

MSc thesis in Geomatics

IndoorGML-Constrained Navigation Across Multiple 3D Gaussian Splatting Scenes for Indoor Environments

Mingjie Teo

2026



MSc thesis in Geomatics

IndoorGML-Constrained Navigation Across Discrete 3D Gaussian Splatting Scenes for Indoor Environments

Mingjie Teo

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

Mingjie Teo: *IndoorGML-Constrained Navigation Across Discrete 3D Gaussian Splatting Scenes for Indoor Environments* (2026)

© This work is licensed under a Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

The work in this thesis was carried out in the:



Geo-Database Management Centre
Delft University of Technology

Supervisors: Ir. Edward Verbree
Dr.ir. Martijn Meijers

Co-reader: Prof. Sisi Zlatanova

Abstract

3D Gaussian Splatting (3DGS) has emerged as a compelling method for scene reconstruction by producing compact, photorealistic representations of physical environments in real-time. Its application to indoor spaces, however, introduces a fundamental challenge: reconstructing a multi-room building as a single coherent scene is undermined by the frequent occlusions that walls introduce, degrading the sparse reconstruction on which 3DGS training depends. Per-room reconstruction sidesteps this problem but produces a set of independent scenes in arbitrary coordinate frames that have no shared spatial reference.

This paper presents a pipeline that addresses these challenges by integrating IndoorGML’s topological model as the organisational backbone for multiple independently reconstructed 3DGS scenes. The pipeline proceeds in four stages. First, per-room 3DGS scenes are reconstructed using Structure from Motion (SfM) and Gaussian training from video-captured image sets. Second, an Umeyama similarity transformation is estimated for each room between its (SfM) coordinate frame and a shared architectural floor plan reference, simultaneously resolving scale ambiguity and establishing a common metric coordinate system across all scenes. Third, the IndoorGML model is supplemented with additional attributes, and ingested into a PostGIS database that preserves the *Cell–Node–Edge* topological structure of the original schema. Fourth, a web-based viewer queries this database at runtime to enforce geometrically derived navigation constraints within each room, and to trigger topologically-guided transitions between rooms as the user crosses door boundaries.

The pipeline is demonstrated on three adjacent rooms at the Faculty of Architecture and the Built Environment, of the Delft University of Technology. The site comprises two lecture halls and a corridor space. The prototype achieves sustained real-time rendering performance above 60 frames per second on consumer-grade hardware, with navigation constraints successfully preventing the camera from reaching scene regions outside the training range. The results confirm that IndoorGML’s topological model, when extended to reference 3DGS scene data and spatial transforms, provides a strong framework for multi-scene indoor navigation.

Keywords: 3D Gaussian Splatting, IndoorGML, topology

Acknowledgements

The first and greatest thanks goes to my supervisors, Edward Verbree and Martijn Meijers. Our regular and creative discussions always lead to something interesting to explore, and ensured consistent progress in the project. It is not an exaggeration to state that the thesis would not have been possible without your advice and guidance.

Next, special thanks to Sherwin and Suat Yen, whose help, support and encouragement directly enabled this entire academic journey.

Acknowledgements also go out to my Geomatics cohort, in particular: Lars, Juan, Alexandre, Alena, Carlo, Ming-Chieh and Hadi. The companionship and debates in the Geolab are constant sources of motivation. Special mention as well to my Ultimate Frisbee friends at Force Elektro.

Lastly, my parents and family, for always being there as an anchor.

Contents

1. Introduction	2
1.1. 3D Gaussian Splatting	2
1.2. Research Gap	5
1.3. Proposed Research Angle	6
2. Related Work	8
2.1. 3DGS	8
2.2. Indoor Navigation	10
2.3. IndoorGML	14
2.4. Summary of Related Works	16
3. Research Objective	18
3.1. Research Questions	18
3.2. Scope	19
4. Methodology	20
4.1. Data Capture and Processing	21
4.2. Scene Alignment and Scaling	25
4.3. Topology Creation	28
4.4. Web Viewer Implementation	30
5. Results	34
5.1. Sparse Reconstruction Results	34
5.2. 3DGS Reconstruction Results	34
5.3. Alignment Results	37
5.4. Web Viewer Implementation	38
6. Discussion	42
6.1. Research Objective	42
6.2. Limitations	44
7. Conclusions and Future Work	48
7.1. Conclusion	48
7.2. Future Work	49
A Declaration of AI/LLM usage	52
B Reproducibility self-assessment	54
C Umeyama Similarity Transformation Results	56
C.1 Point-pair Correspondence Errors	56
C.2 4x4 Umeyama Transformation Matrix	57
D References	58

List of figures

Figure 1.1	3DGS pipeline by Kerbl et al., 2023.	3
Figure 1.2	Author's reconstruction of Playground benchmark dataset, using Lichtfeld Studio.	3
Figure 2.1	Instanced panorama image stations encoded as clickable bubbles within a point cloud. Source: (Autodesk, 2025).	11
Figure 2.2	Indoor navigation graph. Source: (Yang & Worboys, 2015).	13
Figure 2.3	Structured Space Model of two connected rooms. Source: (Li et al., 2019).	15
Figure 4.1	Conceptual pipeline.	20
Figure 4.2	Selected rooms for 3DGS reconstruction. Extracted and adapted from inclusivecampusmap.tudelft.nl	21
Figure 4.3	Sparse reconstructions of 0450 (top left), 0470 (top right) and 0807 (bottom). .	24
Figure 4.4	3DGS scene of Cell 0470 after 15,000 iterations.	25
Figure 4.5	Manual geometry and topology creation using InEditor and InFactory.	28
Figure 4.6	Thin versus thick door approach for modelling connectivity. Source: (Li et al., 2019).	29
Figure 4.7	Output to-scale IndoorGML model.	30
Figure 4.8	Door geometry and their intrusion into the adjacent room(s).	33
Figure 5.1	Screenshot of 3DGS reconstruction of corridor 0807.	35
Figure 5.2	Screenshot of 3DGS reconstruction of room 0470.	36
Figure 5.3	Screenshot of 3DGS reconstruction of room 0450.	36
Figure 5.4	Floater distribution (centroid-only) of corridor 0807.	37
Figure 5.5	Visualisation of aligned sparse point clouds.	38
Figure 5.6	Outlines of geometry constraints in the web viewer.	40
Figure 5.7	Overlay of three 3DGS room scenes.	40

List of tables

Table 1	Relative comparison of various visual representation formats for the built environment.	5
Table 2	Data acquisition specifications.	22
Table 3	Schema and data of Cell table.	31
Table 4	Schema and data of Node table.	31
Table 5	Schema and data of Edge table.	32
Table 6	Sparse reconstruction statistics per Cell.	34
Table 7	3DGS reconstruction statistics per Cell.	35
Table 8	Umeyama transformation summary per Cell.	37
Table 9	File size statistics per Cell.	39
Table 10	Point-pair correspondence alignment errors for C1 (0470).	56
Table 11	Point-pair correspondence alignment errors for C2 (0450).	56
Table 12	Point-pair correspondence alignment errors for C3 (0807).	57
Table 13	4×4 Umeyama transformation matrix for C1 (0470).	57
Table 14	4×4 Umeyama transformation matrix for C2 (0450).	57
Table 15	4×4 Umeyama transformation matrix for C3 (0807).	57

1. Introduction

Urban environments are complex systems with multitudinous objects and data spanning multiple scales. To effectively communicate information, three-dimensional (3D) scene representation techniques are crucial in enabling users to visualise simulated extracts of these urban systems. Existing visualisation methods – such as boundary representations, surface mesh models, point clouds and voxel grids – require the user to balance tradeoffs across multiple conditions, including processing time, visual fidelity, and storage size. However, recent developments in neural radiance field (NeRF) representations highlight 3D Gaussian Splatting (3DGS) as a promising new method that creates highly detailed scenes, enabling immersive visualisation.

1.1. 3D Gaussian Splatting

3D Gaussian Splatting was first proposed by Kerbl et al. (2023) as a breakthrough means to enable two major research areas in computer graphics and vision: photorealistic novel view synthesis and real-time scene rendering. In the former, NeRF representations revolutionised graphics due to its ability to maintain high visual fidelity across new synthesised camera views, while also reproducing lighting and view-dependent effects such as shadows (Mildenhall et al., 2021). However, this required the training of a neural network, and rendering the scene involves per-pixel evaluations and ray marching, such that a single frame could take seconds to render. While developments have significantly improved the training process and rendering times down to several frames per second (Müller et al., 2022), the latter objective is still largely infeasible, preventing NeRF from being applied in real-time interactions.

Instead of representing the scene with a fully implicit neural function, Kerbl et al. proposed the use of 3D anisotropic Gaussian ellipsoids, termed splats. These splats are explicit volumetric primitives that, during rendering, are blended together in the camera view to approximate a continuous scene representation (Chen & Wang, 2025). By using GPU-accelerated Gaussian rasterisation, 3DGS is able to balance real-time novel-view synthesis with visual fidelity comparable to NeRF approaches (Kerbl et al., 2023). This balance between rendering quality and computational efficiency is key to the rapid adoption of 3DGS towards applications even beyond the field of computer graphics.

The general processing pipeline to create a 3DGS scene is shown in Figure 1.1. From a set of input images, a sparse point cloud is first generated through Structure from Motion (SfM). The coordinates of these SfM points serve as the initialised centroids of the 3D Gaussian ellipsoids, which additionally encode the 3D covariance matrix denoting its size and shape, a spherical harmonic function that describes its colour given an angle of view, and its opacity. At each iteration, the ellipsoids are projected onto the image plane, sorted by depth and blended into pixels using alpha compositing. The rendered image is then tiled into rasters, and photometric loss calculated at the pixel level against the observed image. The gradients of the photometric loss is then propagated backwards through the compositing equation of each tile to adjust the Gaussian parameters.

To ensure that the quality of the reconstruction is not limited by the number of input sparse points, adaptive density control occurs at regular intervals of the training process. Ellipsoids that have large gradients are densified via splitting into duplicate ellipsoids with inherited parameters for further refinement. Conversely, Gaussians with negligible contribution to the image are

1.1. 3D Gaussian Splatting

simply pruned and removed. This mechanism allows the training process to dynamically change Gaussian densities to more accurately reflect the complexity of the image sub-space.

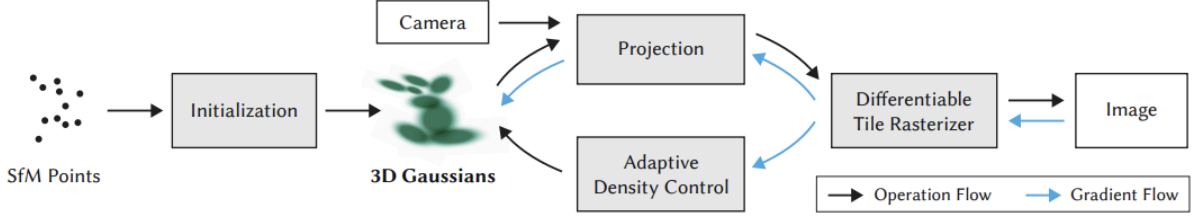


Figure 1.1: 3DGS pipeline by Kerbl et al., 2023.

The resultant 3DGS output is a highly detailed scene with plausible views even at camera angles where there were no source images. An sample 3DGS scene is shown in Figure 1.2, reconstructed from Knapitsch et al. (2017)’s benchmark dataset. The scene was independently recreated using Lichtfeld Studio – a free-to-use application that uses the training process proposed by Kerbl et al. (2023). Objects within the playground are well-captured with texture and shading, especially for the roughness of the sandpit, as well as the specular reflections of the metal slides. Given that the playground is the focus of the source images, objects outside the foreground – such as the house and the trees – still maintain acceptable detail, albeit with noticeable blurring of features. Note that the scene also inherently handles occlusions and uncertainty to an extent – the top of the figure is a data gap in the source image, which the scene represents as a black mask.



Figure 1.2: Author's reconstruction of Playground benchmark dataset, using Lichtfeld Studio.

3D Gaussian Splatting scenes sit in an interesting middle ground between its NeRF predecessor, and geometry-centric representations typically used for visualising the urban environment. NeRFs produce excellent visual quality but suffer from the aforementioned slow training process

and rendering performance. In comparison, 3DGS scenes comprising several hundred images can be trained within an hour, and achieve thirty to sixty frames per second (FPS) when rendering.

Point clouds are essentially sparse samples of the captured geometry, and thus provide highly accurate measurements. 3DGS cannot achieve a comparable level of geometric accuracy; the 3D position of the ellipsoids is itself a parameter to be optimised in the training process. Despite the fact that the initial SfM points typically lie on surfaces, the resultant trained ellipsoids may shift position or hover in empty space.

However for point clouds, as the name suggests, the collection of points is a discrete representation of reality. Increasing point density is the only solution to approximate visual continuity, and this inevitably comes at the cost of file size and rendering performance. In contrast, 3DGS's use of 3D ellipsoids as volumetric primitives means that large surfaces and areas can potentially be represented by a small number of ellipsoids, thus achieving visual continuity with a relatively smaller file size and better rendering performance. Gaussians can also be composited to produce smooth edges and replicate transparency and reflections, whereas point clouds may suffer from sampling artifacts.

Mesh models can provide high visual fidelity, depending on the object in question. However, this method has strong geometric requirements in watertightness, and is laborious to produce. In practice, this means that mesh models are typically handcrafted for a singular object such as a building, thus losing the context of its surrounding objects such as vegetation. BIM and similar models which use boundary representation (BRep) as their geometric foundation are also commonly used to represent key components of the built environment. However, they are typically manually authored or reconstructed and serve as a highly abstracted geometric container for storing design and as-built information.

A relative comparison of the various visualisation methods used in the built environment is suggested, which can be seen in Table 1. For cases where geometric accuracy is not of high importance, 3DGS representations provide a cohesive visual experience for the end-user, whilst also having lower processing and storage costs. It is also important to note that visualisation methods are not mutually exclusive, and 3DGS may complement point clouds to enable accurate measurements while providing a superior visual experience.

Method	Geomet- ric Accu- racy	Process- ing Effort	Visual Fidelity	Realtime Render- ing	Storage Size
Point Cloud	High	High	Medium	Medium	Large
Mesh	High	High	Medium– High	High	Medium
BRep/ BIM	Very High	Very High	Low	High	Small
NeRF	Medium	Very High	Very High	Low	Small– Medium
3D Gaussian Splatting	Low	Medium	Very High	Very High	Medium

Table 1: Relative comparison of various visual representation formats for the built environment.

1.2. Research Gap

Following Kerbl et al.’s paper, a wealth of research continues to improve the shortfalls of the 3DGS method. Various implementations seek to alleviate the computational bottleneck for large scenes, through depth-aware scene partitioning (Hung et al., 2025) or object masking (Wang et al., 2025a). Similarly, methods for improving rendering performance aim to progressively stream important Gaussians (Zoomers et al., 2025), cull occluded Gaussians (Gao et al., 2025a), or provide level-of-detail (LoD) flexibility in selective rendering (see Kulhanek et al. (2025) and Seo et al. (2025)). Multiple splat formats have also been created with varying emphasis on improving compression, load time, rendering quality and memory usage. *.splat* by Kevin Kwok (see <https://github.com/antimatter15/splat>) is lightweight and suitable for web viewing; *.ksplat* by Michael Kellogg (see <https://github.com/mkkellogg/GaussianSplats3D>) has additional cache-optimisation such as splat ordering; *.spz* by NianticLabs (see <https://github.com/nianticlabs/spz>) has 10x compressibility. Regardless, the key trend to note is that research in 3DGS in the past three years focuses on single-scene representations – in other words, how to improve the delivery of a single 3DGS scene.

There is, in general, no current research that investigates how to structure and organise multiple 3DGS scenes. Several researchers have demonstrated means to ingest and process multiple 3DGS scenes. For example, Yuan et al. (2024) uses depth estimates to align overlapping splat files to form a coherent, larger scene. Shen et al. (2026) instead proposes a registration and fusion framework of models with varying lighting conditions of the same scene, thus allowing the scene to take on varying appearances. Eventually, however, these solutions still output a single, albeit enhanced 3DGS scene.

Conceptually, the idea of placing just two 3DGS reconstructions into a singular view seems rather straightforward. Using one reference 3DGS scene, simply import the second scene, then align and scale it using one or multiple known reference points. Indeed, existing commercial

applications such as SuperSplat already provide some functionality to achieve this. However, demonstrations of such functionality are typically limited to placing a singular small reconstructed object – such as a table – in the context of a much larger scene – such as a living room. Intensive manual effort is also required in culling such that the ellipsoids of the reconstructed object lie within the expected geometric bounds of the object itself.

Gaussians that lie outside the object’s expected geometry are known as floaters. As the training process minimises photometric loss instead of geometric error, ellipsoids may deviate from its initial surface-bound position and position themselves in open space to achieve a better photometric result. Furthermore, the spherical harmonic function that describes the expected colour of the ellipsoid is trained on the range of viewing angles of the images in the object, and not on the larger scene. As a result, not removing these floaters of the object will create visual discontinuities and distortions when placed in the larger scene. The issue of floaters is especially pertinent to indoor scene recreations, where complex scene geometry and furniture create large occlusions that result in greater generation of floaters.

1.3. Proposed Research Angle

Despite the proliferation of research into 3D Gaussian Splatting, there is a significant research gap in handling more than a single 3DGS scene. On the other hand, independent 3DGS scenes do not explicitly store external associations with other scenes. A spatial organisation framework is thus necessary to delineate spatial relationships between multiple 3DGS scenes. The ideal context to describe spatial associations lies in indoor environments, where complex subdivision of space and connectivity relationships are crucial for spatial awareness and navigation. This paper thus investigates how existing indoor spatial frameworks – specifically, IndoorGML – can be used to align, organise and structure multiple discrete 3DGS reconstructions of indoor environments.

The proposed research question is thus:

To what extent can discrete 3D Gaussian Splatting scenes be spatially aligned and integrated within an IndoorGML topological model, to enable topologically-constrained and contextually continuous navigation of indoor environments in a web-based viewer?

Current research in 3DGS and indoor navigation are discussed in the following section, to further situate the research angle against contemporary 3DGS solutions and grounded topological frameworks.

1.3. Proposed Research Angle

2. Related Work

The high visual fidelity that 3DGS possesses has obvious advantages for the built environment, some of which are first discussed below. On the other hand, urban structures may also have conditions that make implementing 3DGS visualisation challenging. Unlike large, outdoor places, the compartmentalisation of indoor spaces makes a single monolithic 3DGS scene inefficient to represent the full environment. There is thus potential to explore structuring multiple 3DGS scenes to mimic the distribution of indoor spaces, and how this structure enables navigation between the discrete visualisations. The following sections review related work in 3DGS applications for the built environment, indoor navigation models, and how these concepts may help support structured navigation and interpretation of interconnected 3DGS reconstructions of indoor environments.

2.1. 3DGS

2.1.1. 3DGS and the Urban Environment

Kerbl et al. (2023)’s method sparked a wave of renewed research in Gaussian splats, mostly focused on optimisation at various stages of the 3DGS pipeline (Fei et al., 2025). However, the various advantages of 3DGS also benefit diverse urban applications. New strategies for preserving architectural heritage exploit the visual fidelity and novel-view synthesis of 3DGS to create photorealistic and immersive experiences (see Yu et al. (2025) and Jamil & Brennan (2025)). Further integration with anecdotal narratives enables 3DGS to provide compelling audiovisual storytelling experiences.

In the case of heritage preservation where understanding spatial context and appearance is key, the visual continuity and immersion that 3DGS produces are exceptionally valuable. Gaussian splats enable more intuitive perception of texture, materiality, lighting and spatial enclosure over conventional point cloud or mesh model visualisations. These properties are especially beneficial for urban applications that focus on contextual interpretation of space, such as heritage preservation, facility inspection and management, as well as virtual walkthroughs.

3DGS implementations have also been achieved for large-scale urban scene reconstruction. Yan et al. (2025) finds the method achieves competitive results against traditional photogrammetric methods in terms of reconstruction accuracy, detail fidelity and memory usage. Conversely, 3DGS achieves higher visual fidelity for large urban areas at a much faster speed compared to a Neural Radiance Field (NeRF) approach, at the cost of higher memory usage (Gao et al., 2025b).

While not particularly relevant for indoor spaces, large-scale urban reconstruction efforts highlight an important consideration for 3DGS, that parallels practical limitations seen in point cloud implementations; increasing the scale of capture inevitably causes corresponding increases in memory consumption and storage size. This suggests potential benefits to decomposing a single large-scale scene into multiple smaller scenes to improve the overall rendering and visualisation experience. Since storage space scales with scene size, and indoor spaces are naturally subdivided into multiple smaller spaces, decomposition into multiple discrete 3DGS scenes is both practically advantageous and spatially intuitive.

2.1.2. 3D Splats for Indoor Spaces

Research directions in 3DGS have also overlapped with challenges in the indoor urban environment. Accurate, realistic representation of the real world remains a moving goalpost for 3DGS despite its strength in modelling uncertainties. As such, improving scene reconstruction results remains a popular research angle for 3DGS (Wu et al., 2024).

Indoor environments, however, are fundamentally difficult for image-based reconstruction pipelines. Since 3DGS pipelines are typically initialised from Structure from Motion (SfM) sparse reconstructions, limitations inherent to photogrammetric methods directly affect the quality of Gaussian initialisation. For instance, erroneous initialisation of SfM points is identified to be the cause of excessive floaters and detail loss (Pan et al., 2025). These limitations are pertinent since building interiors comprise large, featureless walls and reflective surfaces, which thus impact the number and quality of feature points that can be extracted from the set of input images. Such low-texture environments require the introduction of artificial texture (Yang & Jiang, 2021) or synthetic noise (Hafeez et al., 2020) to create robust keypoints for the reconstruction process. Similarly, the presence of repetitive objects – such as doors, chairs and tables – also reduces the accuracy of the feature matching process.

To this end, new 3DGS algorithms leveraging the strengths of images and point clouds have emerged. In addition to an image set, Simultaneous Location and Mapping (SLAM) methods collect LiDAR point clouds for accurate point initialisation. For example, LiDGS fuses geometric and depth priors from LiDAR point cloud with multi-view images to improve reconstruction accuracy (Yan et al., 2025). In a similar vein, PlanarGS (Jin et al., 2025) presents a particularly interesting pipeline aiming to resolve ambiguity stemming from planar, low-texture surfaces, which are especially prevalent in indoor spaces. The method uses a pretrained vision-language segmentation model with keyword prompts to segment large planar regions from individual images and provide initial bounding box proposals. These planar proposals are cross-verified across multiple neighbouring views along with depth and normal maps. The verified planar masks then guide the Gaussian optimisation process using co-planarity constraints, resulting in strong alignment with the expected planar geometry in the scene.

These studies indicate a growing trend towards injecting supplementary geometric and semantic structure into the 3DGS training process, albeit primarily focused on improving reconstruction quality at the room scale. However, the same principles may also support the organisation of multiple 3DGS scenes on a macro scale. By segmenting the indoor space into discrete rooms, structural information regarding their adjoining planar boundaries may serve as supplementary information to define and align multiple adjacent rooms.

Additionally, these algorithms aim to solve a second problem: to reduce floaters by constraining the 3D positional parameters of the Gaussians to plausible surfaces. (Wang et al., 2025b) suggests that the Gaussian training phase can trap ellipsoids in a spurious local minimum, due to a gradient vanishing problem during backpropagation. Instead of being culled in the pruning phase, these ellipsoids are thus retained and refined despite being placed incorrectly in open space.

However, the Gaussian optimisation process is intrinsically a minimisation problem for photometric loss. While there is potential to augment the training process with additional geometric constraints, there can still be a multitude of 3D arrangements of ellipsoids to compose a scene with the same image error. The issue of floaters is thus more nuanced than eliminating all non-surface-bound ellipsoids – floaters may still retain meaningful contribution to image composition across multiple viewpoints, based on their spatial relationship to nearby ellipsoids (Yang et al.,

2026). This is particularly relevant for indoor scene reconstructions, as the enclosing geometry of the indoor space, and the presence of various occluding scene furniture mean that a significant proportion of Gaussians are already constrained by a limited set of viewing directions. Considerations in dealing with floaters are an open issue, since pruning floaters may thus significantly degrade the output of the reconstruction process.

2.1.3. 3DGS and Indoor Navigation

Despite the growing popularity of 3DGS visualisation, relatively little work investigates how users should navigate between multiple splat scenes or interpret spatial relationships between disconnected reconstructions. Navigation therefore becomes not only a rendering problem within a single scene, but also a problem of spatial organisation and topology across multiple scenes.

The majority of current research in 3DGS for indoor navigation is mostly confined within the field of computer vision, primarily for the purpose of autonomous robot navigation. This generally entails extracting features from the 3DGS scene and determining whether the feature is a traversable plane or an obstacle. For instance, NVSim constructs navigable ground planes from sparse floor data, and generates connectivity using new rendered views (Jeong et al., 2025). Splat-Nav uses spline trajectories from Gaussian splat scenes to generate corridors and enforce safe flight trajectories for drones (Chen et al., 2025).

This research direction defines navigation primarily as a routing objective performed for autonomous robot movement. 3DGS scenes are created, decomposed and analysed to determine robot-traversal areas and propose potential route trajectories. It is quite distinct from human wayfinding experiences when navigating indoor environments, which is discussed in the following section. Since photorealism is the key strength of 3DGS for urban applications, navigating the 3DGS representation of the indoor environment should be done in a way that preserves the immersion of the user. In other words, the structure for organising multiple 3DGS scenes should aim to replicate how a human would physically move through and view the indoor spaces.

2.2. Indoor Navigation

The fidelity of visualisations – whether 3DGS, point clouds or other representations – increasingly match the real world. It is therefore important to discuss the human experience of orientation and wayfinding, as well as the technical implementations for navigation within these virtual environments, with a particular focus on indoor contexts.

2.2.1. Human Wayfinding and Orientation

Spatial cognition and awareness are crucial requirements to navigate an environment in order to achieve one’s goal within that space (Weisman, 1981). There is thus a two-part process to wayfinding in an indoor space – to contextualise the current location, and thereafter to determine and navigate a route to another location (Jamshidi et al., 2020). Li & Klippel (2014) coined the term Environmental Legibility (EL) to describe the combination of visual factors – architectural features, layout configuration and signage – that influences the user’s ability to understand and navigate the space. These factors are key information sources for users to create their individual cognitive map of the space, which thereby influences the decision-making process of pathfinding.

In a virtual environment where the user is quickly transported from the physical realm to a different disconnected space, these visual features remain fundamental in allowing the user to adapt and familiarise themselves with the new virtual location. The high visual fidelity of 3DGS reconstructions replicates geometry, textures, lighting and reflections to a high degree of detail. Applying 3DGS visualisations in the virtual environment thus enables the recreation

2.2. Indoor Navigation

of pertinent landmarks and details to greatly boost environmental legibility. Indeed, Yunisya et al. (2025) found 3DGS scenes create a highly realistic environment with superior sensory stimuli over traditional virtual reality (VR) simulations. The Gaussian scene is better able to differentiate between architectural features through the inclusion of lighting and materiality contrasts, clear depth-based separation of objects, and legible signage.

It is important to note that Yunisya et al. (2025)’s conclusions are drawn from reconstructions of corridors, passageways, and larger open spaces within a building, with unobstructed movement between these spaces. In the case of adjacent spaces connected by an occluding common wall, key visual features such as the shared door may be necessary to serve as anchors for users to orientate themselves when moving between reconstructions of two different enclosed rooms.

2.2.2. Wayfinding and Navigation in Indoor Contexts

Several commercial solutions exist which enable various methods to “move” across different instances within a virtual scene. For example, Autodesk’s Recap (Professional) software encodes 360° panorama images of scan locations into the point cloud collected via a terrestrial LiDAR scanner (see Figure 2.1). By clicking on individual bubbles, the user’s view teleports to the corresponding scan location. It is important to note that no topological relationship exists between these views – each image is separately loaded and rendered. While the transition between isolated image views is meant to be ameliorated by the context provided by the point cloud, the abrupt changes may still cause disorientation to the user.

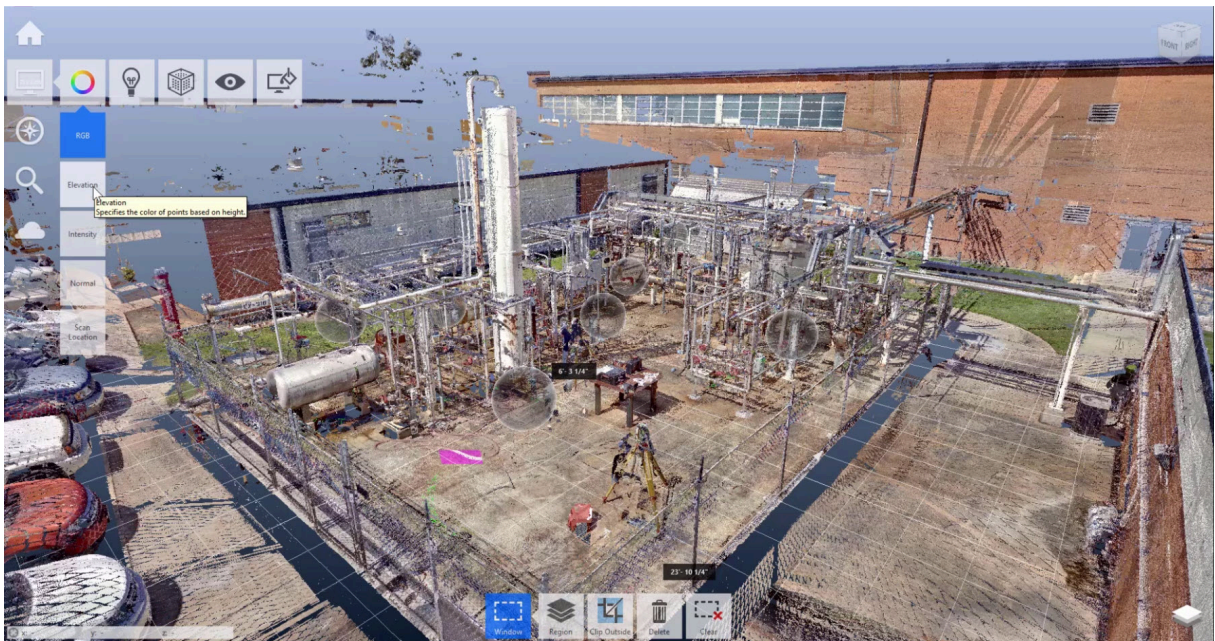


Figure 2.1: Instanced panorama image stations encoded as clickable bubbles within a point cloud. Source: (Autodesk, 2025).

This distinction between visual immersion and spatial understanding is thus particularly important for indoor 3DGS applications. Although a 3DGS scene may visually reproduce an environment with high realism, users may still experience disorientation if relationships between spaces are unclear or transitions between scenes are abrupt. Consequently, navigation between multiple indoor splat scenes likely requires more than visual continuity alone; it also requires explicit modelling of connectivity and adjacency relationships.

Virtual Reality (VR) applications are very similar to 3DGS scenes in terms of the immersive experience they bring to the user. Li et al. (2021) introduced teleport graphs with visibility-

based connectivity relationships between teleport nodes, as a means for effective VR indoor navigation and scene understanding by the user. The ‘point-and-teleport’ method typically used in VR software can cause spatial disorientation, which requires the user to readjust their understanding or perform additional teleportations. The topology of the proposed teleport graph ensures pairwise visibility between connected teleport nodes, allowing the user to traverse the environment in a continuous and systematic manner. The approach demonstrates the value of topology-based navigation models for human understanding and contextualisation of indoor space. Retaining common architectural features or scene furniture across teleport nodes enables the viewer to use the familiar objects – now viewed at a different angle and distance – to reorientate to the new position.

The contrast in implementation between the point cloud and VR methods highlight the fact that transitions that are based on topology can significantly reduce disorientation for the user. The visibility-based relationships proposed in teleport graphs are especially relevant to multi-scene 3DGS environments. If individual splat scenes correspond to rooms or sections of an indoor environment, visibility and accessibility constraints may help determine how transitions between scenes should be presented to the user. Such approaches suggest that topological navigation models may provide a useful framework for organising interconnected 3DGS spaces.

2.2.3. Indoor Navigation Graph

In terms of topological representations of indoor spaces, Yang & Worboys (2015) proposed a graph approach to describe the various indoor interconnections. A combinatorial map of the indoor space is first established from the floor plan, to represent spatial relations between rooms and corridors. The dual to the combinatorial map – known as the dual map – further minimises the spatial representation for connectivity purposes. This is done by reducing geometry into room nodes corresponding to each enclosed indoor space, as well as portal nodes signifying ingress and egress between room nodes. These portal nodes are then linked with navedges to indicate connectivity and traversability between node pairs (see Figure 2.2). The paper provides a formal theoretical foundation in representing indoor connectivity using geometry-derived topology. It is important to note that the approach is largely 2D in nature - connectivity is established floor-by-floor, while vertical connectivity in the case of stairs, ramps, elevators, and multi-floor spaces is treated as a special case.

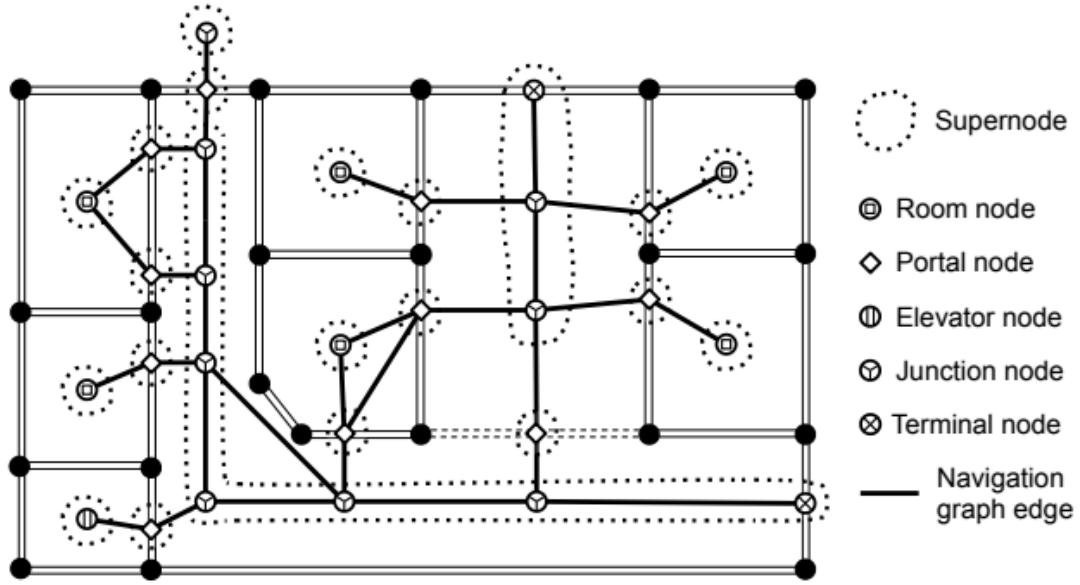


Figure 2.2: Indoor navigation graph. Source: (Yang & Worboys, 2015).

Zhou et al. (2022) proposed a floor-aware hierarchical visibility graph, called HiVG. While it adopts the same method of visibility-based placement of junction nodes, nodes are organised on a hierarchical level that incorporates building semantics – building-level, floor-level, axial-level, etc. This enables efficient querying of traversability across multiple floors.

For multi-scene 3DGS environments, graph-based approaches are particularly relevant because they distinguish navigational relationships from geometric representation. Continuous reconstruction between scenes may have strict geometric requirements in terms of planarity, at which 3DGS is relatively weak in comparison to BRep and mesh models. Instead, encoding connectivity in an abstract fashion through adjacency relationships is a possible alternative, which would then allow geometry and topology to be separately constructed. This aligns well with the concept of multiple discrete 3DGS scenes representing individual indoor spaces.

2.2.4. Semantic Navigation

Semantically informed modelling of the indoor environment further extends the concept of describing topology in multiple independent models. Liu (2017) proposed a two-level routing approach for indoor navigation; a geometric model calculates and visualises a route trajectory, whereas a logical model provides sequential route instructions using space names as points of reference. A combination of both models can enhance wayfinding decisions, based on the user’s preference for processing spatial information to create a cognitive map.

Zhu et al. (2025) demonstrates another possibility of semantic navigation. The method exploits the rich semantic information inherent in Industry Foundation Class (IFC) data to develop IFC-Graph. Instead of the accurate position coordinates typically required to enable point-to-point routing, IFC-Graph enables object-to-object routing; location accuracy thus scales to the corresponding object of interest. This approach is suitable for indoor scenarios for multiple reasons: location accuracy indoors is typically low, and the purpose of routing is generally to move between separate rooms rather than to precise coordinate locations.

2.2.5. Scene Graph

While scene graphs are typically used in computer graphics, Schulz et al. (2023) presents an interesting application for the built environment. Inspired by Linked Data principles, scene

graphs can enable heterogeneous data formats of the same spatial scene to be accurately located and scaled through sequential and hierarchical transform operations. It is therefore possible to describe and align multiple discrete representations, objects, and features using an ordered set of transformation parameters. However, it is important to note that Schulz et al. demonstrates this approach for a single room; the hierarchical nature of scene graphs is unidirectional and limits each child to a single parent node. This limitation does not hold for indoor spaces with multiple, interconnected rooms.

The approach suggests that hierarchical transformation operations may be a potential way to organise multiple 3DGS scenes within a common spatial reference framework. However, the approach needs further refinement to support one-to-many, many-to-many, and bidirectional relationships to better reflect the complex connectivity relationships among indoor spaces.

2.3. IndoorGML

IndoorGML is an Open Geospatial Consortium (OGC) open standard for representing and sharing indoor spatial information, focusing on the topological relationships between indoor spaces. The purpose of the standard is to enable location-based services (LBS) – such as routing, wayfinding, and locating objects for accessibility and asset management – across diverse systems. The model can be considered the culmination of the above modelling approaches to provide a universal, open format for indoor navigation. The IndoorGML 2.0 schema (Open Geospatial Consortium, 2025) comprises two relevant components: the core Structured Space Model and a Navigation Extension Module.

2.3.1. Structured Space Model

The Structured Space Model defines the cellular decomposition of the indoor environment, and the topological relationships between cells based on the Poincaré Duality, akin to Yang & Worboys (2015)’s indoor navigation graph. Physical spaces (defined as primal space) are conceptualised as non-overlapping *Cells* (see Figure 2.3). Each *Cell* is bounded and serves a definable function – for example, a corridor serves as a *TransitionSpace* between multiple *GeneralSpace* rooms. The dual space defines the centroid of each *Cell* as a *Node*, where adjacency relationships are indicated by *CellBoundaries*. These are further subdivided based on navigability. Two *Cells* with a shared and navigable *CellBoundary* are topologically connected, and this connection is defined as an *Edge*. Given the example in Figure 2.3, *Edges* can be defined as $R1 \rightarrow D1 \rightarrow R2$, or $R2 \rightarrow D3 \rightarrow EXT$. The dual space thus serves as the connectivity graph that describes the topological relationships between indoor spaces and their boundaries.

This abstraction is especially valuable for compartmentalised 3DGS environments, where individual scenes may vary in completeness, scale or reconstruction quality. By separating topological relationships from geometric representation, IndoorGML potentially enables multiple heterogeneous 3DGS scenes to remain navigationally coherent despite inconsistencies in the underlying reconstruction.

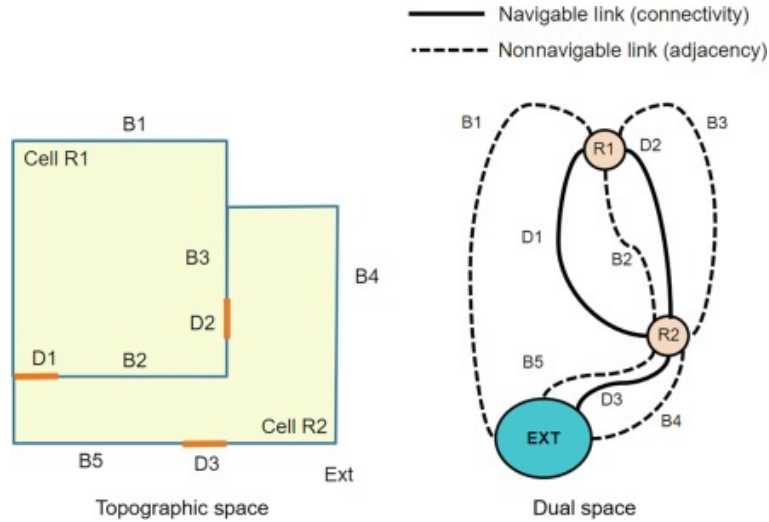


Figure 2.3: Structured Space Model of two connected rooms. Source: (Li et al., 2019).

2.3.2. Navigation Model

The Navigation Extension Module expands upon the core model by defining a logic-based navigation graph supporting various movement types, routing functions and accessibility constraints. Navigability is divided into 3D *CellSpace* and 2D *CellBoundary* categories to define accessible connected volumes and the common plane between them. These concepts align naturally with the problem of navigating between multiple indoor splat scenes. Individual 3DGS scenes may potentially be associated with IndoorGML cells, while transfer spaces and navigable states could define allowable transitions between scenes. Such an approach would enable scene traversal to be modelled independently from the rendering process itself.

Crucially, the 2.0 IndoorGML structure separates topology from geometry. As an abstract framework, the model accepts scenarios where geometry may be incomplete, or irrelevant to the purpose at hand. Geometry – and its visual representation – is thus treated as an optional realisation of *CellSpaces*, allowing multiple representations of the same indoor space while maintaining the integrity of the topological model. This has two implications for a multi-3DGS environment. Since 3DGS scenes prioritise visual appearance over explicit geometric structure, a purely geometry-driven navigation framework may be difficult to establish directly from Gaussian primitives alone. The IndoorGML framework instead provides the opportunity to treat 3DGS scenes as a pure visual representation of the indoor space. However, this also implies that topology cannot be derived or built from the geometry inherent to the 3DGS reconstruction directly; rather, some external reference is required to create the topology to define the *Cells* and their connections.

There have been several successful methods for creating a valid IndoorGML model from various input data. van Steijn (2023) used a benchmark point cloud dataset to create an IndoorGML model in near-real-time. Room geometries were reconstructed using slices of the point cloud, while doors were identified and extracted using the route trajectory of the handheld LiDAR scanner. Jamali et al. (2015) proposed the use of survey points to compensate for inaccurate geometries in existing building models. The survey points are used as location anchors, such that the navigation graph of the rooms and corridors is generated relative to these benchmarks. Both methods require the use of accurate point measurements to rebuild both the geometric and topological aspects of IndoorGML. Perhaps more relevant to photogrammetric methods, Kim & Lee (2015) instead proposed the use of omni-directional images to generate IndoorGML

data. In this case, no geometry is created; topology is created directly from the camera positions of images, as well as the detection of doors as transition nodes within each image.

2.4. Summary of Related Works

In general, there is no relevant work published that provides a topological framework for organising multiple discrete 3D Gaussian Splatting scenes. Current 3DGS research directions instead focus on scarce-input splatting, modelling for temporal variation as 4DGS, dynamic or instant 3DGS reconstruction, and reducing aliasing, artefacts and distortion errors as highlighted by Kerbl (2025).

Conversely, research in indoor topology and navigation modelling provides mature frameworks for representing adjacency, traversability and semantic relationships between spaces, but has not yet been explored in the context of radiance field visualisations. Specifically, IndoorGML models remain difficult to achieve in practice (Ledoux, 2020). Indeed, changes to the schema from version 1.1 to 2.0 (Part 1) aimed to clarify terminology, substantiate descriptions and diversify formats, with the overall objective to increase the universal applicability of IndoorGML to indoor navigation and location-based services (Diakité et al., 2020).

Within both contexts, there is a mutually beneficial opportunity to investigate the fusion of IndoorGML with 3DGS models. IndoorGML’s topological model serves as a consistent structure to spatially organise discrete 3DGS scenes of indoor spaces, and provides an intuitive method to define transitions between these discrete scenes. On the other hand, IndoorGML models are rarely visualised, and even when visualised, only in simple 2D or 3D geometric shapes. The high visual fidelity of 3DGS scenes can be incorporated as a highly immersive visual representation of IndoorGML’s *Cell* geometry, thus enabling greater end-user applications that benefit from photorealistic visualisation, such as indoor heritage tours, facility walkthroughs and museum navigation.

2.4. Summary of Related Works

3. Research Objective

Section 2 puts forth two main findings in the current status of literature: that multi-3DGS implementations remain an unexplored research avenue, and that the IndoorGML format can potentially provide the topological framework to spatially organise these scenes. This paper thus investigates the synergistic opportunities in spatially organising independently reconstructed 3DGS scenes of indoor spaces, to establish topological relationships between connected scenes that can create an immersive navigation experience for the user.

3.1. Research Questions

To reiterate the main research question:

To what extent can discrete 3D Gaussian Splatting scenes be spatially aligned and integrated within an IndoorGML topological model, to enable topologically-constrained and contextually continuous navigation of indoor environments in a web-based viewer?

The research angle is two-part: first define the conceptual pipeline behind the integration of 3DGS scenes into IndoorGML, then develop a prototype web viewer to serve as a proof-of-concept. The question thus naturally decomposes into several sequential components, which are briefly explained with supporting sub-questions (*in italics*):

1. Data Structure

How should the 3DGS scenes be captured and organised to adhere to the IndoorGML topological schema?

IndoorGML’s Structured Space Model abstracts indoor spaces into discrete, non-overlapping *Cells*. The image capture and reconstruction for individual 3DGS scenes should therefore reflect this cell-based organisation, such that each 3DGS scene corresponds to a single navigable unit in the topological model.

2. Geometric Features

What method is most appropriate to obtain a constraining geometry that accurately fits the expected scene bounds?

3DGS reconstructions commonly contain floater Gaussians that are located outside the physical bounds of the scene itself. These floaters may still be useful as they retain significant contribution to the image composition of the scene, but may cause visual degradation and artifacts from viewing angles outside those seen during the training process. A method is therefore needed to define and enforce a geometric boundary that preserves photorealistic quality throughout user navigation.

3. Scene-to-Scene Transitions

How can a shared spatial reference frame be established to enable coordinate conversion between independently reconstructed 3DGS scenes?

3DGS scenes have arbitrary scale, orientation and coordinate systems. While this does not affect topological relationships, geometric operations are challenging to apply across multiple scenes. The alignment of 3DGS scenes to a single common spatial reference thus enables consistent operations in rendering, movement and transitions across connected scenes.

What point-correspondence method produces a robust similarity transform between a 3DGS

3.1. Research Questions

scene's local coordinate system and a common reference frame?

Reliable point correspondences between the common reference point and the 3DGS scene are required to enable scene alignment to the common reference frame. However, the 3D position of Gaussians may not correspond precisely to available reference points. The robustness of the resulting alignment depends on both the transform estimation method used, and the quality and distribution of the point-pairs.

4. Topology Creation

What attributes are required in addition to IndoorGML's core concepts to enable topological linkages between 3DGS scenes and their shared transition spaces?

For instance, the above sub-question already identifies the need for a transform method for scene alignment. Identifying the supplementary information necessary to bridge the gap between IndoorGML and 3DGS splat scenes is therefore crucial to enable the schema to serve as the operational backbone of the web viewer prototype.

5. Web Viewer Implementation

How can the IndoorGML schema be implemented in a database format that best supports the operations needed in a multi-3DGS web viewer?

The IndoorGML schema is an XML-based geospatial standard, which is not efficient for direct use and querying by a web viewer. Translating the schema into a database format that supports querying operations is therefore important in ensuring the performance of the web viewer.

To what extent does the topology-constrained multi-3DGS viewer enforce topological navigation rules while maintaining real-time performance?

The key objective of the paper is ensuring that introducing topology brings benefits to multi-3DGS scenes without decreasing performance. The viewer must therefore enforce topological navigation rules while rendering a photorealistic 3DGS scene in real-time.

3.2. Scope

Based on the gap identified in the reviewed literature, this paper is, to the authors' knowledge, the first to conceptualise, and demonstrate a practical pipeline integrating IndoorGML with independently reconstructed 3DGS scenes. The contribution is scoped as a proof-of-concept prototype.

The following are explicitly within scope:

- Conceptual and practical pipeline to link discrete 3DGS scenes via an IndoorGML topological model.
- Coordinate alignment of independent 3DGS scenes to a shared coordinate reference framework.
- Web-based viewer enforcing topologically-constrained navigation, for a single-floor, multi-room demonstration.

The following are out of scope of the paper, and suitable for discussion and future work:

- Evaluation of the output web viewer through a user study to determine immersiveness.
- Optimisation or modification of the 3DGS reconstruction process itself.
- Automatic generation of the IndoorGML model.
- Multi-floor implementation of the topological constraints.

4. Methodology

The methodology is structured as a sequential pipeline, following the decomposition of the main research question established in Section 3. Each component addresses one sub-question, and the output of each stage feeds into the next (see Figure 4.1). The following subsections detail each modular component of the pipeline. As a practical demonstration, a small section of rooms in Delft University of Technology (TU Delft)’s Faculty of Architecture and the Built Environment (BK) was selected and captured.

Two guiding principles shape the design of the overall pipeline. First, the workflow is intended to demonstrate general applicability, rather than optimisation for a specific case. The intent is that each modular process can be adapted or substituted without invalidating the full pipeline. For example, the use of standard 3DGS reconstruction as per Kerbl et al. (2023) deliberately serves as a baseline choice – the pipeline is not intrinsically dependent on the reconstruction method, and can in principle be replaced with alternatives such as PlanarGS (Jin et al., 2025) which may recover a more accurate scene. Second, all methods and tools used across the pipeline are free and openly accessible (as at time of research). Scripts to process the data were written in Python and Javascript, and also made available for public use at <https://github.com/mjteo95/3dgs-topology>. Due to size constraints, the initial data – videos, image frames and sparse reconstructions – are not included in the repository. However, final and intermediate outputs relevant to the demonstration of the pipeline are included as a GitHub Release, to allow the user to replicate the navigation experience created by the web viewer prototype. This is in line with the proof-of-concept defined in the scope, and lowering the barrier to entry for future work may encourage further adoption of IndoorGML.

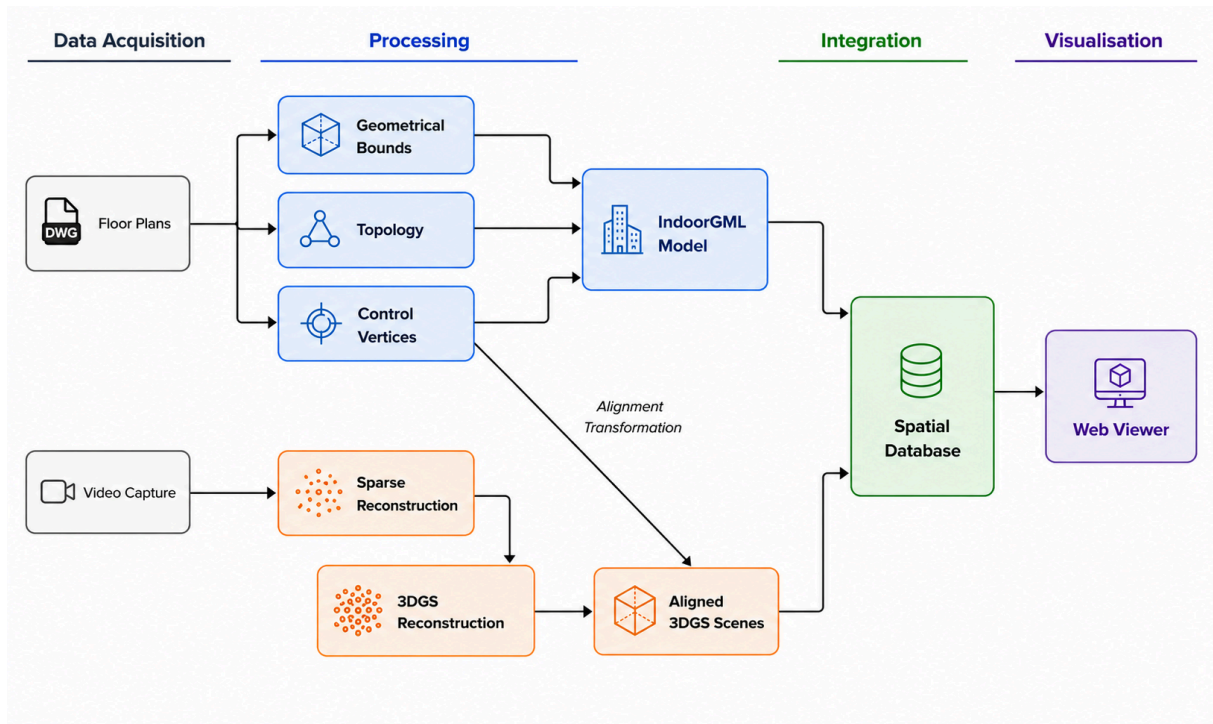


Figure 4.1: Conceptual pipeline.

4.1. Data Capture and Processing

From IndoorGML’s Structured Space Model, any space enclosed by bounding geometry and serving a purpose can be considered a valid *Cell*. A common and intuitive interpretation of this definition is therefore to classify rooms as individual *Cells*.

A one-to-one association of 3DGS scenes with rooms brings multiple advantages over reconstructing a single scene of multiple rooms. A fundamental assumption of most 3DGS pipelines is that all input images belong to a single, coherent scene. Indoor environments typically violate this assumption because walls between rooms serve as large occluders to the scene. The severe occlusion means that limited images, if any, would be available to connect images seen from either side of the wall. As a result, Structure from Motion may struggle with ambiguities in camera pose alignment, leading to poor sparse reconstruction. Conversely, room-based *Cell* classification clearly compartmentalise spaces by walls, doors and partitions. These serve as geometric boundaries to the *Cell*, which naturally coincide with visibility discontinuities. The geometric enclosure ensures all input data captured are relevant to the scene, thus improving reconstruction accuracy.

Per-*Cell* scene reconstruction also aligns well with wayfinding behaviours. Rather than envisioning a building as a single continuous space, cognitive maps typically feature subdivided spaces and navigation linked via landmark features. These are strengths inherent to the IndoorGML topological model; multi-3DGS visualisations simply reap similar benefits when structured accordingly.

4.1.1. Data Acquisition

The pipeline begins with the acquisition of data necessary to create the 3DGS scenes. Three adjacent spaces on the ground floor of the West Wing of the BK faculty were selected as the site of interest (see Figure 4.2). Rooms 0470 and 0450 are lecture halls, while 0807 is part of the corridor space. Neither lecture halls are directly connected to each other, but rather connected to the corridor via two doors each. The room configuration is thus sufficient to demonstrate topological rules: direct navigability between 0470 and 0450 is not possible.

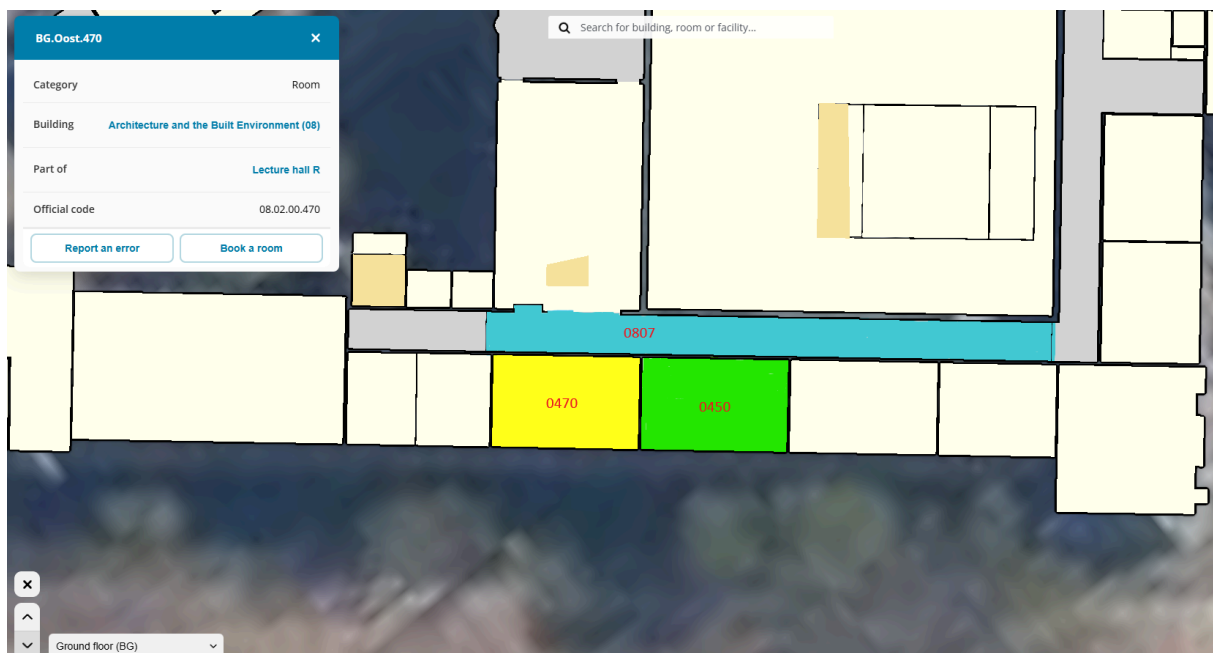


Figure 4.2: Selected rooms for 3DGS reconstruction. Extracted and adapted from inclusivecampusmap.tudelft.nl

For each scene, an image set of the associated room is necessary to create the initial SfM points as well as to determine photometric loss during the Gaussian training phase. To accomplish this, a smartphone camera was used to capture a video per room. Specifications about the data acquisition are listed in Table 2. The video trajectory comprises three loops at low, chest-height and elevated viewpoints using an extendable handheld tripod attachment. This trajectory ensures sufficient coverage of the scene at varying angles to improve reconstruction accuracy in the 3DGS training phase.

Parameter	Specifications
Device	Google Pixel 8 Pro
Camera	Main wide-angle camera
Sensor resolution	50 MP
Video resolution	3840×2160 pixels (4K UHD)
Frame rate	30 fps
Colour profile	HLG HDR
Video format	MP4
Image stabilisation	Optical and electronic stabilisation

Table 2: Data acquisition specifications.

Video acquisition was chosen as the capture method due to its ability to efficiently generate a dense sequence of overlapping viewpoints. Compared to still images, the continuous nature of video recording simplifies data collection, thus reducing the likelihood of coverage gaps. The high overlap between frames corresponds to smoothness in viewpoint transitions, which are beneficial for SfM and 3DGS reconstruction alike. The method also allows retrospective variations in image sampling intervals, thus providing flexibility in balancing the visual quality of reconstruction against computational limitations of the hardware used. Especially in the context of the long narrow corridor of 0807 complete with complex scene furniture, video capture is able to sequentially record the spatial relationships between adjacent image features, which is significantly harder to guarantee by capturing discrete still images.

It is important to acknowledge that video frames have substantially lower image resolution than still photographs, and may be further affected by blurring from excessive camera movement. Ensuring clear frames can be extracted is thus key to this approach. To mitigate the drawback of motion blur, video stabilisation was enabled, and camera movement was carefully controlled to be as slow and gradual as possible during the capture process.

FFmpeg was used to extract usable frames from the captured videos, at a fixed extraction rate of 1.5-3 frames per second, depending on the input video length. This is to ensure a roughly equivalent number of images per room, as the number of input images is proportional to the memory cache required during the reconstruction phases. Additional qualifiers were included to ensure visual quality of the output images. Where transitions between camera viewpoints are slow and thus generate large overlap between extracted frames, a block-level comparison between consecutive frames is calculated, and frames with no significant block differences are dropped. This ensures images have sufficient overlap for sparse reconstruction, while removing redundant images.

4.1.2. Sparse Reconstruction

The extracted frames are then processed to create a sparse set of points based on photogrammetric feature matching. COLMAP was used to perform the sparse reconstruction and generate the SfM points for 3DGS initialisation. This is done in three stages. First, Scale-Invariant Feature Transform (SIFT) points are extracted from each image. Next, images are compared pairwise to determine matching SIFT points and verify geometry. Third, an incremental reconstructor registers the images in a sequential order. This is done by estimating each camera’s pose relative to images that have already been registered, thereby triangulating 3D positions of the matched SIFT points. Throughout the registration process, bundle adjustments are regularly conducted to jointly refine camera poses and point positions. This is done by minimising the reprojection error across all registered image-point observations. The result is a registered set of camera poses and a sparse 3D point cloud.

Since the frames extracted from the input video are already structured in chronological order, it is possible to use sequential matching for the feature matching stage. Candidate pairs are restricted to temporally adjacent frames, based on the valid assumption that the order proximity of images translates to spatial proximity of features within these images. This significantly reduces the computational requirements compared to exhaustive image-pair enumerations. The matching mode also detects the three loops of varying height that were captured at distinct intervals of the video. As such, images are again matched at larger specified intervals to register the loop and improve the accuracy of the recovered camera poses and 3D point positions.

The sparse reconstruction produces two outputs consumed by subsequent stages, which are visualised in Figure 4.3, in terms of coloured points and camera poses in red. The sparse point cloud is used to initialise Gaussian positions in the 3DGS training phase as well as for alignment calculations, while the registered camera poses serve as the training viewpoints from which photometric loss is computed. Indoor environments present well-known difficulties for SfM. Surfaces with low texture – white walls, uniformly coloured floors – produce few stable keypoints. For 0450, repetitive scene furniture in the form of tables and chairs create ambiguous matches which reduce reconstruction accuracy, whereas the long corridor geometry of 0807 makes it challenging to triangulate keypoints along the corridor’s main axis. In these cases, the extraction rate of frames were customised to increase or decrease redundancy in image overlaps accordingly. Per-room reconstruction statistics are reported in Section 5.

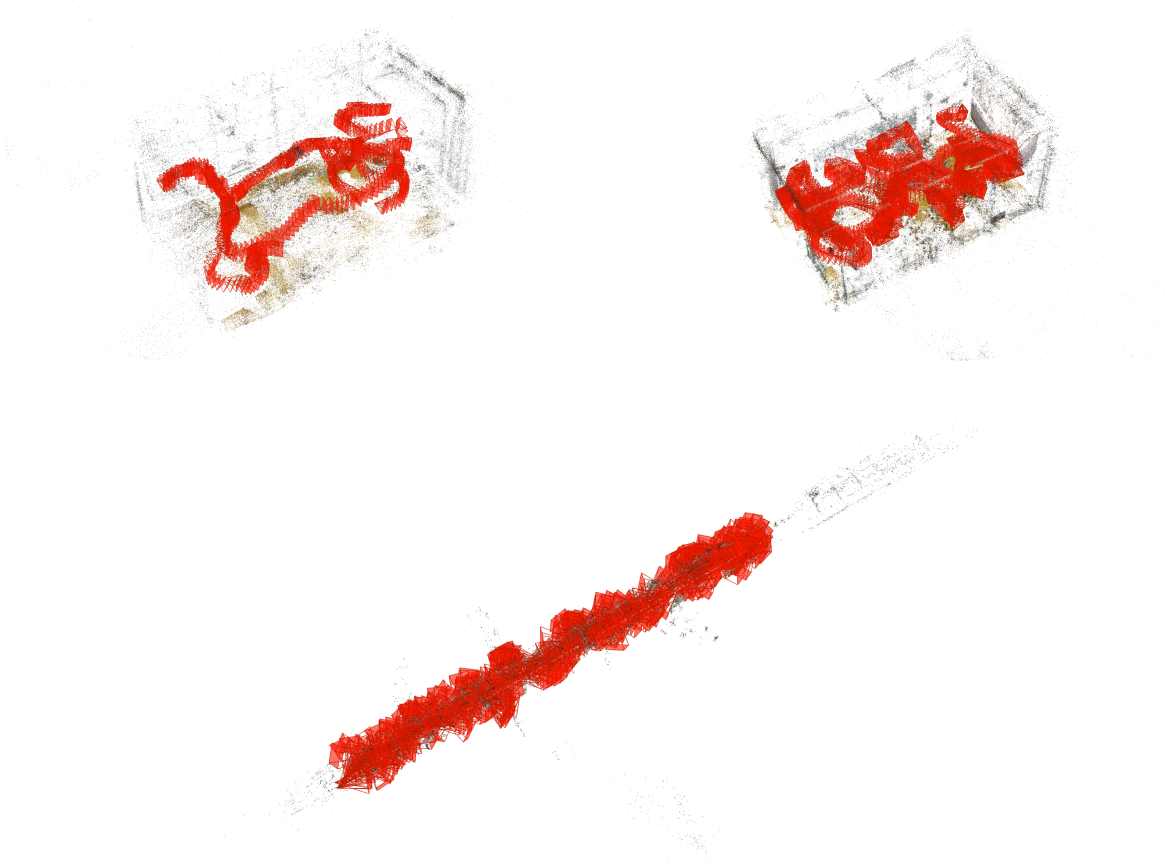


Figure 4.3: Sparse reconstructions of 0450 (top left), 0470 (top right) and 0807 (bottom).

4.1.3. 3DGS Scene Reconstruction

Prior to training, the registered images must first be undistorted to remove the lens distortion in the sparse reconstruction. COLMAP provides an image undistortion pipeline that remaps each image from the simple radial camera profile chosen to represent the smartphone camera, to the pinhole camera model that is typically assumed for 3DGS training.

The undistorted image sets and camera poses were then used as inputs to Lichtfeld Studio, a free application for 3DGS scene training. The training process follows the method of Kerbl et al. (2023). An initial set of Gaussians is seeded from the SfM sparse point cloud, and their parameters iteratively optimised. At each training iteration, the ellipsoids are alpha composited from a registered camera viewpoint and the rendered synthetic image is compared against the corresponding ground-truth image; the resulting photometric loss is backpropagated to update each Gaussian’s position, opacity, anisotropic covariance, and spherical harmonic colour coefficients. Periodic densification and pruning operations adapt the Gaussian count to the scene’s geometric complexity throughout training.

Training was performed on a consumer laptop, which is equipped with a NVIDIA GeForce RTX 3070Ti GPU with 8GB of VRAM and 32GB of system RAM. The hardware is considered broadly accessible at the consumer-level and therefore consistent with the accessibility principle established in Section 3. However, the choice comes with practical limitations for the 3DGS training process. A high quality scene reconstruction generally involves high-resolution images, a large Gaussian count and well-defined third-degree spherical harmonic functions. The 8GB VRAM is insufficient to cache this large per-Gaussian memory footprint.

To mitigate this issue, the number of input images were first regulated in Section 4.1.1 to less than 1000 frames per scene. The maximum Gaussian count during the training phase was

4.1. Data Capture and Processing

then capped at between 3-4 million Gaussians. The spherical harmonic degree was reduced from 3 to 1. Input images were tiled and downsampled to a maximum width of 1920 pixels. Taken together, these constraints reflect a deliberate trade-off between reconstruction fidelity and hardware accessibility. This is deemed an acceptable balance, considering that scene reconstruction quality is not the main focus of the paper, but rather as an enabler to visualise the topological constraints.



Figure 4.4: 3DGS scene of Cell 0470 after 15,000 iterations.

4.2. Scene Alignment and Scaling

The issue of alignment and scaling is pertinent whenever multiple models are to be placed relative to one another’s position. For a multi-room setup, each room is reconstructed independently to produce a single 3DGS scene each. Every 3DGS scene is defined in its own local (COLMAP) coordinate frame. This frame has arbitrary scale, orientation and shares no intrinsic relationship to the coordinate frames of other 3DGS scenes. While pairwise alignment and scaling between adjacent scenes is theoretically possible, introducing an external reference and aligning scenes to this common framework presents a cleaner and more accurate solution. The architectural floor plan is able to serve as this reference, since it expresses the geometry and position of all rooms in a single coordinate system. The issue of alignment here is therefore to find, on a per-room basis, a transformation that maps the COLMAP local coordinate frame to the floor plan’s coordinate system.

A natural approach to deriving this transformation would be to identify geometric features – such as wall surfaces, floor planes and door opening – directly within the trained 3DGS scene, and then match them to corresponding features within the floor plan. However, reliably extracting geometry from Gaussians is problematic. The COLMAP keypoints used to initialise the 3DGS training are typically surface-bound features on walls, floors and objects. They therefore carry some implicit geometric meaning. Since the 3D position of the Gaussians is a learned parameter that is updated at every training iteration, the trained ellipsoids are not guaranteed to remain surface-bound. This issue is further compounded by the adaptive density

control mechanism: densification and pruning operations add new points and subtract existing points throughout the training phase, impacting both the count and indexing of the point set. It is therefore not possible to reliably establish point-to-point correspondences between the initial sparse point cloud and the output Gaussian ellipsoids, nor to assume that any Gaussian lies on a surface that is suitable for extraction and alignment.

The alignment approach here exploits a consequence of such a relationship. Although the position of individual Gaussians cannot be reliably tracked throughout the training process, the overall spatial distribution of the output Gaussians remain generally consistent with the initial sparse points. Since Gaussian parameter updates rely on photometric loss which is based on the same image observations that produce the keypoints, the majority of ellipsoids converge to a position close to their original seeded location. This relationship between the sparse points and the trained Gaussians means that a coordinate transformation calculated from the sparse point cloud, can be directly applied on the trained Gaussians to produce a similar, albeit approximate alignment effect. This allows us to reduce the geometry extraction problem from the Gaussians to the surface-bound sparse point cloud.

4.2.1. Umeyama Similarity Transform

To align the 3DGS scene to the common reference frame of the floor plan, the method is to calculate a coordinate transformation between the initial sparse point cloud and the vertices of the floor plan. This is estimated as a 3D similarity transformation – rotation, translation and uniform scale – which can be calculated using the Umeyama algorithm (Umeyama, 1991).

Given a set of manually selected correspondence points identified in both the COLMAP sparse reconstruction and the floor plan, the algorithm finds the transformation parameters that minimise the sum of squared distances between the corresponding point-pairs. This is expressed in Equation (4.1), where $\mathbf{R}$ is a rotation matrix, c is a uniform scale factor, and $\mathbf{t}$ is a translation vector. The optimal closed-form solution is obtained by calculating the cross-covariance matrix of the two point sets, then performing singular value decomposition (SVD).

$$\frac{1}{n} \sum_{i=1}^n \|\mathbf{y}_i - (c\mathbf{R}\mathbf{x}_i + \mathbf{t})\|^2 \quad (4.1)$$

There are no strict requirements in the position or geometry of the point-pairs other than being visible in both representations simultaneously; in practice within an indoor context, however, prominent corners of walls, and door frame edges serve as reliable candidates. These features produce multiple keypoints in the sparse point cloud, and are also clearly marked vertices in the floor plan geometry. Theoretically, a minimum of three non-collinear point pairs is required to constrain the seven degrees of freedom of a similarity transform in three dimensions; in practice, a larger set is used to reduce sensitivity to individual point localisation error.

The Umeyama method is particularly well-suited to this use case, as it is robust against inexact positional correspondences. Identifying point-pairs between COLMAP sparse reconstruction and 2D floor plans is inherently imprecise. The position and accuracy of the selected sparse point correspondence is highly dependent on the local point density, which in turn relies on the coverage of the input images. There is thus a range of potential sparse points which can potentially represent the thick corner of a room, compared to the singular corner vertex found at the convergence of two infinitely-thin walls of the abstracted floor plan. Since the Umeyama method distributes residual error across all correspondence pairs, the least-squares estimate is less sensitive to imprecise point-pair coordinates, and converges to a transformation that best satisfies all input correspondences.

4.2. Scene Alignment and Scaling

Since the floor plan is defined in 2D, all floor plan vertices are initialised at a uniform height of zero. The COLMAP correspondence points are therefore manually selected – to the best of the situation – to lie on the floor surface of each room. Since both point sets can be assumed to be planar, it is possible to first project the COLMAP space into a 2D coordinate system that is aligned with the floor plane. This is done using the Rodrigues rotation formula (see Equation (4.2), where the floor normal n is rotated onto the Z-axis $k = [0, 0, 1]^T$:

$$R = I + [v]_{\times} + [v]_{\times}^2 \frac{1 - c}{s^2} \quad (4.2)$$

where I is a 3x3 identity matrix, $v = n \times k$, $s = \|v\|$, $c = n \cdot k$, and the skew-symmetric matrix of v is:

$$[v]_{\times} = \begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix} \quad (4.3)$$

The rotation matrix can be expressed in homogeneous coordinates as:

$$T_{\text{level}} = \begin{pmatrix} R_{11} & R_{12} & R_{13} & 0 \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.4)$$

Applying the Rodrigues rotation means the similarity transform between the projected sparse points and the floor plan vertices can be reduced from 3D to 2D, resulting in a more accurate transformation. Note that this is also dependent on the planarity of the chosen sparse points.

The output of the 2D Umeyama estimation can be represented as a 4X4 transformation matrix that encodes the optimal rotation, scale factor and translation for each room, as in Equation (4.5):

$$T_{2D} = \begin{pmatrix} sr_{11} & sr_{12} & 0 & t_x \\ sr_{21} & sr_{22} & 0 & t_y \\ 0 & 0 & -s & sz \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.5)$$

where s is the estimated scale factor, r is the estimated 2D rotation matrix, t is the estimated 2D translation vector, and z is the mean floor elevation of the sparse points. The final 3D transformation matrix incorporating the Rodrigues rotation matrix is therefore :

$$T = T_{2D} T_{\text{level}} \quad (4.6)$$

An additional consideration is the arbitrary axes of the sparse reconstruction. Using the example of 0470 in Figure 4.4, the horizontal plane is more closely defined by the XZ-axis. To reconcile this with the conventional XY-horizontal, Z-vertical axis orientation of the floor plan, all 6 axis permutations are tested in the alignment process, and the axis configuration of the permutation with the best result is stored alongside the transformation matrix itself. Both are then ingested into the database as part of the additional attributes required for the IndoorGML schema described in the following section. These serve as the operational link during runtime between the floor plan reference frame and the 3DGS scene.

4.3. Topology Creation

4.3.1. Manual IndoorGML Modelling

As discussed earlier in Section 2.3.2, the topology linking individual 3DGS reconstructions of each room has to be obtained from an external source. The reference floor plan presents a natural option since it is already used to calculate the alignment of scenes. The IndoorGML topological model was thus constructed manually using a suite of tools from STEMLab: InEditor, InFactory, and InViewer-Desktop. InEditor is a 2D graphical authoring tool designed for the IndoorGML schema, while InFactory connects to Ineditor, providing the RESTful server and library to translate user-drawn geometry and topology into the IndoorGML schema. InViewer-Desktop provides the functionality to inspect IndoorGML models in 3D.

An image-based approach was taken to creating the geometry of the three rooms. A floor plan image is loaded as a reference underlay, against which room boundaries and door openings are hand-traced (see Figure 4.5). Initially, a thin-door approach (see Figure 4.6) would have been considered sufficient for topological navigation. In this case, the connecting doors can simply be defined as 2D LineStrings which form a subsection of the larger LineString representing the shared wall. However, a thick-door approach proved more representative, as the reference floor plan includes wall thickness. Rather than modelling the centre-line of the wall as the shared boundary of two rooms, the geometry of each room can be defined as the inner boundaries of the walls. This is naturally advantageous as the vertices of the room corners and door frames used for the scene alignment all lie on this inner boundary. Consequently, each door also has to be modelled as individual *Cells* complete with their own geometry, to connect the two interior boundaries of adjacent rooms, thus completing the thick-door approach. On the other hand, shared walls do not have to be explicitly modelled as they do not serve any further geometric purpose.

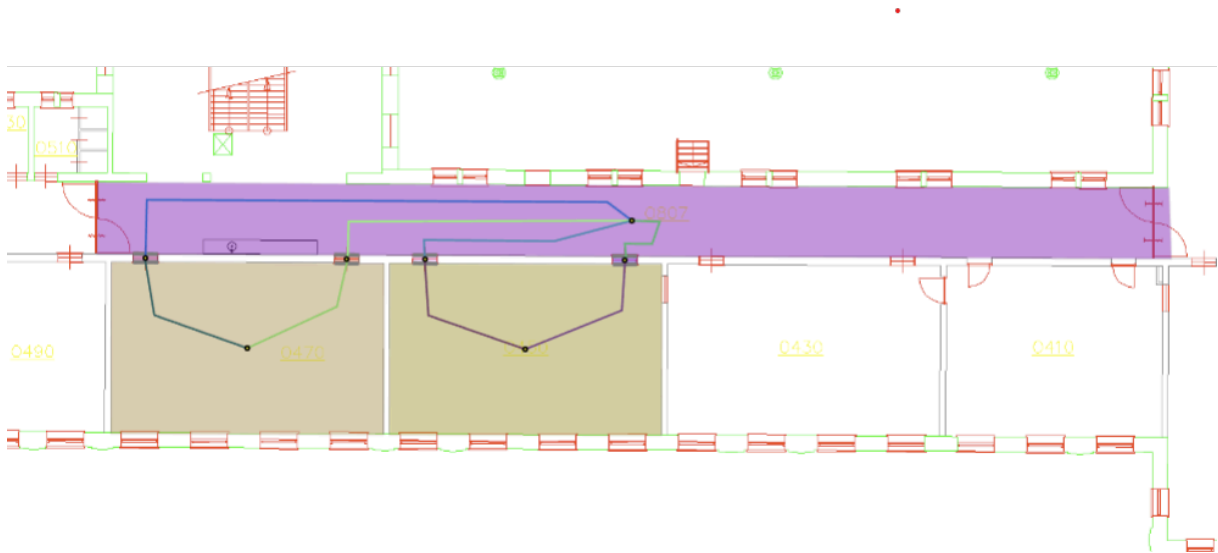


Figure 4.5: Manual geometry and topology creation using InEditor and InFactory.

Since doors now occupy valid space, several considerations need to be addressed. First, *Cells* corresponding to rooms have corresponding 3DGS scenes, while *Cells* corresponding to doors do not. To mitigate this, the topology of the *CellSpace* can be appropriately labelled, and later used to distinguish between *Cells* to trigger operations in the web viewer. This is further discussed in Section 4.4.1. Second, the topological connection between rooms is now extended to intersect the *Node* centroid of the door *Cell*. Where the thin-door model would encode a

4.3. Topology Creation

transition between two rooms as $R1 \rightarrow D1 \rightarrow R2$ (as demonstrated in Section 2.3.1), the thick-door model now encodes the same transition as $R1 \rightarrow B1 \rightarrow D1 \rightarrow B2 \rightarrow R2$, where $B1$ and $B2$ are the respective *CellBoundary* common to $R1 \rightarrow D1$, and $D1 \rightarrow R2$.

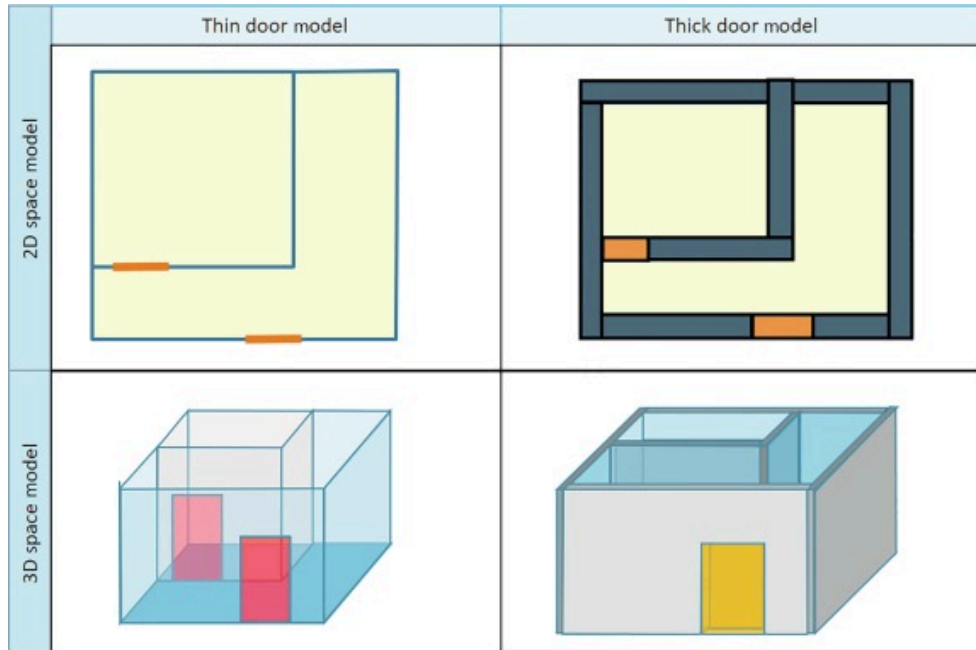


Figure 4.6: Thin versus thick door approach for modelling connectivity. Source: (Li et al., 2019).

4.3.2. Model Realignment

A fundamental limitation of the image-based approach taken above is that the coordinate values written to the output IndoorGML file uses the pixel coordinate system of the reference image, rather than the coordinate reference frame of the floor plan (in DXF) itself. The spatial dimensions of the IndoorGML file are therefore not geometrically accurate: vertex positions correspond to screen coordinates instead of the values used to align the 3DGS scenes, and the scale of the model is ambiguous. This is an inherent characteristic of using image references, and necessitates an intermediate processing step to realign and scale the model to the floor plan, which serves as the authoritative geometric reference throughout the entire pipeline.

To resolve this, a coordinate replacement script was written to parse and overwrite vertex positions for the IndoorGML file. Given the GML file and a JSON file containing the floor plan vertices per *CellSpace*, a global lookup table is created that maps each GML vertex to its corresponding pair in the floor plan. Door *Cells* are first matched since they share common vertices with the adjoining room and corridor *Cells*. These vertices can then be skipped when mapping the vertices of the room and corridor *Cells*. Once the global lookup is fully populated, the vertices of each *LinearRing* of every *CellSpace* can be updated accordingly. Since the Z coordinates were correctly labelled in the manually-created IndoorGML model, and the 2D floor plan does not include height values, these are retained as-is.

The derived geometry corresponding to the dual space equally needs to be updated according to the above changes. *Node* point geometry – corresponding to the centroid of the *CellSpace* – is simply recalculated as the mean of the floor ring vertices. Thereafter, the *Edge* geometry is similarly rebuilt. In this case, each transition is constructed as a three-point *LineString* that initialises at the origin *Node* centroid, passes through the midpoint of the shared *CellBoundary*, then arrives at the centroid of the destination *Node*. Finally, the bounding envelope is recalculated based on the updated coordinates of the full model. A visualisation of the updated

thick-door IndoorGML model is shown in Figure 4.7. The 3D geometry of *Cells* are colour-coded based on their labelled function, and *Edges* are marked as green lines connecting the white points corresponding to *Nodes*.

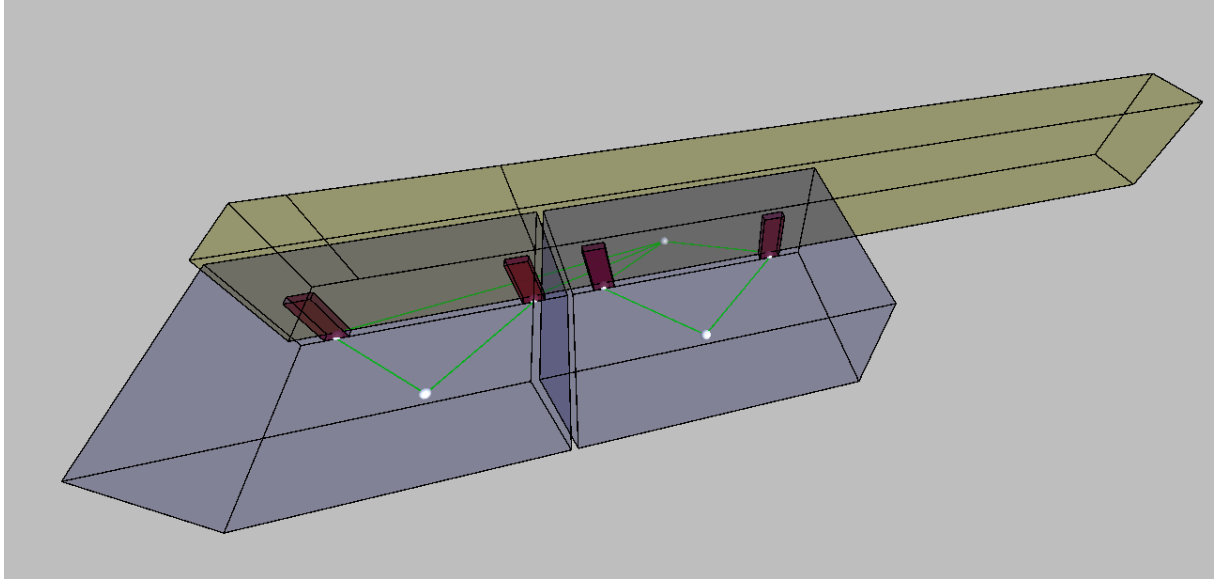


Figure 4.7: Output to-scale IndoorGML model.

4.4. Web Viewer Implementation

At this stage of the pipeline, the IndoorGML model has been created, and each 3DGS scene is accompanied by a transformation matrix that aligns the scene to the geometry in the IndoorGML model. It is therefore possible to use the geometry and topology stored in the IndoorGML model and apply them to the 3DGS scenes. However, the XML format of the model is not suitable for direct ingestion and use for a Javascript-based web viewer, and thus needs to be adapted into a database format that can be efficiently queried.

4.4.1. Database Schema

A PostGIS-enabled PostgreSQL database was chosen, given the need to store and query geometric attributes. The database schema is structured to mirror the *Cell-Node-Edge* tripartite of the IndoorGML core model directly, such that there is one table per concept. This matching schema is intentional: by preserving the topological organisation of the IndoorGML schema in the relational database, information needed by the web viewer can be directly performed using SQL queries. The schemas and stored data are shown in Table 3, Table 4, and Table 5 respectively.

The Room table represents the *CellSpace*, storing the floor polygon geometry, ceiling height, cell type, the path to the associated 3DGS splat file, and a JSONB column that encodes the axis permutation and Umeyama transformation matrix. Note that *Cells* marked as *ConnectionSpace* do not have any splat files or transformation matrix associated with them. These are marked as *null* in the database.

The Node table stores the navigable centroid point associated with each *CellSpace*, linked by a foreign key to the Room table. The Edge table encodes a topological connection as the 3-point LineString earlier described in Section 4.3.2, with foreign keys to the origin and destination *Node*, and an additional attribute for door height. This three-table structure means that a topological query — for example, “Which room is reachable from my current room?” — can

4.4. Web Viewer Implementation

easily be expressed as an SQL query to extract the relevant information from the PostGIS database.

id	room_name	cell_type	geom	height	file_path	transform
C1	0470	GeneralSpace	POLYGON Z((46.65737 ...	5.9	static/ splats/0470.p...	{"perm": [1, 2, 0], ...
C2	0450	GeneralSpace	POLYGON Z((58.87032 ...	5.9	static/ splats/0450.p...	{"perm": [0, 2, 1], ...
C3	0807	TransitionSpace	POLYGON Z((46.02113 ...	5.9	static/ splats/0807.p...	{"perm": [1, 2, 0], ...
C4	0470-0807-D1	ConnectionSpace	POLYGON Z((47.62493 ...	3.0		
C5	0470-0807-D2	ConnectionSpace	POLYGON Z((56.45986 ...	3.0		
C6	0450-0807-D1	ConnectionSpace	POLYGON Z((59.8904 8...	3.0		
C7	0450-0807-D2	ConnectionSpace	POLYGON Z((68.65956 ...	3.0		

Table 3: Schema and data of Cell table.

id	room_id	geom
S1	C1	POINT Z(52.6100225 4.4819825 0)
S2	C2	POINT Z(64.81492 4.427782499999999 0)
S3	C3	POINT Z(63.87645666666666 10.609408333333333 0)
S4	C4	POINT Z(48.1434525 8.4126775 0)
S5	C5	POINT Z(56.978404999999995 8.3764475 0)
S6	C6	POINT Z(60.40894 8.3624025 0)
S7	C7	POINT Z(69.17809 8.3274975 0)

Table 4: Schema and data of Node table.

id	from_state	to_state	geom	dual_boundary	door_height
T1	S1	S4	LINESTRING Z(47.6249...	B1-REVERSE	3.0
T2	S3	S4	LINESTRING Z(47.6289...	B2-REVERSE	3.0
T3	S1	S5	LINESTRING Z(56.4598...	B3-REVERSE	3.0
T4	S5	S3	LINESTRING Z(56.4638...	B4-REVERSE	3.0
T5	S2	S6	LINESTRING Z(59.8904...	B5-REVERSE	3.0
T6	S6	S3	LINESTRING Z(59.8944...	B6-REVERSE	3.0
T7	S2	S7	LINESTRING Z(68.6595...	B7-REVERSE	3.0
T8	S7	S3	LINESTRING Z(68.6635...	B8-REVERSE	3.0
T1-REVERSE	S4	S1	LINESTRING Z(48.6579...	B1	3.0
T2-REVERSE	S4	S3	LINESTRING Z(48.6619...	B2	3.0
T3-REVERSE	S5	S1	LINESTRING Z(57.4929...	B3	3.0
T4-REVERSE	S3	S5	LINESTRING Z(57.4969...	B4	3.0
T5-REVERSE	S6	S2	LINESTRING Z(60.9234...	B5	3.0
T6-REVERSE	S3	S6	LINESTRING Z(60.9274...	B6	3.0
T7-REVERSE	S7	S2	LINESTRING Z(69.6926...	B7	3.0
T8-REVERSE	S3	S7	LINESTRING Z(69.6966...	B8	3.0

Table 5: Schema and data of Edge table.

4.4.2. Web Viewer Server and Scene Rendering

The web viewer is implemented as a single-webpage application, served by a FastAPI backend. A single data endpoint — GET /api/scene-data — is used, which queries all three tables and serialises the result as a GeoJSON FeatureCollection. Each *Cell*, *Node*, and *Edge* is represented as a typed Feature carrying its geometry and all accompanying attributes. This allows the client to load the complete topological and geometric model at startup. The 3DGS splat files themselves are served as static assets directly from the filesystem, with file paths resolved from the `file_path` column of the room table at load time.

Scene rendering is handled by the Gaussian Splats 3D library, a Three.js-based renderer for 3DGS .ply files created by Mark Kellogg. Each 3DGS ply file is instantiated with its independent GS3d.Viewer. The scene corresponding to the current room location is rendered, while remaining scenes are hidden from view. Note that we render the 3DGS splat scene in its original units, which affects the operations discussed below.

The PolygonZ geometry stored in the room table serves as the primary geometric input for constructing navigation constraints. The 2D geometry is rebuilt in 3D after incorporating the corresponding height value. Each room’s 3D polygon is then slightly inset by 0.3m to form a smaller enclosing polygon. This reduction is intentional to prevent the camera from moving too close to the wall space where Gaussians are clustered. To enforce this geometric constraint, a ray-casting point-in-polygon test is performed. If the horizontal coordinates of the camera is found to lie outside the room’s boundary, its position is projected to the nearest polygon edge instead. The vertical position of the camera is independently checked and clamped to the height interval defined by the room’s height attribute. These checks are run on every animation frame. As the Three.js camera position is defined relative to the 3DGS scene, the coordinates are converted to floor plan coordinates to perform the testing, using the axis permutation and the inverse of the transformation matrix. Any corrections in positions are similarly converted back to the 3DGS space before applying it to the camera.

4.4. Web Viewer Implementation

In contrast to the room setups, each door's 3D polygon is instead offset outwards by 0.6m, creating a net 0.3m intrusion into the adjacent rooms. This allows us to use door geometry as detection zones to trigger transitions between scenes, complete with the slight buffer for more reliable detection (see Figure 4.8). When the camera position enters the door geometry, the *from_node* and *to_node* foreign keys of the transition table are used to query the connecting *Node*, as well as the file path to the corresponding splat scene. This scene of the new active room is then loaded, and the camera position is placed at the midpoint of the intruding vertical face of the door polygon. This is done using the same point-in-polygon test to determine which face of the buffered door polygon lies inside the destination room.

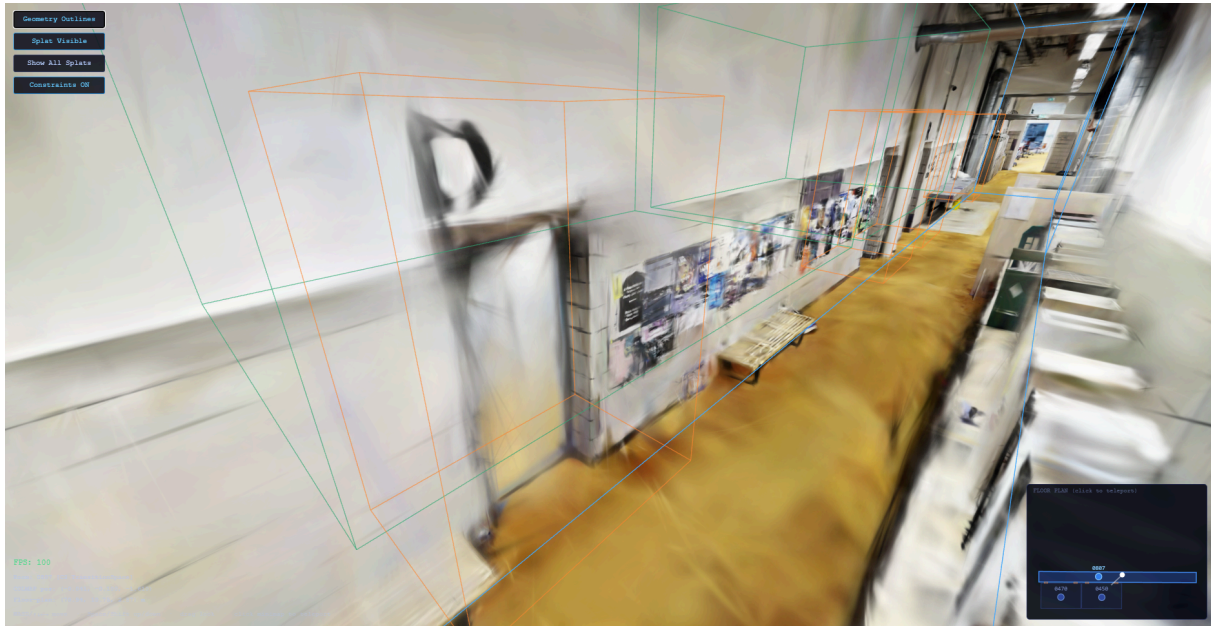


Figure 4.8: Door geometry and their intrusion into the adjacent room(s).

Finally, the camera direction is carried over to the new scene. Since the reference floor plane is equivalent for both rooms, the pitch angle is unaffected and can thus be inherited directly from the origin room. A two-step transformation method is then applied to the horizontal component of the camera direction – first from the origin to the floor plan reference, then subsequently to the coordinate system of the destination *Cell*. This effectively reproduces the same horizontal viewing angle from origin to destination. The transition process mimics the wayfinding behaviour of physically passing through a door to reach the adjoining room. Separately, the geometric boundaries of the scene is then rebuilt using the transformation matrix of the active room.

To prevent accidental triggers of a reverse transition, safeguards are enabled upon transiting into a new scene. Once a transition occurs, further transitions are disabled from occurring until the camera exits the geometry of the door. A short 3-second timer serves as a fallback preventing any transitions, in the event that the exit condition is not satisfied. Additionally, a minimap is rendered that allows the user to directly teleport to any room, without traversing the transition graph.

5. Results

Following the implemented methodology, the results across all stages of the pipeline are discussed. This includes intermediate outputs from the sparse reconstruction, the 3DGS reconstruction, alignment accuracy, and finally the topologically-linked dataset offered by the web viewer prototype.

5.1. Sparse Reconstruction Results

The sparse reconstruction statistics, summarised in Table 6, reflect the varying geometric complexity of the three selected rooms. Room 0470 achieves the highest reconstruction quality across all metrics: a registration rate of 98.81% from 840 input images, a sparse point count of 300,105, and a mean track length of 5.11 images per point, indicating that the majority of reconstructed points are observed from multiple viewpoints and are therefore well-constrained. Room 0450 performs comparably, with a registration rate of 88.93% and a mean track length of 3.73.

Room 0807, the corridor, presents a highly contrasting profile. The registration rate of 68.33% — the lowest of the three rooms — results in only 548 of 802 input images being successfully placed, and the sparse point count of 89,534 is substantially lower than the lecture halls despite the corridor’s longer video length. The lower value for mean observations per image is a direct result of parameter reductions during the feature extraction process: the number of images ingested by the subsequent 3DGS reconstruction phase had to be significantly reduced to fit the memory and computational constraints. Overall, the mean reprojection errors for all three rooms fall within the range of 1.24–1.30 pixels, which is indicative of acceptable camera pose accuracy for use as training viewpoints.

Cell/Parameter	0450	0470	0807
Input video length	2m51s	4m40s	9m35s
Input images	515	840	802
Registered images	458	830	548
Registration rate	88.93%	98.81%	68.33%
Sparse point count	181,364	300,105	89,534
Mean reprojection error	1.30077	1.24322	1.25182
Mean track length	3.72739	5.10869	3.86633
Mean observations per image	1,476.02	1,847.16	630.423

Table 6: Sparse reconstruction statistics per Cell.

5.2. 3DGS Reconstruction Results

The 3DGS training configuration, shown in Table 7, reflects the hardware constraints discussed in Section 4.1.3. A spherical harmonic degree of 1 is applied uniformly across all three rooms, and training is fixed at 30,000 iterations throughout. The image resize factor varies per room in response to available VRAM: 0470, with the largest input image set, is trained at full resolution; 0450 is downsampled by a factor of 2; and 0807 requires a factor of 4 despite having fewer training images, reflecting the increased per-tile rasterisation cost of the visually complex corridor scene.

5.2. 3DGS Reconstruction Results

Cell/Parameter	0450	0470	0807
Training images	458	830	548
Image resize factor	2	1	4
Spherical harmonic degree	1	1	1
Training iterations	30,000	30,000	30,000
Initial Gaussian count	181,364	300,105	89,534
Final Gaussian count	800,000	800,000	1,200,000

Table 7: 3DGS reconstruction statistics per Cell.

All three rooms reach or exceed 800,000 final Gaussians, with 0807 producing 1,200,000 — the highest count — despite its comparatively sparse initialisation. The complex and cluttered distribution of scene furniture (see Figure 5.1) encourages aggressive densification in the training phase. The increased number of Gaussians better models large and sharp variance in colours and structures, thus improving the photometric loss.



Figure 5.1: Screenshot of 3DGS reconstruction of corridor 0807.

0470 presents the most visually complete reconstruction, with little blurring and well-defined edges (see Figure 5.2). This is likely due to the comparatively larger colour variance in the scene, as well as a larger number of reference camera views to better constrain the novel-view synthesis. In comparison, 0450 suffers from excessive blurring and erroneous colour compositing at multiple locations. It is possible that despite its sparse reconstruction statistics being comparable to 0470, the repetitive colour and texture of the chairs, tables and flooring resulted in poor camera pose estimation. Furthermore, the thin structures of the chairs exacerbate the struggles in reconstruction, in contrast to the multi-coloured chairs in 0470 providing distinct keypoint separation.



Figure 5.2: Screenshot of 3DGS reconstruction of room 0470.



Figure 5.3: Screenshot of 3DGS reconstruction of room 0450.

It is also relevant to mention the interactions of the three scenes and highlight the need for topological relationships. Each scene contains floaters which extend beyond the enclosing walls that serve as the expected scene bounds. As an example, the floater ellipsoids are reduced to their centroids and visualised as a point cloud in Figure 5.4. The distribution of these floaters are interesting, as they often drift and expand away from the scene bounds from windows and open areas that were not captured in the route trajectory. The composite of these floaters may then visually represent the outdoor environment, or the view of an adjacent indoor space through a common window. There is thus meaningful value in preserving these floaters, since they contribute to a user's positioning, orientation and contextual awareness of an indoor scene.

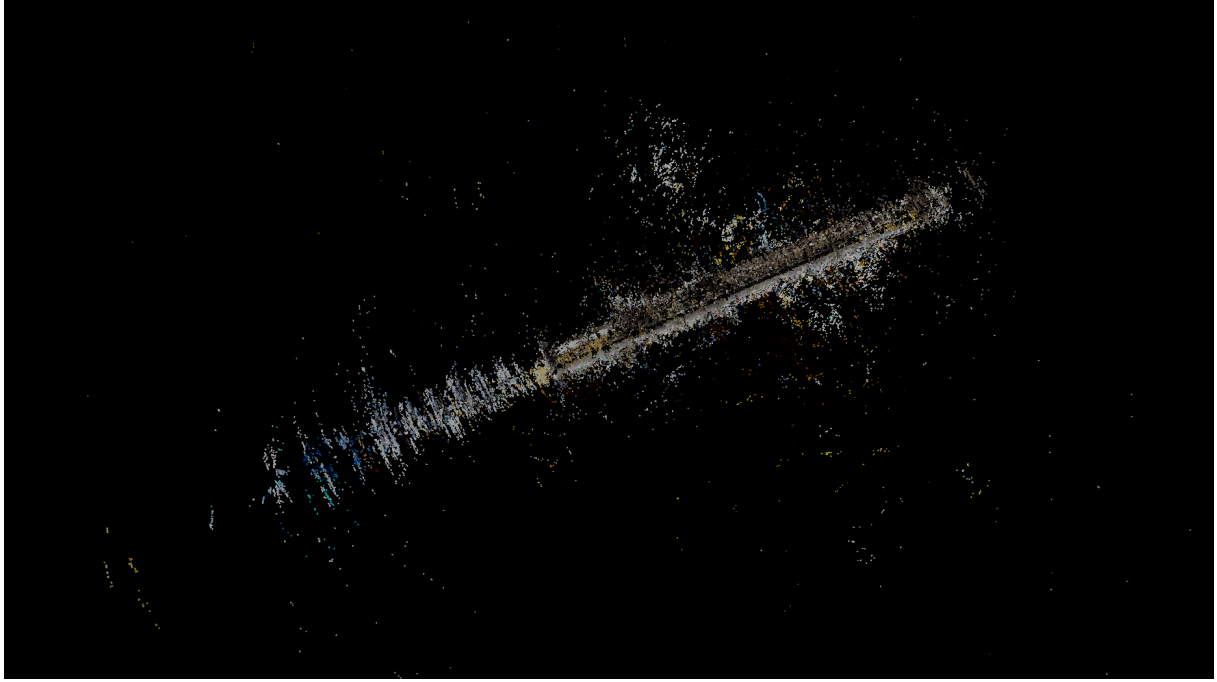


Figure 5.4: Floater distribution (centroid-only) of corridor 0807.

5.3. Alignment Results

The Umeyama transformation parameters and per-correspondence residuals are detailed in Section C. A summary of the transformation results is shown in Table 8. The results across the three rooms differ substantially across various aspects.

Parameter	C1 (0470)	C2 (0450)	C3 (0807)
Correspondences	8	8	14
Scale factor	1.100	1.085	4.884
Tilt ($^{\circ}$)	89.09	35.19	89.72
RMS XY (m)	0.1075	0.1543	0.4132
RMS Z (m)	0.0259	0.0303	0.0292
RMS 3D (m)	0.1114	0.1583	0.4153
Axis permutation	(1,2,0)	(0,2,1)	(1,2,0)

Table 8: Umeyama transformation summary per Cell.

The scale factors recovered for 0470 and 0450 are close to 1, indicating that both sparse reconstructions were relatively close to the metric scale of the floor plan. In contrast, the scale factor of 0807 at 4.884 indicates that the reconstructed geometry was substantially compressed, relative to the reference floor plan. This is consistent with the lower image registration rate and point count in Table 6. The difference in tilt values reflect the arbitrary initialisation of each SfM reconstruction, thus requiring the permutation step to map the different axis ordering for each room.

The per-correspondence XY residuals for 0470 range from 0.072 m to 0.149 m, with a root mean square (RMS) error of 0.1075 m in the horizontal plane. All eight correspondence points contribute residuals of similar magnitude, with no obvious outlier point (see Table 10). 0450 has a slightly higher RMS XY of 0.1543 m, with residuals ranging from 0.030 m to 0.247 m (see Table 11). The largest residual is point 7 with 0.247 m. This point occurs at the corner

of the room nearest the 0807 corridor boundary, where the close proximity to the door and the presence of an occluding cabinet resulted in lower point density and thus a less precise correspondence point chosen. 0807 has a significantly poorer RMS XY error of 0.4132m. The per-point correspondence errors in Table 12 highlight point 5 – 0.883m – and point 6 – 1.114m – as the outlier points. These correspond to one of the shared door frames between corridor 0807 and room 0450. The cause for the large drift is likely attributable to the poor point density throughout the sparse reconstruction of 0807, as well as insufficient coverage during the video capture process.

Overall, the residual errors are acceptable for alignment, as the recovered parameters are sufficient to correctly place each scene in relative orientation to each other at acceptable accuracy tolerances. Figure 5.5 visualises the placement of all three sparse point clouds, after the transformation matrices have been applied. Note that the orientation parallels the floor plan: 0470 in blue and 0450 in orange are correctly placed next to each other, and the large cluster of outlying points to the east and southeast indicate the outdoor-facing windows; 0807 in green is also adjacent to, and correctly orientated in alignment with both 0470 and 0450.

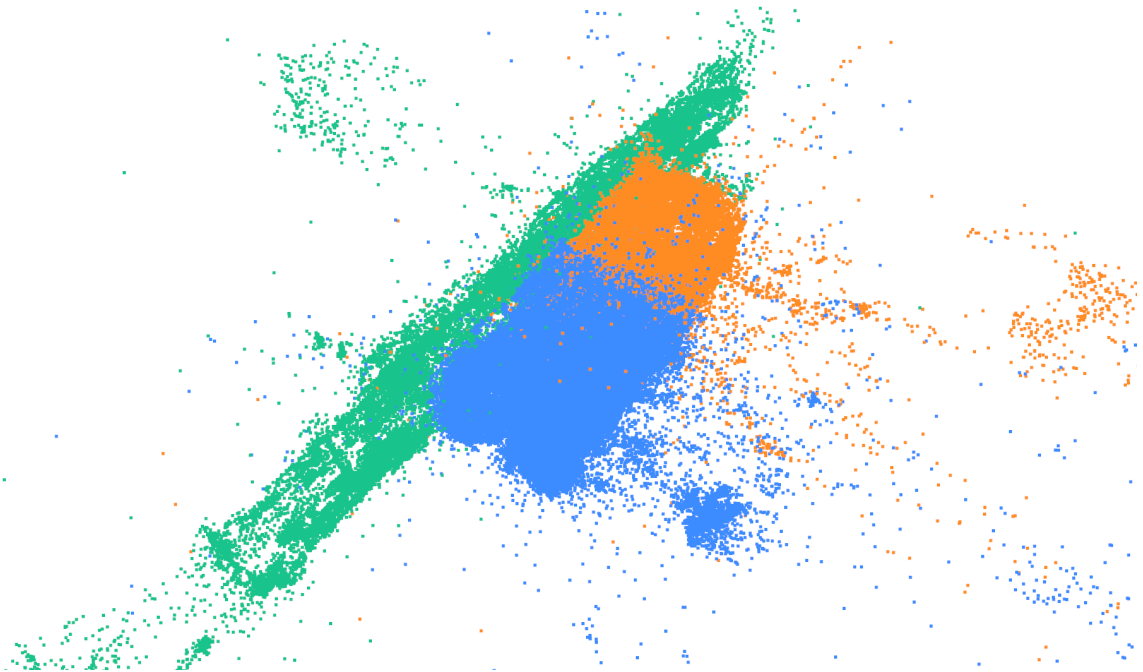


Figure 5.5: Visualisation of aligned sparse point clouds.

5.4. Web Viewer Implementation

The web viewer is evaluated as a functional prototype against the three operational requirements established by the research questions: real-time rendering performance, geometric constraint enforcement, and room transitions that are informed by topology.

5.4.1. Real-time Performance

The file size statistics in Table 9 illustrate the compressive effect of the reconstruction pipeline across its successive stages. The input image sets, extracted from video at a controlled frame rate, already represent a substantial reduction from the raw video captures — from 889 MB to 270 MB for 0450, and from 2.90 GB to 331 MB for 0807. The 3DGS splat files effect a further reduction relative to the input images: 81 MB from 270 MB for 0450 (a 70% reduction), 81 MB from 367 MB for 0470 (78%), and 122 MB from 331 MB for 0807 (63%). Notably, the 3DGS output for 0450 and 0470 is also smaller than their respective sparse reconstruction folders,

5.4. Web Viewer Implementation

despite encoding a substantially richer scene representation that includes per-Gaussian opacity, covariance, and colour coefficients. This compactness has direct implications for the web viewer: each room’s scene data is delivered as a single self-contained PLY file, and file sizes in the 80–120 MB range which are easily loaded and rendered compared to a single, larger scene file.

Cell/Parameter	0450	0470	0807
Video capture	889MB	1.41GB	2.90GB
Input images	270MB	367MB	331MB
Sparse reconstruction	105MB	190MB	63MB
3DGS reconstruction	81MB	81MB	122MB

Table 9: File size statistics per Cell.

The decomposition into discrete 3DGS scenes allow the frame rate of the web viewer to remain around 100 frames per second (FPS) throughout navigation in all three rooms. A brief performance reduction – in the form of a drop to 11 FPS and user input delay – is noticeable at within the few seconds after the initial load. This is attributed to the pre-loading of the remaining scenes into their respective viewer in the background. Once the pre-loading is completed, the frame rate stabilises with no subsequent drops through navigations and transitions, since the scene data of the destination’s room is already loaded and awaiting rendering. The sustained performance of the frame rate indicates the per-room viewer architecture is sufficient to maintain real-time rendering on consumer-level hardware.

5.4.2. Geometric Constraints

The geometric constraint system successfully confines camera movement to the expected navigable volume of each room throughout testing. Figure 5.6 shows the rendered outlines of the 3D volume of the geometry. Note that the camera is positioned in line with a face (in blue) of the geometric constraints; the inward offset of 0.3m prevents the camera from moving further towards the left wall. This prevents the Gaussians along the wall from being viewed at a “behind” angle that was not qualified in the training phase. This is significant for visual quality: 3DGS scenes are known to degrade when rendered from viewpoints far outside the training coverage, producing visible artefacts from floater Gaussians — isolated ellipsoids that accumulate in poorly-observed regions of the scene. By enforcing that the camera remains within the constraint boundary, the viewer prevents the user from encountering these artefacts during normal navigation.



Figure 5.6: Outlines of geometry constraints in the web viewer.

5.4.3. Topological Transitions

A demonstration of the value of topology is shown in Figure 5.7. For visual clarity, Gaussians for all three scenes were reduced to their centroid, and overlaid. Each scene carries its own local coordinate frame. Rather than occupying spatially distinct and adjacent regions as the physical rooms do, the three scenes overlap almost entirely, with different orientations and scale. Note the significant scale difference for corridor 0807, as its Gaussians occupy a substantially smaller spatial extent despite the room’s larger geometric dimensions in the real world. The overlap is an expected outcome of SfM reconstruction, where coordinate frame and scale is arbitrarily determined based on the initialisation of the sparse reconstruction process. This proves that prior to transformation and alignment, no spatial relationship between the raw coordinates of two separate rooms carries physical or topological meaning.



Figure 5.7: Overlay of three 3DGS room scenes.

5.4. Web Viewer Implementation

Room transitions are structured to replicate the physical experience of passing through a doorway. When the camera enters a door trigger zone — defined by the extruded 3D volume of the door *Cell* — the scene first fades very briefly to black. Then, the active viewer is exchanged for that of the destination room. Finally, the camera is placed at the threshold of the door, and the corresponding viewing angle is applied. This arrangement ensures that the transition conveys a spatially coherent sense of movement: the user approaches a door, passes through it, and arrives on the other side facing into the room they have entered, consistent with the topological relationship encoded in the IndoorGML transition record linking the two *Nodes*.

Additionally, the combination of geometric bounds and door-based transitions enforce the topological relationship between the three scenes. Rooms 0470 and 0450 are only able to move to corridor 0807 through the two doors per room. Conversely, there is no direct topological connection between 0470 and 0450, and this is exemplified in the viewer: a camera in 0470 is not able to move beyond the geometric bounds towards the expected space of 0450, and vice versa.

As the individual scenes have been pre-loaded, there is no delay in rendering the new scene. Visually, the transition occurs almost instantaneously, which creates a more immersive experience moving between discrete scenes as compared to loading a separate scene file. Importantly, movement between adjacent scenes are only triggered through the appropriate doors: the geometric constraints for each scene successfully restricts the camera from moving through impermeable walls to view the space of the adjoining *Cell*, although in practice the non-active Gaussians are not rendered.

6. Discussion

Based on the functions described in the preceding chapter, the central research question is revisited to qualify the achievements of the pipeline. Limitations of the proposed implementation are foregrounded, and potential expansions to the work as well as its applications are discussed below.

6.1. Research Objective

The main research objective is to explore how discrete 3DGS scenes can be spatially aligned and integrated within an IndoorGML topological model to enable topologically-constrained and contextually continuous navigation of indoor environments in a web-based viewer. The pipeline is a clear demonstration that this integration is both conceptually possible and practically implementable based on the sequential and modular approach taken. Each stage of the pipeline is informed meaningfully by the sub-research questions, as follows.

6.1.1. Data Capture

How should the 3DGS scenes be captured and organised to adhere to the IndoorGML topological schema?

The choice of a per-room 3DGS reconstruction strategy adheres to IndoorGML’s Structured Space Model and its decomposition of indoor space into discrete, enclosed sub-spaces. A one-to-one association between room *Cells* and 3DGS scenes was adopted, with each scene captured and processed independently using video-derived image sets. This organisation enhances the 3DGS reconstruction – the walls and partitions defined as *CellBoundaries* coincide with the visibility discontinuities that make reconstructions of indoor spaces challenging. In reconstructing a single scene of multiple rooms, the common wall(s) and door(s) present immutable occlusions that weaken feature matching and incremental registration, resulting in poorly aligned sparse reconstructions and unstable camera poses. On the other hand, with the *Cell*-based approach, the clear confined geometry of each room enabled loop closure during the data capture phase, which positively impacted both the sparse reconstruction as well as the Gaussian training process.

6.1.2. Geometric Features

What method is most appropriate to obtain a constraining geometry that accurately fits the expected scene bounds?

The geometry constraint system was derived from the boundaries of each *Cell* – which were manually drafted following the reference floor plan – and given additional buffers during runtime. The viewer operations query this geometry and successfully confined camera movement within the expected navigable volume of each room. This meant that the camera could not reach regions beyond the expected scene bounds where 3DGS reconstruction quality significantly degrades. Importantly, this geometry is a natural output of the topology creation phase, which outputs the IndoorGML model. This means that no additional data collection is required beyond the inputs to the creation of the IndoorGML model.

6.1.3. Scene-to-Scene Transitions

How can a shared spatial reference frame be established to enable coordinate conversion between independently reconstructed 3DGS scenes?

The chosen approach relied on the geometry and topological relationships provided by an

6.1. Research Objective

external reference – the architectural floor plan. Each 3DGS scene is then mapped into this common coordinate frame, using its individual transformation matrix. The common spatial reference frame serves as an intermediary that allows spatial reasoning, positions and directions to be translated across rooms based on a two-step transformation process. The choice of the floor plan is also appropriate as it is independent of the reconstruction process.

What point-correspondence method produces a robust similarity transform between a 3DGS scene’s local coordinate system and a common reference frame?

The Umeyama least-squares similarity transform was used to calculate the transformation matrix between the local 3DGS coordinate system and the common coordinate frame of the floor plan. The least-squares formulation is especially suitable for this context because exact positional correspondence is not guaranteed, and often impossible – the finite density of the sparse point cloud and geometric abstraction of the floor plan each contribute imprecision to the point-pair coordinates. Umeyama’s least-squares approach is thus appropriate since it is robust against small imprecisions in point correspondences. The added initial step of projecting the sparse points to 2D – to match the planarity of the 2D floor plan vertices – reduced the problem to a 2D similarity transform, which further improves the accuracy of the solution.

6.1.4. Topology Creation

What attributes are required in addition to IndoorGML’s core concepts to enable topological linkages between 3DGS scenes and their shared transition spaces?

Several attributes were identified and parsed into the database along with the core attributes of IndoorGML’s Structured Space Model. The file path attribute per relevant *CellSpace* allows the viewer to locate and load the associated 3DGS ply file. Similarly, the transformation matrix and axis permutation were encoded to enable coordinate conversion and calculations between each Gaussian scene at runtime. An additional height attribute was included to allow extrusion of the 2D floor polygon to form 3D volumes for navigation and geometric checks.

6.1.5. Web Viewer Implementation

How can the IndoorGML schema be implemented in a database format that best supports the operations needed in a multi-3DGS web viewer?

The GML file format of IndoorGML is not suited for web viewer implementation, and thus required parsing into a database schema. The core schema of the IndoorGML model was preserved and translated into three tables in PostGIS, mirroring the *Cell-Node-Edge* trifecta directly. The information in both topographic space and dual space of the Structured Space Model is thus stored in a relational database structure, with foreign keys linking individual modules. This allowed topological queries to be directly expressed as standard SQL.

It is relevant to note that previous researchers (Alattas et al., 2021) have developed a database format for combining and representing legal information with IndoorGML (Alattas, 2022). Furthermore, as at the time of writing, the Open Geospatial Consortium (OGC) is developing – but has yet to publish – Part 2 of the IndoorGML version 2.0 schema, which specifically focuses on technical implementation schemas such as SQL, JSON and GML. The *Cell-Node-Edge* relational table structure used in this pipeline was designed to be functional and suited for the purpose of demonstration in the web viewer.

To what extent does the topology-constrained multi-3DGS viewer enforce topological navigation rules while maintaining real-time performance?

The prototype viewer successfully enforced topological connectivity rules throughout testing. Direct navigation between rooms that are not topologically connected is prevented by the

geometric constraints of the scene; scene transitions can only be executed through door trigger zones formed by the extruded *ConnectionSpace* polygons. Consistent frame rates of 100 FPS demonstrate real-time performance, albeit marred by short frame drops and delay caused by background preloading of adjacent splats. Transitions were designed to preserve the camera direction for consistent orientation, although the overall process cannot be said to be seamless and immersive due to both the spatial teleport between scenes, and the brief fade implemented to mask it.

6.2. Limitations

Despite achieving the main research question, limitations to the proposed methodology and proof-of-concept naturally arise as the consequence of design choices. These limitations are identified and alternative solutions are proposed where relevant.

6.2.1. Alignment Strategies

The current alignment pipeline requires a metrically accurate floor plan to serve as the common reference frame. While architectural floor plans are not uncommon, situations with unavailable or erroneous floor plans restrict the applicability of the method. This is pertinent for older buildings, especially from a heritage angle, where topologically-informed 3DGS representations contribute meaningfully to heritage preservation.

In the absence of a reference floor plan, an alternative approach could be to establish local alignment between connected scenes. Pairwise transformations of adjacent scenes can potentially be computed using shared geometric features at their common boundary. In most cases, this would manifest as the door frame, or wall junctions. This could then be encoded as a transformation matrix to each row in the *Edge* table instead. However, the accuracy of such an alignment approach is heavily dependent on the manual selection of point correspondences, as well as the adaptations to account for the geometric distance between the two faces of the shared features – for example, the thickness of the walls and doors. This would likely also necessitate adaptations to the construction of the transition trigger zones.

LiDAR-augmented SLAM – such as XGrids – may well serve as data capture approaches that solve this limitation where floor plans are not available. The high precision of LiDAR points enable more accurate pairwise transformations, and the scan data can simultaneously be used to directly extract the 3D geometry of the *Cell*, thus serving as the boundary constraints.

6.2.2. 3D Volume Encoding

The database schema currently stores a single height value per *CellSpace*, which is then used to extrude the 2D PolygonZ geometry to a 3D volume. This is a deliberate simplification based on the geometric simplicity of the three rooms: horizontal checks can be performed quickly on a 2D polygon compared to querying a full 3D boundary. This approach is naturally inadequate for spaces with varying heights, such as a sloped or tiered ceiling. In these cases, it may be more suitable to encode the *Cell* geometry as a POLYHEDRALSURFACE Z in PostGIS. Height-based constraints would then need to be adapted correspondingly to query from the 3D volume rather than the stored height value.

6.2.3. Manual Topology Creation

The IndoorGML model was manually constructed using the suite of graphical tools from STEMLab, using the architectural floor plans as a reference overlay. While relatively straightforward for a three-room demonstration, the process becomes laborious when scaled to a building with large numbers of well-connected rooms. The proposed pipeline does not simplify

6.2. Limitations

the process of IndoorGML creation; instead, it integrates a visual representation layer that supplements the topological model with strong, immersive visuals. Rather than lowering the cost of entry for IndoorGML creation, it can potentially improve the cost-benefit ratio to investing the time and resources to create a useable IndoorGML model.

Alternatives are possible based on the open-source tools made available for creating IndoorGML from other information models. For example, Diakit  et al. (2022) developed the IFC2IndoorGML tool which extracts the relevant geometric and semantic information in an IFC BIM model, and parses it into the IndoorGML schema. In most cases, existing geometric and topological information is still necessary to create the IndoorGML model.

6.2.4. Transition Animation

The room transition can be improved upon to create a more seamless and immersive cross-over between scenes. The current implementation shifts the camera position to avoid being placed in close proximity to the Gaussians that depict the door. This design choice avoids rendering the door splats at untrained angles to prevent visual breakdown.

An alternative consideration could aim to improve the spherical harmonic function of Gaussians local to the door. The volume of a shared door can be determined as the sum of Gaussians from two scenes, that each describe one face of the door. In theory, it could be possible to confine these Gaussians to a common plane, and train this set of Gaussians on relevant images from both scenes. This means that within the neighbourhood of the door, each scene would contain local Gaussians with a spherical harmonic function that provides accurate colour when viewed from both sides of the common plane. The transition zone can then be reduced to this vertical plane, which should result in an improved room transition experience.

6.2.5. Prototype Scale and Constraints

The pipeline was implemented using consumer-grade hardware. Notably, the 8GB of VRAM proved to be the limiting factor, primarily for the 3DGS training phase. The practical ceiling on Gaussian count, downsampling and tiling of reference images directly affects the quality of the 3DGS reconstruction. Expanding the scale of implementation naturally requires better hardware.

The overall pipeline uses baseline tools and methods to demonstrate the topological linkage of 3DGS scenes. This can be seen as both a limitation, and an advantage. The modular and generic implementation of the pipeline implies that improvements and enhancements can be made at each stage of the process. It should first be noted that the fundamental requirements for the pipeline are the SfM points and an external reference frame. Beyond these prerequisites, each stage can be adapted accordingly. Still images can be used as input to the sparse reconstruction and 3DGS training, and the 3DGS reconstruction is itself independent to the rest of the intermediate output. As such, alternative training methods can be adopted for more accurate reconstruction results. One example is PlanarGS (Jin et al., 2025), in which the known geometry of the room – whether captured by LiDAR SLAM approaches, or via geometric priors – can be used to constraint Gaussians to wall surfaces. This enables a stable training process, reduce the number of floater artefacts, and results in a more geometrically-accurate reconstruction.

The splat files may also be converted to more efficient formats – .splat, .ksplat, or .spz – and the web viewer adapted to ingest these files instead. In the case where an IndoorGML model is already available for the site of interest, the manual model creation can be skipped, and the existing model directly parsed into the database. The pipeline is thus modifiable to suit the particular use case and available tools of the user.

Interestingly, although the demonstration of the pipeline focused on three indoor spaces, the IndoorGML schema can be further exploited to expand beyond purely indoor environments. Using an *AnchorSpace*, a continuous 3D representation for indoor-outdoor transition zones can be created, connecting 3DGS scenes of the exterior facade and the entrance *Cell* in a fluid manner. This thus preserves spatial orientation and awareness when the user navigates across building exterior and interior spaces. Conceptually, the topological schema remain applicable to outdoor environments as well, although clear definitions and data capture approaches are necessary to cleanly demarcate external spaces into discrete *Cells*.

6.2. *Limitations*

7. Conclusions and Future Work

3D Gaussian Splatting has established itself as a powerful method for scene reconstruction, due to its ability to present photorealistic representations of the real world in real-time. Its application to multi-room indoor environments, however, exposes a fundamental structural problem: a singular scene representing the entire space struggles with weak registration and unstable camera poses due to camera occlusions, while independently reconstructed scenes of partitioned sub-spaces have arbitrary scale, orientation and position, with no clear spatial relationships among one another. Without a framework to govern the relationships between scenes, photorealistic indoor reconstruction remains a collection of isolated artefacts rather than a holistic, connected and navigable environment.

7.1. Conclusion

This thesis has demonstrated that IndoorGML’s topological model provides exactly this framework. By extending the core *Cell-Node-Edge* schema with supplementary attributes — scene file references, Umeyama transformation matrices, and room height values — the model acquires the operational information needed to link independently reconstructed 3DGS scenes into a spatially coherent, navigable whole without modifying either the Gaussian representation or the topological schema’s core structure. The pipeline proceeds sequentially through per-room scene reconstruction, spatial alignment to a shared floor-plan reference frame, IndoorGML model construction and coordinate replacement, PostGIS database ingestion, and web viewer implementation — each stage addressing one component of the main research question and feeding directly into the next.

The alignment of each room’s COLMAP coordinate frame to the shared floor-plan reference is achieved through a two-stage Umeyama similarity estimation. By first fitting a floor plane to the manually selected correspondence points and reducing the problem to a two-dimensional similarity solve, the method produces an acceptable transform that also addresses the scale ambiguity inherent to Structure-from-Motion reconstruction. The per-room transformation matrices, stored in the database alongside the topological structure, serve as the operational link between the IndoorGML reference frame and the per-room COLMAP spaces used at runtime.

The web viewer prototype confirms that the integrated pipeline supports real-time topologically-constrained navigation on consumer-grade hardware. Frame rate remains above 60 frames per second throughout normal navigation, geometric constraints derived from the CellSpace boundaries successfully prevent the camera from reaching scene regions outside the training distribution, and room transitions follow the connectivity rules encoded in the IndoorGML transition graph — admitting passage only through door Cells and only in the directions the topology permits. Despite having the most challenges in the reconstruction phases, the corridor scene remains navigable and visually coherent within these constraints, demonstrating that the pipeline tolerates variation in reconstruction quality across Cells.

Taken together, the results confirm the central proposition of this work: that IndoorGML topology and 3DGS scene representation are complementary in precisely the ways each lacks independently. A 3DGS scene without topology is a spatially isolated artefact; an IndoorGML model without photorealistic scenes is a navigational abstraction with limited visual applicability. The mutually beneficial combination of the two fields thus broadens their range of urban environment and geospatial applications.

7.2. Future Work

The paper proposed a viable pipeline for associating multiple 3DGS reconstructions using topological relationships. The proof-of-concept serves as a simplified demonstration of the feasibility of implementing this pipeline for real-world purposes. There is therefore plenty of opportunities to extend the current work and implement additional functionalities, the most pertinent of which are discussed below.

7.2.1. Topology-based Pre-loading

Currently, the prototype first loads and renders the default scene, before sequentially preloading the remaining two scenes in the background. This approach does not scale well to larger topologies as it would incur substantial memory usage. A potential improvement is to implement connectivity-aware preloading. Given an active scene, scenes with direct adjacency can be pre-loaded. Upon a scene transition, a new query can be made to collect and pre-load scenes which are topological neighbours to the newly active scene. The maximum memory usage would thus be determined by the interplay between the *Cells* with the highest first-order connectivity, and the size of their associated 3DGS files.

7.2.2. Multi-Floor Coverage

The IndoorGML model created focused on a single-floor demonstration. The question of multi-floor coverage is thus a natural extension, since IndoorGML also supports vertical topology in the Structured Space Model. In this case, stairs and lifts connecting different floors are representable as *TransitionSpaces* with defined vertical connectivity. The current pipeline remains broadly applicable to multi-floor contexts. Specifically, since a staircase connects two separate floors, a subset of point-pair correspondences is necessary to align the staircase to each floor plan. The arising problem would then be if a sufficient number of stable point correspondences can be identified for the alignment to each floor.

7.2.3. Applications

A set of topologically-linked 3DGS scenes present multiple opportunities for end-user applications. The most direct application is immersive spatial navigation: allowing a user to virtually explore a multi-room environment displayed at photorealistic fidelity. Note that independent 3DGS scenes have already found applications in remote experiential visualisations, such as virtual tours of university campus, hospitals, and shopping malls. However, it can be argued that topological constraints to these 3DGS scenes provide the ability to move between spaces through their physical connections, which ultimately creates a more immersive effect compared to through a menu or map interface to teleport to a different scene.

The most suitable context for such applications lies in heritage architecture and cultural preservation. Photorealistic, navigable digital twins of historically significant buildings both preserve and provide access to important cultural infrastructure. Where physical degradation, restricted access or geographical distance prevent real-life engagement with heritage spaces, a topologically-structured set of 3DGS scenes can be served online to enable remote viewing and consumption. The application can also be enhanced with a temporal aspect; Repeated scene capture at intervals may reflect and narrate significant changes across time, or serve a monitoring function to highlight structural change.

A crucial benefit of topologically-accurate scenes lie in its accurate representation of human navigation. The imitation of how humans physically navigate space – through accessible doors and openings – present realistic wayfinding and route plotting functions. This is particularly important in the case of first responders, where accurate and plausible route information is

paramount for successful operations under emergency situations. While a typical 3DGS reconstruction of the building interior may sufficiently serve for crews to inspect the site and build prior spatial familiarity, it can be argued that the addition of topology offers additional benefits. Not only does the geometric constraints accurately reflect the impassable elements that routes need to avoid, the topology-based transition zones retains the sense of orientation when shifting between indoor spaces. This implies better re-orientation in cognitive mapping, and thus allows the viewer to familiarise with the surrounding at a faster rate.

The common theme across these applications is that the synergistic combination between the photorealistic visual fidelity of 3DGS and the topological structure of IndoorGML has greater applicability than neither component provides independently. To reiterate, a 3DGS scene without topology is a visually immersive but spatially isolated artefact; an IndoorGML bereft of photorealistic scenes provides strong navigational capabilities but lacks in representative visualisations. The integration of both thus considerably broadens the range of contexts they can be applied to.

7.2. *Future Work*

A Declaration of AI/LLM usage

Generative AI tools in the form of Claude and ChatGPT were used in several parts of the thesis process. In the report writing, generative AI was used to provide initial text drafts for some parts of the thesis, and as a directed search engine for sourcing academic papers. All references were vetted, analysed and adapted for final use in the report. The text drafts were extensively modified to fit my English writing style and grammar.

Claude was also used to refine initial code written. These are specifically the `coordinate_replacer.py`, `server.py` and `viewer.html` files at <https://github.com/mjte095/3dgs-topology>. All outputs were verified and tested to serve the purpose of the prompt. Each portion of the generated code was analysed to understand and validate its function, and adapted to form the final output.

All results, analyses and conclusions are my own. I verify that the content is original, and meets academic integrity standards.

B Reproducibility self-assessment

Relevant data and code were made openly available, accessible at <https://github.com/mjte95/3dgs-topology>. Due to size constraints, the input videos, images and raw sparse reconstruction files were not uploaded into the repository. However, intermediate outputs and the final 3DGS scene files were included as a GitHub release, to ensure that the demonstrative web viewer experience can be replicated. The GitHub repository has sufficient documentation and explanation to minimally run the web viewer prototype and demonstrate the proof-of-concept.

I rate the reproducibility of this thesis as **Moderate** according to the provided scale.

C Umeyama Similarity Transformation Results

C.1 Point-pair Correspondence Errors

The reference floor plan coordinates and the residual errors of the transformed point-correspondence are detailed for each *Cell* in , Table 11, and Table 12. The terminology are as follows:

FP = Floor Plan, which is the reference point against which the transformed point is measured. $|XY|$ and $|Z|$ = Horizontal and vertical residuals, which are the Euclidean distances in each respective plane between the sparse point and its floor plan reference, after the estimated similarity transformation has been applied.

All values are expressed in metres based on the reference floor plan.

Index	FP X	FP Y	$ XY $	$ Z $
0	46.614	0.777	0.1224	0.0474
1	58.563	0.724	0.1488	0.0480
2	58.605	8.188	0.0883	0.0240
3	57.493	8.192	0.1270	0.0100
4	56.460	8.197	0.0735	0.0221
5	48.658	8.230	0.0723	0.0121
6	47.625	8.235	0.0718	0.0283
7	46.657	8.239	0.1249	0.0151

Table 10: Point-pair correspondence alignment errors for C1 (0470).

Index	FP X	FP Y	$ XY $	$ Z $
0	58.829	0.721	0.1867	0.0304
1	70.759	0.664	0.1624	0.0304
2	70.802	8.140	0.1081	0.0234
3	69.693	8.144	0.0295	0.0039
4	68.660	8.148	0.0779	0.0717
5	60.923	8.178	0.1063	0.0333
6	59.890	8.182	0.1959	0.0339
7	58.870	8.186	0.2469	0.0151

Table 11: Point-pair correspondence alignment errors for C2 (0450).

Index	FP X	FP Y	XY	Z
0	46.021	8.603	0.2793	0.0035
1	47.629	8.595	0.2240	0.0064
2	48.662	8.590	0.2106	0.0061
3	56.464	8.560	0.0655	0.0020
4	57.497	8.556	0.0607	0.0045
5 [†]	60.927	8.542	0.8833	0.0299
6 [†]	59.894	8.546	1.1141	0.0600
7	68.664	8.511	0.0963	0.0215
8	69.697	8.506	0.1746	0.0240
9	93.024	8.411	0.2160	0.1039
10	93.024	11.486	0.2251	0.0783
11	56.959	11.652	0.0440	0.0115
12	48.200	11.746	0.1102	0.0156
13	46.031	11.758	0.1911	0.0422

Table 12: Point-pair correspondence alignment errors for C3 (0807).

C.2 4x4 Umeyama Transformation Matrix

The best Umeyama similarity transform results for each *Cell* is converted into a 4x4 matrix, as shown in Table 13, Table 14 and Table 15. These are thereafter parsed into the back-end database along with the corresponding axis permutation.

0.048722	0.993214	−0.470278	52.488933
−0.003727	−0.470587	−0.994255	4.415168
−1.098918	0.045632	−0.017478	1.833688
0.000000	0.000000	0.000000	1.000000

Table 13: 4×4 Umeyama transformation matrix for C1 (0470).

0.389061	−0.844970	0.558572	64.356789
1.010122	0.279330	−0.281025	3.513645
−0.075049	−0.620762	−0.886774	1.833166
0.000000	0.000000	0.000000	1.000000

Table 14: 4×4 Umeyama transformation matrix for C2 (0450).

0.437985	−4.830375	−0.576369	63.180888
0.028183	−0.576171	4.850133	10.023060
−4.864559	−0.438246	−0.023794	1.676913
0.000000	0.000000	0.000000	1.000000

Table 15: 4×4 Umeyama transformation matrix for C3 (0807).

D References

- Alattas, A. (2022). The Integration of LADM and IndoorGML to Support the Indoor Navigation Based on the User Access Rights. *A+BE / Architecture and the Built Environment*, 12(5), 1–344. <https://doi.org/10.7480/abe.2022.05.6314>
- Alattas, A., Vries, M. de, Meijers, M., Zlatanova, S., & Oosterom, P. van. (2021). 3D pgRouting and visualization in Cesium JS using the integrated model of LADM and IndoorGML. *FIG Working Week 2021 20-25 June. : Smart Surveyors for Land and Water Management - Challenges in a New Reality*. https://gdmc.nl/publications/2021/WS_03.3_alattas_de_vries_et_al_11114.pdf
- Autodesk. (2025). *Introduction to ReCap Professional: Point Cloud Display and Navigation*. <https://www.autodesk.com/learn/ondemand/course/introduction-to-recap-professional/module/73O0tAqdl09EsaOz8WNkK3>
- Chen, G., & Wang, W. (2025). *A Survey on 3D Gaussian Splatting*. <https://arxiv.org/abs/2401.03890>
- Chen, T., Shorinwa, O., Bruno, J., Swann, A., Yu, J., Zeng, W., Nagami, K., Dames, P., & Schwager, M. (2025). Splat-Nav: Safe Real-Time Robot Navigation in Gaussian Splatting Maps. *IEEE Transactions on Robotics*, 41, 2765–2784. <https://doi.org/10.1109/TRO.2025.3552348>
- Diakit , A., D  az-Vilari  o, L., Biljecki, F., Isikdag,   ., Simmons, S., Li, K., & Zlatanova, S. (2022). IFC2IndoorGML: An Open-Source Tool for Generating IndoorGML from IFC. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 295–301. <https://doi.org/10.5194/isprs-archives-XLIII-B4-2022-295-2022>
- Diakit , A., Zlatanova, S., Alattas, A. F. M., & Li, K. J. (2020). Towards IndoorGML 2.0: Updates and Case Study Illustrations. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 337–344. <https://doi.org/10.5194/isprs-archives-XLIII-B4-2020-337-2020>
- Fei, B., Xu, J., Zhang, R., Zhou, Q., Yang, W., & He, Y. (2025). 3D Gaussian Splatting as a New Era: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 31(8), 4429–4449. <https://doi.org/10.1109/TVCG.2024.3397828>
- Gao, K., Lu, D., He, H., Xu, L., Li, J., & Gong, Z. (2025b). Enhanced 3-D Urban Scene Reconstruction and Point Cloud Densification Using Gaussian Splatting and Google Earth Imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 63, 1–14. <https://doi.org/10.1109/TGRS.2025.3536169>
- Gao, Y., Gong, Y., Liu, Y., Li, J., Zhong, Z., Zhang, D., Zhang, Y., Xu, D., & Sun, X. (2025a). *Proxy-GS: Efficient 3D Gaussian Splatting via Proxy Mesh*. <https://arxiv.org/abs/2509.24421>
- Hafeez, J., Lee, J., Kwon, S., Ha, S., Hur, G., & Lee, S. (2020). Evaluating Feature Extraction Methods with Synthetic Noise Patterns for Image-Based Modelling of Texture-Less Objects. *Remote Sensing*, 12(23). <https://doi.org/10.3390/rs12233886>

- Hung, S.-H., Yen, T.-Y., Sun, W.-F., See, S., Hung, S.-H., & Chu, H.-K. (2025). *LOBE-GS: Load-Balanced and Efficient 3D Gaussian Splatting for Large-Scale Scene Reconstruction*. <https://arxiv.org/abs/2510.01767>
- Jamali, A., Rahman, A. A., Boguslawski, P., & Gold, C. M. (2015). An Automated 3D Indoor Topological Navigation Network Modelling. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 47–53. <https://doi.org/10.5194/isprsannals-II-2-W2-47-2015>
- Jamil, O., & Brennan, A. (2025). Immersive Heritage through Gaussian Splatting: A New Visual Aesthetic for Reality Capture. *Frontiers in Computer Science*. <https://doi.org/10.3389/fcomp.2025.1515609>
- Jamshidi, S., Ensafi, M., & Pati, D. (2020). Wayfinding in Interior Environments: An Integrative Review. *Frontiers in Psychology*. <https://doi.org/10.3389/fpsyg.2020.549628>
- Jeong, M., Kim, E., Park, S., & Choi, A. J. (2025). *NVSim: Novel View Synthesis Simulator for Large Scale Indoor Navigation*. <https://arxiv.org/abs/2510.24335>
- Jin, X., Jin, R., Li, B., Zou, D., & Yu, W. (2025). *PlanarGS: High-Fidelity Indoor 3D Gaussian Splatting Guided by Vision-Language Planar Priors*. <https://arxiv.org/abs/2510.23930>
- Kerbl, B. (2025). *The Impact and Outlook of 3D Gaussian Splatting*. <https://arxiv.org/abs/2510.26694>
- Kerbl, B., Kopanas, G., Leimkühler, T., & Drettakis, G. (2023). 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4), Article No.: 193, pp 1–14. <https://doi.org/10.1145/3592433>
- Kim, M., & Lee, J. (2015). Developing a Method to Generate IndoorGML Data from the Omni-Directional Image. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 17–19. <https://doi.org/10.5194/isprsarchives-XL-2-W4-17-2015>
- Knapitsch, A., Park, J., Zhou, Q.-Y., & Koltun, V. (2017). Tanks and Temples: Benchmarking Large-scale Scene Reconstruction. *ACM Trans. Graph.*, 36(4). <https://doi.org/10.1145/3072959.3073599>
- Kulhanek, J., Rakotosaona, M.-J., Manhardt, F., Tsalicoglou, C., Niemeyer, M., Sattler, T., Peng, S., & Tombari, F. (2025). *LODGE: Level-of-Detail Large-Scale Gaussian Splatting with Efficient Rendering*. <https://arxiv.org/abs/2505.23158>
- Ledoux, H. (2020). Are Your IndoorGML Files Valid?. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 109–118. <https://doi.org/10.5194/isprs-annals-VI-4-W1-2020-109-2020>
- Li, C., Huang, H., Lien, J.-M., & Yu, L.-F. (2021). Synthesizing Scene-aware Virtual Reality Teleport Graphs. *ACM Transactions on Graphics*, 40, 1–15. <https://doi.org/10.1145/3478513.3480478>
- Li, K.-J., Conti, G., Konstantinidis, E., Zlatanova, S., & Bamidis, P. (2019). 10 - OGC IndoorGML: A Standard Approach for Indoor Maps. In J. Conesa, A. Pérez-Navarro, J. Torres-Sospedra, & R. Montoliu (Eds.), *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation* (pp. 187–207). Academic Press. <https://doi.org/https://doi.org/10.1016/B978-0-12-813189-3.00010-1>

- Li, R., & Klippel, A. (2014). Wayfinding Behaviors in Complex Buildings. *Environment and Behavior*, 48, 0013916514550243. <https://doi.org/10.1177/0013916514550243>
- Liu, L. (2017). *Indoor Semantic Modelling for Routing: The Two-Level Routing Approach for Indoor Navigation* (Issue 17). TU Delft Open. <https://doi.org/10.7480/abe.2017.17>
- Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Commun. ACM*, 65(1), 99–106. <https://doi.org/10.1145/3503250>
- Müller, T., Evans, A., Schied, C., & Keller, A. (2022). Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics*, 41(4), 1–15. <https://doi.org/10.1145/3528223.3530127>
- Open Geospatial Consortium. (2025). *OGC IndoorGML 2.0 Part 1 – Conceptual Model* (Technical Report No. 22–45r5). <https://docs.ogc.org/is/22-045r5/22-045r5.html>
- Pan, W., Zhang, X., Zhai, H., Xiang, X., Jiang, H., & Zhang, G. (2025). Liberated-GS: 3D Gaussian Splatting Independent from SfM Point Clouds. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, , 26675–26685. <https://doi.org/10.1109/ICCV51701.2025.02476>
- Schulz, O., Werbrouck, J., & Beetz, J. (2023). *Towards Scene Graph Descriptions for Spatial Representations in the Built Environment*. https://www.ucl.ac.uk/bartlett/sites/bartlett/files/towards_scene_graph_descriptions_for_spatial_representations_in_the_built_environment.pdf
- Seo, Y., Choi, Y. S., Son, H., & Uh, Y. (2025). FLoD: Integrating Flexible Level of Detail into 3D Gaussian Splatting for Customizable Rendering. *ACM Transactions on Graphics*, 44(4), 1–16. <https://doi.org/10.1145/3731430>
- Shen, L., Chow, H. N., Wang, L., Zhang, T., Wang, M., Li, X., Li, Y., & Han, Y. (2026). GStitch: Spatial-temporal Fusion of 3D Gaussian Splattings for Scalable 3D Reconstruction. *Pattern Recognition*, 173, 112826. <https://doi.org/https://doi.org/10.1016/j.patcog.2025.112826>
- Umeyama, S. (1991). Least-squares Estimation of Transformation Parameters between Two Point Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4), 376–380. <https://doi.org/10.1109/34.88573>
- van Steijn, D. (2023). *Building an IndoorGML Model in (Near) Real-time* (p. 67) [Master's thesis]. <https://www.gdmc.nl/publications/2023/MScThesisDannyVanSteijn.pdf>
- Wang, L., Ren, Q., Liao, K., Wang, H., Chen, Z., & Tang, Y. (2025b). *StableGS: A Floater-Free Framework for 3D Gaussian Splatting*. <https://arxiv.org/abs/2503.18458>
- Wang, T., Wang, X., Hou, Y., Xu, Y., Zhang, W., & Zhan, Z. (2025a). *PG-SAG: Parallel Gaussian Splatting for Fine-Grained Large-Scale Urban Buildings Reconstruction via Semantic-Aware Grouping*. <https://arxiv.org/abs/2501.01677>
- Weisman, J. (1981). Evaluating Architectural Legibility: Way-Finding in the Built Environment. *Environment and Behavior*, 13(2), 189–204. <https://doi.org/10.1177/0013916581132004>
- Wu, T., Yuan, Y.-J., Zhang, L.-X., Yang, J., Cao, Y.-P., Yan, L.-Q., & Gao, L. (2024). Recent Advances in 3D Gaussian Splatting. *Computational Visual Media*, 10(4), 613–642. <https://doi.org/10.1007/s41095-024-0436-y>

- Yan, L., Song, J., Xie, H., Wei, P., Li, G., Zhu, L., Fan, Z., & Gong, S. (2025). LiDGS: An efficient 3D Reconstruction Framework Integrating Lidar Point Clouds and Multi-view Images for Enhanced Geometric Fidelity. *International Journal of Applied Earth Observation and Geoinformation*, 142, 104730. <https://doi.org/10.1016/j.jag.2025.104730>
- Yan, Z., Yin, M., Shao, Y., & Remondino, F. (2025). Evaluating 3D Gaussian Splatting for Urban Scene Reconstruction. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 251–258. <https://doi.org/10.5194/isprs-archives-XLVIII-1-W6-2025-251-2025>
- Yang, L., & Worboys, M. (2015). Generation of Navigation Graphs for Indoor Space. *International Journal of Geographical Information Science*, 29(10), 1737–1756. <https://doi.org/10.1080/13658816.2015.1041141>
- Yang, S., Im, C., Lee, J. W., & Choi, J. B. (2026). *TIDI-GS: Floater Suppression in 3D Gaussian Splatting for Enhanced Indoor Scene Fidelity*. <https://arxiv.org/abs/2601.09291>
- Yang, X., & Jiang, G. (2021). A Practical 3D Reconstruction Method for Weak Texture Scenes. *Remote Sensing*, 13(16). <https://doi.org/10.3390/rs13163103>
- Yu, Y., Verbree, E., Oosterom, P. van, & Pottgiesser, U. (2025). 3D Gaussian Splatting for Modern Architectural Heritage: Integrating UAV-Based Data Acquisition and Advanced Photorealistic 3D Techniques. *AGILE: Giscience Series*, 6, 51. <https://doi.org/10.5194/agile-giss-6-51-2025>
- Yuan, Z., Zhang, T., Johnson-Roberson, M., & Zhi, W. (2024). *PhotoReg: Photometrically Registering 3D Gaussian Splatting Models*. <https://arxiv.org/abs/2410.05044>
- Yunisya, A. N., Verbree, E., & Oosterom, P. van. (2025). Exploring the Potential of Gaussian Splatting Environment for Indoor Wayfinding Simulation. *28th AGILE Conference on Geographic Information Science*. <https://www.gdmc.nl/publications/2025/GS4WayFinding.pdf>
- Zhou, Z., Weibel, R., Richter, K.-F., & Huang, H. (2022). HiVG: A Hierarchical Indoor Visibility-based Graph for Navigation Guidance in Multi-storey Buildings. *Computers, Environment and Urban Systems*, 93, 101751. <https://doi.org/10.1016/j.compenvurbsys.2021.101751>
- Zhu, J., Wong, M. O., Nisbet, N., Xu, J., Kelly, T., Zlatanova, S., & Brilakis, I. (2025). Semantics-based Connectivity Graph for Indoor Pathfinding Powered by IFC-Graph. *Automation in Construction*, 171, 106019. <https://doi.org/10.1016/j.autcon.2025.106019>
- Zoomers, B., Wijnants, M., Molenaers, I., Vanherck, J., Put, J., Jorissen, L., & Michiels, N. (2025). PRoGS: Progressive Rendering of Gaussian Splats. *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 3118–3127. <https://doi.org/10.1109/wacv61041.2025.00308>

