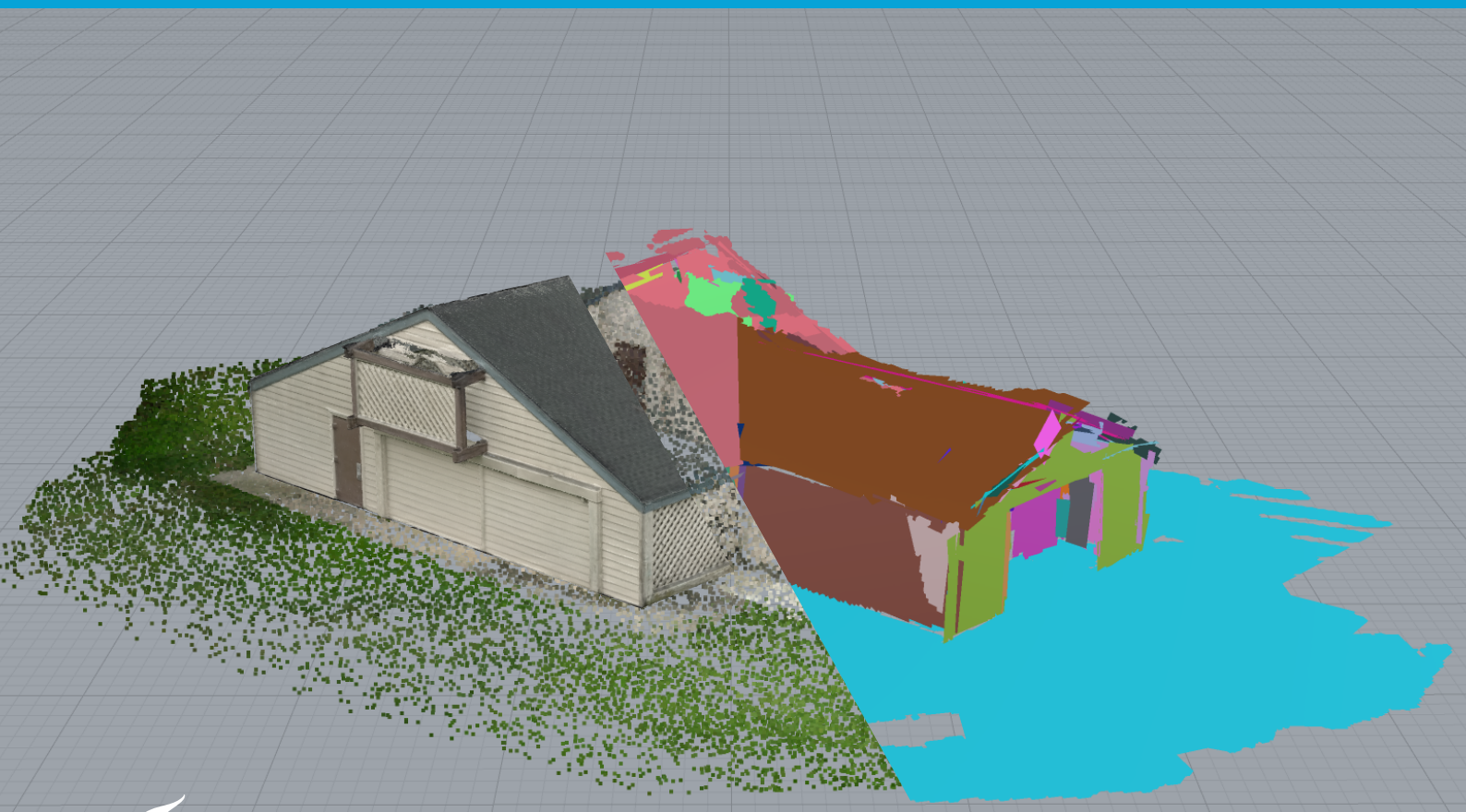
MSc thesis in Geomatics

Adaptive Plane Splatting for 3D Building Reconstruction

Ming-Chieh Hu

2026



TU Delft

MSc thesis in Geomatics

Adaptive Plane Splatting for 3D Building Reconstruction

Ming-Chieh Hu

June 2026

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

The work in this thesis was carried out in the:

3D geoinformation group
Delft University of Technology

Supervisors: Dr. Liangliang Nan
Dr. Michael Weinmann

Abstract

In the last three years, 3D Gaussian Splatting (3DGS) has attracted widespread attention due to its fast training and high-quality rendering, leading to numerous surface reconstruction studies proposing geometric constraints to extract high-quality meshes. Although these methods demonstrate potential for building model reconstruction, modern 3D building applications increasingly require watertight, Boundary Representation (B-rep) models for analytical tasks rather than the standard triangular meshes typically generated. On the other hand, traditional piecewise-planar reconstruction methods that rely on point clouds are often computationally heavy and unstable when generating plane hypotheses from noisy data. To address these limitations, this thesis proposes AdaptivePS, an adaptive, image-to-plane splatting pipeline for multi-view indoor and outdoor scene surface reconstruction. Designed to function as the foundational step in a broader "image to watertight building model" pipeline, it outputs planar primitives ready to be plugged into a piecewise-planar reconstructor. AdaptivePS extends the baseline PlanarSplatting method to outdoor environments by introducing a foreground mask generator and a novel prior generator that jointly recovers camera poses, depth, and normal maps in a single inference—bypassing Structure-from-Motion (SfM) entirely while normalizing scenes to a consistent scale. Additionally, the pipeline employs a mask-guided densification and pruning strategy to adaptively split primitives at object boundaries and remove background noise, alongside a mask-guided trimming mechanism applied to sampled points for sharper boundary delineation. Experiments demonstrate that AdaptivePS achieves sufficient geometric quality for outdoor scenes while running 2x as fast as the baseline framework. The code is available at https://github.com/MCHU-1999/AdaptivePS.

Keywords: 3D Gaussian Splatting, Plane Splatting, Surface Reconstruction, Piecewise-Planar Reconstruction, AdaptivePS.

Acknowledgements

First and foremost, I'd like to express my gratitude to my supervisors Liangliang Nan and Michael Weinmann. I'm very much aware that I am nothing close to a researcher, but I was treated with patience and respect. Sometimes I was stubborn and did plenty of side quests, but you just let me do my work at my own pace. It surprises me that doing research can be this enjoyable, and I think that's because you both have done a great job of mentoring me. I would also like to acknowledge Delft University of Technology, particularly the MSc Geomatics program, as a great environment is a prerequisite to making good work.

My deepest gratitude goes to my lovely family. The study is only made possible because of your support, but it's way more than just that. Because of you, I know I'm free to do anything, go wherever I want, both mentally and physically. It is truly a privilege to know that there are people supporting me by default. I look forward to seeing you soon and telling you more stories in person.

Thanks to my friends in the distance. You probably don't know how helpful it is just to see everyone progressing in their lives. It's such a motivation to me even with all the distance between us. Special tribute to Yuchi, who inspired me to come to the Netherlands, believed in my potential, and showed me possibilities in life. Thanks to my roommate Move, who kept me supported with her optimism and fed me the best food on earth.

Thanks to all the geo-magicians. It has been a great two years with all of you. I came here to study Geomatics only, but I ended up learning way more from you.

To everyone who has been part of this journey: It's difficult to measure how much impact those surrounding me can have. Sometimes it was merely a smile, sometimes words of appreciation, and sometimes honest but difficult advice. No matter the form, I want you to know that I'm genuinely grateful for all of it.

Contents

List of Figures

List of Tables

Acronyms

3DGS 3D Gaussian Splatting v
AI Artificial Intelligence 71
API Application Programming Interface 30
DA3 Depth Anything 3 22
GML Geography Markup Language 1
GS Gaussian Splatting 2
IDR Implicit Differentiable Renderer 5
JSON JavaScript Object Notation 1
KSR Kinetic Shape Reconstruction xiii
LiDAR Light Detection and Ranging 1
LLM Large Language Model 71
MVS Multi-View Stereo 1
NeRF Neural Radiance Fields 1
NVS Novel View Synthesis 1
OOM Out-Of-Memory 62
SAM Segment Anything Model 7
SDF Signed Distance Function 1
SfM Structure-from-Motion v
TnT Tanks and Temples xi
TSDF Truncated Signed Distance Function 1

1. Introduction

1.1. Motivation

In the last decades, 3D building models have been predominantly used for visualization; however, today they are being increasingly employed in a number of domains and for a large range of tasks beyond visualization. Consequently, most of these modern applications require B-rep, water-tight building models to effectively leverage geometric properties like area, volume, orientation, punctures, or openings [8]. Common examples include: solar irradiation analysis, energy demand estimation, shadow impact assessment, noise propagation modeling, computational fluid dynamics and natural hazard mitigation.

However, manually creating these watertight B-rep models at an urban scale is prohibitively time-consuming. Concurrently, many scene and surface reconstruction methods have been proposed. Classical scene reconstruction methods include SfM and Multi-View Stereo (MVS) [51, 18]. Common surface reconstruction methods include algorithms such as Marching Cubes [40], Poisson Surface Reconstruction [27], and Signed Distance Function (SDF) reconstruction [22]. Recent advances in learning-based Novel View Synthesis (NVS), such as Neural Radiance Fields (NeRF) [41] and 3DGS [28], have significantly enhanced the capability of the "photogrammetry to 3D model" pipeline.

A critical trade-off exists in current reconstruction workflows between efficiency and structural utility. While NeRF-based approaches achieve high fidelity, they suffer from prohibitive training times. GS-based reconstruction methods resolve the speed bottleneck and offer real-time rendering. But even with their explicit nature, they are constrained because the Gaussians do not directly represent the underlying surface geometry. To address this, existing GS-based methods enforce geometric constraints and utilize Truncated Signed Distance Functions (TSDFs) [15, 44] to extract surfaces from the rendered depth maps.

However, TSDF typically generate dense, unstructured triangular meshes that are structurally inefficient for building scenes and prone to incompleteness in occluded regions. These topological defects present significant challenges for downstream applications; specifically, non-manifold geometry and holes create ambiguity in representation and storage [2]. As a result, such outputs fail to meet the watertightness and manifold requirements of the ISO 19107 geometry model, which underpins standard urban data formats like CityGML [32] and CityJSON [35].

In contrast, piecewise-planar reconstruction algorithms—such as PolyFit [43] and KSR [7]—can generate ideal representations required for building analysis. However, these methods generally rely on high-precision, dense point clouds acquired via LiDAR systems or traditional MVS pipelines.

Both approaches present challenges to scalability and widespread adoption. While LiDAR provides exceptional geometric accuracy, it is costly and logistically demanding to deploy at an urban scale. Conversely, MVS algorithms, such as COLMAP, prioritize absolute geometric

accuracy through exhaustive multi-view patch matching. This makes them highly robust, but computationally intensive and slow. Furthermore, these algorithms are often hindered by foreground clutter, unwanted objects, or non-optimal capture angles, which introduce noise into the subsequent plane detection and optimization phases.

This work aims to leverage the advantages of Gaussian Splatting (GS)-based methods and piecewise-planar reconstruction algorithms. Instead of treating splatted primitives as a proxy to generate point clouds or TSDF meshes, this work leverages architectures natively designed for planar geometry. Specifically, methods such as PlanarSplatting [52], originally developed for structured indoor scenes, replace standard Gaussian ellipsoids with bounded 2D planar primitives. By optimizing these planar primitives, this work investigates a more direct pathway for piecewise-planar reconstruction algorithms, aiming to bypass traditional plane-detection phases.

Exploiting this requires overcoming a significant domain gap. PlanarSplatting is highly sensitive to the unconstrained complexity of outdoor environments, where atmospheric interference, skyboxes, and foreground clutter disrupt geometric optimization. Therefore, the primary goal of this thesis is to generalize these indoor-centric mechanisms to adapt them to outdoor building reconstruction. By integrating 2D semantic masking, this work aims to directly translate image captures into clean, explicit planes. The long-term vision for this pipeline is to generate lightweight, watertight, and topologically correct 3D building models that fulfill the strict manifold requirements of modern urban analysis.

1.2. Research Objectives

The primary goal of this research is to develop and evaluate a pipeline that leverages the explicit nature and runtime advantage of splatting-based methods to reconstruct structural geometry. To achieve this, the project outlines three main objectives:

1. establishing a robust core splatting framework adapted for outdoor building scenes,
2. performing a thorough quantitative and qualitative evaluation of the resulting reconstructions, and
3. conducting ablation tests on the proposed modifications.

1.2.1. Research Questions

The main research question for this thesis is:
How to optimize the Gaussians (or primitives) towards clusters of bigger bounded planes or polygons?

On top of the main question, the following sub-questions are defined:

1. *What additional information, training loss or post-processing steps are required?*
2. *What level of accuracy could be achieved compared to other scene reconstruction methods?*
3. *To what extent can this method be optimized for computation time?*

1.2.2. Scope

This thesis restricts its focus strictly to the extraction of structured, explicit planar surfaces. Rather than generating final watertight, manifold polygonal models, this project focuses on the prerequisite step: bridging the gap between splatted primitives and solid geometry. While a preliminary proof-of-concept demonstrating the integration of these surfaces into downstream reconstruction methods like KSR is provided, establishing a fully robust, automated hand-off remains outside the primary scope.

Additionally, aside from the core splatting optimization loop, this work strictly excludes the training or fine-tuning of deep neural networks; all external semantic and depth priors rely exclusively on off-the-shelf, pre-trained models. The piecewise-planar assumption remains central to this framework, meaning curved or highly organic geometries are excluded. Finally, the computational scale of the pipeline is bounded; this work does not address optimization strategies for city-scale environments, limiting evaluation datasets to 50 to 500 images per scene.

1.3. Thesis Outline

This thesis is organized into six chapters, including this introduction. The remainder is structured as follows:

Chapter 2 reviews the relevant literature, tracing from neural implicit representations to GS-based surface reconstruction. It also examines existing approaches in both indoor planar reconstruction and general point to piecewise-planar reconstruction to contextualize the theoretical gap this thesis addresses.

Chapter 3 establishes the theoretical background, detailing the mathematical and structural foundations of 3DGS [29] and the PlanarSplatting [52] architecture, which serve as the foundation for the proposed pipeline.

Chapter 4 details the core methodology. It presents a preliminary analysis diagnosing the specific domain gaps when applying indoor models to outdoor environments, and details the proposed architecture designed to overcome these challenges.

Chapter 5 presents datasets utilized, the experiments and results. It defines the evaluation metrics and demonstrates the effectiveness of the proposed method through benchmarking against the baseline model, supported by ablation studies to validate individual proposed modules. Finally, it includes a preliminary proof-of-concept demonstrating the potential of integrating the pipeline's extracted planes into downstream piecewise-planar reconstruction methods.

Chapter 6 summarizes the key findings of the research, answers the research questions, discusses the limitations of the proposed pipeline, and outlines potential pathways for future work.

2. Literature Review

This chapter provides a comprehensive review of the literature, framing the evolution of scene representations and surface reconstruction techniques. Each section details a specific lineage of development, followed by a dedicated discussion that contextualizes how this work builds upon, departs from, or addresses the limitations of paradigms.

The review begins in Section 2.1 with the foundational progression of NeRF-based NVS methods. Next, Section 2.2 shifts the focus to 3DGS methodologies. The field of indoor planar reconstruction—as a domain that deeply aligned with the planar assumptions of this thesis—is isolated in Section 2.3. Lastly, Section 2.4 concludes the chapter by evaluating piecewise-planar surface reconstruction methods.

2.1. Neural Implicit Surface Reconstruction

Volumetric representations for novel-view synthesis were revolutionized by NeRF [41]. Following its success, several methods emerged to address limitations in speed, anti-aliasing, and unbounded rendering through techniques like conical frustum tracing, explicit sparse grids, and fast hash-grid encodings (e.g., Mip-NeRF [4], Mip-NeRF 360 [5], Plenoxels [67], and Zip-NeRF [6]). However, despite these advancements, they remain fundamentally volumetric and lack the surface constraints required for accurate surface reconstruction.

To address these geometric limitations, surface representations replace volume density with SDFs or occupancy fields, defining surfaces as zero-level sets to allow for watertight mesh extraction via Marching Cubes [40]. Early approaches like Implicit Differentiable Renderer (IDR) [63] and UNISURF [45] explored this geometry-appearance disentanglement. The major breakthrough came with VolSDF [62] and NeuS [56], which transformed SDFs into volume densities for standard volumetric rendering. This line of work culminated in approaches like Neuralangelo [37], which achieved state-of-the-art detail by combining SDF-based rendering with multi-resolution hash grids [42].

While surface representations improve reconstruction, they often degrade in regions where photometric consistency alone is insufficient. To address this, researchers introduced structural priors to regularize the underlying SDF. Early works leveraged the "Manhattan World" assumption to align surface normals with dominant structural axes [19]. Alternatively, MonoSDF [69] proposed utilizing monocular normal and depth predictions to guide optimization. This significantly improved untextured areas and established a paradigm widely adopted by subsequent indoor reconstruction methods [55, 53, 13, 21].

While this research adopts the explicit representation to overcome the aforementioned bottlenecks, the geometric principles established in this domain remain fundamental. The regularization strategies pioneered in the implicit domain-—most notably the Manhattan and monocular priors—serve as the direct conceptual blueprint for the constraints I propose to adapt to the Gaussian Splatting framework.

2.2. Splatting-Based Surface Reconstruction

Following the introduction of 3DGS [28], recent works have focused on transforming the sparse Gaussian point cloud into high-fidelity surface meshes. As the foundation of the proposed method, the background knowledge of 3DGS method will be detailed in Section 3.1.

Mesh Extraction Strategies Initial approaches like SuGaR [20] incorporated depth and normal regularization to align Gaussians with the surface, extracting meshes via Poisson reconstruction on the resulting density field. However, this method suffers from high-frequency geometric noise due to the discrete, unconstrained nature of 3D Gaussian primitives, and the Poisson reconstruction algorithm tends to over-smooth sharp edges. To address this, GS2Mesh [59] proposed a workaround. By fusing multi-view depth maps rendered from Gaussians via TSDF Integration [15, 44], it mitigates the artifacts inherent to density fields, establishing a robust surface extraction pipeline for later works.

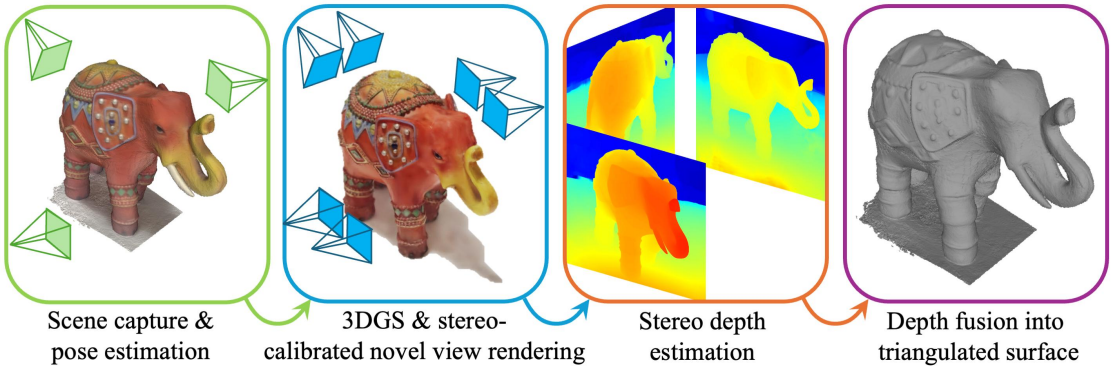


Figure 2.1.: Pipeline of the TSDF integration framework introduced by Wolf et al. [59].

Geometric Constraints and Regularization To improve surface alignment, 2DGS [24] replaced 3D ellipsoids with flat 2D disks (surfels) to enforce local planarity, utilizing self-supervised depth distortion and normal consistency losses. GOF [70] similarly applied geometric losses but introduced an implicit opacity field for mesh extraction via marching tetrahedra; however, the implicit formulation adds complexity and the result still showed some surface ripples (inconsistent normals).

Recent works focus on refining the depth formulation itself. RaDe-GS [71] introduced ray-based depth rendering to reduce volumetric bias. Currently, PGSR [11] represent the state-of-the-art in this domain. PGSR incorporates unbiased depth rendering with novel geometric constraints, specifically occlusion estimation, exposure compensation, and forward-backward projection errors. These terms enforce multi-view geometric consistency, significantly reducing artifacts in complex occluded regions.

Another line of work, such as DN-Splatter and AGS-Mesh [53, 49], leverages monocular depth and normal priors from pre-trained networks [17, 23]. However, the performance of these methods is highly dependent on the quality of the predicted priors, which are often not consistent across different views and need extra scaling process.

Bridging these approaches, PlanarGS [26] incorporates the geometric framework used by PGSR, but tailors the application toward structured indoor environments. PlanarGS utilizes monocular depth and normal priors to synthesize more complex planar information. This is achieved through a language-prompted planar priors (LP3) pipeline based on Grounded SAM [48], which generates semantic planar masks to guide a co-planarity constraint. By integrating these masks, PlanarGS has obtained even better results than PGSR in indoor scenes.

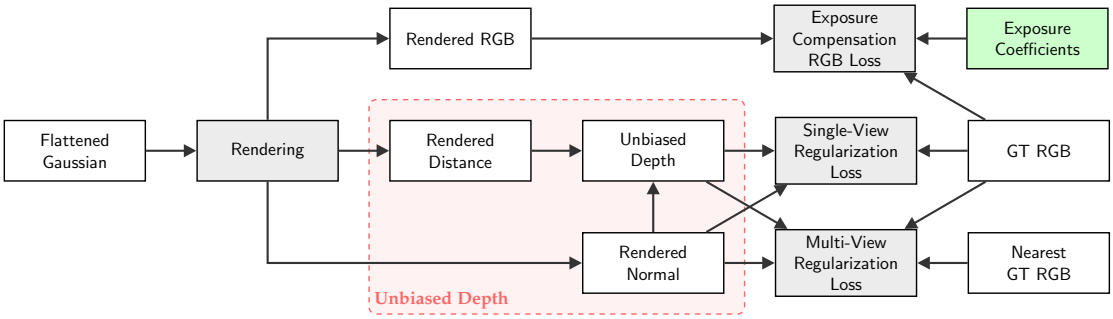


Figure 2.2.: PGSR [11] rendering pipeline with its regularization losses.

Semantic Integration The integration of semantic information has enabled object-level control over the reconstruction. Methods like Gaussian Grouping [65] and TSGaussian [72] employ off-the-shelf segmentation models such as Segment Anything Model (SAM) [30, 47, 9] and DEVA [14]. By adding a learnable 2D Identity Loss and 3D Regularization Loss, these methods can cluster Gaussians into semantic entities. While primarily used for scene editing and masking, this semantic awareness provides a foundation for class-specific filtering—a feature that's critical for outdoor scenes.

While PGSR and PlanarGS achieve impressive geometric fidelity, bridging the gap from unstructured Gaussians to explicit planes remains challenging. Initial tests applying traditional Region Growing directly to PGSR Gaussians (based on size and orientation) proved highly sensitive to primitive density and distribution; the approach requires extensive hyperparameter tuning that is difficult to generalize across diverse outdoor scenes.

To manage the complexity of unconstrained environments, SAM3 is employed to generate 2D masks, isolating architectural geometry from unwanted clutter, unlike methods such as Gaussian Grouping or TSGaussian—which embed semantics directly into 3D primitives. This ensures the subsequent planar optimization operates exclusively on clean, relevant structural data.

2.3. Indoor Planar Reconstruction

Indoor planar reconstruction is a well-established domain with a bigger literature base than its outdoor counterpart. Driven by Augmented and Virtual Reality (AR/VR) applications, the primary objective is to recover 3D planar embeddings from 2D images or video sequences to allow virtual agents to interact realistically with floors and walls. Because the focus is strictly on structural boundaries, these methods inherently tolerate geometric incompleteness in non-planar or heavily cluttered regions.

Within this category, learning-based methods such as PlanarRecon [60], AirPlanes [58], and NeuralPlane [64] have demonstrated impressive results in reconstructing clean geometry. However, these architectures rely heavily on large-scale, manually annotated 2D or 3D plane data for supervision, which is notoriously difficult and expensive to acquire.

In contrast, PlanarSplatting [52] introduces a specialized splatting-based paradigm for indoor planar reconstruction. Tailored for structured indoor environments dominated by large, axis-aligned planes such as walls, floors, and ceilings, it replaces standard 3D Gaussian ellipsoids with bounded 2D planar primitives (rectangles). PlanarSplatting discards standard RGB supervision entirely; instead, it relies exclusively on depth and normal maps generated from pre-trained monocular networks to supervise optimization. This has proven highly effective for initializing other GS pipelines or generating planar priors.

Building upon PlanarSplatting's rasterizer, PLANA3R [39] is a feed-forward plane splatting method integrating Vision Transformers [16]. Once trained, PLANA3R computes camera poses and planar primitives in a single feed-forward inference. Within this architecture, the mechanics of PlanarSplatting function as a alternative to manually annotated planar datasets. Also, by utilizing PlanarSplatting's rasterizer to directly splat 3D planar primitives into dense depth and normal maps, PLANA3R can be trained end-to-end using only monocular depth and normal labels, without requiring explicit plane annotations.

These models are typically optimized for bounded, high-density indoor captures. When applied to outdoor settings, they face challenges such as atmospheric interference, irregular building geometries (e.g., curved surfaces), and foreground clutter, which are not typically addressed in indoor-centric workflows.

While the aforementioned learning-based methods achieve clean plane extraction, their reliance on vast annotated datasets makes them infeasible for outdoor building reconstruction. Conversely, while PlanarSplatting is strictly an indoor-specific model, its underlying geometric mechanics offer a direct answer to this thesis's research question: *How to optimize the Gaussians (or primitives) towards clusters of bigger planes or polygons?*

Therefore, the PlanarSplatting architecture is selected as this research's fundamental backbone. The core objective of this work is to bridge the indoor-outdoor gap, introducing the necessary semantic filtering and regularization to adapt this indoor-specific representation for the complex, cluttered, and unbounded nature of outdoor architectural scenes.

2.4. Piecewise-Planar Surface Reconstruction

Piecewise-planar surface reconstruction focuses on extracting simplified, polygonal representations from unorganized 3D point clouds. It is still an underexplored field, yet these methods represent the gold standard for creating lightweight building models.

Methods like PolyFit [43] and Chauve's method [10] established strong baselines for piecewise-planar reconstruction. However, they rely on exhaustively slicing the bounding box with infinite planes. This approach generates overly dense partitions, limiting scalability; PolyFit requires excessive computation time for inputs exceeding one hundred shapes, while Chauve's method faces significant memory constraints.

In contrast, KSR [7] introduced a kinetic partitioning strategy, reducing partition complexity by an order of magnitude. Notably, PolyFit and KSR both guarantee the production of watertight, intersection-free meshes. This distinguishes them from methods like BSP-Net [12],

which may output intersecting convex polytopes. More recently, GoCoPP [68] demonstrated significantly higher accuracy in primitive fitting across complex object categories. However, GoCoPP functions as a primitive detection and optimization algorithm, and it ultimately relies on the KSR framework to execute the final geometric assembly.

PolyFit relies on Fast RANSAC [50] for planar primitive extraction, while KSR initializes primitives via Region Growing [46]. For both methods, the extraction of this initial primitive configuration relies heavily on the quality and density of the input point cloud. When connecting these pipelines with modern GS-based representations, researchers face a fundamental architectural detour. Converting explicitly learned spatial primitives (splats) back into dense, unstructured point clouds simply to feed them into Region Growing algorithms diminishes the advantages of the neural representation.

To evaluate the potential of optimizing splatted primitives directly into planes, this work investigates an alternative to this point-based detour. By utilizing bounded planes natively optimized during the splatting process, this approach aims to provide an explicit input for the plane-fitting stages of pipelines like PolyFit and KSR. This strategy seeks to preserve the benefits of the explicit representation, bypass the redundant conversion steps, and investigate the topological trade-offs of extracting robust planar priors directly from the rendering pipeline.

2.5. Positioning of the Thesis

In general, this thesis aligns with the current development trajectory in 3D vision shifting from implicit to explicit representations. This transition allows for a wider array of constraints to be utilized with ease. Moreover, explicit representations facilitate direct semantic filtering and pruning, removing the need to encode category information into feature tensors. This work is hugely inspired by the integration of monocular geometric priors and multi-view semantic information to guide optimization.

While numerous GS-based reconstruction methods have demonstrated high geometric fidelity, converting their outputs into watertight solid models typically requires sampling the learned primitives into dense point clouds. This intermediate conversion step leaves the final output dependent on traditional point-based plane detection and diminishes the native structural benefits of the explicit representation. By adopting a planar assumption, indoor planar reconstruction works provide a highly compatible baseline for this research, enabling an ”image-to-planes” workflow that attempts to bypass this redundant intermediate conversion. To provide a clear overview, Figure 2.3 illustrates how the proposed pipeline differs from common image-to-surface reconstruction frameworks.

The envisioned reconstruction pipeline aims to extract planar primitives directly from unconstrained images of outdoor building scenes, investigating an alternative to the plane detection steps of classical pipelines like KSR [7] and PolyFit [43]. This thesis provides a preliminary proof-of-concept, demonstrating the integration of these planar outputs into a downstream piecewise-planar optimization framework. However, establishing a fully robust, automated integration remains an area for future work.

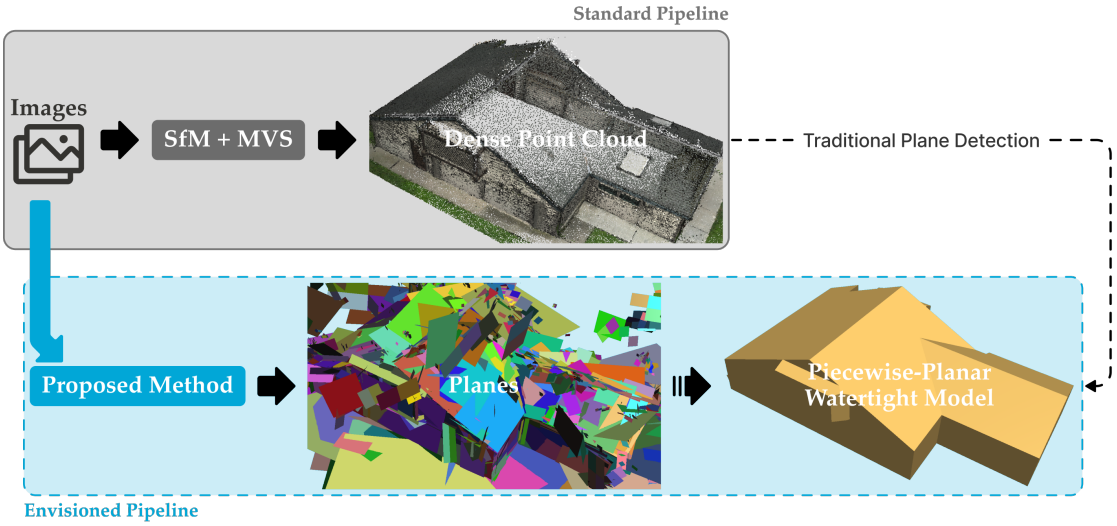


Figure 2.3.: Illustration depicting the envisioned pipeline compared to standard image-to-surface reconstruction pipelines.

3. Theoretical Background

This chapter establishes the architectural and mathematical foundations underpinning the proposed reconstruction pipeline. Section 3.1 describes the original, vanilla 3DGS [29]. Building upon this foundation, Section 3.2 presents the baseline PlanarSplatting [52]. Together, these sections define the operational baseline from which this thesis derives its outdoor building reconstruction methodology.

3.1. 3D Gaussian Splatting

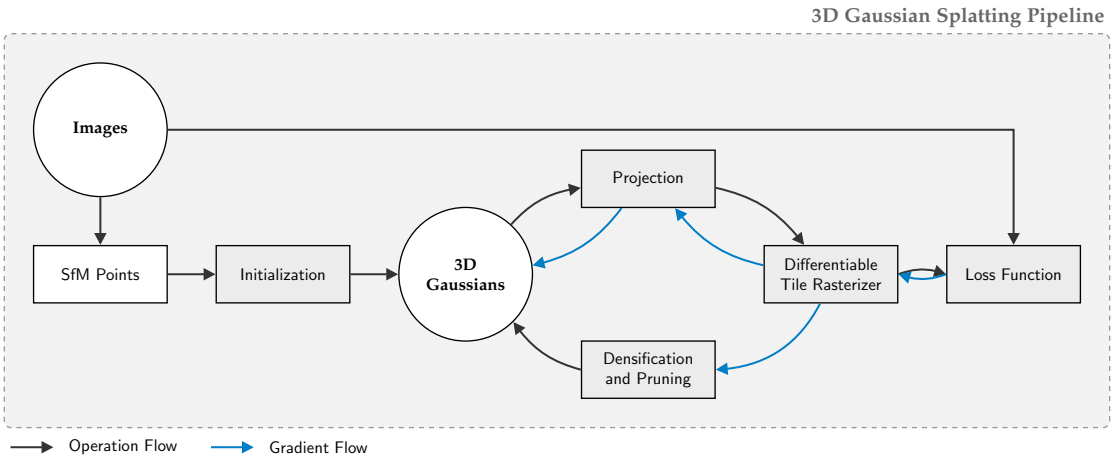


Figure 3.1.: Overview of the vanilla 3DGS [29] pipeline.

To enable end-to-end optimization of scene geometry, 3D Gaussian Splatting [29] formulates each primitive as a differentiable unit, allowing back-propagation. The pipeline begins with an initialization step where sparse point clouds from SfM serve as the initial Gaussian centers. To bridge the gap between the 3D representation and 2D image space, a fast differentiable rasterizer acts as the backbone infrastructure. Finally, the optimization is guided by a composite loss function and an extra control layer to densify or prune the Gaussian primitives. The pipeline of vanilla 3DGS is shown in Figure 3.1.

3.1.1. Gaussian Primitive Formulation

The scene geometry is represented as a set of 3D Gaussians $\{G_i\}$, eliminating the requirement for normal information. Each Gaussian is defined by a full 3D covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{3\times3}$

centered at a mean position $\boldsymbol{\mu}_i \in \mathbb{R}^3$:

$$G(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right). \tag{3.1}$$

3.1.2. Differentiable Primitive Rendering

3DGS builds upon previous point-based differentiable rendering techniques [66] to create a faster, higher-quality approach in NVS. Following the typical neural point-based approach [34, 33], the final pixel color $\boldsymbol{C} \in \mathbb{R}^3$ is computed using α-blending, sorting the Gaussians in a front-to-back order, blending $\mathcal{N}$ ordered primitives overlapping the pixel:

$$\boldsymbol{C} = \sum_{i\in\mathcal{N}} T_i \boldsymbol{c}_i \alpha_i, \quad T_i = \prod_{j=1}^{i-1}(1-\alpha_j). \tag{3.2}$$

Here, $\boldsymbol{c}_i$ denotes the view-dependent color of the i-th primitive, and T_i (Transmittance) represents "how much light has not been blocked by the Gaussians in front of the i-th." And α_i represents the influence of the i-th Gaussian at the current pixel location $\boldsymbol{p} = [x', y']^T$. It is the product of a learned per-Gaussian opacity o_i and the spatial weight of the projected 2D Gaussian $G(\boldsymbol{p}|\boldsymbol{\mu}_i', \boldsymbol{\Sigma}_i')$:

$$\alpha_i = o_i \cdot \exp\left(-\frac{1}{2}(\boldsymbol{p}-\boldsymbol{\mu}_i')^T\boldsymbol{\Sigma}'^{-1}(\boldsymbol{p}-\boldsymbol{\mu}_i')\right), \tag{3.3}$$

where $\boldsymbol{\mu}_i' \in \mathbb{R}^2$ is the projected 2D mean point and $\boldsymbol{\Sigma}' \in \mathbb{R}^{2\times2}$ is the projected 2D covariance matrix. Following the EWA splatting formulation [73], given a viewing transformation matrix $\boldsymbol{W}$ and an intrinsic matrix $\boldsymbol{K}$, the projected $\boldsymbol{\mu}_i'$ and $\boldsymbol{\Sigma}_i'$ can be computed as:

$$\boldsymbol{\Sigma}_i' = \boldsymbol{J}\boldsymbol{W}\boldsymbol{\Sigma}_i\boldsymbol{W}^T\boldsymbol{J}^T, \quad \boldsymbol{\mu}_i' = \boldsymbol{K}\boldsymbol{W}[\boldsymbol{\mu}_i, 1]^T \tag{3.4}$$

where $\boldsymbol{J}$ is the Jacobian of the affine approximation of the projective transformation.

Finally, the optimization of the 3D covariance $\boldsymbol{\Sigma}_i$ is constrained to ensure semi-definite positiveness. This is achieved by decomposing $\boldsymbol{\Sigma}$ into a scaling matrix $\boldsymbol{S}_i \in \mathbb{R}^{3\times3}$ and a rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times3}$:

$$\boldsymbol{\Sigma}_i = \boldsymbol{R}_i\boldsymbol{S}_i\boldsymbol{S}_i^T\boldsymbol{R}_i^T \tag{3.5}$$

where $\boldsymbol{S}$ and $\boldsymbol{R}$ are learned independently.

3.1.3. Optimization

To manage the distribution of primitives, 3DGS utilizes an adaptive density control mechanism. It identifies regions that are either under-reconstructed or suffer from over-reconstruction, applying specific operations to optimize the Gaussian distribution:

- Cloning: In under-reconstructed areas where Gaussians are small but exhibit high positional gradients, the method clones the existing primitives. A copy of the Gaussian is created and shifted in the direction of the gradient to better represent the geometry.

- Splitting: Conversely, in regions where large Gaussians cover areas of high spatial variance, the method splits them into smaller components. The original Gaussian is replaced by two new primitives, with their scales reduced by a factor of $\phi = 1.6$.

The total rendering loss combines an $\mathcal{L}_1$ term and a D-SSIM term [3]:

$$\mathcal{L}_{total} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{D-SSIM}}. \tag{3.6}$$

3.2. PlanarSplatting

Significantly differing from standard 3DGS formulations, PlanarSplatting [52] completely discards RGB photometric supervision. Instead, the optimization process is driven exclusively by monocular depth and normal priors generated by a pre-trained neural network such as Metric3Dv2 [23].

Because the method was explicitly developed for structured indoor environments, it inherently relies on strong geometric assumptions regarding scene planarity. The limitations of these indoor assumptions when applied to unconstrained outdoor environments will be analyzed in Section 4.2.

As illustrated in Figure 3.2, given a set of posed multi-view images, PlanarSplatting reconstructs the scene by optimizing a collection of learnable 3D planar primitives. Utilizing a custom differentiable planar rasterizer, the model explicitly aligns these primitives to recover the scene geometry. Following the optimization phase, these individual primitives are merged to extract 3D plane instances. The primitive and rendering formulation are explained in the following paragraphs.

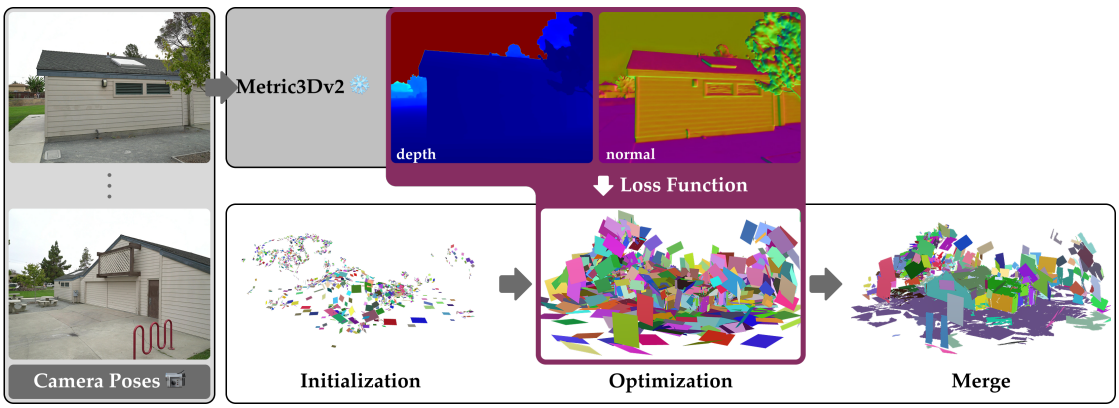


Figure 3.2.: Illustration of the PlanarSplatting architecture. The method renders depth and normal maps from splatted bounded 3D planar primitives, which are then optimized using monocular geometric priors to recover the scene geometry. Reproduced from [52].

3.2.1. Planar Primitive Formulation

The scene geometry is represented as a set of 3D rectangles $\{\pi_i\}$. A planar primitive π_i is formulated as a 3D rectangle, with learnable parameters including the plane center $\boldsymbol{\mu}_i \in \mathbb{R}^3$, the quarternion plane rotation $\boldsymbol{q}_i \in \mathbb{R}^4$ and the plane radii $\boldsymbol{r}_i$.

To accommodate asymmetric primitive extents, a dual-radii formulation is employed:

$$\boldsymbol{r}_i = \{r_i^{u+}, r_i^{u-}, r_i^{v+}, r_i^{v-}\} \in \mathbb{R}_+^4, \tag{3.7}$$

where $r_i^{u+}, r_i^{u-}, r_i^{v+}, r_i^{v-}$ are the radii defined on the positive/negative direction of the local U and V axes, respectively (as illustrated in Figure 3.3).

Let $\boldsymbol{R}(\boldsymbol{q}_i) \in \mathbb{R}^{3\times3}$ denote the rotation matrix derived from the quaternion parameters. The local orthonormal basis vectors $\boldsymbol{u}_i, \boldsymbol{v}_i \in \mathbb{R}^3$ and the surface normal $\boldsymbol{n}_i \in \mathbb{R}^3$ correspond to the columns of this rotation matrix:

$$\boldsymbol{R}(\boldsymbol{q}_i) = \begin{bmatrix} | & | & | \\ \boldsymbol{u}_i & \boldsymbol{v}_i & \boldsymbol{n}_i \\ | & | & | \end{bmatrix}. \tag{3.8}$$

which can be explicitly computed via directional projections:

$$\boldsymbol{u}_i = \boldsymbol{R}(\boldsymbol{q}_i)[1,0,0]^T, \quad \boldsymbol{v}_i = \boldsymbol{R}(\boldsymbol{q}_i)[0,1,0]^T, \quad \boldsymbol{n}_i = \boldsymbol{R}(\boldsymbol{q}_i)[0,0,1]^T. \tag{3.9}$$

During optimization, these learnable parameters enable the planar primitives to translate, rotate, and deform to conform to the scene geometry.

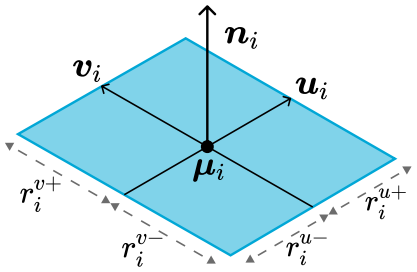


Figure 3.3.: Representation of the 3D plane primitive with learnable shape parameters.

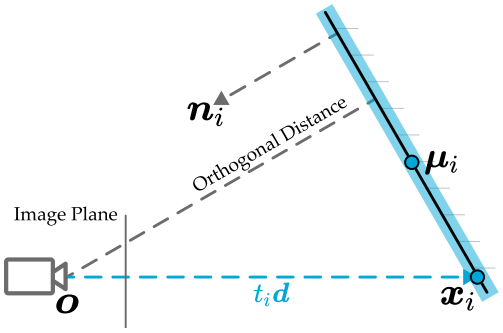


Figure 3.4.: Ray to plane intersection.

3.2.2. Differentiable Planar Primitive Rendering

To project the 3D planar primitives onto the 2D image plane, this work finds the ray-primitive intersections across the pixel grid, as depicted in Figure 3.4. Given a pixel coordinate $\boldsymbol{p}$ on the image plane, we cast a ray defined by the camera center $\boldsymbol{o} \in \mathbb{R}^3$ and the pixel-specific viewing direction $\boldsymbol{d} \in \mathbb{R}^3$. Its unique intersection point $\boldsymbol{x}_i \in \mathbb{R}^3$ with a specific planar primitive π_i is calculated as:

$$\boldsymbol{x}_i = \boldsymbol{o} + t_i \cdot \boldsymbol{d} = \boldsymbol{o} + \underbrace{\left[\frac{(\boldsymbol{\mu}_i - \boldsymbol{o}) \cdot \boldsymbol{n}_i}{\boldsymbol{d} \cdot \boldsymbol{n}_i}\right]}_{t_i} \cdot \boldsymbol{d} \tag{3.10}$$

where t_i represents the ray-space depth of the intersection relative to the camera origin.

Conventionally, the spatial influence of an anisotropic Gaussian primitive is modeled via a continuous volumetric function:

$$w_G(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i)\right). \tag{3.11}$$

However, because volumetric Gaussians generate soft, structurally ambiguous boundaries near edges, PlanarSplatting replaces them with a sharp, distance-based rectangular splatting weight function.

For a given intersection point $\boldsymbol{x}_i$, its localized projection distances $\mathcal{P}_{U,i}, \mathcal{P}_{V,i} \in \mathbb{R}$ are evaluated along the local orthonormal axes of the primitive:

$$\mathcal{P}_{U,i} = (\boldsymbol{x}_i - \boldsymbol{\mu}_i) \cdot \boldsymbol{u}_i, \quad \mathcal{P}_{V,i} = (\boldsymbol{x}_i - \boldsymbol{\mu}_i) \cdot \boldsymbol{v}_i. \tag{3.12}$$

The independent boundary weights along the local axes are regulated via a scaled logistic sigmoid $\sigma(\cdot)$. The weight along the U-axis is given by:

$$w_U(\boldsymbol{x}|\boldsymbol{\mu}_i, \pi_i) = \begin{cases} 2\sigma(5k(r_i^{u+} - |\mathcal{P}_{U,i}|)), & \text{if } \mathcal{P}_{U,i} > 0 \\ 2\sigma(5k(r_i^{u-} - |\mathcal{P}_{U,i}|)), & \text{otherwise} \end{cases}. \tag{3.13}$$

And similarly for V-axis:

$$w_V(\boldsymbol{x}|\boldsymbol{\mu}_i, \pi_i) = \begin{cases} 2\sigma(5k(r_i^{v+} - |\mathcal{P}_{V,i}|)), & \text{if } \mathcal{P}_{V,i} > 0 \\ 2\sigma(5k(r_i^{v-} - |\mathcal{P}_{V,i}|)), & \text{otherwise} \end{cases}. \tag{3.14}$$

And the final splatting weight is calculated as:

$$w(\boldsymbol{x}|\boldsymbol{\mu}_i, \pi_i) = \begin{cases} w_U, & \text{if } w_U < w_V \\ w_V, & \text{otherwise} \end{cases} \tag{3.15}$$

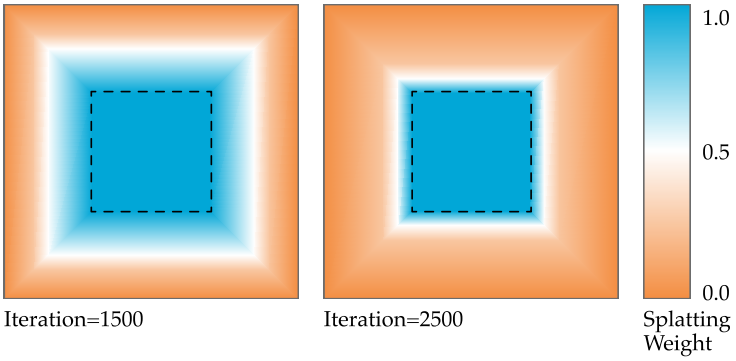


Figure 3.5.: Illustration of the proposed plane splatting function. The proposed plane splatting function approximates the rectangular boundary as the number of iterations increases, allowing for stable scene fitting.

As illustrated in Figure 3.5, the splatting function progressively approximates a strict rectangular boundary as the sharpening hyperparameter k increases. During optimization, k is dynamically scaled via an exponential schedule up to a maximum value of 300:

$$k = \min\left(20e^{-(1-0.001 \cdot ite)}, 300\right), \tag{3.16}$$

where ite denotes the current optimization iteration index.

To render the discrete pixel values, all valid ray-to-plane intersections along the viewing path are filtered by their splatting weights and sorted by depth in a front-to-back order. Let $\tau(j)$ be a mapping function yielding the global primitive index of the j-th closest intersected plane along the ray. The M nearest valid intersections are retained for rendering ($M = 30$).

The composite depth and normal values for a given pixel $\boldsymbol{p}$ in the image $\boldsymbol{I}$ are accumulated via alpha-blending:

$$\boldsymbol{D}_{render}(\boldsymbol{p}) = \sum_{j=1}^{M} T_j \cdot w(\boldsymbol{x}|\boldsymbol{\mu}_{\tau(j)}, \pi_{\tau(j)}) \cdot t_j \tag{3.17}$$

$$\boldsymbol{N}_{render}(\boldsymbol{p}) = \sum_{j=1}^{M} T_j \cdot w(\boldsymbol{x}|\boldsymbol{\mu}_{\tau(j)}, \pi_{\tau(j)}) \cdot \boldsymbol{n}_{\tau(j)} \tag{3.18}$$

where $t_{\tau(j)}$ and $\boldsymbol{n}_{\tau(j)}$ represent the depth and normal of the sorted primitive, respectively, and T_j represents the accumulated visibility transmittance:

$$T_j = \prod_{i=1}^{j-1}(1 - w(\boldsymbol{x}|\boldsymbol{\mu}_{\tau(i)}, \pi_{\tau(i)})) \tag{3.19}$$

3.2.3. Optimization

The optimization framework is supervised exclusively using monocular geometric constraints. The normal loss enforces angular consistency and L1 alignment with the monocular surface normals $\boldsymbol{N}(\boldsymbol{p})$:

$$\mathcal{L}_{normal} = \sum_{\boldsymbol{p}\in\boldsymbol{I}} \|1 - \boldsymbol{N}_{render}(\boldsymbol{p})^T \boldsymbol{N}(\boldsymbol{p})\|_1 + \sum_{\boldsymbol{p}\in\boldsymbol{I}} \|\boldsymbol{N}_{render}(\boldsymbol{p}) - \boldsymbol{N}(\boldsymbol{p})\|_1. \tag{3.20}$$

Similarly, the depth loss penalizes discrepancies relative to the monocular depth map $\boldsymbol{D}(\boldsymbol{p})$:

$$\mathcal{L}_{depth} = \sum_{\boldsymbol{p}\in\boldsymbol{I}} \|\boldsymbol{D}_{render}(\boldsymbol{p}) - \boldsymbol{D}(\boldsymbol{p})\|_1. \tag{3.21}$$

The total objective function is minimized using a weighted joint loss formulation:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{normal} + \lambda_2 \mathcal{L}_{depth}, \tag{3.22}$$

where the balancing hyperparameters are set to $\lambda_1 = 5$ and $\lambda_2 = 2$.

4. Methodology

4.1. Overview

This chapter details the methodology for adapting PlanarSplatting to unconstrained outdoor environments. To provide a clear overview, this chapter distinguishes between the iterative **research methodology** used to develop the solution and the **technical methodology** of the final proposed solution.

As illustrated in Figure 4.1, the development of this work followed an iterative research progression:

1. **Preliminary Analysis (Baseline Verification):** To motivate the architectural changes, the baseline PlanarSplatting model is first evaluated on unconstrained outdoor scenes. This diagnostic stage identifies the specific operational bottlenecks and failure cases inherent to the original indoor-specific method. This analysis is documented in Section 4.2.
2. **Iterative Development:** Guided by the gaps identified in the preliminary phase, new algorithmic solutions were continuously planned, implemented, and evaluated to bridge the outdoor domain gap.

The final result of this cyclical research process is the proposed solution: **AdaptivePS**. The explicit mechanics of this architecture are detailed in Section 4.3, which encompasses the integration of 2D semantic masking to isolate structural features, followed by geometric regularizations to stabilize planar optimization in outdoor environments.

Finally, detailed implementation specifications, including software configurations, parameter settings, and hardware setups required to reproduce the proposed pipeline, are provided in Appendix A.

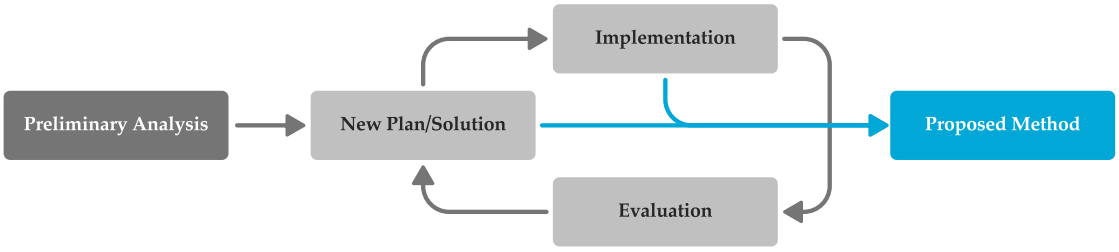


Figure 4.1.: Iterative research methodology used to develop the proposed AdaptivePS pipeline.

4.2. Preliminary Analysis: The Outdoor Domain Gap

To establish a generic baseline and motivate the necessity of the proposed architecture, the original PlanarSplatting framework was evaluated in both controlled synthetic environments and unconstrained real-world outdoor scenes. The objective of this diagnostic phase was to isolate the specific variables that cause indoor-specific models to fail in outdoor domains.

4.2.1. Diagnostic Verification via Synthetic Control

Before evaluating the model on real-world data, a synthetic control experiment was conducted to verify the fundamental viability of the core planar splatting mechanism. A simplified, bounded 3D box scene (detailed in Section 5.1.1) was generated to provide the pipeline with perfect ground-truth inputs. By supplying flawless camera poses, depth maps, and normal maps, this experiment intentionally bypassed the baseline's neural prior generation network. This isolated the performance of the differentiable rasterizer under ideal, clutter-free conditions.

As shown in Figure 4.2, this strictly controlled version of the PlanarSplatting successfully reconstructed the piecewise-planar geometry of the box with high fidelity. Crucially, the reconstruction was completely free of floating background planes or topological artifacts. This diagnostic test isolates the variables of failure, proving that the underlying differentiable plane rasterizer is mathematically sound. Consequently, any degradation observed in subsequent real-world tests must be attributed strictly to the complexities of the input data, the inaccuracy of generated monocular priors, and the lack of scene bounding.

4.2.2. The Methodological Gap

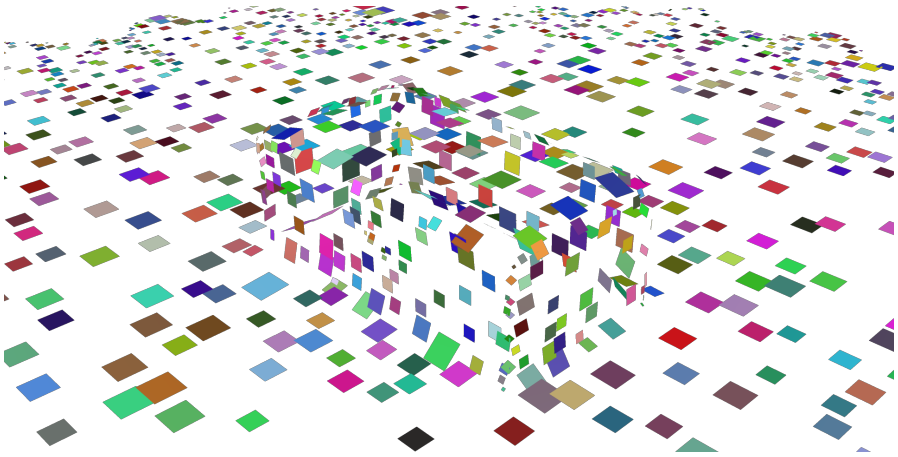
While the framework succeeds in a perfect environment, real-world captures do not satisfy these conditions. When applied to real-world scenes (such as the TnT *Barn* scene), the vanilla pipeline suffers from severe geometric degradation. This failure is rooted in fundamental methodological discrepancies between indoor and outdoor environments.

Indoor scenes are inherently bounded and structurally contiguous; mesh fusion algorithms can safely operate under the assumption that the environment is closed by walls, floors, and ceilings. In contrast, outdoor scenes are characterized by unbounded backgrounds, larger depth ranges (e.g., skyboxes), and other unwanted clutter (e.g., vegetation, vehicles).

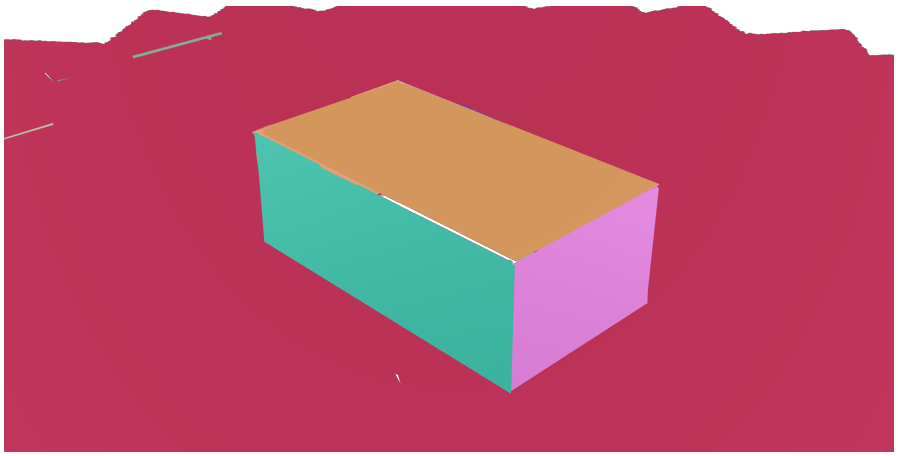
As shown in Figure 4.3, when applied to unconstrained environments, the model attempts to fit planar primitives to distant, unconstrained regions or background elements. This generates massive amounts of unwanted, disconnected planar artifacts that corrupt the visual fidelity and break the topological requirements necessary for piecewise-planar building extraction.

The primary goal of this research is to resolve these identified domain gaps. To achieve this, the development of the proposed AdaptivePS pipeline is detailed in the following section.

(a) Ground truth 3D model



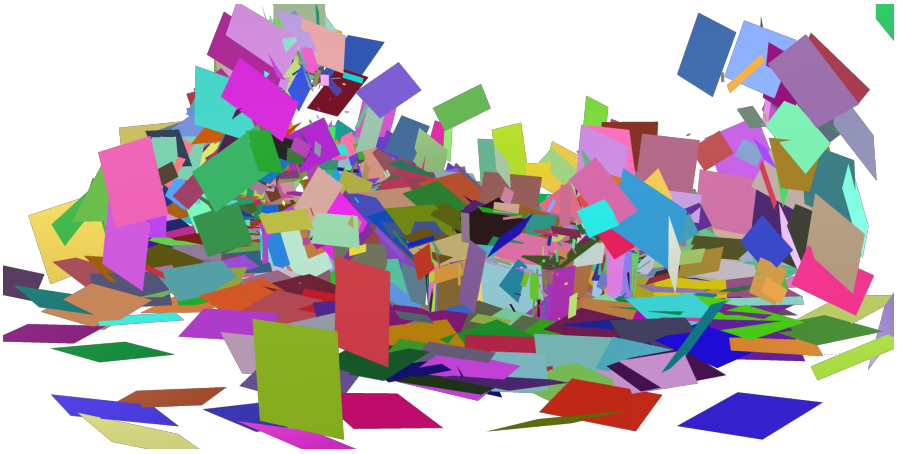
(b) Plane initialization



(c) Merged planes

Figure 4.2.: Reconstruction of a synthetic, clutter-free box scene using perfect geometric priors (planes colored randomly). The clean result verifies the functional integrity of the core splatting mechanism.

(a) Plane initialization



(b) Optimized planes (in the last iteration)

(c) Merged planes

Figure 4.3.: Results of baseline PlanarSplatting, with TnT [31] *Barn* scene as example (planes colored randomly).

4.3. The Proposed AdaptivePS

As established in Section 4.2, the core differentiable plane rasterizer and optimization strategy of the baseline PlanarSplatting provide a robust geometric foundation. However, its indoor-centric assumptions lead to severe degradation when exposed to the unbounded depth and unconstrained clutter of outdoor environments.

To generalize this framework for outdoor building reconstruction, the proposed AdaptivePS pipeline introduces targeted modifications to overcome these specific domain gaps.

The new pipeline:

- upgrades the modules for pose-recovering and geometric prior generating;
- integrates adaptive semantic masking to isolate the target geometry;
- introduces a novel semantic-guided densification and pruning strategy.

The workflow of the proposed method is illustrated in Figure 4.4.

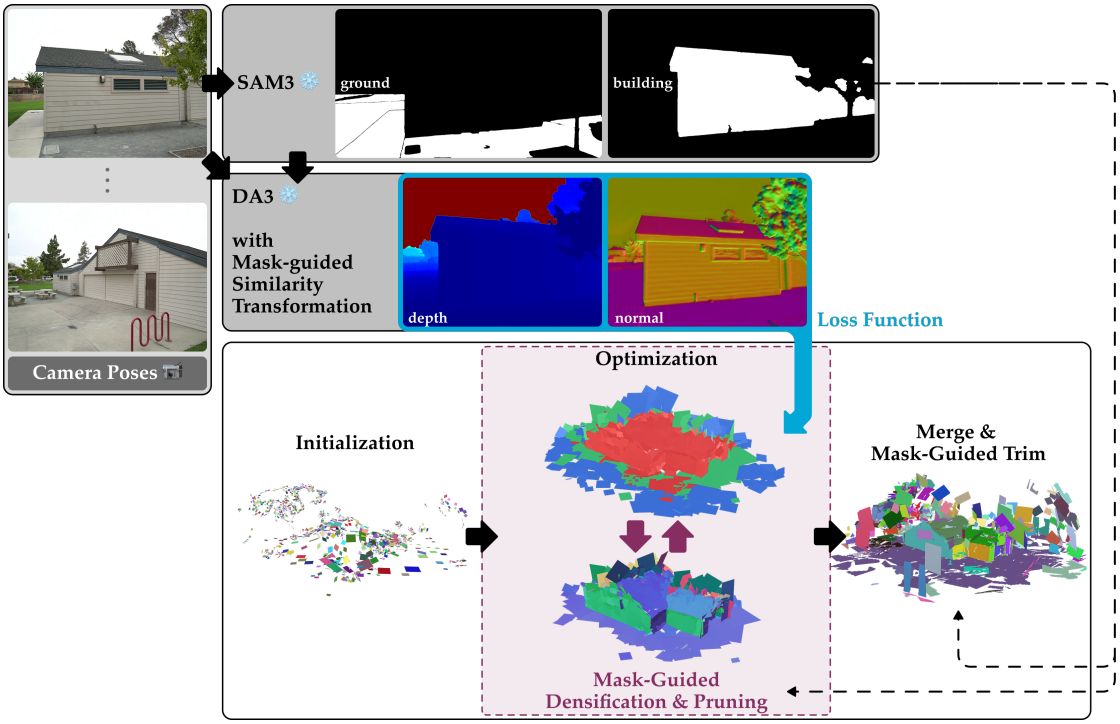


Figure 4.4.: Flowchart of the proposed AdaptivePS.

4.3.1. Foreground Mask Generation

The preliminary analysis demonstrated that heuristic depth truncation is insufficient for completely filtering out unwanted clutter. To enforce a clean optimization and restrict planar primitives from fitting to the sky or unwanted objects, the pipeline introduces an adaptive semantic masks generated by SAM3 [9]. By providing targeted prompts, SAM3 generates foreground

masks that explicitly isolate the target building (or any user-defined object) from the messy outdoor environment.

For each scene, 2 series of masks are generated, namely ground mask and building mask. The ground and building masks are used later in prior generation (Section 4.3.2), and in the densification and pruning (Section 4.3.3) stage only the building masks are used. The specific prompts utilized in this work are detailed in Appendix B.

4.3.2. Prior Generation

Prior Generator The baseline PlanarSplatting[1] assumes the inputs are posed images but also allow for pose-free multi-view inputs. The proposed AdaptivePS provides the same options with the use of Depth Anything 3 (DA3) [38].

For the use case of outdoor drone shots, a heavy reliance on traditional SfM pipelines like COLMAP is often unstable. As architectural structures frequently exhibit repetitive elements (e.g., concrete, brickwork textures) and reflective materials (e.g., glass windows), which pose significant challenges for traditional feature-matching algorithms.

DA3 provides a all-in-one solution to these challenges, as it jointly optimizes metric depth and relative camera poses. It demonstrates superior performance compared to its predecessors, such as DUSt3R [57] and MASt3R [36]. And it also exhibits better multi-view consistency than those from Metric3Dv2.

Despite the aforementioned advantages, processing massive image datasets simultaneously through DA3 is highly memory-intensive and requires substantial GPU VRAM. As a result the resolution of normal maps might be reduced. Also, because DA3 is not explicitly trained to predict surface normals, these must be derived analytically from the depth map gradients. The derived normal $\boldsymbol{N}_d(\boldsymbol{p})$ is computed via the cross-product of neighboring 3D points back-projected from the depth map:

$$\boldsymbol{N}_d(\boldsymbol{p}) = \frac{(\boldsymbol{P}_2 - \boldsymbol{P}_1) \times (\boldsymbol{P}_4 - \boldsymbol{P}_3)}{|(\boldsymbol{P}_2 - \boldsymbol{P}_1) \times (\boldsymbol{P}_4 - \boldsymbol{P}_3)|} \tag{4.1}$$

where $\{\boldsymbol{P}_j \mid j = 1, 2, 3, 4\}$ are the unprojected 3D coordinates corresponding to the four adjacent orthogonal pixel neighbors of $\boldsymbol{p}$.

To compensate for the lack of native normal prediction, AdaptivePS provides two pipeline configurations:

- **Hybrid Configuration:** Employs DA3 solely for camera tracking and metric depth generation, while relying on Metric3Dv2 to estimate surface normals. This configuration yields a direct monocular prediction target, setting $\boldsymbol{N}(\boldsymbol{p}) = \boldsymbol{N}_{\text{Metric3D}}(\boldsymbol{p})$ in the optimization loss.
- **Unified Configuration:** Utilizes DA3 for depth generation and computes the target normal maps analytically via local gradients, setting $\boldsymbol{N}(\boldsymbol{p}) = \boldsymbol{N}_d(\boldsymbol{p})$. This eliminates secondary model dependencies and accelerates the overall prior generation phase.

The performance trade-offs between these two configurations are evaluated in Chapter 5.

[1] Despite not explicitly mentioned in their publication, the official implementation of PlanarSplatting (https://github.com/ant-research/PlanarSplatting) supports pose-free multi-view inputs based on VGGT [54].

Mask-Guided Similarity Transformation A fundamental challenge in photogrammetry and image-to-3D reconstruction is scale ambiguity. In outdoor environments, the scale of target buildings varies significantly more than in standard indoor rooms. Because the baseline PlanarSplatting architecture relies on numerous scale-dependent metric hyperparameters, these variations can severely disrupt the optimization process. Therefore, it is highly beneficial to normalize the metric scale of the outdoor scene prior to reconstruction.

This normalization is achieved through a similarity transformation (scaling, rotation, and translation) guided by building and ground masks generated by SAM3. The transformation first analyzes the isolated building points and scales the target structure to a user defined value (the other parameters need to be updated accordingly). Next, utilizing the ground masks, the algorithm estimates the ground plane and rotates the entire scene to align the ground normal vector with the Z-axis $(0, 0, 1)$. Finally, it translates the centroid of the building's footprint, snapping it to the origin $(0, 0, 0)$ of the coordinate system. The effect of this transformation is illustrated in Figure 4.5.

It also significantly reduces manual labor by eliminating the need for per-scene calibration of the bounding and geometric parameters, such as truncation distances and camera ranges (detailed in Table A.3) that are otherwise required to prevent geometric failure or memory overflow.

It should be noted that this similarity transformation is primarily necessary for data lacking reliable camera extrinsics, such as typical unconstrained outdoor captures. For controlled laboratory environments or real-world captures with high-precision GPS localization, this normalization step can be bypassed, provided that the pipeline's hyperparameters are calibrated to appropriately match the true metric scale of the scene.

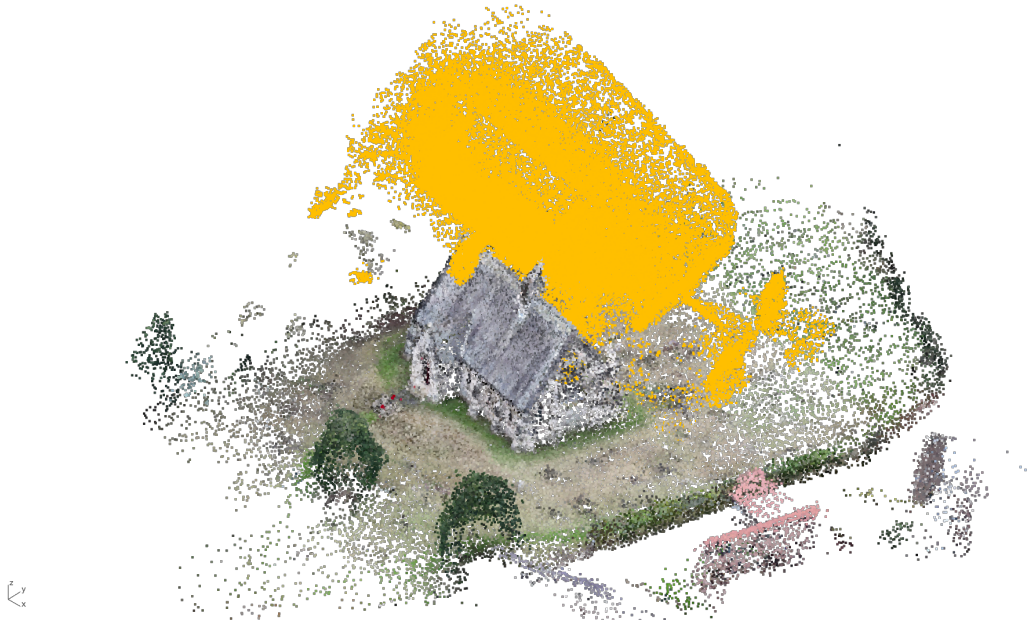


Figure 4.5.: Similarity transformation applied on a scene with camera parameters unknown. The yellow point cloud represents the scene before transformation (scene: church-cadeby from Pexels dataset).

4.3.3. Densification and Pruning

Since the introduction of the original 3DGS [28], it has been a standard convention to control the growth of primitives via adaptive densification and pruning. It is concluded in their work that the positional gradients can reflect both the "under-reconstruction" and "over-reconstruction" regions. Standard 3DGS addresses these via cloning and splitting respectively.

The baseline PlanarSplatting employs a modified densification strategy: it utilizes splitting only and disables cloning. This restriction is highly practical for outdoor scenes, in which the primary challenge is an abundance of geometry (background clutter) rather than insufficient plane coverage. While under-reconstructed areas occasionally occur due to sparse camera angles or occlusions, it is inevitable, and generating excessive background planes is the more critical failure mode.

During optimization, the baseline PlanarSplatting executes a splitting operation based on the spatial gradients of a plane's radii to better fit the scene geometry. If the average radii gradients along the U-axis (r^{u+} and r^{u-}) exceed 0.1, the plane is split along the V-axis. Conversely, planes are split along the U-axis if their V-axis gradients (r^{v+} and r^{v-}) exceed the 0.1 threshold. This splitting operation is performed every 1,000 iterations. Concurrently, the algorithm prunes planes that fall below the minimum spawning size or those that remain entirely invisible to the cameras (contributing zero gradient to the total loss), which is also done every 1,000 iterations. The exact configurations and values for these hyperparameters[2] are detailed in Table A.4.

Mask-Guided Densification and Pruning The baseline pruning strategy proves insufficient for unconstrained outdoor environments, as large planes frequently anchor to background clutter or straddle the boundary between the target building and the sky.

To overcome this, a novel densification and pruning strategy guided by semantic foreground masks (generated by SAM3) is proposed. By intersecting the gradients with the binary semantic masks, the proposed method tracks the spatial distribution of each plane across all camera views.

For a given plane, if its projected gradients fall within the foreground mask, a foreground hit (H_{fg}) is recorded. If the gradients fall within the background, a background hit (H_{bg}) is recorded. Note that a single plane can accumulate both foreground and background hits depending on its size, location and orientation. Crucially, projections that fall outside the camera frame are ignored during hit accumulation. This ensures that planes observed in close-up captures are not falsely penalized as background geometry.

Once hits are accumulated across all available camera frames, a foreground ratio ($\mathcal{R}_{\text{fg}}$) is computed for all planes where $H_{\text{fg}} + H_{\text{bg}} > 0$:

$$\mathcal{R}_{\text{fg}} = \frac{H_{\text{fg}}}{H_{\text{fg}} + H_{\text{bg}}} \tag{4.2}$$

[2] `split_thres`, `process_plane_freq_ite`, `check_vis_freq_ite`

Based on this score, a simple thresholding operation is applied to dynamically categorize and manage the primitives:

$$\text{Action} = \begin{cases} \text{Retain}, & \text{if } \mathcal{R}_{\text{fg}} > 0.6 \quad \text{(foreground)} \\ \text{Split (both } U \text{ and } V\text{)}, & \text{if } 0.3 < \mathcal{R}_{\text{fg}} \leq 0.6 \quad \text{(ambiguous)} \\ \text{Prune}, & \text{if } \mathcal{R}_{\text{fg}} \leq 0.3 \quad \text{(background)} \end{cases} \tag{4.3}$$

This thresholding enforces a strict semantic boundary. Planes falling primarily in the foreground ($\mathcal{R}_{\text{fg}} > 0.6$) are preserved as structural geometry. And those falling primarily in the background ($\mathcal{R}_{\text{fg}} \leq 0.3$) are aggressively pruned, effectively eliminating outdoor clutter.

Most importantly, planes that are classified as *ambiguous* ($0.3 < \mathcal{R}_{\text{fg}} \leq 0.6$) represent primitives that bridge the target building and the background. Rather than discarding them entirely (which could create holes in the building facade) or keeping them (which introduces background noise), the proposed method splits these planes along both the U and V axes. And in subsequent optimization cycles, these smaller sub-planes will cleanly separate—falling decisively into either the foreground to be retained, or the background to be pruned.

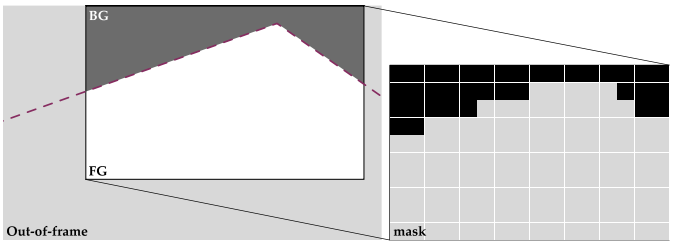


Figure 4.6.: Diagram illustrating the classification of spatial projections into Foreground (FG), Background (BG), and Out-of-Frame (OOF) regions for semantic hit accumulation.

4.3.4. Plane Merging and Trimming

Following the optimization phase, the pipeline transitions to a post-processing stage designed to extract macroscopic architectural surfaces from the optimized planar primitives. This is achieved through point-level trimming, followed by co-planar merging.

First, the optimized primitives must be refined to trim over-extended geometry. The baseline PlanarSplatting accomplishes this via a point-level, mesh-distance trimming strategy:

1. A coarse reference mesh is fused via TSDF integration using the depth maps rendered from the optimized planes.
2. The planar primitives are discretized into a dense set of sampled points, governed by a defined voxel resolution.
3. The Euclidean distance between each sampled point and the reference mesh is calculated.
4. Any point whose distance exceeds a predefined tolerance threshold is permanently removed, trimming the boundaries of the plane to match the multi-view consensus geometry.

Mask-Guided Point Trimming To enforce precise silhouette adherence and prevent planes from overhanging into the background, this work introduces a novel mask-guided trimming mechanism immediately following the baseline trim. This method adopts the exact pruning logic established in Section 4.3.3, but shifts the evaluation target from the planes to the sampled points.

Each sampled point $\boldsymbol{P}_i$ is projected onto all available camera frames. By intersecting these projections with the binary SAM3 masks, foreground hits (h_{fg}) and background hits (h_{bg}) are accumulated specifically for that individual point.

Similar to the primitive-level evaluation, a point-specific semantic foreground ratio ($\mathcal{R}'_{\text{fg}}$) is computed:

$$\mathcal{R}'_{\text{fg}} = \frac{h_{\text{fg}}}{h_{\text{fg}} + h_{\text{bg}}} \tag{4.4}$$

However, unlike planar primitives which can be subdivided during optimization, discretely sampled points are indivisible. Therefore, the ternary classification is replaced with a binary thresholding criteria for point retention:

$$\text{Action} = \begin{cases} \text{Retain}, & \text{if } \mathcal{R}'_{\text{fg}} > 0.6 \quad \text{(Foreground)} \\ \text{Prune}, & \text{if } \mathcal{R}'_{\text{fg}} \leq 0.6 \quad \text{(Background)} \end{cases} \tag{4.5}$$

Sampled points that fail to meet this threshold are trimmed. This final semantic refinement ensures that the boundaries of the primitives perfectly conform to the isolated target building, eliminating any residual overhang or background artifacts before the final mesh generation.

Co-Planar Merging and Extraction Following the semantic trimming phase, the surviving points and their associated primitives are evaluated for consolidation. The merging process follows a traditional co-planarity heuristic: primitives are clustered and merged if their normal angular deviation is below a specified threshold (25°) and their spatial offset is within a distance threshold (0.1cm).

Finally, these consolidated points are reconstructed into macroscopic solid meshes. Instead of relying on isolated planar data structures, the output is formatted as a unified polygonal mesh, with each distinct contiguous surface assigned a unique plane instance label and randomized color for visualization and downstream application.

The exact numerical values for these sampling and merging parameters[3] are detailed in Appendix A.

4.3.5. Other Adaptations

Mesh Post-processing In the final stages of the baseline PlanarSplatting, the pipeline merges and trims bounded planes by checking their distance to a coarse reference mesh generated via TSDF integration. TSDF integration is highly valuable here because it enforces multi-view geometric consensus, inherently averaging out high-frequency noise and discarding view-inconsistent floating artifacts.

[3] `voxel_length`, `normal_angle_thresh`, `dist_thresh`

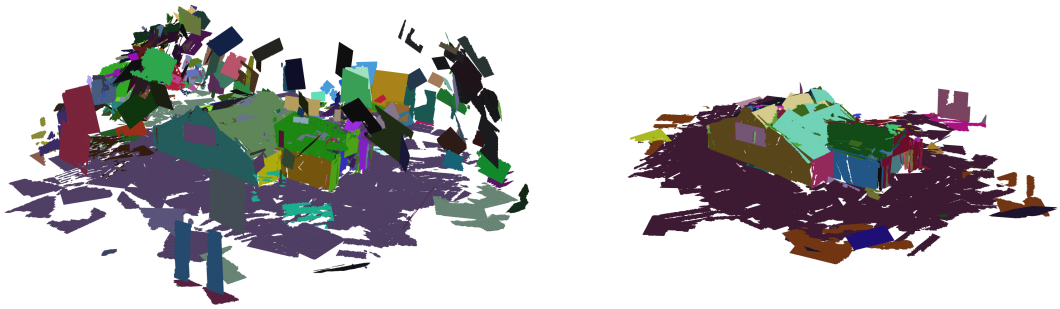
To further suppress outdoor background clutter, an explicit mesh post-processing function was introduced to the baseline. This function isolates and retains only the largest connected mesh component, effectively discarding background geometry. This connected-component filtering is applied at two critical junctures in the pipeline:

1. **Initialization:** Providing a cleaner structural base to ensure planar priors are initialized on the target area.
2. **Merging and Trimming:** Delivering a heavily filtered reference mesh to ensure final planar primitives are not erroneously anchored to outdoor clutter.

Figure 4.7 demonstrates the effect of this mesh post-processing function. As shown, the target building geometry is preserved and noise is removed. Consequently, Figure 4.8 illustrates the impact of having cleaner reference mesh on the final planar reconstruction, significantly reducing the presence of floating artifacts compared to the unmodified version.

(a) Without mesh post-processing (b) With mesh post-processing

Figure 4.7.: Effect of the proposed mesh post-processing function on initialization mesh



(a) Without mesh post-processing (b) With mesh post-processing

Figure 4.8.: Effect of the proposed mesh post-processing function on final reconstruction

5. Experiments and Results

This chapter presents a comprehensive evaluation of the proposed pipeline, structured to systematically validate its geometric accuracy, structural robustness, and generalization capabilities. The evaluation begins in Section 5.1 by detailing the spectrum of utilized datasets, which range from controlled environments (DTU MVS, TnT) to uncalibrated real-world drone footage (Pexels). Section 5.2 introduces the core quantitative and quantitative metrics and outlines experimental design. Sections Section 5.3, Section 5.4, and Section 5.5 present the core experimental results and analyses, encompassing direct baseline benchmarking, targeted ablation studies, and broader cross-category comparisons. Finally, Section 5.6 concludes the chapter with a preliminary proof-of-concept demonstrating the integration of the pipeline's extracted planes into KSR [7].

5.1. Datasets

To evaluate the proposed AdaptivePS pipeline across different domains, four datasets representing varying levels of environmental control, calibration quality, and geometric complexity are utilized:

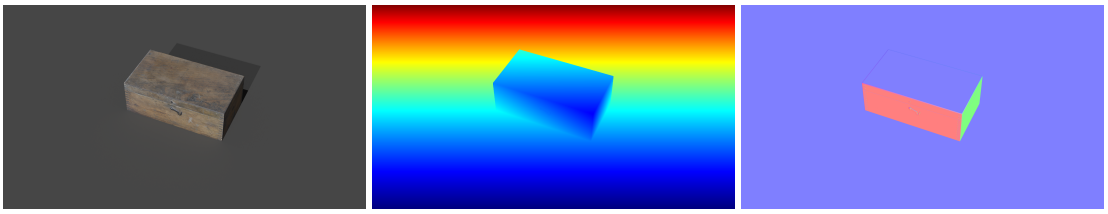
- **Blender Synthetic Box:** An idealized, synthetic environment providing flawless ground-truth depth, normal maps, and camera poses. It is used strictly for the initial verification phase of the baseline PlanarSplatting mechanism.
- **DTU MVS Dataset [25]:** A highly controlled laboratory dataset. The scenes are explicitly filtered into *building* and *others* categories, with evaluation restricted to the building-centric subsets to align with the assumptions of this framework.
- **Tanks and Temples Dataset [31]:** A standard real-world benchmark derived from high-resolution video tracks. While a six-scene subset is standardly used in the literature, only the piecewise-planar *Barn* scene is evaluated here.
- **Pexels Dataset** A custom, uncalibrated dataset compiled from royalty-free online video footage. Lacking ground truth or pre-existing camera parameters, it serves as a qualitative testbed representing consumer-grade drone captures.

The evaluation is strictly focused on isolated architectural structures with clearly visible facades; large-scale urban environments and dense cityscapes are outside the scope of this work.

5.1.1. Blender Synthetic Box

This dataset was synthetically generated using Blender[1] and its Python API to serve as the idealized control environment for the diagnostic verification detailed in Section 4.2. The data generation scripts are publicly available[2].

The dataset consists of a multi-view turntable camera orbit rendering a single textured box[3] on a flat ground plane, completely free of background clutter or open skyboxes. As shown in Figure 5.1, the script exports synchronized high-fidelity RGB frames, noise-free depth maps, and analytical surface normal maps alongside exact camera intrinsic and extrinsic parameters, providing the perfect geometric inputs required for core pipeline isolation.



rendered image | rendered depth | rendered normal

Figure 5.1.: Rendered views of the synthetic box scene from blender.

5.1.2. DTU MVS Dataset

The DTU MVS dataset [25] is a widely recognized benchmark standard for training and evaluating NVS and 3D reconstruction frameworks. The complete dataset comprises 124 distinct scenes containing various object categories captured under highly controlled laboratory conditions.

The dataset was acquired using an industrial robot arm equipped with a structured-light scanner, providing precise ground-truth geometry and camera extrinsics. Each scan consists of either 49 or 64 RGB images captured at a resolution of 1600×1200, and is pre-processed to eliminate dark current noise and defective pixels. While the dataset provides 8 lighting configurations for each scene, this work utilizes the most diffuse (minimize harsh reflections and shadows) illumination setup[4] to isolate structural optimization from complex specular artifacts.

To align with the architectural focus of this thesis, the scenes were explicitly filtered and grouped into two sub-categories: *building* and *others*. Only scenes falling into the *building* category are evaluated, as displayed in Figure 5.2.

[1] https://www.blender.org
[2] https://github.com/SherAndrei/blender-gen-dataset
[3] "Old Storage Wooden Box" (https://skfb.ly/o9DXt) by carlcapu9 (https://sketchfab.com/carlcapu9) is licensed under Creative Commons Attribution (https://creativecommons.org/licenses/by/4.0).
[4] Denoted by the identifier "_3_"

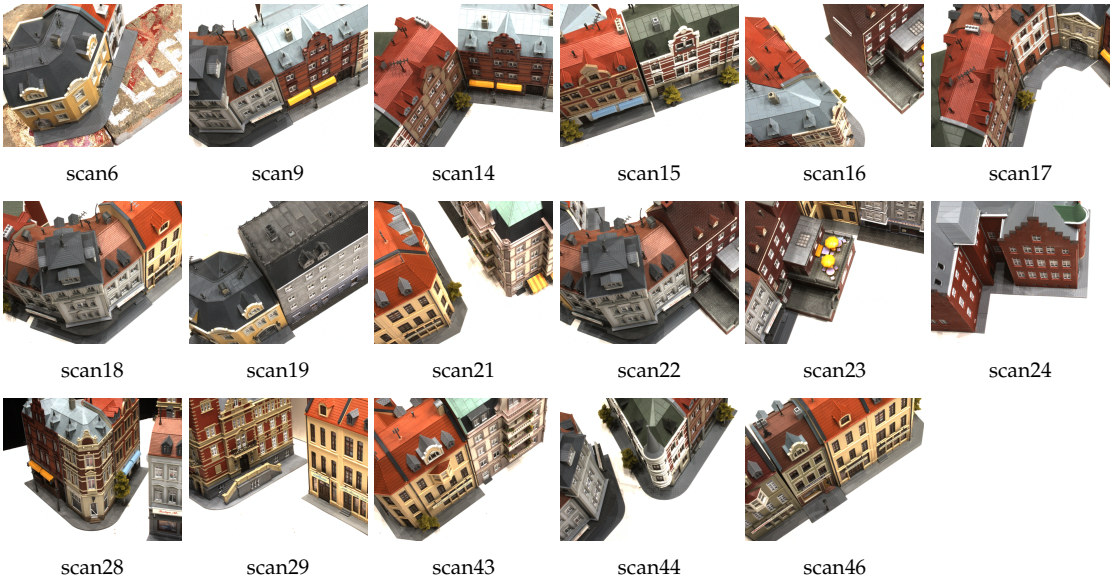


Figure 5.2.: Scenes from the DTU MVS [25] *building* subset.

5.1.3. Tanks and Temples

The TnT dataset [31] is another widely recognized benchmark, designed for evaluating large-scale, image-based 3D reconstruction and NVS algorithms. Captured under realistic conditions, the dataset includes both expansive outdoor tracks and complex indoor environments. The ground-truth geometry was acquired using an industrial laser scanner, providing a high-precision spatial baseline for verification.

The input data is derived from high-resolution video sequences, with extracted evaluation frames utilizing a resolution of 1957×1090. In the NeRF-based and GS-based works, it is standard practice to benchmark algorithms across a common subset of scenes, typically including *Barn*, *Courthouse*, *Caterpillar*, *Ignatius*, *Meetingroom*, and *Truck*, as illustrated in Figure 5.3.

However, because this thesis focuses explicitly on piecewise-planar architectural reconstruction, the majority of the scenes in the benchmark violate the assumptions of the pipeline. Consequently, evaluation within this work is strictly restricted to the *Barn* scene. This scene features prominent planar surfaces along its primary facade and roof sections, making it an ideal testbed for evaluating the proposed adaptations.

5.1.4. Pexels Dataset

The Pexels drone orbit Footage (Pexels dataset) was compiled from royalty-free stock drone footage[5] to test the pipeline on real-world drone captures. The dataset consists of 8 landscape scenes captured at a resolution of 1280×720.

This dataset features no ground-truth geometry, camera calibration parameters, or predefined camera poses. Each scene consists of one to five video clips recorded at 30 frames per second.

[5] https://www.pexels.com

Barn (selected)

Courthouse

Caterpillar

Ignatius

Meetingroom

Truck

Figure 5.3.: Scenes from TnT [31] dataset.

Frames were extracted at a fixed frequency of one frame every 20 video frames, with the total number of images capped at 200 per scene. Due to the absence of reference data, these scenes are used exclusively for qualitative evaluation. The compiled scenes are shown in Figure 5.4.

Figure 5.4.: Scenes from Pexels dataset.

5.2. Evaluation Metrics & Experiment Design

5.2.1. Evaluation Metrics

Since the objective of this thesis is geometric fidelity rather than NVS, standard photometric metrics used in GS frameworks are omitted from this evaluation. Instead, performance is quantified strictly through geometric benchmarks that measure the accuracy and completeness of the reconstructed 3D surfaces against ground truth data.

Qualitative evaluation is conducted across four core criteria: *Artifact Suppression*, *Edge Sharpness*, *Detail Preservation*, and *Coplanar Consolidation*. These specific aspects were selected to evaluate how well the pipeline addresses the inherent challenges of unconstrained outdoor built environments while verifying its ability to generate clean planar primitives suitable for downstream applications.

Quantitative evaluation is executed using open-source toolboxes made for respective datasets: the DTU evaluation protocol is adapted from a Python implementation[6] of the official evaluation code, and the evaluation of TnT relies on the official Python toolbox[7]. These scripts compute two primary geometric metrics: Chamfer Distance and F1-score, alongside with runtime and amount of planes.

Chamfer Distance The Chamfer Distance (d_{CD}) evaluates the global spatial discrepancy between the reconstructed 3D point set P (sampled from the final mesh) and the ground-truth point cloud G. It is calculated as the sum of the average closest-neighbor distances from the

[6] https://github.com/jzhangbs/DTUeval-python
[7] https://github.com/isl-org/TanksAndTemples/tree/master/python_toolbox

reconstruction to the ground truth (accuracy) and from the ground truth to the reconstruction (completeness):

$$d_{CD}(P, G) = \frac{1}{|P|} \sum_{\boldsymbol{x} \in P} \min_{\boldsymbol{y} \in G} \|\boldsymbol{x} - \boldsymbol{y}\|_2 + \frac{1}{|G|} \sum_{\boldsymbol{y} \in G} \min_{\boldsymbol{x} \in P} \|\boldsymbol{x} - \boldsymbol{y}\|_2 \tag{5.1}$$

where $\| \cdot \|_2$ denotes the Euclidean L_2 norm. A lower Chamfer Distance indicates a tighter geometric fit to the ground-truth surface.

F1-Score Using the same point sets defined previously, for a reconstructed point $\boldsymbol{x} \in P$, its distance to the ground-truth cloud G is defined as:

$$e_{\boldsymbol{x} \to G} = \min_{\boldsymbol{y} \in G} \|\boldsymbol{x} - \boldsymbol{y}\|_2 \tag{5.2}$$

These point-to-cloud distances are aggregated to define the Precision $\mathrm{Pre}(d)$ of the reconstruction for a given distance threshold d:

$$\mathrm{Pre}(d) = \frac{1}{|P|} \sum_{\boldsymbol{x} \in P} [e_{\boldsymbol{x} \to G} < d] \tag{5.3}$$

where $[\cdot]$ denotes the Iverson bracket, which outputs 1 if the condition inside is true and 0 otherwise. Precision serves as a direct indicator of geometric accuracy.

Similarly, for a ground-truth point $\boldsymbol{y} \in G$, its distance to the reconstructed point set P is formulated as:

$$e_{\boldsymbol{y} \to P} = \min_{\boldsymbol{x} \in P} \|\boldsymbol{y} - \boldsymbol{x}\|_2 \tag{5.4}$$

The Recall $\mathrm{Rec}(d)$ of the reconstruction, which quantifies structural completeness at the threshold d, is defined as:

$$\mathrm{Rec}(d) = \frac{1}{|G|} \sum_{\boldsymbol{y} \in G} [e_{\boldsymbol{y} \to P} < d] \tag{5.5}$$

Finally, Precision and Recall are combined via a harmonic mean to yield the summary F1-score metric:

$$F(d) = \frac{2 \cdot \mathrm{Pre}(d) \cdot \mathrm{Rec}(d)}{\mathrm{Pre}(d) + \mathrm{Rec}(d)} \tag{5.6}$$

For evaluating DTU MVS dataset, the thresholding distance is set to $d = 2$mm[8]. For evaluating TnT dataset, the preset values from the official toolbox is used, which is $d = 1$cm for *Barn* scene.

In addition to these quantitative metrics, a comprehensive qualitative analysis is performed across all datasets. This qualitative assessment focuses explicitly on identifying domain gaps, evaluating surface smoothness, and detecting the presence of floating background artifacts to differentiate the performance of the baseline and the proposed framework.

[8]Note that when evaluating MVS models on the DTU dataset, the standard thresholding value (d) used to calculate the F1-score is 0.5 mm [61].

5.2.2. Experiments Design

To rigorously assess the performance of the proposed pipeline, the evaluation is structured around 18 total scenes, comprising 17 scans from the DTU MVS *building* subset and the *Barn* scene from the TnT dataset. The evaluation is divided into three distinct testing protocols, mapped out in Table 5.1.

Evaluation Protocol	Type	Specific Scenes / Scans Evaluated
Baseline Comparison	Quantitative	All 17 DTU *building* scans and TnT *Barn*
	Qualitative	All 17 DTU *building* scans, TnT *Barn*, and Pexels dataset
Ablation Studies	Quantitative	All 17 DTU *building* scans
Cross-Category Comparison	Quantitative	DTU scan24 and TnT *Barn*

Table 5.1.: Mapping of evaluated analysis types and specific scenes across the three experimental protocols.

Baseline Comparison A comprehensive qualitative and quantitative comparison is conducted against the baseline PlanarSplatting, accompanied by visual analysis of the reconstruction results. While qualitative evaluations are performed across all designated datasets, quantitative benchmarking is restricted to scenes with precise ground-truth data. Specifically, quantitative evaluations are executed over the *building* subset of the DTU MVS dataset and the TnT *Barn* scene to leverage their precise ground truth.

Ablation Studies To isolate the exact performance gains of each architectural modification, a series of ablation experiments are performed by systematically disabling or altering individual modules. The ablation matrix evaluates the following components:

- **Geometric Prior Source:** Comparing the impact of analytical surface normals derived natively from DA3 depth gradients against the sharper, high-resolution normals predicted by Metric3Dv2.
- **Mesh Post-Processing:** Evaluating reconstruction cleanliness with and without filtering the largest-connected-component during TSDF mesh fusion.
- **Iterative Semantic Pruning:** Assessing optimization stability by enabling and disabling the mask-guided plane splitting and pruning routines detailed in Section 4.3.3.
- **Final Semantic Trimming:** Verifying boundary precision by comparing results with and without the point-level mask-guided trimming executed during the merging stage detailed in Section 4.3.4.

Cross-Category Comparison To contextualize the performance of AdaptivePS within the wider scope of 3D vision, the pipeline is evaluated alongside contemporary methods from differing categories, such as standard GS-based and NeRF-based frameworks. Because these approaches are fundamentally engineered for high-fidelity NVS rather than piecewise-plnar surfaces, this evaluation is not intended as a direct state-of-the-art benchmark. Instead, this

cross-category comparison serves to highlight how different optimization objectives manifest in the final geometry, demonstrating the trade-offs.

5.3. Baseline Comparison

Here is the comparison between the baseline PlanarSplatting and the proposed AdaptivePS. The qualitative part is discussed in Section 5.3.1 , and the quantitative in Section 5.3.2.

5.3.1. Qualitative Analysis

Across all evaluated datasets and scenes, the proposed AdaptivePS method demonstrates a clear and consistent advantage over the baseline framework, yielding significant visual and structural improvements across all qualitative metrics.

DTU MVS Dataset As illustrated by the reconstruction results in Figure 5.5, 5.6, and 5.7, the primary objective of adapting planar splatting for outdoor architectural scenes is successfully realized on the DTU *building* subset. The outcome of AdaptivePS is a substantial upgrade over the baseline, yielding clean, piecewise-planar structures that are suitable to serve as direct inputs for downstream solid optimization frameworks like KSR.

A detailed visual analysis highlights several key improvements:

- **Artifact Suppression:** Background clutter and floating artifacts are effectively eliminated.
- **Edge Sharpness:** As a result, the structural boundaries of the main building geometries maintain sharp, well-defined edges.
- **Detail Preservation:** Intricate architectural elements, (e.g., chimneys, dormers, towers, and spires) are reconstructed with higher fidelity. This suggests that the source of depth prior for supervision in AdaptivePS is more accurate and consistent across multiple views.
- **Coplanar Consolidation:** The proposed merging strategy performs as intended; surfaces that belong to a single structural plane are successfully consolidated, avoiding the fragmented patching observed in the baseline.

Despite these improvements, a minor limitation is observed regarding smaller-scale architectural features. For elements such as windows, doors, or complex dormer geometries, the planar boundaries can occasionally appear ambiguous, resulting in skewed shapes. Nevertheless, the reconstructions on the DTU *building* subset remains highly robust and a significant advancement over the baseline.

TnT Barn Scene The TnT *Barn* scene represents the most complete real-world outdoor capture with available ground truth evaluated in this work, it serves as a highly representative benchmark for the intended use case. To thoroughly assess the performance disparities between the baseline and AdaptivePS on this critical asset, this scene is rendered and evaluated from multiple viewing angles, as shown in Figure 5.8.

A detailed visual analysis yields the following observations:

- **Artifact Suppression:** The background suppression remains consistent with the DTU results, distant background geometry and sky artifacts are effectively eliminated. On top of this, the proposed mask-guided pruning strategy successfully removes unwanted foreground occlusions (e.g., trees, tables, and stools).
- **Edge Sharpness:** The boundaries of the extracted planes are notably less sharp than those observed in the DTU scans. However, the extent of the building mass remain clearly preserved.
- **Detail Preservation:** There is a visible drop in recall, with certain planes (as shown in the 5^{th} row) missing entirely from the final output.
- **Coplanar Consolidation:** The merging quality varies across the structure; as shown in the 6^{th} row, one facade of the building is reconstructed with noticeably lower planar cohesion in more challenging regions of the scan.

The results on the *Barn* scene reveal a drop in geometric fidelity when compared to DTU scenes. Despite these limitations (namely: reduced sharpness, varying planar cohesion, and minor missing planes) the extraction of the primary architectural surfaces remains robust. The generated planar primitives maintain sufficient structural coherence to serve as viable inputs for downstream solid optimization algorithms.

Pexels Dataset As an uncalibrated, real-world dataset, the Pexels scenes introduce intense environmental complexities, including highly cluttered backgrounds and incomplete multi-view coverage stemming from distant drone flight paths. Consequently, the quality of the resulting reconstructions varies significantly across different scenes, as illustrated in Figure 5.9.

The *church-cadeby* scene preserves the highest level of detail, recovering complex geometric features such as side buttresses. Its overall reconstruction quality satisfies the four evaluation criteria to a degree comparable with the highly controlled DTU *building* scans. In contrast, scenes such as *church-chesterfield*, *killingbeck-chapel*, *tower-court*, and *elbphilharmonie* yield intermediate results comparable to the *Barn* scene. For these structures, the primary planar enclosures and global volumetric shapes are recovered; one can clearly identify the building footprint and rough shape. However, finer building elements are lost, resulting in geometry that more closely resembles architectural massing models.

Other scenes highlight specific vulnerabilities in the pipeline. The *moskee-haarlem* reconstruction is incomplete because the upstream SAM3 masking pipeline missed a significant portion of the structure where the building visually blends with the ground plane. Additionally, its small glass dome is overly simplified due to the dome's low pixel coverage across the source views, which fails to generate a sufficient loss gradient to drive optimization. Finally, scenes featuring highly complex geometries captured from extreme distances, namely *wotrubakirche* and *krasna-horka-castle*, represent failure cases. Because the cameras are situated far from the target, the pipeline lacks the necessary pixel-level structural data to resolve crisp planar intersections, causing the reconstructions to degenerate into vague, noisy, and fuzzy planes.

Discussion Across all evaluated scenes, a clear trend emerges: the geometric fidelity and detail preservation of the proposed pipeline are heavily dependent on spatial resolution and camera proximity, with close-up captures consistently yielding vastly better planar reconstructions. On top of that, it is shown that in every scene the background and foreground clutters are removed.

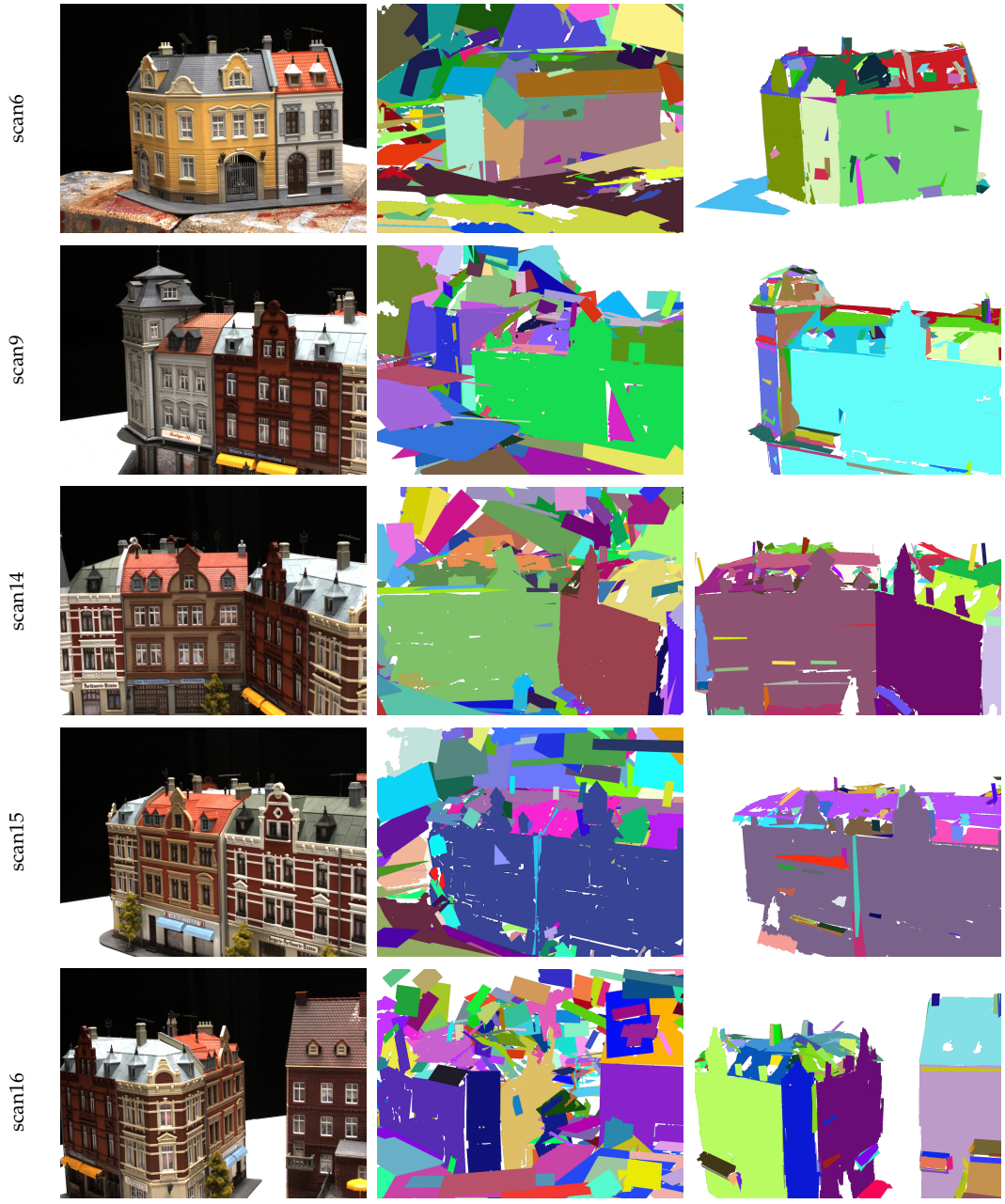


Figure 5.5.: Qualitative comparison on DTU MVS [25] *building* subset.

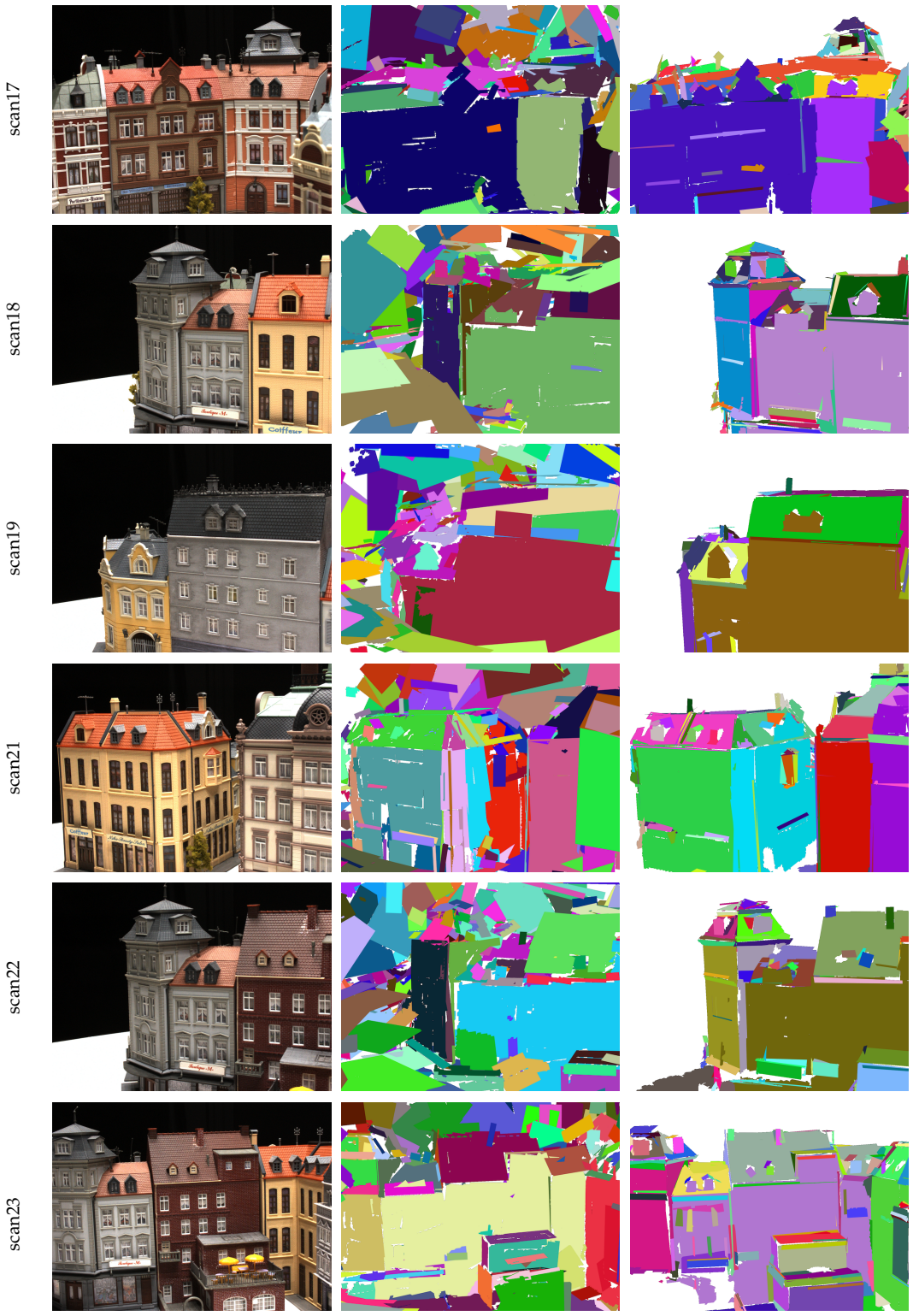


Figure 5.6.: Qualitative comparison on DTU MVS [25] *building* subset (continued).

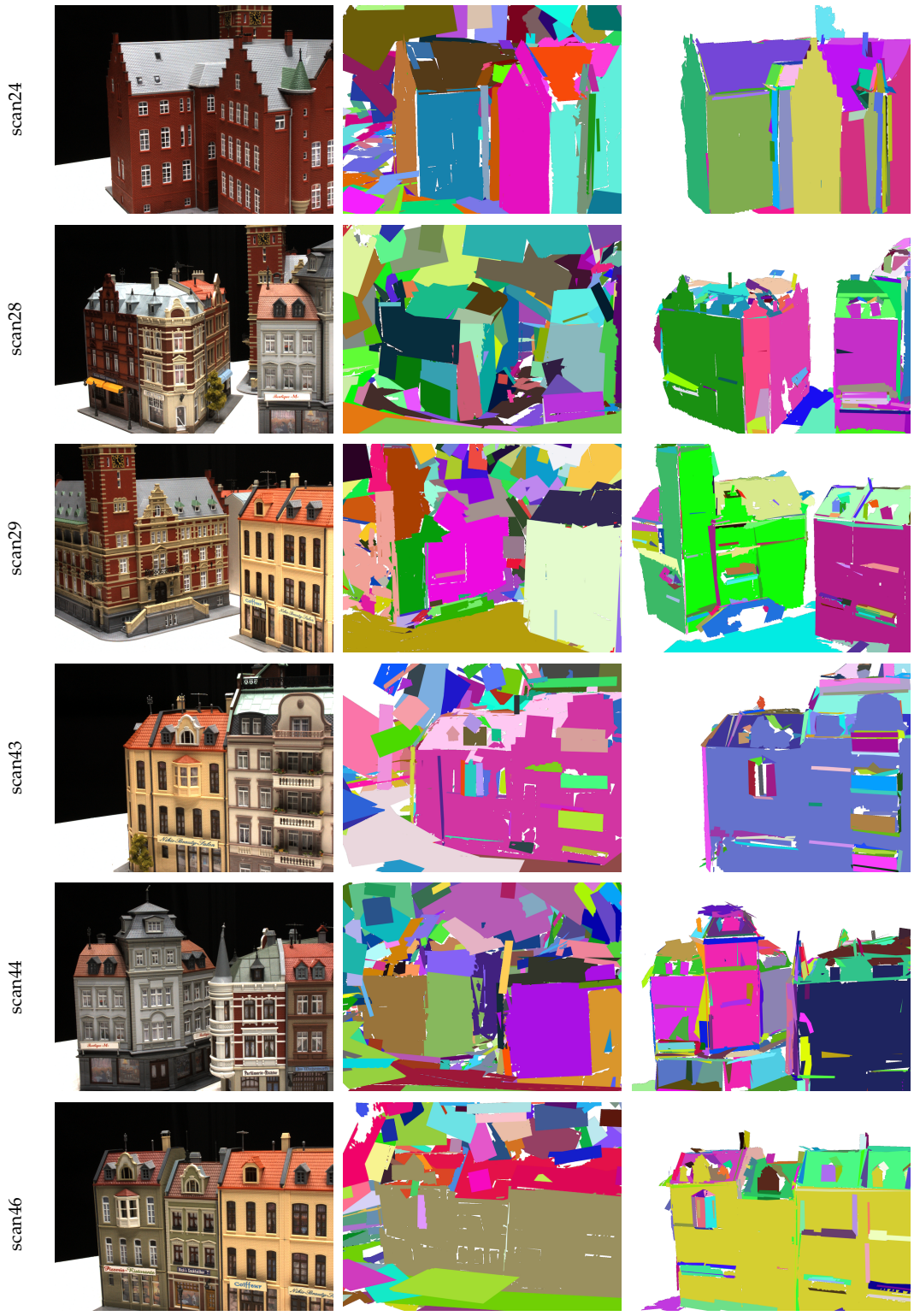
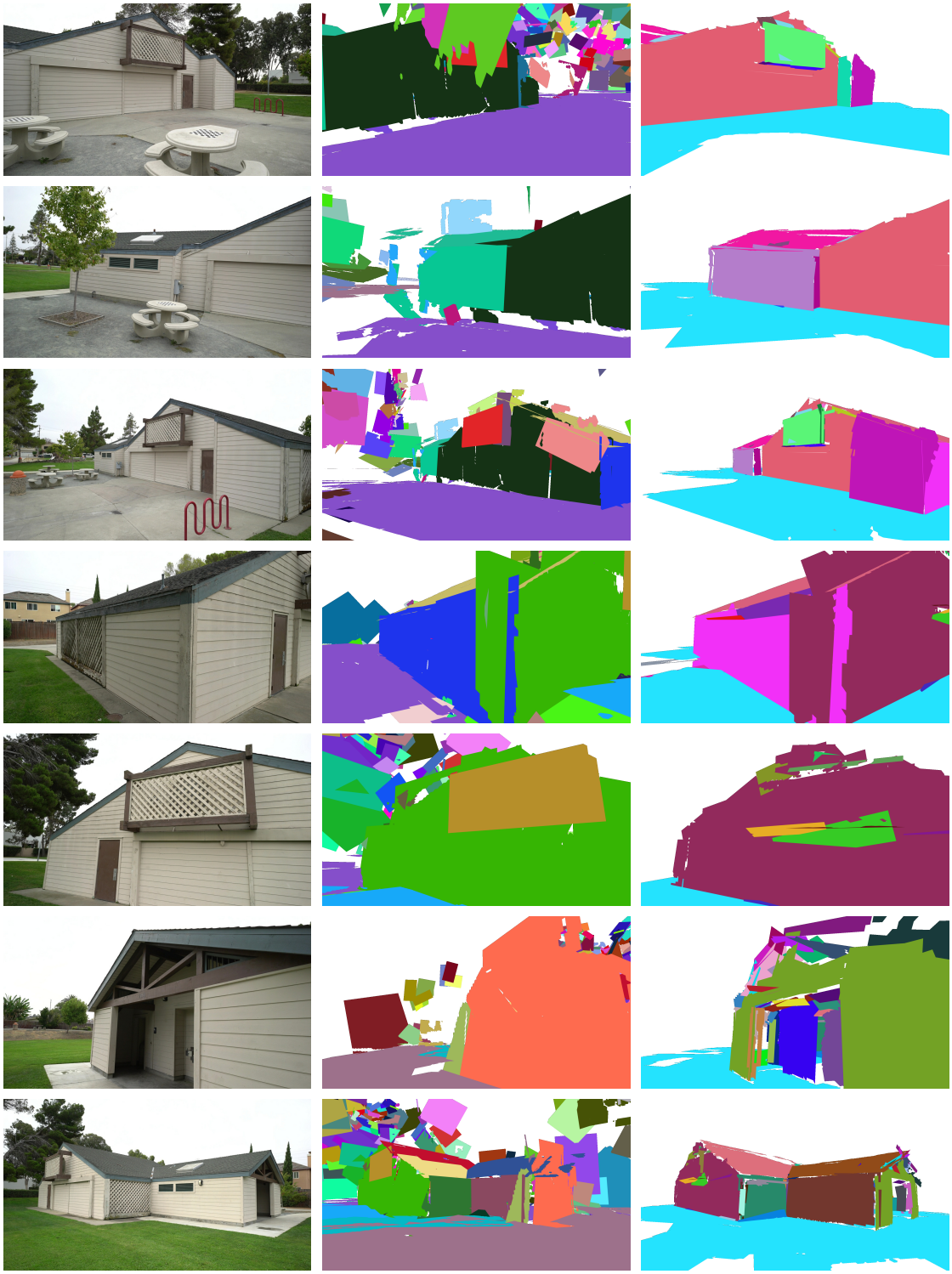


Figure 5.7.: More qualitative comparison on DTU MVS [25] *building* subset (continued).



Input image Baseline AdaptivePS

Figure 5.8.: Qualitative comparison on TnT [31] *Barn* scene.

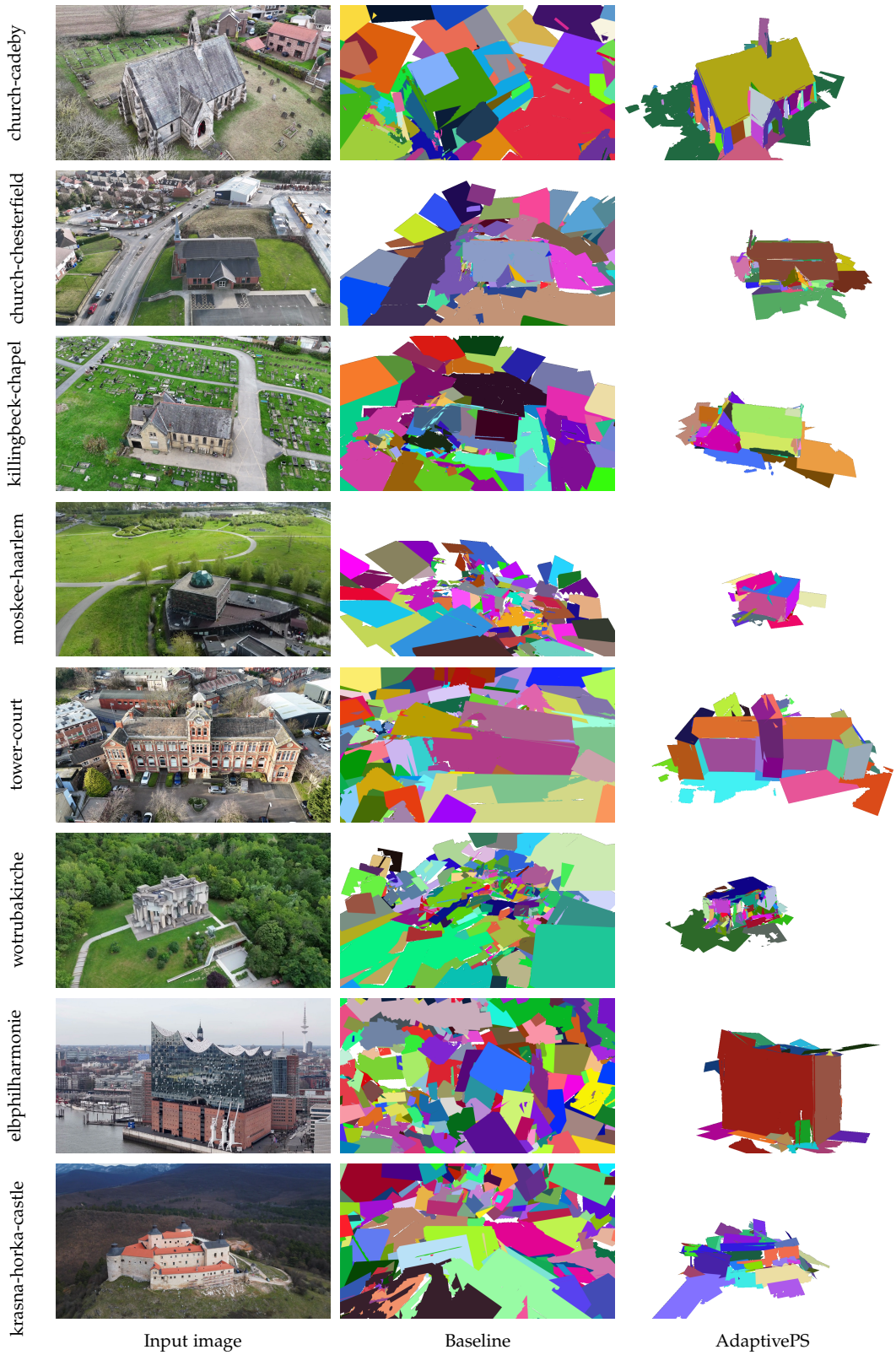


Figure 5.9.: Qualitative comparison on Pexels dataset.

5.3.2. Quantitative Analysis

As established in the experimental design, the quantitative benchmarking is evaluated on the DTU *building* subset and the TnT *Barn* scene to ensure reliable comparisons against high-precision ground-truth data.

DTU MVS Dataset Table 5.2 presents the performance across the four primary metrics (Final Planes, Chamfer Distance, F1-score, and Runtime), and Table 5.3 decomposes the Chamfer Distance into its Accuracy and Completeness components for a deeper structural analysis.

Across nearly all metrics, AdaptivePS outperforms the baseline PlanarSplatting by a significant margin:

- **Chamfer Distance:** On average, the global geometric discrepancy is reduced by more than half, yielding a 2x improvement over the baseline.
- **F1-Score:** Structural integrity and completeness are substantially improved, with the F1-score averaging a 2x increase.
- **Runtime:** Efficiency is nearly doubled, resulting in a 2x faster execution time on average.

The only exceptions are *scan17*, *scan23*, and *scan46*, where AdaptivePS generated more planes than the baseline. However, while minimizing the plane count is a core objective for lightweight modeling, it is not an absolute indicator of reconstruction quality. In this case, a slightly higher plane count correlates with the extraction of complex geometric details that the baseline aggressively smoothed over or failed to capture. These specific structural differences are illustrated from a bird's-eye perspective in Figure 5.10.

Tanks and Temples: Barn Scene Table 5.4 and Table 5.5 detail the quantitative performance on the unconstrained TnT *Barn* scene. Here, the proposed method surpasses the baseline by an even bigger margin:

- **Chamfer Distance:** The Chamfer Distance is drastically reduced from 0.371 m (37.1cm) down to just 0.065 m (6.5cm). Figure 5.11 visualizes the accuracy and completeness score.
- **F1-Score:** The precision and recall metric sees a massive 6.4x increase.
- **Runtime:** The overall processing time is halved, completing in just under 17 minutes compared to the baseline's nearly 36 minutes.

Discussion Given that the core splatting optimization mechanics remain largely unchanged from the baseline, it was initially hypothesized that the Accuracy (Acc.) scores between the two methods would be comparable. However, the decomposed Chamfer metrics clearly show that AdaptivePS achieves massive gains in both Accuracy and Completeness. This confirms that the architectural shift to a superior depth prior source directly influences the geometric fidelity of the final splatting process.

Furthermore, the 2x speedup observed across both datasets is primarily attributed to the new prior generation module. While the baseline framework processes network inferences sequentially, AdaptivePS executes prior generation in a single inference. This efficiency allows the proposed pipeline to process a standard DTU scan (49 images) in just 10 to 20 seconds, and the highly dense *Barn* scene (410 images) in only 75 seconds.

Table 5.2.: Quantitative comparison on the DTU [25] *building* subset. Chamfer Distance is measured in millimeters. Shaded cells indicate the top performance for each metric.

	Final Planes ↓		d_{CD}(mm)↓		F1-score @ 2mm ↑		Runtime ↓	
Scan	Baseline	AdaptivePS	Baseline	AdaptivePS	Baseline	AdaptivePS	Baseline	AdaptivePS
6	150	94	4.48	1.99	0.23	0.60	488s	225s
9	142	135	4.88	2.70	0.26	0.37	525s	254s
14	152	121	5.72	3.24	0.21	0.27	505s	226s
15	141	102	4.41	2.33	0.26	0.52	501s	226s
16	238	118	6.61	2.49	0.14	0.45	451s	229s
17	138	**191**	6.32	2.87	0.20	0.40	448s	241s
18	169	156	4.66	1.65	0.21	0.72	451s	232s
19	151	63	4.82	2.44	0.29	0.41	406s	224s
21	223	179	3.73	1.78	0.38	0.71	348s	230s
22	231	174	4.94	1.55	0.23	0.76	330s	234s
23	147	**176**	3.97	2.08	0.36	0.63	338s	251s
24	140	82	4.52	1.65	0.21	0.69	335s	229s
28	190	127	6.87	2.84	0.09	0.39	317s	247s
29	184	127	7.09	3.50	0.09	0.31	321s	229s
43	190	121	4.33	1.72	0.24	0.70	335s	215s
44	213	179	5.80	2.33	0.19	0.54	332s	213s
46	122	**134**	4.88	2.06	0.21	0.64	328s	216s
Mean	171.8	134.1	5.18	2.31	0.22	0.53	408s	231s

Table 5.3.: Detailed Chamfer Distance breakdown on the DTU [25] *building* subset. Acc. (accuracy) and Comp. (completeness) are measured in mm. Shaded cells indicate top performance.

	Acc. ↓		Comp. ↓		Overall ↓	
Scan	Baseline	AdaptivePS	Baseline	AdaptivePS	Baseline	AdaptivePS
6	3.87	1.32	5.09	2.67	4.48	1.99
9	5.04	2.41	4.71	2.99	4.88	2.70
14	5.33	2.74	6.12	3.75	5.72	3.24
15	4.21	1.92	4.61	2.74	4.41	2.33
16	6.96	2.30	6.25	2.68	6.61	2.49
17	5.95	2.53	6.68	3.21	6.32	2.87
18	4.66	1.41	4.66	1.89	4.66	1.65
19	5.23	2.13	4.40	2.76	4.82	2.44
21	3.95	1.50	3.51	2.06	3.73	1.78
22	4.68	1.39	5.21	1.71	4.94	1.55
23	4.24	1.55	3.70	2.60	3.97	2.08
24	4.59	1.45	4.46	1.86	4.52	1.65
28	6.04	2.66	7.70	3.01	6.87	2.84
29	6.99	2.92	7.19	4.08	7.09	3.50
43	4.07	1.30	4.58	2.13	4.33	1.72
44	5.47	2.11	6.12	2.54	5.80	2.33
46	4.64	2.11	5.11	2.01	4.88	2.06
Mean	5.06	1.98	5.30	2.63	5.18	2.31

Table 5.4.: Quantitative comparison on the TnT [31] dataset. Chamfer Distance is measured in meters. Shaded cells indicate the top performance for each metric.

	Final Planes ↓		CD (cm) ↓		F1-score @ 1cm ↑		Runtime ↓	
Scan	Baseline	AdaptivePS	Baseline	AdaptivePS	Baseline	AdaptivePS	Baseline	AdaptivePS
Barn	402	98	37.1	6.5	0.0062	0.04	35.75m	16.85m

Table 5.5.: Detailed Chamfer Distance breakdown on the TnT [31] dataset. Acc. (accuracy) and Comp. (completeness) are measured in cm. Shaded cells indicate top performance.

	Acc. ↓		Comp. ↓		Overall ↓	
Scene	Baseline	AdaptivePS	Baseline	AdaptivePS	Baseline	AdaptivePS
Barn	54.7	5.6	19.5	7.4	37.1	6.5

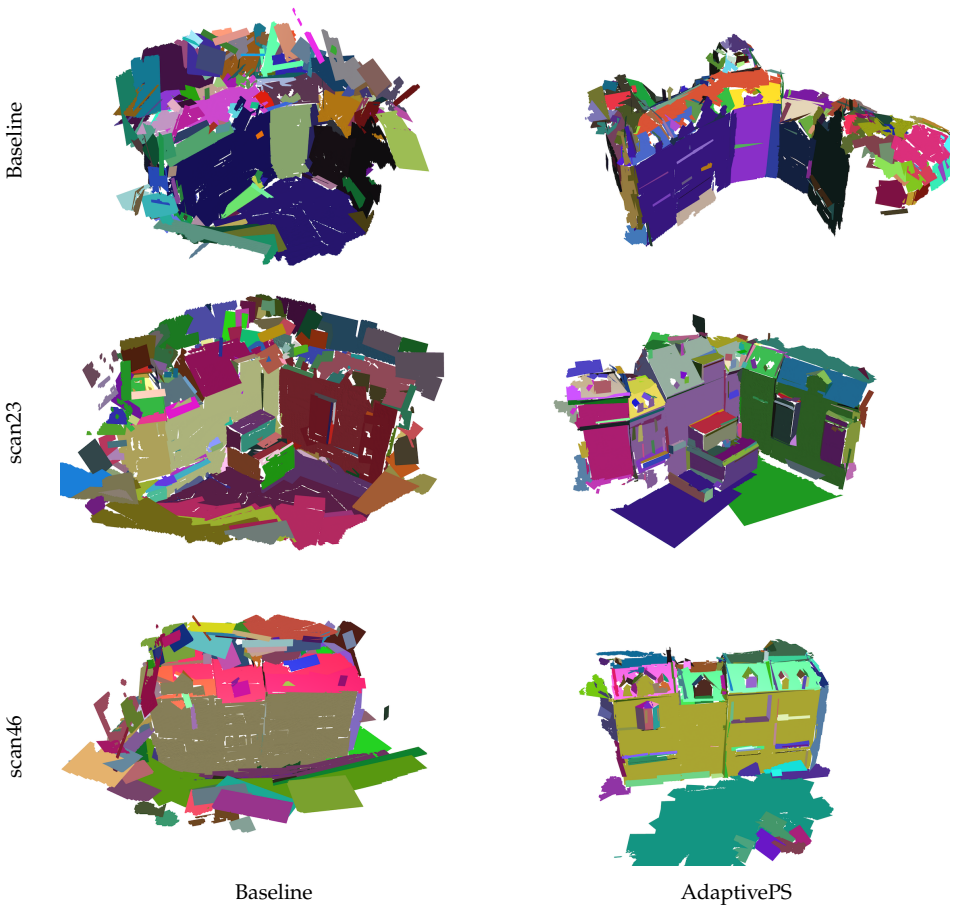
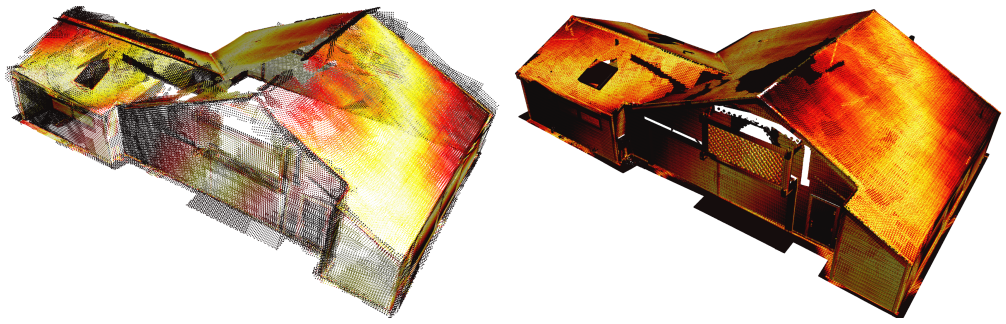


Figure 5.10.: Bird view of the three spacial cases in DTU *building* subset (where the proposed method have more planes compare to baseline).



(a) precision (accuracy) (b) recall (completeness)

Figure 5.11.: Visualizations of TnT [31] *Barn* scene precision and recall. The point cloud is color-coded by distance error, where white and yellow represent low errors, orange and red represent moderate errors, and black represents high errors ($> 6.5cm$).

5.4. Ablation Studies

This section presents an ablation study conducted on the DTU *building* subset and the TnT *Barn* scene to isolate and quantify th impact of key modifications. Specifically, three core proposed adaptations are evaluated: mesh post-processing (detailed in Section 4.3.5), mask-guided densification and pruning (Section 4.3.3), and final mask-guided trimming (Section 4.3.4).

The quantitative results are summarized in Table 5.6 and Table 5.7. Because the *Barn* scene has real-world complexities that are absent from the controlled DTU environment, its metrics are considered a more reliable indicator of practical robustness. Therefore, an accurate assessment requires synthesizing the trends across both tables simultaneously.

Core Module Contributions As anticipated, the full model achieves the highest overall geometric accuracy. Conversely, the "leave-all-out" configuration (where the 3 proposed modules are disabled) yields the poorest results. It should be noted that even with all three modules disabled, the pipeline is not equalvalent to the exact baseline PlanarSplatting, as other adjustments (such as the underlying normal prior source) remain active in the ablation baseline.

An interesting trend emerges regarding the semantic pruning/trimming components: if either the mask-guided densification & pruning or the final mask-guided trimming is disabled, the method still functions at a comparable level. This is expected, as both modules fundamentally target the removal of geometry at different stages of the pipeline. However, the final trimming operates on discretely sampled points from the extracted planes rather than prune/split entire planar primitives during the optimization phase. This allows for more intricate and precise boundary. While the metrics from the DTU *building* subset suggest that the final mask-guided trim (applied post-optimization) is more important than the densification-stage pruning, the controlled nature of the DTU environment suggests caution in over-interpreting this specific margin; both remain critical for unconstrained scenes.

The Normal Prior Discrepancy The most surprising observation is from the source of the normal priors. The first row in both tables reflects a full-model run where the normal priors (derived natively from the DA3 depth gradients) are swapped back to the original Metric3Dv2 [23] normal predictions used by the baseline.

As discussed in Section 4.3.2, GPU VRAM constraints forced a resolution reduction for the DA3 depth maps. Because the derived normals are calculated directly from these reduced depth gradients, they are visibly inferior, both in terms of resolution and edge sharpness. Consequently, utilizing the derived normals was originally hypothesized to degrade performance.

Instead, as demonstrated in the tables, utilizing the highly detailed Metric3Dv2 normals damages the reconstruction accuracy. Visual comparisons of these normal priors and their corresponding reconstruction results are shown in Figure 5.12 and 5.13. A possible explanation for this counterintuitive result is that the sharp details captured by Metric3Dv2 force the planar primitives to falsely fit to micro-structures; and the down-sampled, lower resolution maps implicitly did the jod of smoothing the surfaces.

Table 5.6.: Ablation study on the DTU [25] *building* subset (taking mean values). "Red", "Orange" and "Yellow" denote the top 1-3 results.

Model Setting	Planes no. ↓	CD (mm)↓	F1-score @ 2mm ↑
Replace normal source (to Metric3Dv2 [23])	101.8	2.88	0.45
Leave-one-out:			
w/o Mesh post-processing	137.8	2.37	0.53
w/o Mask-Guided Densification & Pruning	134.9	2.31	0.54
w/o Final Mask-Guided Trim	146.5	2.36	0.51
Isolation:			
Only Mesh post-processing	155.6	2.52	0.49
Only Mask-Guided Densification & Pruning	167.4	2.39	0.52
Only Final Mask-Guided Trim	140.4	2.33	0.54
Leave-all-out:			
All 3 modules disabled	167.4	2.54	0.49
Full model	134.1	2.31	0.53

Table 5.7.: Ablation study on TnT [31] *Barn* (taking mean values). "Red", "Orange" and "Yellow" denote the top 1-3 results.

Model Setting	Planes no. ↓	CD (cm)↓	F1-score @ 1cm ↑
Replace normal source (to Metric3Dv2 [23])	68	9.20	0.02
Leave-one-out:			
w/o Mesh post-processing	144	9.37	0.03
w/o Mask-Guided Densification & Pruning	104	6.68	0.03
w/o Final Mask-Guided Trim	98	6.59	0.04
Isolation:			
Only Mesh post-processing	115	8.47	0.02
Only Mask-Guided Densification & Pruning	184	9.49	0.03
Only Final Mask-Guided Trim	211	10.81	0.03
Leave-all-out:			
All 3 modules disabled	222	13.83	0.02
Full model	98	6.53	0.04

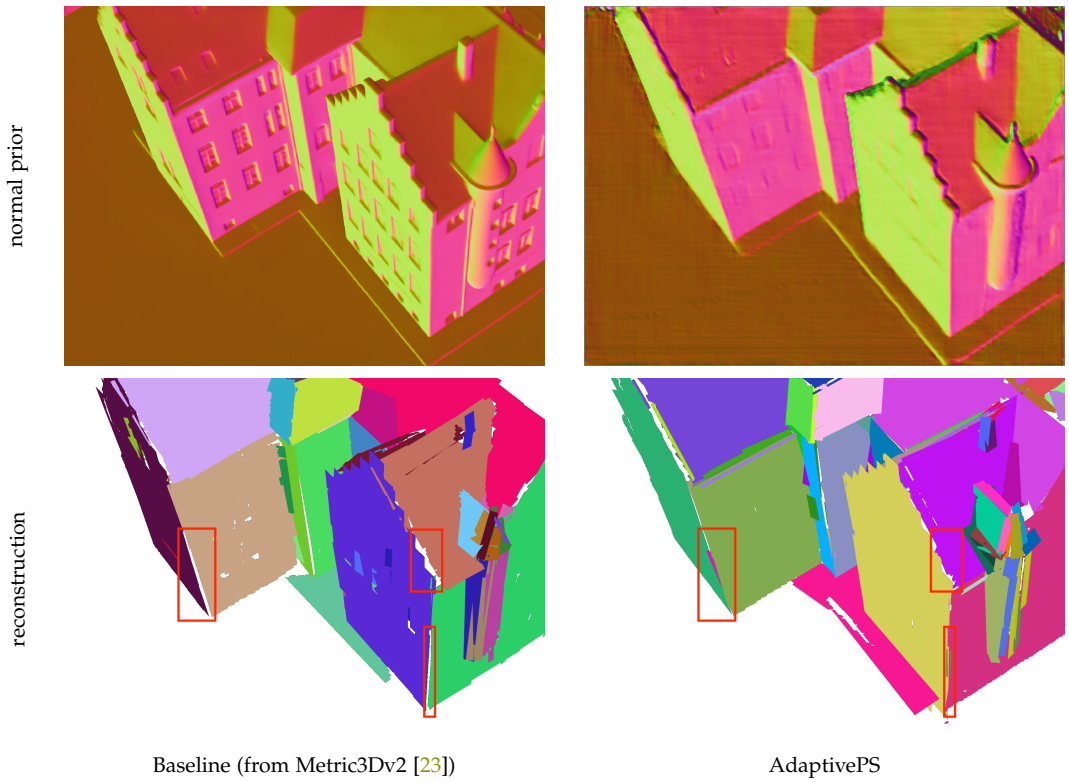


Figure 5.12.: Visualizations of the 2 normal prior source. Sampled from *scan24*.

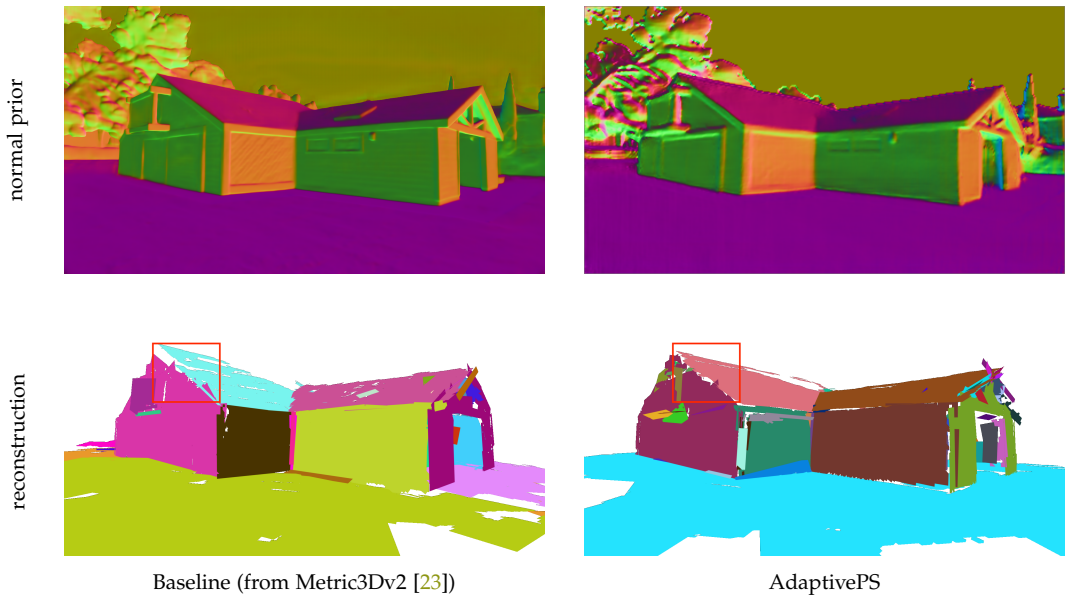


Figure 5.13.: Visualizations of the 2 normal prior source. Sampled from *Barn*.

5.5. Cross-Category Comparison

This section evaluates the proposed pipeline alongside NeRF-based and GS-based NVS and surface reconstruction frameworks. As established in Section 5.2.2, this cross-category comparison is restricted to DTU *scan24* and the TnT *Barn* scene due to the limited evaluation overlap available in the literature.

The benchmarking includes classical neural implicit methods (NeuS [56], Neuralangelo [37]) and recent splatting-based architectures (3DGS [28], SuGaR [20], 2DGS [24], GOF [70], RaDe-GS [71], and PGSR [11]). For *scan24*, the comparison relies solely on Chamfer Distance. In contrast, only the F1-score at a 1cm threshold is reported for the *Barn* scene.

It must be acknowledged that the proposed piecewise-planar representation inherently trades localized geometric accuracy for structural simplicity. Methods generating dense triangular meshes can naturally conform to any surface topographies, even the planar ones. In contrast, forcing geometry into planar constraints is inherently a downsampling operation. Moreover, it is important to note that the proposed pipeline discards standard RGB supervision entirely. Because it optimizes solely on depth and normal priors, it fundamentally cannot leverage the photometric consistency losses and multi-view constraints utilized by frameworks like PGSR and RaDe-GS to refine details. Consequently, as the quantitative results in Table 5.8 and Table 5.9 demonstrate, the proposed method is significantly outperformed across standard accuracy metrics by these works.

However, while AdaptivePS ranks lower in comparison, evaluating the absolute metric values provides crucial context regarding its practical viability. On *scan24*, the reconstructed planar surfaces deviate from the ground truth by only 1.65mm (and 2.31mm on average across the entire DTU building subset). Similarly, on the TnT *Barn* scene, the average deviation is merely 6.5cm. These absolute values demonstrate that while the proposed method cannot compete with the micro-surface fidelity of state-of-the-art dense mesh extractors, its macroscopic structural accuracy remains highly robust and tightly aligned with the physical reality of the scenes.

Table 5.8.: Cross-category comparison on the DTU [25] dataset. Metrics denote Chamfer Distance in mm. ”Red”, ”Orange” and ”Yellow” denote the top 1-3 results.

	NeRF-based		Splatting-based						
scan24	NeuS	Neuralangelo	3DGS	SuGaR	2DGS	GOF	RaDe-GS	PGSR	Proposed
$d_{CD}\downarrow$	1.00	0.37	2.14	1.47	0.48	0.50	0.40	0.31	1.65

Table 5.9.: Cross-category comparison on the TnT [31] dataset. Metrics denote F1-score @ 1cm. ”Red”, ”Orange” and ”Yellow” denote the top 1-3 results.

	NeRF-based		Splatting-based						
Barn	NeuS	Neuralangelo	3DGS	SuGaR	2DGS	GOF	RaDe-GS	PGSR	Proposed
F1-Score↑	0.29	0.70	0.13	0.14	0.36	0.51	0.43	0.66	0.04

5.6. Proof of Concept: Downstream Integration with KSR

To evaluate the viability of the proposed method, this section presents a preliminary proof-of-concept connecting the outputs of AdaptivePS directly to a downstream manifold solver. The evaluation utilizes the official CGAL implementation of Kinetic Shape Reconstruction[9]. Testing is restricted to the TnT *Barn* scene and the unconstrained Pexels dataset, as the **DTU!** (**DTU!**) building scenes lack the closed, 360-degree coverage necessary for watertight manifold reconstruction.

To comprehensively analyze the pipeline's performance and the structural impact of the intermediate data formats, three distinct data sources were evaluated:

- **COLMAP SfM + MVS:** The standard pipeline, producing a highly dense, unstructured point cloud. This was processed using the original, unmodified KSR algorithm.
- **AdaptivePS-Sampled:** A dense point cloud resampled from the explicit planes generated by AdaptivePS. This is processed via the original KSR algorithm to test if Region Growing handles sampled splats better than identified planes.
- **AdaptivePS (Direct):** The explicit planar instances and mesh vertices generated by the proposed pipeline. This required a modified version of the KSR algorithm engineered to ingest planar instances directly, bypassing the Region Growing initialization entirely.

All configurations utilized the same set of empirical parameters (detailed in the supplementary repository). Runtime metrics were recorded using identical hardware setups, with COLMAP benefiting from GPU acceleration.

Tanks and Temples: Barn Scene As shown in Table 5.10 and visually confirmed in Figure 5.14, the integration reveals a computational and geometric trade-off.

Geometrically, the traditional COLMAP-MVS pipeline is vastly superior. It successfully captures finer topological details, such as the window fence structure on the building. This outcome aligns with the benchmark quality demonstrated in the official KSR documentation. In contrast, both the direct AdaptivePS and the resampled AdaptivePS outputs function as second-tier approximations. They successfully preserve the dominant macro-structures (e.g., footprint, primary roof planes and facades), but fail to resolve finer geometric intersections.

However, the computational cost of COLMAP's precision is severe. As shown in Table 5.10, generating the prerequisite data via COLMAP requires over three hours of processing time, whereas the AdaptivePS framework extracts the planar geometry in under 17 minutes—an acceleration of an order of magnitude.

Crucially, the direct injection of AdaptivePS planes performs comparably to the resampled point cloud. This indicates that while the modified KSR integration functions mechanically, the intermediate planar instances produced by AdaptivePS currently lack the strict topological completeness required to rival traditional MVS. The pipeline is not yet fully optimized to guarantee intersection-free adjacency graphs. Consequently, while this proof-of-concept successfully demonstrates that explicit splatted planes can directly drive piecewise-planar solvers at high speeds, refining these outputs to meet production-level geometric standards remains a necessary focus for future work.

[9] https://doc.cgal.org/latest/Kinetic_surface_reconstruction/index.html

Table 5.10.: Runtime comparison for prerequisite data generation prior to KSR optimization.

	COLMAP MVS	AdaptivePS	AdaptivePS (resampled)
Runtime (data creation)	3h 15m 25s	16m 51s	16m 51s

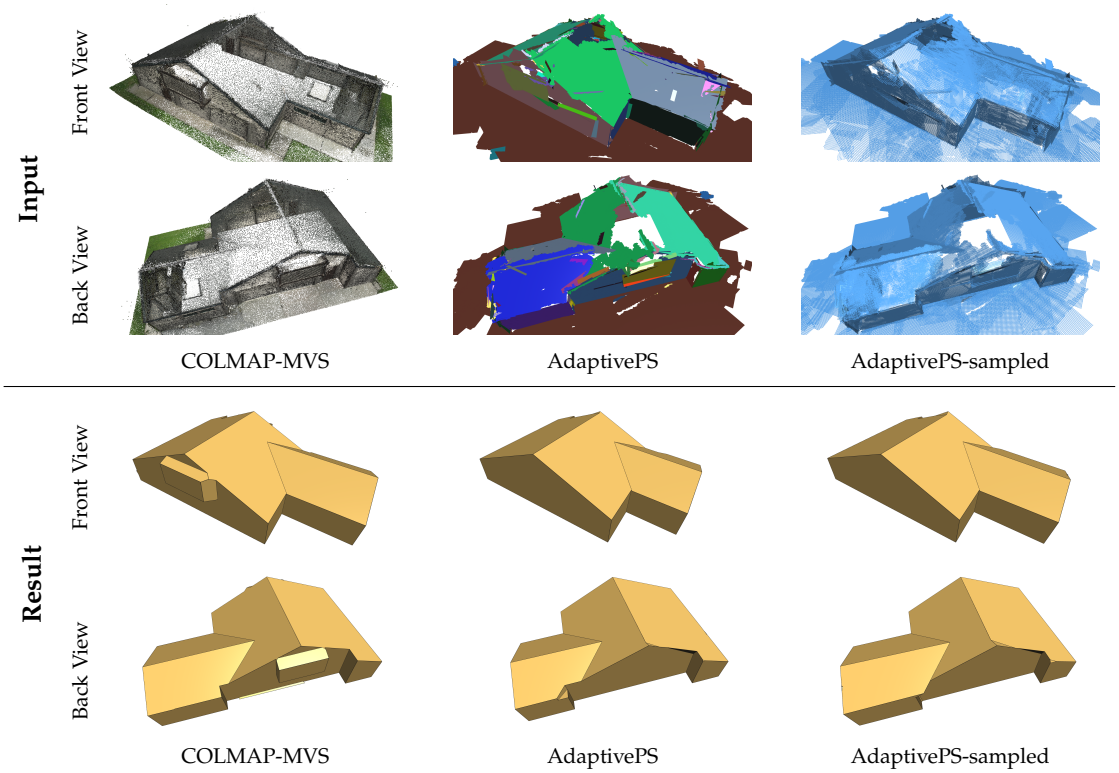


Figure 5.14.: Qualitative comparison of downstream piecewise-planar reconstruction on the *Barn* scene. All models were generated using the same set of fixed parameters.

Pexels Dataset As established in the qualitative analysis (Section 5.3.1, Figure 5.9), the efficacy of planar reconstruction is highly dependent on dataset capture quality and camera proximity. Furthermore, specific Pexels scenes, such as *wotrubakirche* and *krasna-horka-castle*, lack the closed, 360-degree coverage required for watertight manifold extraction. As a result, four representative, fully-closed scenes were selected to showcase this downstream integration.

As shown in Table 5.11 and Figure 5.15, the scenes *church-cadeby* and *killingbeck-chapel* yielded the highest quality planar priors from AdaptivePS. For these scenes, the resulting KSR outputs are the most comparable to the COLMAP-MVS baseline, successfully capturing the dominant architectural volumes, albeit with a slight loss of detail. Conversely, scenes like *church-chesterfield* and *elbphilharmonie* suffered from inadequate initial planar reconstructions, which directly propagated into the final manifold outputs.

In general, the results on Pexels dataset reflects the same performance ranking observed in the TnT *Barn* scene, but the disparity is more pronounced. This gap occurs because the splat-

ted planes struggle to capture architectural details. And this degradation is primarily driven by oblique capture angles, distant camera placements, and the tendency of lower-resolution depth and normal priors to over-smooth intricate geometric features. The runtime metrics also remain highly consistent. As shown in Table 5.11, the proposed method accelerates the prerequisite data generation by up to an order of magnitude compared to COLMAP.

Table 5.11.: Runtime comparison for prerequisite data generation across the Pexels dataset scenes.

Pexels Scene	COLMAP MVS	AdaptivePS	AdaptivePS (resampled)
church-cadeby	25m 52s	5m 49s	5m 49s
church-chesterfield	1h 16m 48s	13m 53s	13m 53s
elbphilharmonie	1h 11m 34s	9m 57s	9m 57s
killingbeck-chapel	39m 04s	6m 56s	6m 56s

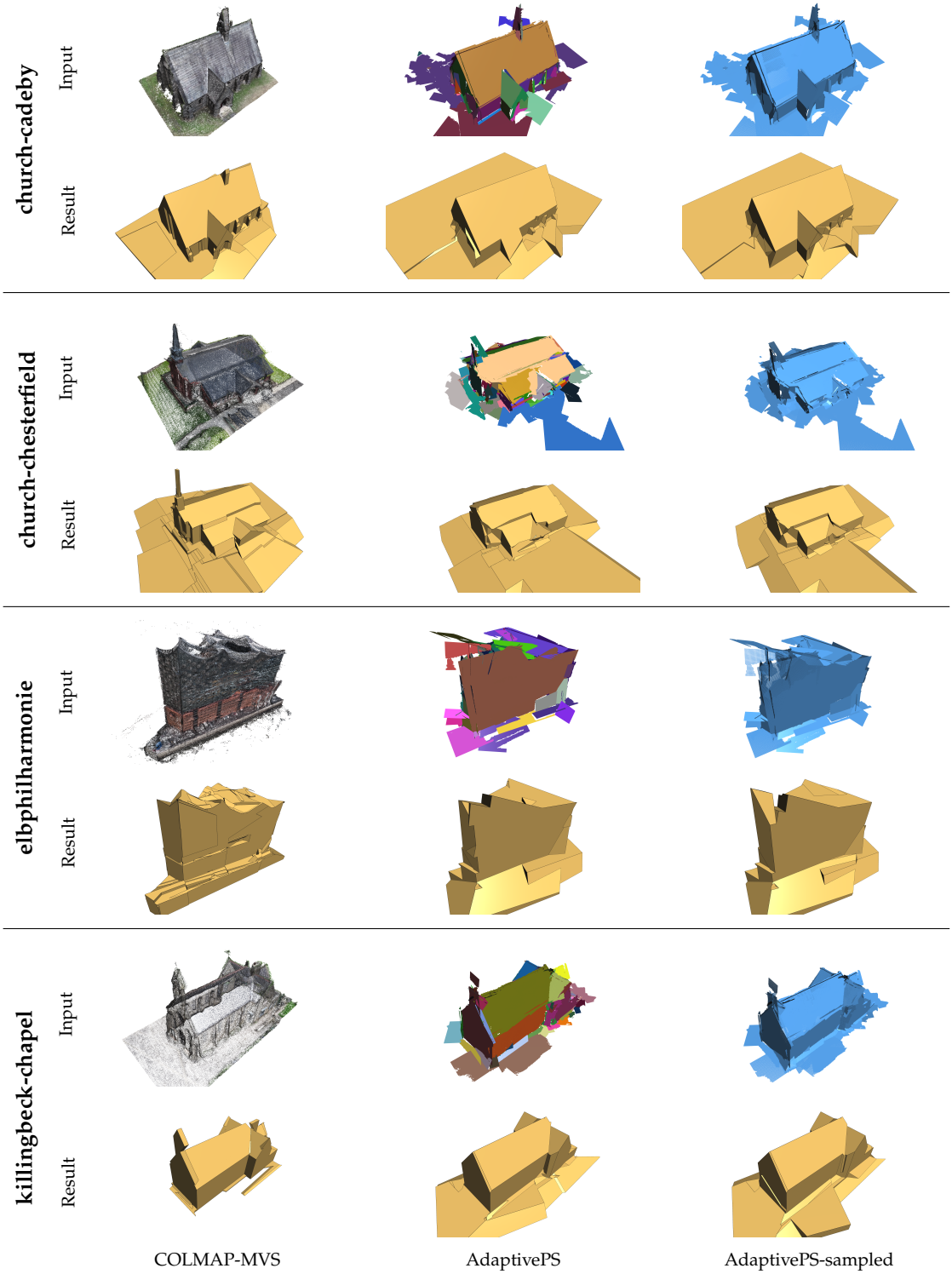


Figure 5.15.: Qualitative comparison of downstream piecewise-planar reconstruction across four scenes from the Pexels dataset.

6. Conclusions

6.1. Research Summary

This section concludes the findings of the research by systematically addressing the main research question and three sub-questions, as initially outlined in Section 1.2.1.

Main Research Question The primary objective of this thesis was driven by the overarching question: *How to optimize the Gaussians (or primitives) towards clusters of bigger bounded planes or polygons?*

A conventional approach to this problem involves applying traditional plane detection algorithms to unstructured point clouds prior to solid optimization. However, this intermediate step introduces significant vulnerabilities.

For instance, PolyFit [43] relies on Fast RANSAC [50] for planar primitive extraction, and KSR [7] initializes primitives via Region Growing [46]. Although their downstream solid optimization algorithms function reliably, the initial primitive extraction degrades significantly when applied to data with occlusions. Additionally, these heuristic methods require extensive hyperparameter tuning to accommodate varying input scales, point densities, and target resolutions. Furthermore, these pipelines face a point-density dilemma: sparse point clouds could yield unstable plane hypotheses, whereas overly dense clouds make the computational cost prohibitive.

To optimize splatted primitives into planes, this work abandons the traditional 3D or 2D Gaussian formulation in favor of the PlanarSplatting architecture, which employs bounded rectangular planes as its base primitive. Optimizing these planes solely through depth and normal priors encourages them to merge and align into broader structural surfaces, rather than fragmenting to capture colored texture. Because these bounded planes are natively optimized during the splatting process, the approach functions as an explicit alternative to the traditional plane-fitting stages required by pipelines such as PolyFit and KSR. While this strategy seeks to bypass the redundant point-conversion detour, preliminary integrations reveal a fundamental computational trade-off: bridging discrete planar splats and manifold solvers sacrifices the exhaustive geometric precision of traditional methods for massive accelerations in processing speed.

Sub-Question 1 *What additional information, training loss or post-processing steps are required?*

After the baseline planar formulation and implementation were established, it became evident that generating planes from outdoor scene captures is more challenging due to the unconstrained nature of such environments. Foreground and background clutter, together with unknown sky regions, can severely degrade reconstruction quality through corrupted depth and normal priors.

To resolve these vulnerabilities, the proposed AdaptivePS introduces several major adaptations and post-processing steps compared to the baseline:

- **Foreground Mask Generation:** A semantic mask generator utilizing SAM3 [9] is introduced to provide reliable references for other downstream modules (Section 4.3.1).
- **Prior Generation:** A faster, unified prior generator based on the feed-forward MVS method, DA3 [38], replaces the baseline source. It jointly recovers camera poses alongside depth and normal maps in a single inference. It also normalizes all scenes to a reasonable, consistent scale; eliminating the need to fine-tune trivial spatial thresholds for plane merging and trimming (Section 4.3.2).
- **Mask-Guided Densification and Pruning:** An adaptive strategy is implemented based on the foreground (building) semantic masks. It adaptively splits primitives at target object boundaries, prunes planes floating in background areas, and splits ambiguous primitives crossing semantic boundaries (Section 4.3.3).
- **Mask-Guided Trimming:** To further eliminate hanging planes post-optimization, a trimming mechanism is applied. This mechanism operates similarly to the densification strategy but it is applied to sampled points from the extracted planes rather than the planar primitives themselves, allowing for intricate boundary delineation (Section 4.3.4).
- **Mesh Post-Processing:** A customized mesh post-processing function is integrated to structurally support the final trimming phase and ensure cleaner outputs (Section 4.3.5).

As demonstrated in Section 5.3, these adaptations have proven vastly superior to the baseline framework. The improvement is visually evident in the qualitative analyses and strongly validated by the gains across all quantitative metrics.

Sub-Question 2 *What level of accuracy could be achieved compared to other scene reconstruction methods?*

As detailed in the cross-category comparison (Section 5.5), the pipeline achieves an accuracy level that is appropriate for its designated task. When benchmarked directly against the baseline PlanarSplatting, AdaptivePS achieves massive performance gains. On the controlled DTU *building* scenes, the average Chamfer Distance is reduced to 2.31mm, representing a 2x improvement. On the unconstrained TnT *Barn* scene, the average Chamfer Distance drops to 6.5cm, marking an approximate 5x improvement over the baseline.

It must be acknowledged that the accuracy of the proposed method is lower when compared to unconstrained, dense NVS and mesh reconstruction frameworks (such as Neuralangelo or PGSR). However, this gap is an expected consequence of the strict plane representation. The absolute metric values, combined with the visual analysis, confirm that while AdaptivePS trades detail for structural simplicity, it provides reliable macroscopic accuracy for modeling applications.

Sub-Question 3 *To what extent can this method be optimized for computation time?*

As reported in the quantitative analysis, AdaptivePS requires only 3.85 minutes (231 seconds) to fully reconstruct a DTU *building* scene on average, and 16.85 minutes for the *Barn* scene from TnT. This represents a roughly 2x speedup over the measured runtime of the baseline PlanarSplatting framework.

The splatting optimization phase accounts for a similar duration in both methods. The reduction in recorded runtime is mostly driven by the modernized prior generator. By leveraging the DA3 [38] feed-forward architecture, the pipeline generates depth maps in a single inference. Furthermore, by analytically deriving surface normals from the DA3 depth gradients, AdaptivePS eliminates the need for a computationally expensive neural inference pass (such as Metric3Dv2 [23]).

Additionally, despite not explicitly factored into the baseline's measured runtime, DA3 jointly recovers global camera poses alongside the depth data. This bypasses the traditional COLMAP SfM process required by most standard pipelines. Therefore, the practical, end-to-end time savings in real-world deployment can be considerably larger than the recorded metrics suggest.

6.2. Contributions

The contributions of this thesis are strictly limited to the adaptations and discoveries made over the baseline frameworks. The core contributions of this work are as follows:

- **Mask-Guided Optimization for Outdoor Scenes:** By introducing a semantic mask-guided pruning and trimming strategy, this work successfully adapts the PlanarSplatting framework to handle the unconstrained outdoor environments. This establishes a reliable, end-to-end pipeline for converting raw 2D images directly into bounded 3D planes. The extracted planar primitives can serve as robust priors for downstream piecewise-planar meshing algorithms, such as PolyFit [43] or KSR [7]. This establishes a high-speed alternative to traditional plane-detection methods.
- **Implicit Smoothing via Low-Res Normal Maps:** This work identifies that the planar-centric optimization fundamentally benefits from the implicit surface smoothing provided by down-sampled, lower-resolution normal maps. By proving that these low-resolution, smoothed, analytically derived normals naturally prevent planar primitives from falsely overfitting to detailed micro-elements, this work improves global structural alignment. Simultaneously, this discovery eliminates a computationally expensive neural inference step, resulting in massive speed improvements.
- **Scale Normalization and Parameter Reduction:** By globally normalizing the spatial scale across all scenes via the new prior generator, the proposed pipeline reduces the user's burden of empirical hyperparameter tuning. The method functions across diverse, uncalibrated datasets with a minimized set of static parameters, improving usability and consistency compared to both the baseline PlanarSplatting framework and traditional heuristic plane-detection algorithms.
- **Direct Integration Proof-of-Concept:** This work establishes a preliminary bridge between planar splatting and KSR. By modifying KSR to directly ingest explicit planar primitives, this work proves the mechanical viability of bypassing the point-conversion detour.

6.3. Limitations

While the proposed AdaptivePS framework successfully extracts planar primitives from unconstrained outdoor scenes, several fundamental and operational limitations must be acknowledged:

- **Strict Planar Constraints:** The core assumption of piecewise-planar geometry inherently restricts the work's applicability. Complex, non-planar architectural features——such as curved domes, cylindrical towers, or highly intricate spires——are unavoidably smoothed over or structurally simplified, making the method unsuitable for specific architectural typologies.
- **Absence of Photometric Supervision:** By entirely discarding RGB input to foster planar clustering, the pipeline sacrifices the ability to leverage multi-view photometric constraints. Without color supervision to guide optimization, the method cannot refine micro-surface geometries to the accuracy levels achieved by state-of-the-art dense mesh extractors.
- **Sensitivity to Spatial Resolution:** The geometric fidelity of the reconstruction is heavily dependent on camera proximity and capture quality. As demonstrated by the distant drone captures in the Pexels dataset, insufficient coverage directly prevents the pipeline from resolving crisp planar intersections, leading to a severe decay in extraction quality.
- **Vulnerability to Upstream Semantic Failures:** The pipeline's structural robustness is limited by the reliability of the upstream semantic mask generator (SAM3). Because it relies on user-guided, prompt-based instance isolation, ambiguous text prompts or visual blending at structural boundaries (e.g., building bases blending into the ground) can result in incomplete masking. These upstream errors could corrupt the downstream planar optimization.
- **Geometric Coarseness in KSR Integration:** While a preliminary proof-of-concept demonstrated that the extracted bounded planes can be directly ingested by downstream solid optimization frameworks (i.e., KSR [7]), the explicit planar outputs currently lack the fine-grained geometric resolution inherently provided by dense point clouds. And the final watertight models fail to match the precision of traditional multi-view stereo methods, limiting their utility for high-fidelity architectural reconstruction.

6.4. Future Works

From the progression of this work, two future trajectories have emerged: completing the immediate operational pipeline, and exploring an ambitious architectural shift for planar reconstruction.

- **Enhancing Planar Granularity for Downstream Solvers:** While the preliminary proof-of-concept established the mechanical viability of directly feeding splatted planes into piecewise-planar algorithms, the geometric coarseness of the input planes currently limits the final output quality. Future research should focus on enhancing the granularity of the planar reconstruction phase itself. Developing optimization strategies to capture finer architectural details within the explicit planar framework—without sacrificing the computational speed of the pipeline—is necessary to close the geometric precision gap between splat-based planes and traditional dense point clouds.

- **A Unified Planar Foundation Model:** Currently, the proposed AdaptivePS framework relies on a modular, sequential integration of distinct feed-forward networks (SAM3 for semantic masking and DA3 for geometric priors). An ambitious future direction is to transcend this modularity through deep neural integration. Inspired by the Vision Transformer architectures, future work could explore training a unified foundation model dedicated to 3D planar surface detection and even optimizing to solid models. Rather than relying on optimization guided by priors from yet another source, such a deeply integrated model could potentially learn to predict and extract bounded architectural planes from unconstrained 2D images in a single feed-forward pass.

A. Implementation Details

A.1. Hardware Setup

All experiments were conducted on the Delft AI Cluster[1]. The underlying system runs on `CentOS 7`, while all experimental environments were containerized using Apptainer[2] with an `Ubuntu 22.04` base environment. Unless stated otherwise, each experiment was executed using a single NVIDIA A40 GPU and 8 CPU threads.

A.2. Plane Model

This section defines the initial state and geometric constraints for the planar primitives. The splats begin as a set of 2,000 samples and evolve through size-clamping milestones. These milestones ensure that the primitives remain within a manageable size range.

Table A.1.: Hyperparameters controlling plane model

Parameter	Value	Description
`init_plane_num`	2000	Initial number of planar primitives to sample and initialize.
`radii_init`	0.1	Initial size assigned to newly generated planar splats.
`radii_max_list`	[0.5, 2]	Maximum allowed plane sizes across milestone stages.
`radii_min_list`	[0.01, 0.01]	Minimum allowed plane sizes across milestone stages.
`radii_milestone_list`	[0, 1000]	Iteration milestones at which radii constraints dynamically update.

A.3. Training

The training hyperparameters govern the temporal flow and optimization weights of the learning process. The model balances monocular depth and normal priors using specific coefficients—λ_1 for normals and λ_2 for depth. A relatively short duration of 5,000 iterations

[1] https://daic.tudelft.nl
[2] https://apptainer.org

is sufficient for convergence, provided the learning rates for centers, rotations, and radii are properly balanced.

Table A.2.: Hyperparameters controlling training process

Parameter	Value	Description
`max_total_iters`	5000	Total training iterations.
`weight_mono_normal`	5.0	Loss weight for the monocular normal consistency constraint (λ_1).
`weight_mono_depth`	2.0	Loss weight for the monocular depth consistency constraint (λ_2).
`lr_radii`	0.002	Learning rate for optimizing plane sizes.
`lr_center`	0.002	Learning rate for optimizing plane center positions.
`lr_rot_normal`	0.002	Learning rate for optimizing plane normal rotations.
`lr_rot_xy`	0.002	Learning rate for optimizing in-plane rotations.

A.4. Geometric Constraints

In the original PlanarSplatting framework, parameters such as `voxel_length`, `sdf_trunc`, and `depth_trunc` dictate the resolution and spatial extent of the TSDF volume. These settings have a direct and significant impact on memory consumption; improper calibration in large-scale outdoor environments frequently leads to Out-Of-Memory (OOM) errors. Furthermore, `depth_trunc` serves as a primary gate for the TSDF outcome, determining which depth values are integrated into the reference mesh.

A major distinction of the proposed pipeline is the removal of the requirement per-scene hyperparameter tuning. While the baseline requires these parameters to be updated according to the specific metric scale of each scene, the similarity transformation introduced in Section 4.3.2 allows for the use of standardized, universal values across diverse datasets.

Additionally, this work identifies and resolves a specific limitation in the baseline regarding the `max_depth` parameter. In the baseline, this acts as a hard limit on loss functions, where only geometry within this range contributes to optimization. In outdoor environments, it traps the clutters by allowing unconstrained artifacts to persist without gradient correction. AdaptivePS resolves this by utilizing semantic masks to ensure the optimization remains focused on the target architecture while maintaining fixed, robust values for `voxel_length` and `sdf_trunc`.

Finally, `init_sphere_radius` defines the bounding volume utilized for initialization. In the proposed pipeline, this serves primarily as a fallback mechanism for instances where a valid TSDF reference mesh cannot be generated.

Table A.3.: Hyperparameters controlling geometric constraints

Parameter	Value	Description
`init_sphere_radius`	6	Radius of the bounding sphere used for initial scene scaling.
`voxel_length`	0.04	Voxel size used in TSDF fusion (`refuse_mesh`) for initial mesh generation.
`sdf_trunc`	0.16	Truncation distance for the TSDF volume integration.
`depth_trunc`	40.0	Maximum depth value integrated into the TSDF volume. It was the only way to
`max_depth`	200.0	Maximum ray-traced depth considered during training and rendering.

A.5. Densification and Pruning

As established in Section 4.3.3, managing the population of primitives is the primary defense against background artifacts. The parameters in Table A.4 define the frequency at which the model "checks" its work. Notably, the `check_plane_vis_freq` also triggers the proposed mask-guided logic, ensuring that semantic gating happens in sync with the baseline's geometric pruning.

Table A.4.: Hyperparameters controlling densification and pruning behaviors

Parameter	Value	Description
`split_thres`	0.10	Average radii gradient magnitude threshold; planes exceeding this are split.
`process_plane_freq`	1000	Frequency of executing the baseline plane processing (split/prune) routines.
`check_plane_vis_freq`	1000	Frequency of checking and updating baseline plane visibility (to prune occluded geometry) and to execute the proposed mask-guided densification and pruning.
`coarse_stage_ite`	1000	Number of iterations for the initial coarse training stage before refinement/splitting begins.
`split_start_ite`	1000	Iteration at which plane splitting operations commence.

A.6. Plane Merging and Trimming

The final stage of the pipeline merges optimized primitives into macroscopic surfaces through two consecutive rounds of refinement (Section 4.3.4). It involves executing the complete "trim and merge" sequence twice to ensure geometric and semantic consistency. The process begins by discretizing the planar primitives into sampled points at a spatial resolution defined by `space_resolution`. These points are then filtered against both the TSDF reference mesh and the semantic masks before being merged based on co-planarity thresholds.

Table A.5.: Hyperparameters controlling plane merging and trimming

Parameter	Value	Description
`normal_angle_thresh`	15.0 / 5.0	Maximum angle difference (degrees) between adjacent coplanar primitives to allow merging.
`dist_thresh`	0.1	Maximum point-to-plane distance for clustering points to a primitive.
`mesh_dist_thresh`	0.01	Distance threshold used to associate reference mesh vertices to a plane.
`mesh_dist_thresh_2`	0.02	Relaxed distance threshold for mesh vertex association in secondary refinement passes.
`space_resolution`	0.02	Spatial resolution parameter used to sample points from planar primitives.
`voxel_size`	0.015	Voxel size used for determining neighbors in the merging step.
`bg_trim_enabled`	True	Toggles whether to prune background points and faces that don't belong to valid primitives.

B. SAM3 Prompts

This appendix details the specific text prompts utilized to generate semantic masks across the datasets. The exact prompts applied to each scene are listed in Table B.1, B.2, and B.3.

To accommodate varying environmental complexities, the building mask extraction was executed in two distinct modes. For the DTU MVS *building* subset, all detected target instances within a single frame are merged together to form one unified mask. For the TnT *Barn* scene and the Pexels dataset, the extraction captures at most one primary instance per frame. This ensures the central target structure is cleanly isolated from clutters.

The generation of ground masks follows an aggregation strategy identical to the first mode. Across all datasets, multiple descriptive prompts may be utilized to identify ground regions, with all detected instances merged into one ground mask per frame.

Table B.1.: Scene names and prompts (DTU [25] *building*)

Scene Name	**Building Prompt**	**Ground Prompt**
All scans	”houses/buildings”	-

Table B.2.: Scene names and prompts (TnT [31] dataset)

Scene Name	**Building Prompt**	**Ground Prompt**
Barn	”The barn house”	”ground”, ”grass”, ”pavement”

Table B.3.: Scene names and prompts (Pexels dataset)

Scene Name	**Building Prompt**	**Ground Prompt**
church-cadeby	”that stone masonry church building”	”ground”, ”grass”
church-chesterfield	the red building with a spire in center of frame	”grass”, ”road”, ”pavement”
killingbeck-chapel	”that stone masonry church building”	”ground”, ”grass”, ”road”
moskee-haarlem	that building in the center of frame”	”water”, ”grass”
tower-court	”that building with clock-tower	”ground”, ”road”, ”pavement”
wotrubakirche	”the modernism concrete building”	”grass”, ”road”, ”pavement”
elbphilharmonie	”Elbphilharmonie, that modernism red-brick and glass building”	”water”, ”road”, ”pavement”
krasna-horka-castle	”that castle building”	”ground”, ”grass”, ”pavement”
clocktower	”that clocktower”	”ground”, ”pavement”

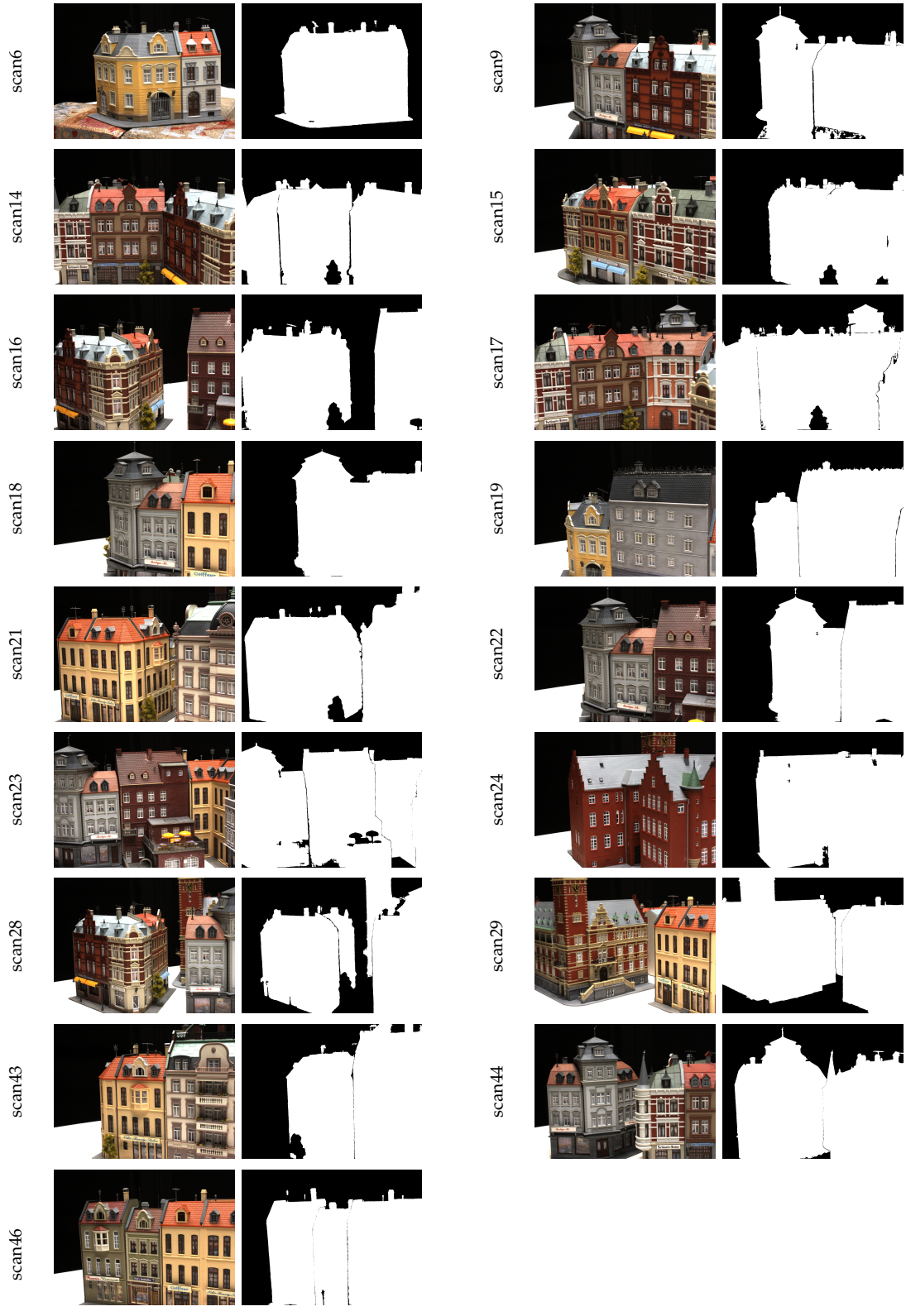
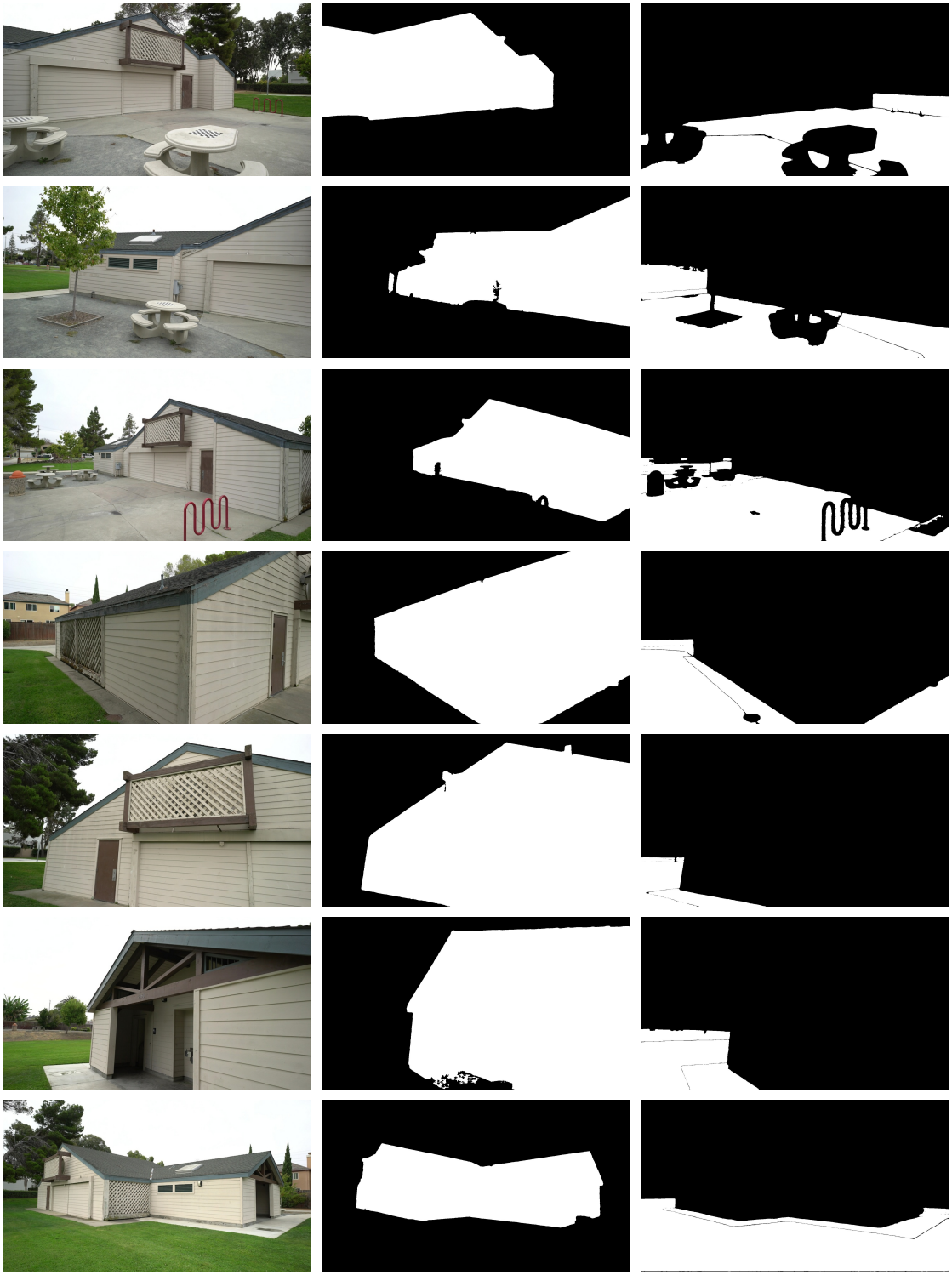


Figure B.1.: Generated masks for DTU MVS [25] *building* subset.



Input image | Building mask | Ground mask

Figure B.2.: Generated masks for TnT [31] *Barn* scene.

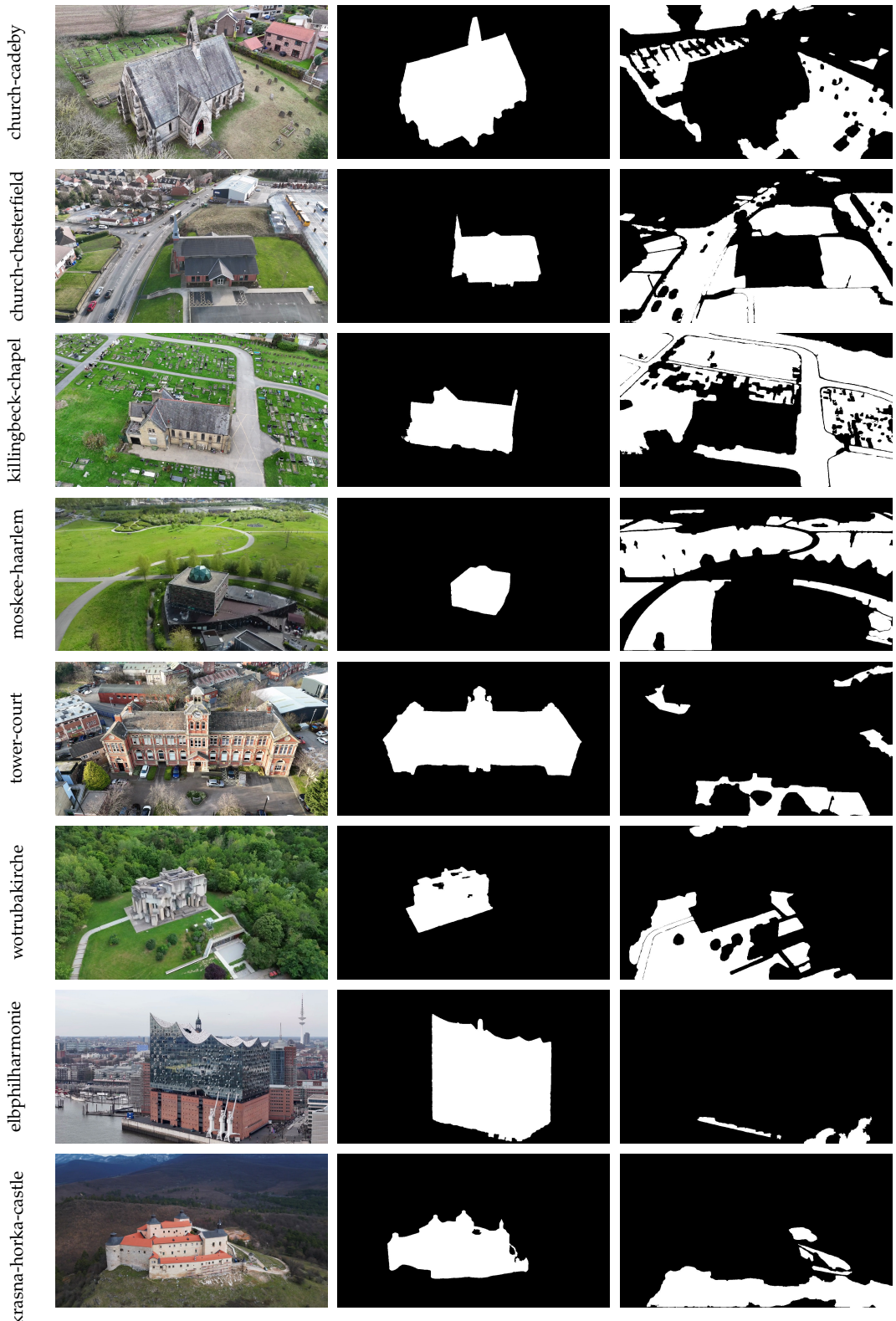


Figure B.3.: Generated masks for Pexels dataset.

C. DTU MVS Dataset Pre-Processing

The DTU MVS dataset originally provides camera calibration as 3×4 projection matrices in millimeters, with slight principal point offsets in the raw images. To prepare the DTU dataset for the 3D reconstruction pipeline, a two-step preprocessing procedure was implemented to extract standardized camera parameters and align the image spaces.

Camera Decomposition The provided camera calibration files contain combined projection matrices $\boldsymbol{P} \in \mathbb{R}^{3\times4}$. For my reconstruction pipeline, it is necessary to explicitly separate the intrinsic camera properties from the extrinsic poses.

RQ decomposition (via `cv2.decomposeProjectionMatrix()`) is applied to factorize the projection matrix into the intrinsic matrix $\boldsymbol{K} \in \mathbb{R}^{3\times3}$ and the extrinsic matrix $[\boldsymbol{R}|\boldsymbol{t}] \in \mathbb{R}^{4\times4}$. Since the same camera was used to capture all viewpoints in a scene, a single, shared intrinsic matrix $\boldsymbol{K}$ is maintained across all views.

Furthermore, the raw DTU dataset represents world-space translations in millimeters. To improve numerical stability and matches the expectations of most neural and geometric pipelines, the translation components of the extrinsic matrices are scaled to **decimeter**:

$$\boldsymbol{t}_{dm} = \boldsymbol{t}_{mm} \times 10^{-2} \tag{C.1}$$

The rotation matrices $\boldsymbol{R}$ (dimensionless) and intrinsic parameters (in pixel units) remain unaffected by this scaling.

Image Alignment Analysis of the extracted intrinsic matrix $\boldsymbol{K}$ reveals that the optical center (principal point) (c_x, c_y) is not perfectly aligned with the geometric center of the given 1600×1200 images ($c_x \approx 823, c_y \approx 619$). As many view-synthesis and standard projection frameworks implicitly assume a perfectly centered principal point, a deliberate crop is done to rectify this offset, as shown in Figure C.1.

A target resolution of 1554×1162 is set. The image cropping boundaries $(left, top)$ are calculated to center the original principal point within the new image dimensions:

$$left = \text{round}(c_x) - \frac{W_{target}}{2}, \quad top = \text{round}(c_y) - \frac{H_{target}}{2}. \tag{C.2}$$

By applying these crop boundaries, the new principal point mathematically shifts to exactly half of the target resolution:

$$c'_x = c_x - left = 777.0, \quad c'_y = c_y - top = 581.0. \tag{C.3}$$

Following the image cropping, the shared intrinsic matrix K is updated with these new (c'_x, c'_y) values.

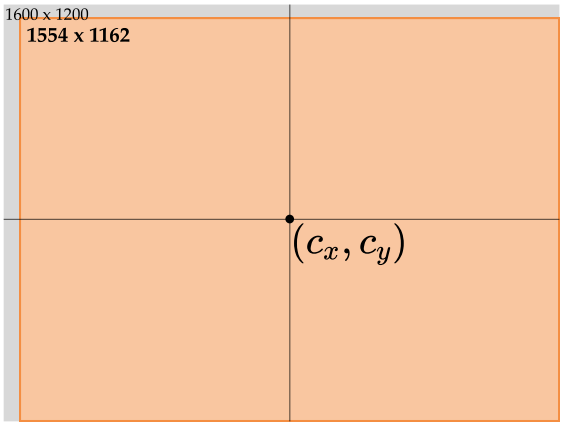


Figure C.1.: Illustration of the clipping operation applied to DTU [25] scenes. Images were cropped to align the image center to camera center (without changing the camera center in world coordinates).

D. Declaration of AI/LLM Usage

In accordance with institutional guidelines, this appendix details the use of Artificial Intelligence (AI) and Large Language Models (LLMs) during the research and writing of this Master's thesis.

Tools During the preparation of this thesis, I utilized Google Gemini[1] and GitHub Copilot[2]. GitHub Copilot was integrated into Visual Studio Code (VSCode) and leveraged underlying models including ChatGPT, Claude, and Gemini.

Tools Usage These tools were employed for specific, targeted tasks:

- Thesis Writing and Formatting: Both Gemini and GitHub Copilot were used to help with LaTeX syntax formatting. Gemini was primarily used to revise manuscript drafts, improve sentence structure, and correct grammatical errors.
- Literature Search: Gemini served as an searching tool to help identify relevant papers and refine search terms when keywords were ambiguous.
- Programming: GitHub Copilot was utilized within VSCode to trace execution paths and improve my comprehension of the codebase. It was also used to generate specific code snippets based on direct prompts.

Extent of Use The tools were used regularly as assistance rather than main content (or code) generators. For programming, Copilot was used strictly on-demand for isolated code snippets; no autonomous coding agents were created or deployed to write software independently. For writing, Gemini provided regular drafting and editing assistance but did not author original academic arguments.

Adherence to Academic Integrity I confirm that all LLM-generated outputs were reviewed, fact-checked, and verified before inclusion in this work. I confirm that it contains no falsified, fictional, or plagiarized material. Furthermore, I verified that all code snippets generated and incorporated into the codebase adhere to open-source licensing requirements.

[1] Google Gemini: https://gemini.google.com/app
[2] Github Copilot: https://github.com/features/copilot

E. Reproducibility Self-Assessment

All source code and analysis scripts related to this thesis are openly accessible at https://github.com/MCHU-1999/AdaptivePS. Complete instructions and a sample workflow are provided in the project README. The experiments are based on the DTU MVS dataset [25, 1] and the TnT benchmark [31].

DTU MVS dataset is available at https://doi.org/10.1109/CVPR.2014.59 and https://doi.org/10.1007/s11263-016-0902-9. TnT dataset is available at https://doi.org/10.1145/3072959.3073599. I rate the reproducibility of this thesis as **High** according to the provided scale.

The overall reproducibility is self-rated using the following scale:

1. **High:** All data and code fully available, well-documented, and tested to reproduce main results.
2. **Moderate:** Most data and code available, basic documentation, some steps may require clarification.
3. **Low:** Limited or no data/code sharing, minimal documentation, reproduction unlikely.

Bibliography

[1] Aanæs H, Jensen RR, Vogiatzis G, Tola E, and Dahl AB (2016). Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, pages 1–16.

[2] Arroyo Ohori K, Ledoux H, and Peters R (2024). *3D modelling of the built environment*. Delft University of Technology. Version 0.9.

[3] Baker AH, Pinard A, and Hammerling DM (2023). DSSIM: a structural similarity index for floating-point data. doi:10.48550/arXiv.2202.02616. ArXiv:2202.02616 [stat].

[4] Barron JT, Mildenhall B, Tancik M, Hedman P, Martin-Brualla R, and Srinivasan PP (2021). Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. doi:10.48550/arXiv.2103.13415. ArXiv:2103.13415 [cs].

[5] Barron JT, Mildenhall B, Verbin D, Srinivasan PP, and Hedman P (2022). Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. doi:10.48550/arXiv.2111.12077. ArXiv:2111.12077 [cs].

[6] Barron JT, Mildenhall B, Verbin D, Srinivasan PP, and Hedman P (2023). Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. doi:10.48550/arXiv.2304.06706. ArXiv:2304.06706 [cs].

[7] Bauchet JP and Lafarge F (2020). Kinetic Shape Reconstruction. *ACM Transactions on Graphics*, 39(5):1–14. ISSN 0730-0301, 1557-7368. doi:10.1145/3376918.

[8] Biljecki F, Stoter J, Ledoux H, Zlatanova S, and Çöltekin A (2015). Applications of 3d city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4):2842–2889. ISSN 2220-9964. doi:10.3390/ijgi4042842.

[9] Carion N, Gustafson L, Hu YT, Debnath S, Hu R, Suris D, Ryali C, Alwala KV, Khedr H, Huang A, Lei J, Ma T, Guo B, Kalla A, Marks M, Greer J, Wang M, Sun P, Rädle R, Afouras T, Mavroudi E, Xu K, Wu TH, Zhou Y, Momeni L, Hazra R, Ding S, Vaze S, Porcher F, Li F, Li S, Kamath A, Cheng HK, Dollár P, Ravi N, Saenko K, Zhang P, and Feichtenhofer C (2025). SAM 3: Segment Anything with Concepts. doi:10.48550/arXiv.2511.16719. ArXiv:2511.16719 [cs].

[10] Chauve AL, Labatut P, and Pons JP (2010). Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1261–1268. IEEE, San Francisco, CA, USA. ISBN 978-1-4244-6984-0. doi:10.1109/CVPR.2010.5539824.

[11] Chen D, Li H, Ye W, Wang Y, Xie W, Zhai S, Wang N, Liu H, Bao H, and Zhang G (2025). PGSR: Planar-based Gaussian Splatting for Efficient and High-Fidelity Surface Reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 31(9):6100–6111. ISSN 1077-2626, 1941-0506, 2160-9306. doi:10.1109/TVCG.2024.3494046. ArXiv:2406.06521 [cs].

[12] Chen Z, Tagliasacchi A, and Zhang H (2020). BSP-Net: Generating Compact Meshes via Binary Space Partitioning. doi:10.48550/arXiv.1911.06971. ArXiv:1911.06971 [cs].

[13] Chen Z, Wu X, and Zhang Y (2024). Nc-sdf: Enhancing indoor scene reconstruction using neural sdfs with view-dependent normal compensation.

[14] Cheng HK, Oh SW, Price B, Schwing A, and Lee JY (2023). Tracking Anything with Decoupled Video Segmentation. doi:10.48550/arXiv.2309.03903. ArXiv:2309.03903 [cs].

[15] Curless B and Levoy M (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 303–312. Association for Computing Machinery, New York, NY, USA. ISBN 0897917464. doi:10.1145/237170.237269.

[16] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, and Houlsby N (2021). An image is worth 16x16 words: Transformers for image recognition at scale.

[17] Eftekhar A, Sax A, Bachmann R, Malik J, and Zamir A (2021). Omnidata: A Scalable Pipeline for Making Multi-Task Mid-Level Vision Datasets from 3D Scans. doi:10.48550/arXiv.2110.04994. ArXiv:2110.04994 [cs].

[18] Goesele M, Snavely N, Curless B, Hoppe H, and Seitz SM (2007). Multi-view stereo for community photo collections. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. doi:10.1109/ICCV.2007.4408933.

[19] Guo H, Peng S, Lin H, Wang Q, Zhang G, Bao H, and Zhou X (2022). Neural 3D Scene Reconstruction with the Manhattan-world Assumption. doi:10.48550/arXiv.2205.02836. ArXiv:2205.02836 [cs].

[20] Guédon A and Lepetit V (2023). SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering. doi:10.48550/arXiv.2311.12775. ArXiv:2311.12775 [cs].

[21] Han L, Zhang X, Song H, Shi K, Liu YS, and Han Z (2025). Sparserecon: Neural implicit surface reconstruction from sparse views with feature and depth consistencies.

[22] Hoppe H, DeRose T, Duchamp T, McDonald J, and Stuetzle W (1992). Surface reconstruction from unorganized points. *SIGGRAPH Comput. Graph.*, 26(2):71–78. ISSN 0097-8930. doi:10.1145/142920.134011.

[23] Hu M, Yin W, Zhang C, Cai Z, Long X, Wang K, Chen H, Yu G, Shen C, and Shen S (2024). Metric3Dv2: A Versatile Monocular Geometric Foundation Model for Zero-shot Metric Depth and Surface Normal Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10579–10596. ISSN 0162-8828, 2160-9292, 1939-3539. doi:10.1109/TPAMI.2024.3444912. ArXiv:2404.15506 [cs].

[24] Huang B, Yu Z, Chen A, Geiger A, and Gao S (2024). 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–11. doi:10.1145/3641519.3657428. ArXiv:2403.17888 [cs].

[25] Jensen R, Dahl A, Vogiatzis G, Tola E, and Aanæs H (2014). Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE.

[26] Jin X, Jin R, Li B, Zou D, and Yu W (2025). PlanarGS: High-Fidelity Indoor 3D Gaussian Splatting Guided by Vision-Language Planar Priors. doi:10.48550/arXiv.2510.23930. ArXiv:2510.23930 [cs].

[27] Kazhdan M, Bolitho M, and Hoppe H (2006). Poisson surface reconstruction.

[28] Kerbl B, Kopanas G, Leimkuehler T, and Drettakis G (2023). 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14. ISSN 0730-0301. doi:10.1145/3592433.

[29] Kerbl B, Kopanas G, Leimkühler T, and Drettakis G (2023). 3d gaussian splatting for real-time radiance field rendering.

[30] Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, Gustafson L, Xiao T, Whitehead S, Berg AC, Lo WY, Dollár P, and Girshick R (2023). Segment Anything. doi:10.48550/arXiv.2304.02643. ArXiv:2304.02643 [cs].

[31] Knapitsch A, Park J, Zhou QY, and Koltun V (2017). Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4).

[32] Kolbe TH (2009). Representing and exchanging 3d city models with citygml. In *3D geo-information sciences*, pages 15–31. Springer.

[33] Kopanas G, Leimkühler T, Rainer G, Jambon C, and Drettakis G (2022). Neural point catacaustics for novel-view synthesis of reflections. *ACM Transactions on Graphics (SIGGRAPH Asia Conference Proceedings)*, 41(6):201.

[34] Kopanas G, Philip J, Leimkühler T, and Drettakis G (2021). Point-based neural rendering with per-view optimization.

[35] Ledoux H, Arroyo Ohori K, Kumar K, Dukai B, Labetski A, and Vitalis S (2019). Cityjson: A compact and easy-to-use encoding of the citygml data model. *Open Geospatial Data, Software and Standards*, 4(1):1–12.

[36] Leroy V, Cabon Y, and Revaud J (2024). Grounding image matching in 3d with mast3r.

[37] Li Z, Müller T, Evans A, Taylor RH, Unberath M, Liu MY, and Lin CH (2023). Neuralangelo: High-Fidelity Neural Surface Reconstruction. doi:10.48550/arXiv.2306.03092. ArXiv:2306.03092 [cs].

[38] Lin H, Chen S, Liew J, Chen DY, Li Z, Shi G, Feng J, and Kang B (2025). Depth Anything 3: Recovering the Visual Space from Any Views. doi:10.48550/arXiv.2511.10647. ArXiv:2511.10647 [cs].

[39] Liu C, Tan B, Ke Z, Zhang S, Liu J, Qian M, Xue N, Shen Y, and Braud T (2025). PLANA3R: Zero-shot Metric Planar 3D Reconstruction via Feed-Forward Planar Splatting. doi:10.48550/arXiv.2510.18714. ArXiv:2510.18714 [cs].

[40] Lorensen WE and Cline HE (1987). Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, page 163–169. Association for Computing Machinery, New York, NY, USA. ISBN 0897912276. doi:10.1145/37401.37422.

[41] Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, and Ng R (2020). NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. doi:10.48550/arXiv.2003.08934. ArXiv:2003.08934 [cs].

[42] Müller T, Evans A, Schied C, and Keller A (2022). Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics*, 41(4):1–15. ISSN 0730-0301, 1557-7368. doi:10.1145/3528223.3530127. ArXiv:2201.05989 [cs].

[43] Nan L and Wonka P (2017). PolyFit: Polygonal Surface Reconstruction From Point Clouds. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2353–2361.

[44] Newcombe RA, Izadi S, Hilliges O, Molyneaux D, Kim D, Davison AJ, Kohi P, Shotton J, Hodges S, and Fitzgibbon A (2011). Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136. doi:10.1109/ISMAR.2011.6092378.

[45] Oechsle M, Peng S, and Geiger A (2021). UNISURF: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction. doi:10.48550/arXiv.2104.10078. ArXiv:2104.10078 [cs].

[46] Rabbani T, {van den Heuvel} F, and Vosselman G (2006). Segmentation of point clouds using smoothness constraints. In Maas H and Schneider D, editors, *ISPRS 2006 : Proceedings of the ISPRS commission V symposium Vol. 35, part 6 : image engineering and vision metrology, Dresden, Germany 25-27 September 2006*, volume 35, pages 248–253. International Society for Photogrammetry and Remote Sensing (ISPRS). ISPRS commission V symposium : image engineering and vision metrology, ISPRS 2006 ; Conference date: 25-09-2006 Through 27-09-2006.

[47] Ravi N, Gabeur V, Hu YT, Hu R, Ryali C, Ma T, Khedr H, Rädle R, Rolland C, Gustafson L, Mintun E, Pan J, Alwala KV, Carion N, Wu CY, Girshick R, Dollár P, and Feichtenhofer C (2024). SAM 2: Segment Anything in Images and Videos. doi:10.48550/arXiv.2408.00714. ArXiv:2408.00714 [cs].

[48] Ren T, Liu S, Zeng A, Lin J, Li K, Cao H, Chen J, Huang X, Chen Y, Yan F, Zeng Z, Zhang H, Li F, Yang J, Li H, Jiang Q, and Zhang L (2024). Grounded sam: Assembling open-world models for diverse visual tasks.

[49] Ren X, Turkulainen M, Wang J, Seiskari O, Melekhov I, Kannala J, and Rahtu E (2024). AGS-Mesh: Adaptive Gaussian Splatting and Meshing with Geometric Priors for Indoor Room Reconstruction Using Smartphones. doi:10.48550/arXiv.2411.19271. ArXiv:2411.19271 [cs].

[50] Schnabel R, Wahl R, and Klein R (2007). Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26(2):214–226.

[51] Snavely N, Seitz SM, and Szeliski R (2006). Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846. ISSN 0730-0301. doi:10.1145/1141911.1141964.

[52] Tan B, Yu R, Shen Y, and Xue N (2024). PlanarSplatting: Accurate Planar Surface Reconstruction in 3 Minutes. doi:10.48550/arXiv.2412.03451. ArXiv:2412.03451 [cs].

[53] Turkulainen M, Ren X, Melekhov I, Seiskari O, Rahtu E, and Kannala J (2024). DN-Splatter: Depth and Normal Priors for Gaussian Splatting and Meshing. doi:10.48550/arXiv.2403.17822. ArXiv:2403.17822 [cs].

[54] Wang J, Chen M, Karaev N, Vedaldi A, Rupprecht C, and Novotny D (2025). Vggt: Visual geometry grounded transformer.

[55] Wang J, Wang P, Long X, Theobalt C, Komura T, Liu L, and Wang W (2022). Neuris: Neural reconstruction of indoor scenes using normal priors.

[56] Wang P, Liu L, Liu Y, Theobalt C, Komura T, and Wang W (2023). NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. doi:10.48550/arXiv.2106.10689. ArXiv:2106.10689 [cs].

[57] Wang S, Leroy V, Cabon Y, Chidlovskii B, and Revaud J (2024). Dust3r: Geometric 3d vision made easy.

[58] Watson J, Aleotti F, Sayed M, Qureshi Z, Aodha OM, Brostow G, Firman M, and Vicente S (2024). AirPlanes: Accurate Plane Estimation via 3D-Consistent Embeddings. doi:10.48550/arXiv.2406.08960. ArXiv:2406.08960 [cs].

[59] Wolf Y, Bracha A, and Kimmel R (2024). GS2Mesh: Surface Reconstruction from Gaussian Splatting via Novel Stereo Views. doi:10.48550/arXiv.2404.01810. ArXiv:2404.01810 [cs].

[60] Xie Y, Gadelha M, Yang F, Zhou X, and Jiang H (2022). PlanarRecon: Real-time 3D Plane Detection and Reconstruction from Posed Monocular Videos. doi:10.48550/arXiv.2206.07710. ArXiv:2206.07710 [cs].

[61] Yang J, Mao W, Alvarez JM, and Liu M (2020). Cost volume pyramid based depth inference for multi-view stereo.

[62] Yariv L, Gu J, Kasten Y, and Lipman Y (2021). Volume Rendering of Neural Implicit Surfaces. doi:10.48550/arXiv.2106.12052. ArXiv:2106.12052 [cs].

[63] Yariv L, Kasten Y, Moran D, Galun M, Atzmon M, Basri R, and Lipman Y (2020). Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. doi:10.48550/arXiv.2003.09852. ArXiv:2003.09852 [cs].

[64] Ye H, Liu Y, Liu Y, and Shen S (2025). NEURALPLANE: STRUCTURED 3D RECONSTRUCTION IN PLANAR PRIMITIVES WITH NEURAL FIELDS.

[65] Ye M, Danelljan M, Yu F, and Ke L (2024). Gaussian Grouping: Segment and Edit Anything in 3D Scenes. doi:10.48550/arXiv.2312.00732. ArXiv:2312.00732 [cs].

[66] Yifan W, Serena F, Wu S, Öztireli C, and Sorkine-Hornung O (2019). Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics*, 38(6):1–14. ISSN 1557-7368. doi:10.1145/3355089.3356513.

[67] Yu A, Fridovich-Keil S, Tancik M, Chen Q, Recht B, and Kanazawa A (2021). Plenoxels: Radiance Fields without Neural Networks. doi:10.48550/arXiv.2112.05131. ArXiv:2112.05131 [cs].

[68] Yu M and Lafarge F (2022). Finding Good Configurations of Planar Primitives in Unorganized Point Clouds. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6357–6366. IEEE, New Orleans, LA, USA. ISBN 978-1-6654-6946-3. doi:10.1109/CVPR52688.2022.00626.

[69] Yu Z, Peng S, Niemeyer M, Sattler T, and Geiger A (2022). MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction. doi:10.48550/arXiv.2206.00665. ArXiv:2206.00665 [cs].

[70] Yu Z, Sattler T, and Geiger A (2024). Gaussian Opacity Fields: Efficient Adaptive Surface Reconstruction in Unbounded Scenes. doi:10.48550/arXiv.2404.10772. ArXiv:2404.10772 [cs].

[71] Zhang B, Fang C, Shrestha R, Liang Y, Long X, and Tan P (2024). RaDe-GS: Rasterizing Depth in Gaussian Splatting. doi:10.48550/arXiv.2406.01467. ArXiv:2406.01467 [cs].

[72] Zhao L, Bao Z, Xie Y, Chen H, Chen Y, and Li W (2025). TSGaussian: Semantic and depth-guided Target-Specific Gaussian Splatting from sparse views. *Image and Vision Computing*, 162:105706. ISSN 02628856. doi:10.1016/j.imavis.2025.105706.

[73] Zwicker M, Pfister H, Van Baar J, and Gross M (2001). Surface splatting. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 371–378. ACM. ISBN 978-1-58113-374-5. doi:10.1145/383259.383300.

Colophon

This document was typeset using LaTeX, using a modified the KOMA-Script class `scrbook` available at https://github.com/tudelft3d/msc_geomatics_thesis_template. The main font is Palatino.