generate initial point cloud for 2DGS; Testing of 2DGS and 2DGS Meshing, Research and creation of Post-Processing Script; Writing Report

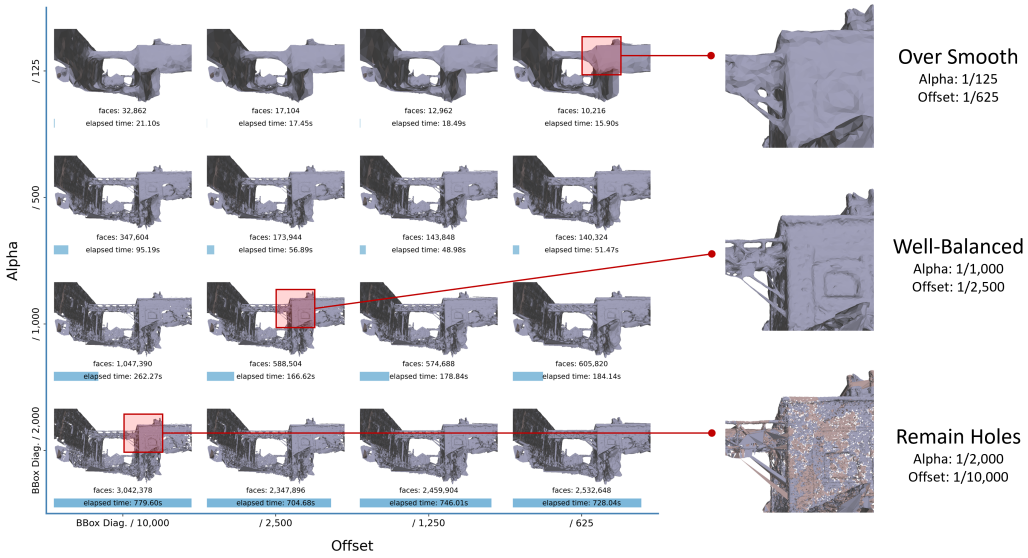

Figure 4. Parameter effects of Alpha Wrapping: impact of Alpha and Offset parameters on mesh quality and hole filling