A Framework for Integrating and Managing Heterogeneous 3D Geospatial Data in Urban Digital Twins

Jeson Lonappan[1], Olga Shkedova[1], Udo Feuerhake[1], Monika Sester[1]

[1] Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, Germany - (jeson.lonappan, olga.shkedova, udo.feuerhake, monika.sester)@ikg.uni-hannover.de

Keywords: Urban Digital Twins, 3D Data Integration, Voxelization, Spatial Database, CityGML, Mobile Laser Scanning

Abstract

Urban Digital Twins (UDT) require systematic integration of heterogeneous 3D geospatial data sources, but existing integration methods struggle with semantic information loss during fusion, geometric precision degradation through format conversions, and limited storage scalability. This paper presents a modular, database-centric framework achieving bidirectional semantic enrichment through semantically enriched voxelization. The framework integrates CityGML building models, Mobile Mapping System (MMS) point clouds, and Digital Terrain Models (DTM) using PostgreSQL/PostGIS database system with pgPointCloud, the point cloud extension of Postgres for patch-based storage. A two-stage refinement pipeline is applied to align MMS point clouds to CityGML wall surfaces using Random Sample Consensus (RANSAC) and Iterative Closest Point (ICP) algorithms. To integrate the terrain, Constrained Delaunay Triangulation (CDT) algorithm is applied with building footprints as constraints. All datasets are independently voxelized at a common configurable resolution, with voxels enriched via custom pgPointCloud schemas storing multi-source attributes. A unified voxel table merges layers using priority-based conflict resolution. The framework is evaluated in terms of computational performance, registration precision, and storage efficiency, demonstrating feasibility and correctness of the integration pipeline on a representative urban test case. This paper presents a proof-of-concept evaluated on a small urban area in Hannover, Germany, demonstrating the framework’s potential for further development.

1. Introduction

UDTs have emerged as essential platforms for smart city applications, enabling simulation-based decision-making, infrastructure management, and urban planning (Batty, 2018). Their effectiveness fundamentally depends on systematic integration of heterogeneous 3D geospatial data sources, including semantic building models, high-resolution point clouds, and terrain representations (Biljecki et al., 2015).

Traditional data integration methods in the geospatial domain exhibit several critical limitations. During fusion processes, format conversions and geometric transformations often fail to preserve attribution data, leading to severe semantic information loss. Simultaneously, repeated coordinate transformations degrade overall geometric precision. Furthermore, storage inefficiency becomes prohibitive with city-scale datasets, particularly for volumetric representations that potentially generate billions of discrete elements. Finally, existing integration pipelines typically perform one-way data transformations rather than exploiting the complementary strengths of heterogeneous sources through mutual enrichment.

While standard CityGML LoD2 models provide essential volumetric and roof topology information, they inherently lack high-frequency façade-level details such as windows, doors, and architectural façade indentations. To bridge this gap, Mobile Mapping Systems (MMS) have emerged as a critical technology for UDTs. Due to their capacity for rapid, large-scale data acquisition, MMS point clouds are invaluable for the continuous temporal updating of dynamic urban environments (Brenner, 2016). By fusing dense MMS data with existing generalized building models, flat LoD2 façades can be systematically enriched with rich geometric and RGB attributes, effectively paving the way for automated updates toward LoD3 representations. However, achieving this requires a framework capable of handling the massive scale and distinct data structures of both formats.

Voxels offer a promising common spatial representation for integrating these heterogeneous 3D models. However, realizing their full potential requires moving beyond purely geometric translations to incorporate robust semantic enrichment. While the conceptual need for semantically enriched voxels is clear, practical implementations are heavily bottlenecked; managing massive, multi-source volumetric grids generates prohibitively large datasets, severely limiting storage efficiency and database scalability.

To address these challenges, this paper presents a database-centric voxelization framework that utilizes pgPointCloud[1] (Ramsey et al., 2021) dimensional compression to ensure scalable voxel storage within UDTs. While our previous study briefly introduced a possibility of multi-source data integration based on the PostgreSQL/PostGIS spatial database (Shkedova et al., 2026), the current research provides a detailed description of the heterogeneous data integration pipeline, with the primary novelty lying in the application of pgPointCloud patch storage to voxel datasets. Whereas pgPointCloud traditionally manages LiDAR point clouds, our work demonstrates that voxels exhibit similar spatial locality and compression characteristics, making 600-voxel patches an effective storage unit that enables native PostgreSQL integration. By implementing a modular pipeline for bidirectional semantic enrichment, the approach integrates heterogeneous data sources including CityGML, MMS point clouds, and DTMs through voxel-based processing. Furthermore, priority-based integration with provenance tracking offers a systematic approach to multi-source conflict resolution:

[1] https://github.com/pgpointcloud/pointcloud

unlike binary "first-wins" or "last-wins" strategies, accuracy-based prioritization preserves the data quality hierarchy while retaining multi-source metadata for validation. Ultimately, this allows buildings to acquire geometric detail and RGB attributes while simultaneously endowing point clouds with semantic classifications.

This paper is structured as follows: Section 2 reviews related work on 3D city model integration, voxel representations, and semantic enrichment. Section 3 details the framework architecture, refinement pipelines, voxelization algorithms, and pgPointCloud storage design. Section 4 presents implementation details, quantitative validation results, and performance benchmarks. Section 5 discusses contributions, limitations, and practical implications. Section 6 concludes the paper.

2. Related Work

CityGML has emerged as the international standard for semantic 3D city models, providing hierarchical LoD specifications (Gröger et al., 2012). The 3DCityDB database schema enables efficient storage and management of CityGML datasets (Yao et al., 2018). The recent release of 3DCityDB version 5 introduces simplified geometric mapping that significantly improves its functionality (Yao et al., 2025).

Systematic reviews identify persistent obstacles in multi-source 3D city model integration (Syed Abdul Rahman et al., 2024). A comprehensive analysis of UDT data integration across lifecycle phases reveals that heterogeneous data fusion remains a critical bottleneck, with semantic interoperability and geometric consistency identified as primary challenges (Jeddoub et al., 2025). Existing approaches typically sacrifice either semantic information for geometric precision or vice versa. Format-based integration methods (e.g., conversion to IFC, OBJ) incur semantic loss, while database-centric approaches often lack efficient storage mechanisms for massive point-based representations (Mazzetto, 2024).

Voxelization provides a unified discrete space representation suitable for multi-source integration. Conventional approaches focus on geometric conversion: mesh voxelization (Schwarz and Seidel, 2010), surface sampling (Laine, 2013), or ray-casting methods (Haumont et al., 2003). However, these methods treat voxels as purely geometric primitives without semantic enrichment. Recent work introduces semantically enriched voxels combining geometric discretization with attribute preservation (Heeramaglore and Kolbe, 2022). This concept enables voxels to carry CityGML semantics, material properties, and multi-source provenance. A recent application of this concept is VoxCity (Fujiwara et al., 2025), which generates grid-based semantic 3D city models from open data for seamless urban environment simulation.

Registration of MMS point clouds to semantic models typically involves a robust pipeline that combines RANSAC for coarse alignment (Fischler and Bolles, 1981) with ICP for local refinement (Besl and McKay, 1992). Despite recent advances in deep learning-based registration (Zhang et al., 2020), geometric constraints remain important for obtaining accurate alignment at the building scale. Recently, the focus has shifted from uni-directional mapping to bidirectional enrichment workflows. The CityGML 3.0 standard (Kutzner et al., 2020) now natively supports mutual attribute inheritance via its PointCloud module (Beil et al., 2021), and modern pipelines have demonstrated reciprocal updating using semantic models to classify raw points while simultaneously utilizing point cloud geometry to refine LoD models (Zhu et al., 2024). However, these advances largely exist as discrete workflows; systematic UDT frameworks that manage this bidirectional flow, specifically using voxel-based architectures as a unifying medium remain absent.

Efficient storage of massive volumetric data is critical for performance. pgPointCloud extends PostgreSQL/PostGIS with patch-based storage, clustering nearby points into compressed binary blobs ("patches") to optimize I/O operations. Research has demonstrated its utility for managing LiDAR datasets, showing that patch-based organization significantly outperforms flat-table storage for spatial queries (Meyer and Brunn, 2019). Within the context of semantic point clouds, enhanced patch schemas have been introduced to more effectively manage complex attribute data (Poux et al., 2017). Similarly, recent autonomous driving mapping frameworks have utilized pgPointCloud to manage environmental measurements, though they note efficiency bottlenecks when data scales extremely high without aggressive tiling or snapshotting strategies (Andert et al., 2025). Despite its provenance in LiDAR management, the application of pgPointCloud's patch compression specifically for voxelized urban data remains under-investigated.

3. Framework

3.1 Architecture and Database Design

Figure 1 illustrates the overall system design comprising four major components: data sources, refinement pipelines, voxelization modules, and integration layer.

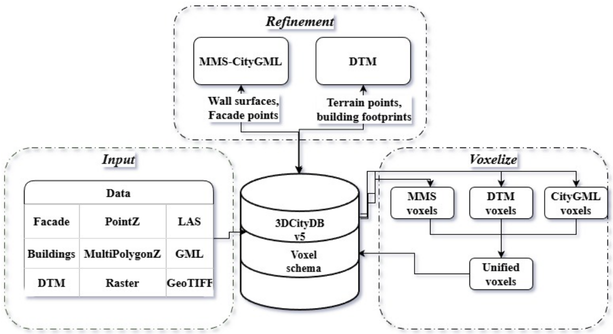

Figure 1. Integration framework showing data input, refinement pipelines, database architecture, and voxelization-integration workflow.

The framework employs PostgreSQL/PostGIS for spatial operations, 3DCityDB v5 for CityGML management, and pgPointCloud for voxel storage. This database-centric design minimizes intermediate file proliferation and enables efficient spatial queries on integrated voxel layers. Three custom PointCloud schemas (PCID) define layer-specific attribute storage: buildings (X, Y, Z, BuildingID), MMS (X, Y, Z, Classification, R, G, B), and terrain (X, Y, Z, TerrainType). All schemas use SRID=25832 (UTM Zone 32N) with dimensional compression. Voxels are stored as `PcPATCH` columns with GIST spatial indexing on `PC_EnvelopeGeometry` for bounding-box queries.

3.2 Data Sources and Preparation

For evaluating the methodology, datasets of a small area of Linden-Nord neighborhood in Hannover, Germany are selected. The selected data is visualized in Figure 2. Semantic building models and terrain models are obtained from official OpenGeoData[2] portal of Lower Saxony, Germany and stored in 3DCityDB. Terrain data has a 1m grid spacing, which is sampled from aerial surveying point clouds with a geometric resolution of at least 4 points per m^2. The data were initially stored in TIFF format and subsequently converted to a Triangulated Irregular Network (TIN) in CityGML format.

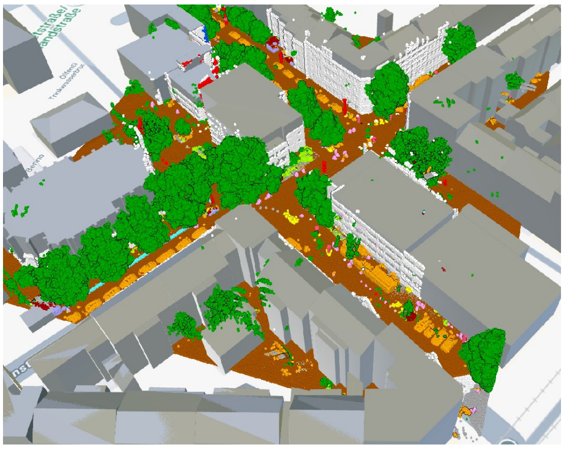

Figure 2. Input data containing terrain, buildings and point clouds with label colors.

The CityGML LoD2 building models provide essential semantic and macroscopic structural boundaries. However, as is standard for LoD2, they inherently lack detailed façade structures (such as window reveals, doors, and balconies). These models are generalized representations, typically generated by extruding 2D cadastral footprints to align with standardized roof geometries. Due to this automated generation process, the models frequently exhibit topological anomalies and geometric errors, such as non-watertight polygon seams, unclosed volumes, and slightly intersecting surfaces. Consequently, the LoD2 model’s positional precision reflects that of the source building footprints, while its absolute vertical accuracy is roughly 1 meter.

In contrast, the point clouds, acquired via a Riegl VMX-250 MMS, provide highly detailed as-built geometries. While the local relative accuracy of these point clouds is at the centimetre level, GNSS constraints in this dense urban test area result in an absolute positional accuracy of 20 to 30 centimetres. The point cloud data contain XYZ coordinates, RGB colors derived from cameras, and point-wise semantic classifications encompassing various urban features, including buildings, roads, fences, pedestrians, vehicles, tree canopies, and tree trunks.

3.3 Refinement Pipelines

The integration of these highly heterogeneous urban data sources into a unified voxel space requires strict spatial and topological congruence. Because these datasets are acquired via different sensors and methodologies, they inherently possess conflicting geometric accuracies, resolutions, and structural representations. Therefore, dedicated harmonization pipelines are required to bridge the gap between as-built point clouds and generalized 3D city models.

If fused directly into a rigid voxel grid without prior harmonization, these inherent discrepancies create severe spatial conflicts. For instance, MMS data often suffers from local positioning drift due to GNSS multipath effects in urban canyons, whereas CityGML models exhibit global positioning offsets and simplified planar boundaries. Misalignment between these two sources forces the voxelizer to arbitrarily assign overlapping semantic labels or create duplicate walls. Furthermore, geometrically matching planar building base polygons to continuously sloped 2.5D terrain surfaces often causes buildings to either partially sink into the ground or float above it (Yan et al., 2019). Sinking geometries cause terrain to bleed into ground floors, while floating foundations create unclassified voxel gaps. To prevent these voxel-level semantic ambiguities and establish an accurate spatial correspondence, the following robust pre-alignment and topological refinement pipelines are executed prior to voxelization.

MMS-CityGML Alignment: To establish spatial correspondence, only building-classified MMS points are loaded, and CityGML walls are spatially filtered to those intersecting the MMS survey extent. To extract the point cloud corresponding to each specific façade, a volumetric buffer of 2.0 m is applied around the CityGML wall planes. This 2.0 m search distance is geometrically necessary to absorb the typical global positioning offsets inherent in generalized CityGML LoD2 models, ensuring the true as-built MMS façade falls within the extraction zone.

However, because street-level MMS data is highly susceptible to occlusions from different urban features, simply clipping the data does not guarantee a viable surface for registration. Therefore, each candidate wall undergoes two strict validation stages. First, a *density gate* requires at least 2,000 points within the buffer. While MMS point clouds are inherently dense, this threshold acts as a critical minimum lower-bound to filter out heavily occluded walls or false-positive planar structures, ensuring a statistically significant sample size for reliable plane fitting. Second, a *uniformity gate* requires at least 70% of 20 equal horizontal wall segments to contain points. Point density alone is insufficient for registration; if thousands of points are clustered in a single visible corner of a heavily occluded wall, the subsequent alignment quality will degrade, causing the geometry to hinge and rotate incorrectly. The uniformity gate ensures the retained points are spatially distributed across the entire façade, providing a geometrically stable baseline for parallel alignment.

Accepted points are clipped to each wall via the buffer and vertically aligned using the building’s reference ground elevation from CityGML. This multi-stage selection ensures that only densely sampled, geometrically consistent façade segments enter registration, excluding heavily occluded surfaces, other urban objects, and classification errors.

RANSAC segments each wall’s clipped MMS point cloud by fitting the dominant plane $\boldsymbol{\pi} : ax + by + cz + d = 0$, classifying points as inliers if their perpendicular distance falls below the threshold $\tau = 0.1$ m as in Equation (1):

$$\mathcal{I} = \left\{ \mathbf{p}_i \;\middle|\; \frac{|a\,x_i + b\,y_i + c\,z_i + d|}{\sqrt{a^2 + b^2 + c^2}} < \tau \right\}, \tag{1}$$

[2] https://lgln-geodaten.niedersachsen.de/

where $\mathbf{p}_i = (x_i, y_i, z_i)$ are MMS point coordinates. The threshold of 0.1 m is geometrically motivated by standard architectural dimensions and the necessity to prevent registration bias. CityGML LoD2 representations geometrically simplify façades as perfectly flat polygons. Therefore, the MMS data must be strictly filtered to isolate the true coplanar wall surface. The 10 cm RANSAC distance threshold is an established parameter for façade plane extraction (Xu and Stilla, 2019). It is sufficiently large to absorb typical MMS sensor noise and physical surface roughness (e.g., masonry, cladding) while systematically excluding non-structural architectural elements. As established in knowledge-based façade reconstruction, features such as window reveals, doors, and drainage pipes typically recess or protrude by more than 15 cm from the primary structural plane (Pu and Vosselman, 2009). Consequently, the RANSAC-fitted plane reliably separates the dominant façade surface (inliers) from these features (outliers), preventing recessed or protruding geometries from biasing the subsequent ICP alignment. ICP then registers the inlier point cloud to a densely sampled representation of the corresponding CityGML wall surface (at 0.01 m resolution) by minimising Equation (2):

$$E_{\text{ICP}} = \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \text{closest}(\mathbf{p}_i, \mathcal{S})\|^2, \tag{2}$$

where $\mathcal{S}$ is the set of points sampled on the CityGML wall surface and $\text{closest}(\mathbf{p}_i, \mathcal{S})$ returns the nearest point in $\mathcal{S}$. Walls failing the quality gates are excluded. Semantic attributes (building IDs, surface types) are transferred to aligned point clouds while clipping to achieve bidirectional enrichment.

DTM Refinement: To topologically harmonize the terrain with the building models prior to voxelization, CDT is applied (Shewchuk, 1996). CDT incorporates building footprints as hard constraints, forcing triangle edges to respect building boundaries. This eliminates terrain interpolation artifacts within building footprints, resolving the ”floating and sinking” conflicts, and ensures mutually exclusive geometric consistency between the terrain and building models at the foundation level.

3.4 Voxelization

After the refinement pipelines, the different input data are spatially consistent for voxelization. All datasets are independently voxelized at a configurable resolution r within a common 3D grid defined by the study area extent. Each source is discretized into the same voxel space, enabling consistent spatial overlay during integration. We evaluate two resolutions: $r = 0.10$ m and $r = 0.05$ m.

Building layer: To account for geometric errors and topological gaps inherent in CityGML boundary surfaces, a robust voxelization strategy is applied to each building individually. Rather than relying purely on direct surface rasterization, which can leave holes at non-watertight polygon seams, we employ an intermediate solid-to-shell approach based on volumetric repair techniques (Nooruddin and Turk, 2003), which have proven effective for resolving misclosures in defective 3D city models (Steuer et al., 2015).

First, CityGML boundary surfaces are voxelized through multi-orientation ray casting, utilizing established 3D scan-conversion principles (Kaufman, 1987). For each surface, the optimal projection plane is selected based on the surface normal orientation. Ray casting determines interior points by applying an even-odd Point-in-Polygon test, and the third coordinate is interpolated from the surface plane equation. Next, vertical column filling generates temporary solid building interiors; this sweep-line approach ensures gap-free solid volumes and has been successfully adapted for the topological voxelization of geospatial data (Nourian et al., 2016, Huang et al., 1998).

This temporary solid volume is then hollowed out via a morphological shell extraction filter as in Equation (3), which retains only those voxels that possess at least one empty face-adjacent neighbour:

$$\text{shell}(v) = \text{solid}(v) \wedge \exists n \in N_6(v) : \neg\text{solid}(n), \tag{3}$$

where $\text{solid}(v)$ is a boolean predicate indicating whether voxel v lies inside the building volume, and $N_6(v)$ denotes its six face-adjacent neighbours. Checking the 6-connected neighbourhood for empty space guarantees the extraction of a topologically closed, watertight 26-connected boundary. This process produces hollow architectural shells, achieving a 40–60% storage reduction compared to storing the full solid volume. Finally, each shell voxel inherits the semantic building identifiers from its parent CityGML object.

MMS layer: MMS point clouds are mapped into the same voxel grid. For each point with coordinates p_i along axis $i \in \{x, y, z\}$, the integer voxel index v_i is computed via Equation (4):

$$v_i = \lfloor (p_i - p_{i,\min})/r \rfloor, \quad i \in \{x, y, z\}, \tag{4}$$

where $p_{i,\min}$ is the minimum coordinate of the study area extent along axis i and r is the voxel resolution as defined in Section 3.4. Multiple points falling within the same voxel are aggregated, RGB color values are averaged and classification codes are assigned by majority voting. This achieves significant data compression while preserving photogrammetric color fidelity.

Terrain layer: The refined DTM TIN is voxelized by sampling terrain elevation at each voxel center. For each grid position (x, y), the enclosing triangle is identified via spatial indexing, and elevation z is interpolated using barycentric coordinates. Each terrain voxel is assigned a TerrainType attribute distinguishing surface from optional subsurface extension to achieve a volumetric layer.

3.5 Storage and Integration

Voxels from each layer are grouped into batches of 600 elements and stored as `PcPATCH` columns using `pgPointCloud`'s dimensional compression. The patch size of 600 is deliberately chosen to optimize database I/O and spatial indexing performance. Because PostgreSQL allocates data in rigid 8 KB memory pages, exceeding this size forces the database to use inefficient out-of-line storage, which severely degrades data retrieval speeds and spatial index efficiency. Empirical evaluations of point cloud database storage confirm that, when utilizing dimensional compression, a capacity of 400 to 600 elements per patch maximizes the payload within a single 8 KB page without triggering these performance penalties (Meyer and Brunn, 2019).

A table to store the unified voxel scene's metadata records layer name, voxel resolution, spatial extent, coordinate offset, and processing provenance for each layer, enabling cross-layer queries and reproducibility tracking. The unified voxel table merges the three independent layers using a deterministic, priority-based conflict resolution strategy in which MMS takes the highest priority, followed by buildings, then terrain. This hierarchical heuristic is based on domain logic and the intrinsic geometric accuracy of the input sensors. MMS point clouds provide the highest real-world geometric fidelity at centimetre-level accuracy and represent as-built conditions, thereby rightfully superseding generalised CityGML geometries where spatial overlaps occur. CityGML buildings provide critical semantic structure and take precedence over the DTM. Furthermore, because the terrain and building footprints are already topologically harmonised during the CDT refinement stage, structural conflicts at the building–terrain intersection are inherently minimised prior to voxelization. Because MMS point clouds may capture ephemeral objects such as vehicles or pedestrians, only building-classified points are retained prior to voxelization, ensuring that non-building artifacts do not propagate into the unified model. For voxels at identical (x, y, z) coordinates, the higher-priority source overwrites core attributes while retaining multi-source provenance metadata for quality assessment.

4. Results and Validation

The experiments were done on a system with AMD Ryzen APU and 32GB unified RAM. The software stack includes PostgreSQL 14.5, PostGIS 3.3, pgPointCloud 1.2, Python 3.12.

4.1 Voxelization Statistics

Table 1 presents the layer-wise voxelization results at both resolutions. Building voxelization produces the largest layer due to the dominance of building surfaces in the study area. MMS voxelization compresses 8.5 M input points into voxels through spatial aggregation. Terrain voxelization generates voxels from TIN triangles via surface interpolation. Increasing the resolution from 0.10 m to 0.05 m increases the total voxel count by a factor of 5, consistent with the expected surface-area scaling for shell-based building representations.

Table 1. Layer-wise voxelization statistics at 0.10 m and 0.05 m resolution.

	0.10 m		0.05 m	
Layer	**Voxels (M)**	**Patches**	**Voxels (M)**	**Patches**
Buildings	4.18	6,969	24.50	40,837
MMS	1.18	1,962	3.53	5,883
Terrain	1.54	2,572	6.17	10,279
Total Input	**6.90**	**11,503**	**34.20**	**56,999**
Unified	**3.02**	**5,027**	**16.40**	**27,337**

4.2 Registration Fitting Residuals

The MMS–CityGML alignment pipeline processes 113 wall surfaces across 36 buildings. Because the evaluation compares aligned MMS points against the simplified planar LoD2 surfaces used as registration targets, the reported metrics represent fitting residuals rather than absolute positional accuracy. The alignment achieves a median residual of 0.05 m, with per-wall residuals ranging from 0.016 m on uniform planar façades to a maximum of 0.328 m on highly complex exterior surfaces (as shown in Figure 3).

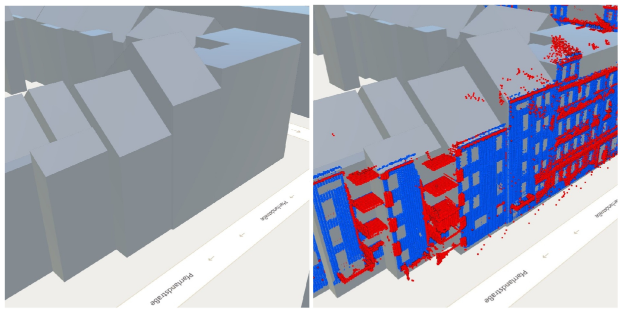

Figure 3. Alignment of MMS point clouds to generalized CityGML LoD2 models. (Left) The native LoD2 geometry assuming perfectly flat planar façades. (Right) The overlaid MMS points following RANSAC-based plane extraction. Blue points represent inliers used for ICP registration, while red points indicate outliers, successfully capturing pronounced architectural features such as balconies and window recesses absent in the generalized model.

Of the 113 aligned surfaces, 93.8% achieve residuals below 0.10 m, confirming reliable ICP convergence on well-sampled walls.

4.3 Storage and Performance

Table 2. Database storage and processing time by layer at 0.10 m and 0.05 m resolution.

	0.10 m			0.05 m	
Layer	**Vox (M)**	**MB**	**Time (s)**	**Vox (M)**	**MB**
Buildings	4.18	5.0	265.7	24.50	41.2
MMS	1.18	11.1	148.5	3.53	16.8
Terrain	1.54	2.2	70.0	6.17	7.1
Unified	3.02	10.6	281.5	16.40	36.9
Total	–	**28.9**	**765.7**	–	**102.0**

Table 2 reports per-layer storage footprint and processing time. At 0.10 m resolution, the complete pipeline executes in 765.7 s, with building voxelization as the most time-consuming stage (265.7 s) due to shell extraction complexity, followed by integration (281.5 s) which loads, merges, and re-inserts all layers. Database insertion dominates all stages, accounting for 84% of building voxelization time. pgPointCloud patches achieve compact storage through dimensional compression, totalling 28.9 MB at 0.10 m and 102.0 MB at 0.05 m for the complete scene. Increasing the resolution to 0.05 m increases the total voxel count by approximately $5\times$ but storage grows only $3.5\times$, demonstrating sub-linear storage scaling due to improved patch compression at higher density. Processing time scales approximately linearly with voxel count, with the 0.05 m pipeline completing in approximately $4\times$ the 0.10 m time.

4.4 Integration Results

Unified integration merges input voxels from all three layers into a deduplicated output.

In semantic enrichment, building voxels acquire RGB colors and façade features from MMS overlap regions, while MMS voxels retain CityGML building identifiers for semantic classification. As visualized in Figure 4, this priority-based integration yields a highly detailed UDT, where the flat, generalized façades of native LoD2 models are successfully replaced with as-built voxel geometry, while maintaining the structural integrity of the CityGML roofs and the continuous DTM base.

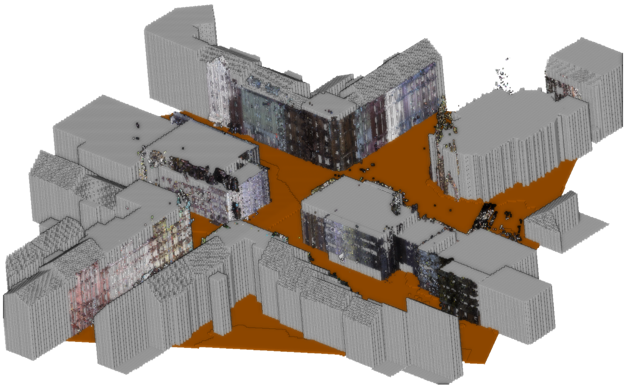

Figure 4. Visualization of the unified, semantically enriched voxel model, with façade geometry and RGB colors.

Multi-source voxels occur primarily at building-terrain interfaces and MMS-building overlaps, enabling quality assessment through geometric consistency checks. To quantify the extent of multi-source data fusion, spatial intersections between the independent voxel layers were analyzed. Out of the 3.02 million unified voxels, a total of 148,748 voxels contain overlapping data from multiple sources as visualized in Figure 5.

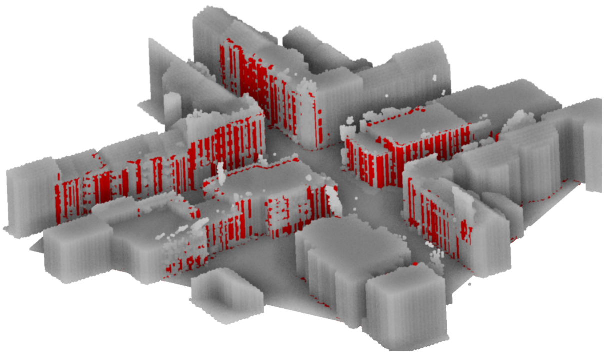

Figure 5. Spatial distribution of voxel conflicts within the unified 3D grid. Gray voxels represent single-source data, while red voxels indicate conflicts. Overlaps are heavily concentrated on building façades and at structural foundations.

The vast majority of these conflicts occur between the CityGML and MMS layers. As illustrated by the red voxel clusters in Figure 5, these overlaps are systematically distributed across the façade surfaces where dense, as-built point clouds intersect the generalized CityGML LoD2 planar shells. The intersection between Building and Terrain layers accounts for 27.8%, strictly localized at the building foundation lines. In all overlapping cases, the multi-source provenance metadata was successfully recorded in the pgPointCloud schema prior to priority-based integration.

4.5 Spatial-Semantic Query Performance

To validate the practical utility of the database-centric framework, two spatial-semantic queries were evaluated natively within database on the unified voxel table.

- **Q1: Neighbourhood Analysis:** Extracts all unified voxels (building, MMS, terrain) within a 50 m radius of a target building centroid, producing a multi-layer volumetric context export. The spatial index on `PC_EnvelopeGeometry` pre-filters 600-voxel patches, avoiding exhaustive point-level distance calculations.
- **Q2: Façade Protrusion Detection:** Identifies MMS voxels that lie outside the CityGML building footprint (convex hull) yet within 2.0 m of its boundary, isolating architectural protrusions such as balconies, bay windows, and entrance canopies that are absent from the generalised LoD2 model. This cross-layer query combines semantic filtering (MMS source) with geometric containment tests.

Table 3 summarizes measured execution times. The results confirm that the patch-based architecture supports analytical queries on millions of voxels directly within the database, without requiring data export to external tools.

Table 3. Spatial-semantic query performance on unified voxels.

Query	Time (s)	Voxels
Q1: Neighbourhood	7.3	882,670
Q2: Protrusion	3.6	5,059

5. Discussion

The database-centric framework successfully addresses the challenges identified in Section 1. pgPointCloud dimensional compression retains all attribute fields without conversion loss, preserving semantic information throughout the pipeline. Direct voxelization from native formats eliminates intermediate transformations that would degrade geometric precision. Patch-based storage is architecturally suited for larger deployments, as demonstrated by the compact footprint observed in this prototype, while priority-based integration achieves mutual benefit across heterogeneous sources through bidirectional enrichment.

The quantitative overlap analysis reveals that while only 5% of the unified voxel space contains multi-source conflicts, these intersections are highly concentrated at critical semantic boundaries. The overwhelming dominance of Building-MMS overlaps underscores the necessity and effectiveness of the proposed priority-based heuristic. By strictly enforcing MMS precedence at these locations, the framework systematically overwrites generalized LoD2 geometry with high-fidelity voxels, achieving the desired geometric and RGB enrichment. Furthermore, the Building-Terrain overlaps validate the pre-processing CDT refinement stage. Because the DTM was constrained by building footprints prior to voxelization, terrain interpolation artifacts inside buildings were eliminated. Consequently, the remaining voxel overlaps at the 0.1 m resolution represent legitimate topological boundaries, which are cleanly resolved by prioritizing structural building voxels over generalized terrain.

Several limitations warrant consideration. The resolution parameter controls the trade-off between geometric detail and computational cost: moving from 0.10 m to 0.05 m increases voxel count by 5× and processing time by 4.3×, while storage grows only 3.5× due to improved compression. Finer resolutions would further increase computational cost, whereas coarser resolutions would lose architectural detail captured by the MMS. The same trade-off causes the voxel “staircase effect,” where limited grid resolution makes smooth or slanted surfaces appear as stepped, block-like boundaries as visible in Figure 4.

Furthermore, the alignment metrics reported in Section 4.3 require careful interpretation. Because RANSAC-ICP aligns MMS points to generalized LoD2 wall surfaces, the algorithm naturally fits to the dominant wall plane. However, real-world façades contain non-planar sub-structures such as balconies, ledges, and deep window recesses that introduce systematic residuals. Consequently, higher error values reflect alignment to a simplified reference rather than a lack of true positional accuracy. On façades with sparse point coverage or pronounced geometric detail, this planar assumption becomes less representative, increasing overall alignment uncertainty. Quantifying this effect, for example through per-façade residual distributions or separate evaluation against LoD3 reference models, remains a direction for future work.

While the current implementation explicitly filters MMS data to building classes, the priority-based integration schema is readily extensible to non-building semantic classes (e.g., vegetation, street furniture, and ground). In future, these unconstrained MMS classes can be seamlessly integrated by inserting them into the conflict resolution hierarchy below CityGML buildings but above the generalized DTM. This rule-based extension ensures that classified MMS ground points can naturally refine generalized terrain geometries, while preventing non-building objects from overwriting structural building models.

6. Conclusions

This paper presented a modular, database-centric framework for integrating heterogeneous 3D geospatial data in UDTs through semantically enriched voxelization. The framework achieves bidirectional enrichment across CityGML, MMS, and DTM sources using pgPointCloud patch-based storage, demonstrating real-world applicability on a dataset from Hannover, Germany. At 0.10 m resolution, the unified model comprises 3.0M voxels in 28.9 MB; at 0.05 m, it scales to 16.4M voxels in 102 MB with sub-linear storage growth. By addressing systematic gaps in 3D city model integration identified in recent reviews, the framework provides a replicable, modular workflow for future city-scale deployment including energy simulation, flood modeling, and temporal change detection.

References

Andert, F., Böttcher, O., Mushyam, A., Schmälzle, P., 2025. Semantic lidar point cloud mapping and cloud-based slam for autonomous driving. *2025 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, IEEE, 853–860.

Batty, M., 2018. Digital twins. *Environment and planning B: Urban analytics and City Science*, 45(5), 817–820.

Beil, C., Kutzner, T., Schwab, B., Willenborg, B., Gawronski, A., Kolbe, T. H., 2021. Integration of 3D point clouds with semantic 3D city models–providing semantic information beyond classification. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 8, 105–112.

Besl, P. J., McKay, N. D., 1992. Method for registration of 3-d shapes. *Sensor fusion IV: control paradigms and data structures*, 1611, Spie, 586–606.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A., 2015. Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4), 2842–2889.

Brenner, C., 2016. Scalable estimation of precision maps in a mapreduce framework. *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 1–10.

Fischler, M. A., Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.

Fujiwara, K., Tsurumi, R., Kiyono, T., Fan, Z., Liang, X., Lei, B., Yap, W., Ito, K., Biljecki, F., 2025. VoxCity: A seamless framework for open geospatial data integration, grid-based semantic 3D city model generation, and urban environment simulation. *arXiv preprint arXiv:2504.13934*.

Gröger, G., Kolbe, T. H., Nagel, C., Häfele, K.-H., 2012. OGC city geography markup language (CityGML) encoding standard.

Haumont, D., Debeir, O., Sillion, F., 2003. Volumetric cell-and-portal generation. *Computer Graphics Forum*, 22number 3, Wiley Online Library, 303–312.

Heeramaglore, M., Kolbe, T. H., 2022. Semantically enriched voxels as a common representation for comparison and evaluation of 3D building models. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10, 89–96.

Huang, J., Yagel, R., Filippov, V., Kurzion, Y., 1998. An accurate method for voxelizing polygon meshes. *Proceedings of the 1998 IEEE symposium on Volume visualization*, 119–126.

Jeddoub, I., Nys, G.-A., Hajji, R., Billen, R., 2025. Data integration across urban digital twin lifecycle: a comprehensive review of current initiatives. *Annals of GIS*, 31(3), 367–386.

Kaufman, A., 1987. Efficient algorithms for 3D scan-conversion of parametric curves, surfaces, and volumes. *ACM Siggraph Computer Graphics*, 21(4), 171–179.

Kutzner, T., Chaturvedi, K., Kolbe, T. H., 2020. CityGML 3.0: New functions open up new applications. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 88(1), 43–61.

Laine, S., 2013. A topological approach to voxelization. *Computer Graphics Forum*, 32number 4, Wiley Online Library, 77–86.

Mazzetto, S., 2024. A review of urban digital twins integration, challenges, and future directions in smart city development. *Sustainability*, 16(19), 8337.

Meyer, T., Brunn, A., 2019. 3D Point Clouds in PostgreSQL/PostGIS for Applications in GIS and Geodesy. *GISTAM*, 1, 154–163.

Nooruddin, F. S., Turk, G., 2003. Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics*, 9(2), 191–205.

Nourian, P., Gonçalves, R., Zlatanova, S., Ohori, K. A., Vo, A. V., 2016. Voxelization algorithms for geospatial applications: Computational methods for voxelating spatial datasets of 3D city models containing 3D surface, curve and point data models. *MethodsX*, 3, 69–86.

Poux, F., Neuville, R., Hallot, P., Billen, R., 2017. Model for semantically rich point cloud data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4, 107–115.

Pu, S., Vosselman, G., 2009. Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(6), 575–584.

Ramsey, P., Blottiere, P., Brédif, M., Lemoine, E. et al., 2021. pgpointcloud-a postgresql extension for storing point cloud (lidar) data.

Schwarz, M., Seidel, H.-P., 2010. Fast parallel surface and solid voxelization on GPUs. *ACM transactions on graphics (TOG)*, 29(6), 1–10.

Shewchuk, J. R., 1996. Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. *Workshop on applied computational geometry*, Springer, 203–222.

Shkedova, O., Lonappan, J., Sester, M., 2026. ”colorful smoke” visual metaphor for temporal occupancy in urban digital twins. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-4/W19-2025, ISPRS, 127–134.

Steuer, H., Machl, T., Sindram, M., Liebel, L., Kolbe, T. H., 2015. Voluminator—approximating the volume of 3d buildings to overcome topological errors. *AGILE 2015: Geographic Information Science as an Enabler of Smarter Cities and Communities*, Springer, 343–362.

Syed Abdul Rahman, S. A. F., Abdul Maulud, K. N., Ujang, U., Wan Mohd Jaafar, W. S., Shaharuddin, S., Ab Rahman, A. A., 2024. The digital landscape of smart cities and digital twins: A systematic literature review of digital terrain and 3D city models in enhancing decision-making. *Sage Open*, 14(1), 21582440231220768.

Xu, Y., Stilla, U., 2019. Contour extraction of planar elements of building facades from point clouds using global graph-based clustering. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4, 211–219.

Yan, J., Zlatanova, S., Aleksandrov, M., Diakite, A., Pettit, C., 2019. Integration of 3D objects and terrain for 3D modelling supporting the digital twin. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4, 147–154.

Yao, Z., Nagel, C., Kendir, M., Willenborg, B., Kolbe, T. H., 2025. The New 3D City Database 5.0-Advancing 3D City Data Management based on CityGML 3.0. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10, 241–248.

Yao, Z., Nagel, C., Kunde, F., Hudra, G., Willkomm, P., Donaubauer, A., Adolphi, T., Kolbe, T. H., 2018. 3DCityDB-a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. *Open Geospatial Data, Software and Standards*, 3(1), 1–26.

Zhang, Z., Dai, Y., Sun, J., 2020. Deep learning based point cloud registration: an overview. *Virtual Reality & Intelligent Hardware*, 2(3), 222–246.

Zhu, J., Wysocki, O., Holst, C., Kolbe, T. H., 2024. Enriching thermal point clouds of buildings using semantic 3D building models. *arXiv preprint arXiv:2407.21436*.