

GEOMAPI: Processing close-range sensing data of construction scenes with semantic web technologies

Maarten Bassier ^{a,*}, Jelle Vermandere ^a, Sam De Geyter ^{a,b}, Heinder De Winter ^{a,c}

^a Department of Civil Engineering, TC Construction - Geomatics, KU Leuven - Faculty of Engineering Technology, Ghent, Belgium

^b MEET HET BV, Mariakerke, Belgium

^c DIRK BAUWENS NV, Evergem, Belgium



ARTICLE INFO

Keywords:

Geomatics
Semantic Web Technologies
Construction
Close-range sensing
BIM
Point clouds
Photogrammetry

ABSTRACT

The AEC industry struggles with integrating close-range sensing data like photogrammetric and LiDAR outputs due to the vast volume of unstructured 2D and 3D data, hindering tasks like digital twinning and construction monitoring. Although semantic web technologies have effectively managed diverse resources in other domains, their potential in handling construction-related sensing data remains unrealized, further complicated by the absence of a comprehensive data management framework for construction projects. This paper bridges the gap between various close-range sensing resources and AEC industry tasks by integrating Linked Data RDF Graphs with advanced geomatics algorithms. Experimental results demonstrate that this close-range linked data methodology effectively combines sensory data processing with RDF Graphs, thereby enabling more efficient resolution of construction process queries. The GEOMAPI framework significantly improves access to close-range sensing data for construction stakeholders, facilitates complex geomatics computations, and stores multi-temporal analysis results, thereby advancing the capabilities in the AEC sector.

1. Introduction

With the rise of increasingly faster and cheaper close-range sensing data acquisition tools, the demand for construction digitization is rapidly increasing in the Architecture, Engineering, and Construction (AEC) industry. Never before have construction assets been captured in such high detail and point density. Every year, numerous laser scanning and photogrammetric projects are initiated on construction sites and existing structures in infrastructure and buildings [1]. The amount of data captured has become a key bottleneck in the state of the art (SOTA). The data captured in a single documentation of an average two-story building of 30,000 m² with LiDAR technologies can easily amount to over 200 scans (>40 GB of point cloud data) or around 10,000 pictures in a photogrammetric pipeline, resulting in a point cloud of approximately 100 million points [2]. Current processing methods, including registration, analysis, and modeling routines, massively downsample and tile the data, causing a significant loss of information [3,4]. This is especially true for segmentation techniques using innovative deep learning, which are currently constrained to only a few tens of thousands of points per batch [5]. There is currently a massive gap between the very accessible close-range sensing documentation tools and the various needs of construction digitization processes [6].

A second key obstacle is the limited longevity of captured data and the specificity of the data modalities. Most approaches do not reuse past observations of facilities for their analyses. Methods proposed for construction modeling, project planning, and facility management typically require a completely new dataset of the asset, as they are bound to a single data modality and are not robust to changes in the scene [7]. This is particularly true for construction site methods that demand periodic data to assess progress and quality on-site [8]. Proposed methods struggle to handle time-series of close-range sensing observations coming from different sources, and their integration with Building Information Models (BIM) and CAD are inadequate [9,10]. Instead, these methods necessitate a complete redocumentation of the site and discard prior measurements. This significantly reduces the lifespan of close-range sensing data, which is extremely wasteful given that most of the asset documentation remains valid. Furthermore, this practice adversely affects the detection rate of analyses that could benefit from an accumulation of information in consecutive measurement epochs.

The goal of this work is to address both of the aforementioned issues using semantic web technologies. Specifically, we propose representing close-range sensing information as an RDF Graph for the multi-temporal management of construction observations. Our approach surpasses the current state of the art by standardizing geomatics operations for these uniform resources, enabling seamless multi-modal

* Corresponding author.

E-mail address: maarten.bassier@kuleuven.be (M. Bassier).

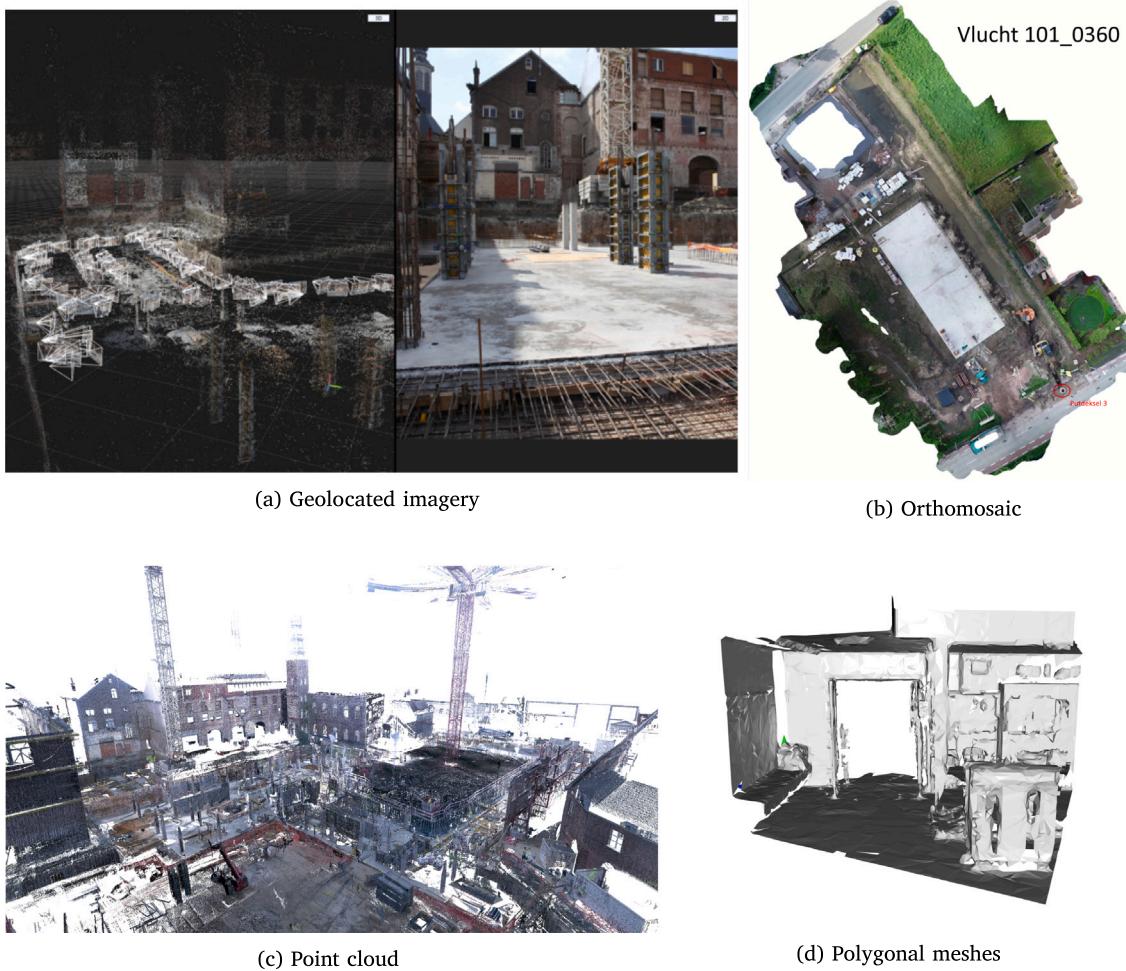


Fig. 1. Overview of the prominent close-range sensing modalities used within the construction domain.

geomatics analyses through Graph operations. In this endeavor, we not only represent geospatial resources as links to native 2D and 3D information, as is currently practiced [11], but we also extract the pertinent geospatial information and incorporate it directly into the Graph. The proposed methods are encapsulated in an Open Source API named GEOMAPI. This API defines the concepts for prominent construction geospatial resources, including BIM, CAD, point clouds, polygonal meshes, and geolocated image information (Fig. 1). In summary, the primary contributions of this work are:

1. Literature study on close-range sensing processing in the construction industry.
2. Framework to jointly process multi-temporal and multi-modal geospatial information as linked data resources.
3. Geomatics methodology to operate on RDF Graphs.
4. Empirical test cases on the use of RDF Graphs for close-range sensing in construction applications.

The remainder of this work is structured as follows.

1. **Background:** Review of related work on close-range sensing in construction applications and multi-modal analyses.
2. **Methodology:** Description of the GEOMAPI technology, explaining Linked Data and Geomatics concepts.
3. **Testcases:** Evaluation of the framework in three distinct construction tasks throughout a construction's life-cycle.
 - **Construction sites:** Progress estimation on conventional construction sites.

- **Dataset updating:** Facility documentation using periodic mappings.
- **Road Construction:** Monitoring volume changes on road construction.

4. **Contributions & Limitations:** Exploration of the advantages and shortcomings of the proposed technology.

5. **Conclusions:** Summary of findings and future work in geomatics and linked data concepts.

2. Background & Related work

In this section, the related work for the key aspects of this research are discussed: (1) the current processing of close-range sensing data in the construction industry, (2) state-of-the-art multi-source and multi-temporal processes (3) semantic web technology initiatives in the construction industry.

2.1. Close-range sensing in the construction industry

Construction tasks, including asset digitization, facility management, and construction monitoring, are increasingly being addressed through close-range sensing observations [12]. Depending on the type of asset and the specific application, a variety of LiDAR and photogrammetric close-range sensing techniques are proposed to provide reliable geometric and textural information [13,14] (Fig. 1). The preference for either approach varies depending on the domain. Photogrammetric approaches, employing handheld cameras, footage, or even tower crane

cameras, yield highly detailed textures. Conversely, LiDAR approaches, utilizing terrestrial laser scanners, mobile mappers, or stop-and-go systems, offer high metric accuracy [15].

For example, structural elements are typically analyzed based on accurate geometry to assess tolerances [16,17]. This is also true for structure-related elements in the architecture domain, such as walls, ceilings, beams, and columns [18]. In the rest of the architecture domain, a combination of both methods yields the highest detection rate for mid-sized objects like doors and ceilings [19]. For small-scale objects, photogrammetric approaches are preferred due to their high level of detail and texture documentation. In the HVAC and plumbing domains, which contain highly occluded objects, reliance on both methods is also seen. Pipe and installation dimensions and locations are best determined using LiDAR [20], but the increased coverage and detailing from photogrammetric techniques are invaluable for identifying different components [21]. For smaller objects, such as architectural ornaments or electrical equipment in the electrical domain, there is a clear preference for photogrammetry, where computer vision is essential for detecting these assets [22].

Overall, a plethora of works describe data acquisition metrics for construction applications [23,24]. However, current methods typically rely on a single data source or measurement epoch [25]. For instance, close-range sensing has been proposed as the baseline for tracking progress, quality, and stockpiles on-site compared to the as-designed BIM, its exchange format IFC, or CAD [26,27]. There are several promising approaches, ranging from UAV and body-cam videogrammetry to Simultaneous Localization and Mapping (SLAM) mobile laser scanning approaches [28,29]. Several works report accuracies of LOA20 [$\sigma \leq 0.05$ m] and even LOA30 [$\sigma \leq 0.015$ m] [30] for the evaluation of structural elements all the way to MEP upon installation [31]. However, no reports are found where consecutive datasets were integrated for joint analysis, nor where LiDAR and photogrammetric outputs were used together in such applications, despite the obvious benefits.

Similarly, the utilization of multiple data sources is not well understood, despite their clear advantages in enhancing detection rates across various construction applications. Fusion techniques are increasingly recognized in interpretation algorithms, but they have yet to be proposed for construction applications [32]. As a result, most close-range sensing data is used only once, typically by the party that acquires the data or conducts the analysis. This represents a significant waste of digital assets, particularly given the potential of close-range sensing data as one of the industry's most promising and versatile repositories. The primary challenge lies in the lack of tools to effectively manage close-range sensing data throughout the life-cycle of a construction project [13].

2.2. Multi-modal analyses

This work introduces a framework for managing close-range sensing data from multiple sources. Consequently, it is necessary to consider the types of multi-modal frameworks currently being explored. This area represents a highly innovative field within construction digitization, with origins in Earth Observation (EO). In recent years, research has focused on developing fusion techniques to comprehensively analyze and interpret strongly heterogeneous data. This includes the combination of multi-and hyperspectral data with InSAR, LiDAR, and RGB data [33].

The transition of fusion techniques from Earth Observation to close-range sensing is particularly noticeable in semantic segmentation procedures [34]. Until recently, semantic segmentation predominantly operated on either image-based (e.g., ResNet) or point cloud-based (e.g., PointNet++ [35]) backbones, a trend many semantic segmentation networks continue to follow [36]. However, recent studies have begun exploring both late and early data fusion techniques [37]. Late fusion involves combining the results of separate classification methods on each dataset, such as merging the labels of already interpreted

images and point clouds [38]. In contrast, early data fusion merges either the input data or the extracted features before they are processed by a single deep learning model, showing significant promise [39]. A popular data fusion technique, for example, involves projecting 2D pixel information onto 3D points or vice versa [6]. The most common link is between RGB imagery and point cloud data, though the fusion of polygonal meshes with thermal imagery is also being explored [40]. Additionally, method and hybrid fusion techniques are under investigation: the former links methods in series, such as point-wise classification followed by line extraction [41], while the latter combines process parameters at key stages, thereby gradually integrating the modalities [32]. However, like other approaches, these techniques often do not incorporate processing parameters, timestamps, and metadata of the analyses into the results, which limits their reusability.

The structuring framework for data, features, or methods – whether in early or late fusion, as well as the serialization of results and analysis parameters – represents one of the key objectives that GEOMAPI seeks to optimize. This aspect is critical, as most current methods prioritize performance but often overlook the need for an effective methodology to store intermediate steps and outputs. Such a methodology is essential for facilitating their use in parallel or subsequent processes, while avoiding the unnecessary replication and multiplication of data.

2.3. Multi-temporal analyses

GEOMAPI aims to revolutionize the processing of close-range sensing observations from multiple measurement epochs. In EO, time-series are utilized to store and manage such observations [42]. These measurements are compressed into multi-dimensional (nD) arrays or data cubes, and are managed through their metadata, such as the source, time of collection, spatial resolution, and any preprocessing applied to the data [43]. In the construction industry, this approach is currently limited to building performance monitoring from IoT sensors, where the data is sufficiently lightweight [44]. However, for comprehensive point clouds, as required by change detection, no such data management system has yet been proposed [45]. Present analyses typically restrict themselves to comparing changes between only two datasets and do not utilize these results to inform subsequent analyses [9,46–48].

A significant bottleneck in the construction industry is the prevalent file-based system, which hampers stakeholders' ability to organize close-range sensing data within an efficient spatiotemporal structure. This system often leads to challenges like excessive data overlap, redundant measurements, and single-use analyses [49]. The shift from traditional file-based data structures to attribute-based data cubes is particularly promising for the fusion of multi-modal and multi-temporal data in close-range sensing applications. However, data cubes generally necessitate a rigid data definition. In contrast, this work proposes leveraging semantic web technologies to facilitate a flexible metadata structuration for close-range sensing. This approach is designed to accommodate a wide range of inputs, addressing the limitations of conventional data management practices.

2.4. Semantic web technology initiatives

Semantic Web Technologies and Linked Data, in particular, embody a set of design principles for sharing machine-readable, inter-linked data assets. Although these principles are predominantly used for lightweight online resources, they can also be effectively applied to organize data within local networks. At the heart of this technology is the Resource Description Framework (RDF),¹ which employs a data model representing a directed graph of resources or nodes. Each resource

¹ <https://www.w3.org/TR/rdf11-concepts/>.

within this graph is uniquely identified by a subject, namely, a Uniform Resource Identifier (URI), and is characterized by a set of triples. These triples are composed of the subject, a predicate (representing a relationship or attribute), and an object (or property value). Objects can range from simple variables such as floats and strings to more complex classes, as defined in RDF Schema (RDFS)² and Web Ontology Language (OWL)³ frameworks. These standards are instrumental in providing the necessary terminology to create web ontologies and vocabularies. These ontologies conceptually describe a specific domain of interest by defining classes, properties, and their formal logics. A key advantage of this technology is its ability to describe almost any resource in a standardized manner through ontologies. Consequently, the diverse and heterogeneous data of construction and close-range sensing can be articulated in a common language, enabling efficient querying and management [11].

In the realm of construction data management, two fundamental pillars form the foundation of GEOMAPI. The first pertains to the description of building information, encompassing BIM, CAD, and related details, a domain that is already well-understood. A variety of ontologies have been proposed to serialize building information. These range from ifcOWL [50], which focuses on a precise translation of the IFC schema into RDF, to OPM for building property sets [51], SEAS for generic features of interest [52], and BRICK, which is designed for operational building systems including sensors, controls, and actuators [53]. Particularly noteworthy are the Building Topology Ontology (BOT) [54], which details the spatial topology of object interactions, and the more recent Linked Product Data ontology (BPO) [55], featuring domain-specific modules. These ontologies incorporate geometric modules and, in the case of BPO, utilize OMG/FOG method as dedicated connector ontologies [56,57]. Additionally, there is GEOM [58] and OpenCASCADE Ontology (OCC)⁴ for geometry description.

Geometry description is critical for reasoning with geospatial information, yet its representation varies considerably across different ontologies. For example, BOT uses GEOSPARQL concepts like polygons for space definitions, while ifcOWL mirrors IFC geometry classes but can create significant overhead in geometry creation. GEOM adopts geocab, which is limited in geospatial usability due to its specific coordinate system. OCC represents OpenCASCADE CAD geometries, suitable for parametric geometries. There are also ontologies like 3DMO [59], OntoBREP [60], and OntoSTEP [61], each tailored to their respective geometries. As Pauwels et al. [62] point out, most of these schemas generate an excess of triples to describe geometry, making them ill-suited for tessellated and close-range sensing geometries. The prevailing recommendation is to retain these files in their native formats and utilize links to reference them within the RDF Graph.

The second pillar in construction data management is the description of close-range sensing geometries, an area that remains a focus of ongoing research with no straightforward implementations, such as for ontology-driven analysis of indoor point clouds [63]. While some ontology-driven semantic segmentations of point clouds do extract semantic information, aligning with some of the aforementioned ontologies, they typically do not integrate the point clouds directly into the Graph [64]. There is the now-outdated Arpenteur ontology [65], which delineates concepts for documenting an SfM (Structure from Motion) photogrammetric process. More closely related is the Geometry Metadata Ontology (GOM) [56], developed by our co-authors, which focuses on describing deviation analysis results in relation to point cloud and polygonal mesh geometries. This concept was further developed in our VforDesign (V4D) ontology [11], aimed at facilitating the reuse of pre-existing close-range sensing data. In this work, we expand upon this specific ontology to manage the metadata of close-range sensing inputs more effectively.

Table 1
Used SOTA toolboxes for close-range sensing parsing of construction data.

	Point clouds	Polygonal meshes	Imagery	BIM	CAD
Open3D [67]	✓	✓	✓	-	-
Pye57 [68]	✓	-	-	-	-
Laspy [69]	✓	-	-	-	-
OpenCV [70]	-	-	✓	-	-
PIL	-	-	✓	-	-
Trimesh [71]	✓	✓	-	-	-
IfcOpenShell [72]	-	-	-	✓	-
Ezdx [73]	-	-	-	-	✓

Overall, there exists a significant gap in the application of close-range sensing observations within the framework of semantic web technologies. Typically, datasets are incorporated merely through static links, carrying minimal semantic information. This limited integration does not enhance the usability of the information, leading to the previously discussed challenges of multi-modality, longevity, and multi-temporality in construction data management.

A potential solution to these challenges is the RDF Data Cube vocabulary initiative [66]. This initiative advocates for the publication of statistical data using Semantic Web standards, specifically RDF and SPARQL as published by the W3C, to facilitate the analysis of multidimensional models. Demonstrations on the Barcelona Open Data platform have shown that enriching the entire lifecycle of data publication is feasible by leveraging previously static PDF and CSV files. Building on this concept, GEOMAPI seeks to harness the metadata of geospatial data files to achieve a comparable level of data enrichment.

3. GEOMAPI methodology

This section presents the methodology for standardizing and automating the processing of close-range sensing data in the AEC industry through Linked Data. Specifically, GEOMAPI⁵ defines a schema and methods to manage this information as linked data resources, represented as RDF Graphs (Fig. 2). The framework inputs include various types of close-range sensing data, such as geolocated imagery, point clouds, polygonal meshes, as well as construction data comprising BIM and CAD geometries. From these inputs, a series of metadata is extracted and serialized into RDF Graphs (Section 3.1). To facilitate this process, an RDF ontology is developed to encapsulate the concepts needed to store and query the metadata (Section 3.2). Additionally, a numerical Node system is implemented to translate Graph queries into mutation methods. These methods extract relevant geospatial information from the close-range sensing inputs and convert it into an appropriate set of features for analysis (Section 3.3). Upon the conclusion of an analysis, the methodology incorporates the used method, parameters, and results into the Graph structure. This integration enables future analyses to build upon previous results. The details of the methodology are explained in this section, while the analyses are included as part of the test cases in Section 4.

3.1. Nodes and metadata extraction

At the heart of GEOMAPI is the capability to process geometries and textures derived from a variety of sources and collected at different measurement epochs. To facilitate this, we have developed parsing methods capable of handling key data formats, including point clouds, imagery, polygonal meshes, as well as BIM and CAD information. This is achieved by extending the functionalities of the most efficient state-of-the-art libraries, which are elaborated upon in Table 1.

² <https://www.w3.org/TR/rdf-schema/>.

³ <https://www.w3.org/TR/owl2-overview/>.

⁴ <https://w3id.org/occ>.

⁵ <https://github.com/KU-Leuven-Geomatics/geomapi>.

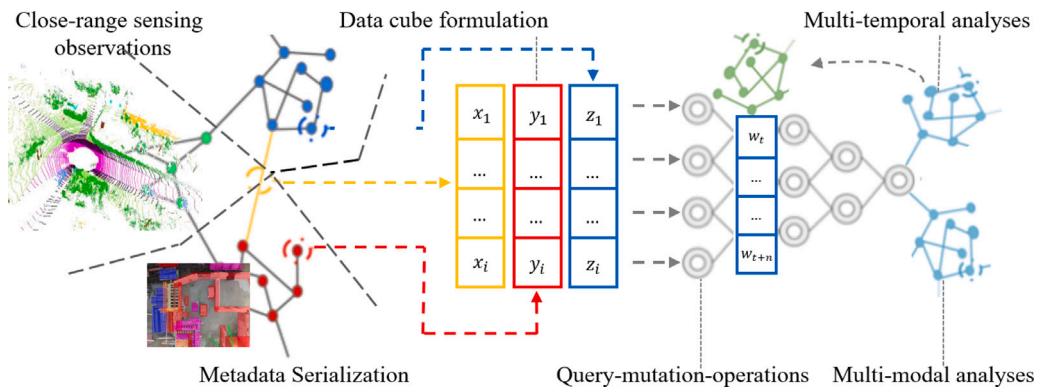


Fig. 2. Overview of the GEOMAPI methodology to serialize close-range sensing metadata as RDF graphs and use it to efficiently manage point cloud, polygonal mesh, etc. and BIM and CAD construction geometries for AEC industry applications.

Secondly, relevant metadata that describes each asset as a geospatial resource is extracted. This step is vital for Graph querying and the multi-modal methodology, as the extracted metadata governs the search space for the close-range sensing resources. The metadata includes both metric and non-metric information. As different functionalities are required to extract and homogenize the metadata, a Node system has been developed that reflects the predominant geospatial resources. GEOMAPI currently defines a general Node type with seven actual data classes governed by two supertypes, one for geometric inputs and one for image inputs (Fig. 3). Each data class inherits functionality from the supertypes and extracts the metadata of its respective resource.

Nodes store the following metric information in accordance with the conceptual framework outlined in the OpenLabel Standard⁶ which is still under development (Fig. 4). The pose is stored as the Cartesian transform $T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$. The Cartesian bounds are stored as a tuple $(x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})$ with the minima and maxima aligned with the cardinal axes. The oriented Bounding Box is stored as a 9×1 matrix $(x, y, z, R_x(\theta), R_y(\phi), R_z(\psi), s_x, s_y, s_z)$ which includes the location (x, y, z) , the rotation around the three cardinal axes applied in the order $R_z(yaw), R_y(pitch), R_x(roll)$ and the size (s_x, s_y, s_z) in each direction. The convex Hull is also stored using its bounding points as an $n \times 3$ matrix $\begin{bmatrix} x_1 & y_1 & z_1 \\ \dots & \dots & \dots \\ x_i & y_i & z_i \end{bmatrix}$ where at least a minimum of 4 non-planar points are required to define the alpha shape. Non-metric information includes the timestamp, the RDF URI, the type of coordinate system, the path to the resource, and additional resource-specific details such as image width.

3.1.1. Geometry nodes

The geometry classes in GEOMAPI specify data and metadata for resources characterized by exact geometric properties, i.e., those whose boundaries are precisely defined by their geometric resource. Currently, we include point clouds (PointCloudNode), polygonal meshes (MeshNode), BIM geometries (BIMNode), and CAD geometries (LineSetNode) in our descriptions. For point clouds, attributes like location, color, optional normal channels, and any scalars in the resource are defined separately. Polygonal meshes are similarly described, with vertices, edges, normals, and color channels. Texture files linked to the material properties are associated with each vertex through texture coordinates. It is important to note that while the texture itself is an image resource, its relation to the shape is unique and does not share properties with image Nodes.

The BIMNode stores geometry as polygonal meshes, along with identification information to link it to the original IFC file. Thus, the BIMNode comprises both a metric resource (a TriangleMesh) and an IFC pointer for non-metric metadata storage. This approach avoids unnecessary duplication of information, with additional details retrieved via Graph querying as discussed in Section 3.3. A similar logic is applied to LineSetNodes for CAD information, but geometries are grouped per layer or connected component. This grouping reduces RDF Graph serialization clutter from potentially hundreds of thousands of non-significant triples and enhances metadata extraction robustness, requiring at least four points for the convex hull.

A key aspect of resource definition is the smallest subdivision in a resource format. For example, an e57 point cloud is defined as a collection of PointCloudNodes, each with its own transform and metadata. This approach is also applied to polygonal meshes with submeshes and BIM with its geometry definitions. To maximize the advantages of Graph querying, unstructured formats, especially large point clouds, are tiled and serialized separately.

3.1.2. Image nodes

The image classes in GEOMAPI are designed to define data and metadata for geolocated image-based resources, such as those processed by Structure-from-Motion (SfM) pipelines or Mobile Mapping Systems (MMS). We distinguish between conventional imagery (CameraNode), panoramic imagery (PanoNode), and orthomosaics (OrthoNode), as each type employs distinct camera models (as depicted in Fig. 5). Specifically, by utilizing the Cartesian transform of each image, we extract the convex hull and other metric information as follows.

For the CameraNode, we use the camera pinhole model and the intrinsic camera matrix $K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$ to compute the viewing frustum for the camera. If a depth map is present, we create a convex hull up to the median depth. Otherwise, a default depth is considered. Likewise, for the PanoNode, we compute a 3D ellipsoid $\frac{(x)^2}{a^2} + \frac{(y)^2}{b^2} + \frac{(z)^2}{c^2} = 1$ with a default or median depth for $\{a, b = a, c = \frac{a}{2}\}$ along the respective axes X, Y and Z. Note that by default, c is considered half of a as the distance from the observer to the ground and potential ceiling is significantly smaller. Especially in indoor scenes, a spherical geometry would collide with numerous nodes on different floors, which increases the combination complexity unnecessarily. For the OrthoNode, we compute a rectangular cuboid from the u, v dimensions of the image multiplied with the Ground Sampling Distance (GSD) and again a default or median depth. Analogous to the above nodes, the RDF Graph serialization only includes the metadata and not the actual image, as including the image would defeat the purpose of the linked data approach.

⁶ <https://www.asam.net/>.

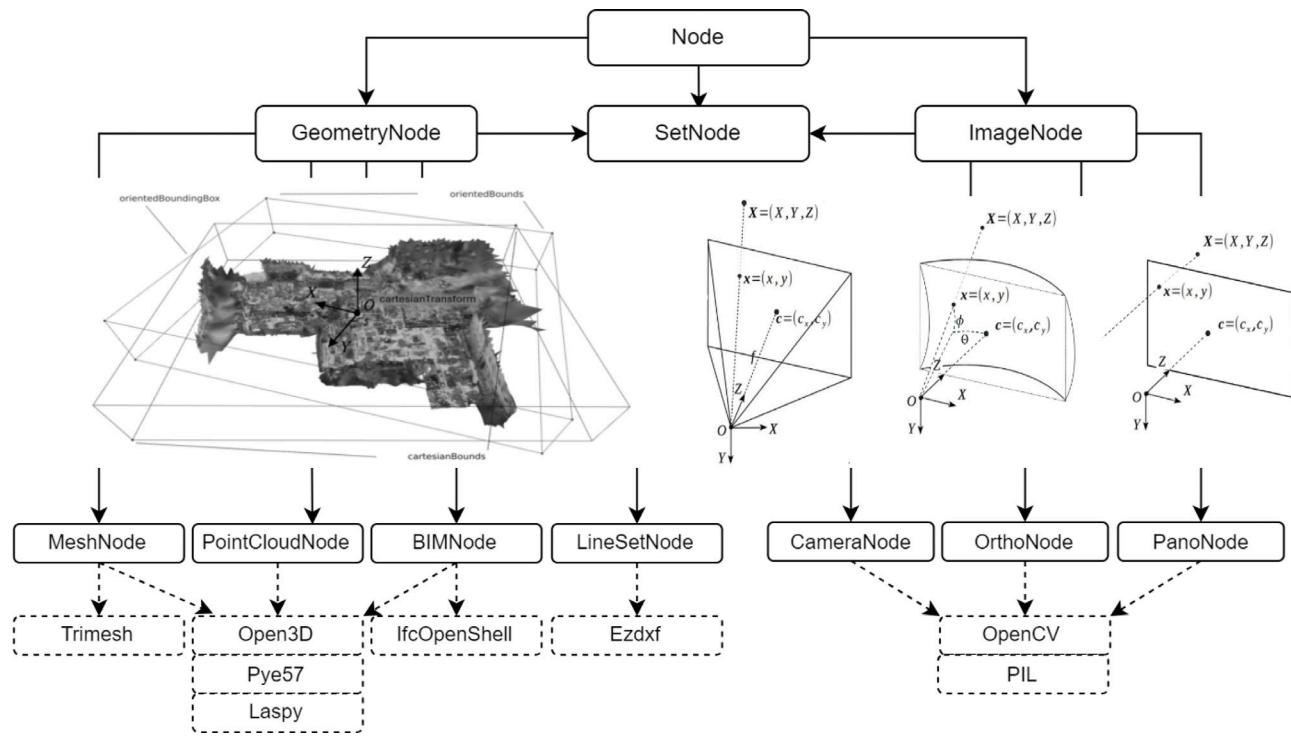


Fig. 3. Overview of the GEOMAPI Nodes and data classes built on top of existing geometry and texture libraries.

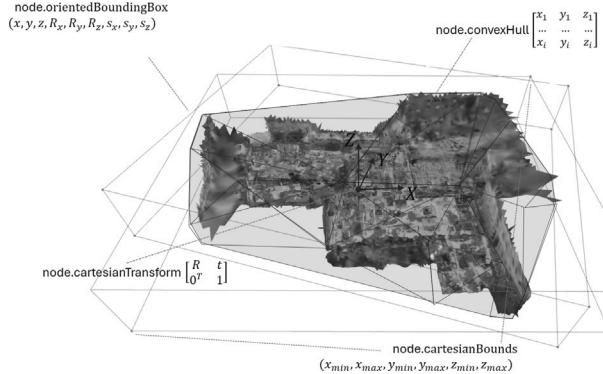


Fig. 4. Overview of the metric metadata extracted from the geometric close-range sensing resources.

3.2. Ontology

The metadata extracted from the close-range sensing data and construction geometries is serialized into RDF Graphs for efficient querying and management. To facilitate this, we have developed an ontology that standardizes the metadata associated with these observations. This ontology builds upon our previous work [11], which initially outlined concepts for describing Structure-from-Motion (SfM) pipelines, mesh and point cloud processing, and semantic enrichment processes, including semantic segmentation and instance segmentation. This ontology is now further developed to (1) define the close-range sensing nodes with the appropriate metric and non-metric metadata, (2) define the relationships to efficiently retrieve the geospatial resources and (3) define the concepts to store multi-modal and multi-temporal analyses and results. The ontology reuses existing concepts wherever possible (see Table 2) and is validated through GraphDB, showing that the schema is consistent.

Table 2

Listing of the used existing ontologies and prototype V4D ontology modules.

Prefix	Namespace
General	
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
owl	http://www.w3.org/2002/07/owl#
xsd	http://www.w3.org/2001/XMLSchema#
dcterms	http://purl.org/dc/terms/
voaf	http://purl.org/voccommons/voaf#
vann	http://purl.org/vocab/vann/
dbp	http://dbpedia.org/ontology/
Geometry	
geo	http://www.opengis.net/ont/geosparql#
bot	https://w3id.org/bot#
bpo	https://w3id.org/bpo#
omg	https://w3id.org/omg#
fog	https://w3id.org/fog#
geom	http://ontology.eil.utoronto.ca/icity/Geom/
ifc	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL
dggs	https://w3id.org/dggs/as
Texture	
xcr	http://www.w3.org/1999/02/22-rdf-syntax-ns#
exif	http://www.w3.org/2003/12/exif/ns#
Coordinate systems	
gom	https://w3id.org/gom#
epsg	http://www.opengis.net/def/crs/EPSG/0

3.2.1. Nodes

As highlighted in Section 2, the current state of the art has limitations in representing close-range sensing data. The OMG ontology [57], extended by FOG [56], offers the most comprehensive concepts for close-range sensing definitions, including concepts to describe file formats and transformations. Following the general recommendation in the field, we do not incorporate actual geometries into our system, as these typically represent instances of big data. Instead, GEOMAPI stores

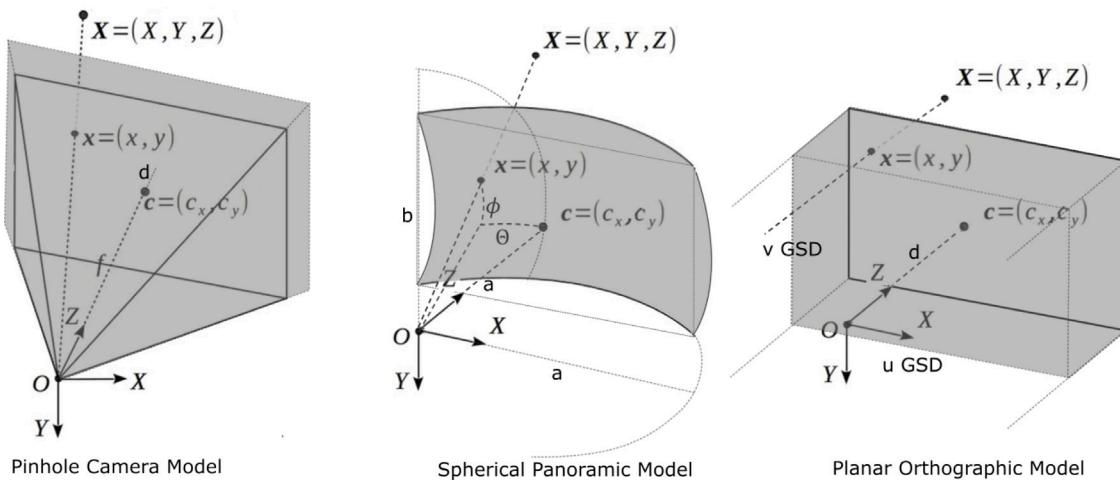


Fig. 5. Overview of the metric metadata extracted from the image-based close-range sensing resources according to the various camera models.

only the metadata outlined above and provides links to the original files (Fig. 6).

In the SOTA, geometry descriptions in RDF Graphs are commonly queried using GeoSPARQL or 3D WKT geometry literals. However, dedicated matrix definitions for the necessary metadata in geospatial queries are lacking. Therefore, in our GEOMAPI ontology, we have extended these concepts to align with the OMG and FOG ontology. For matrix representations themselves, we utilize the gom:rowMajorArray and gom:columnMajorArray datatypes to create new datatypes suited for our needs. For point cloud information, we rely on the E57 and LAS specifications to accurately describe the relevant metadata.

In managing building information within the BIMNode, GEOMAPI establishes links to the BOT, PRODUCT, and IFCOWL ontologies. However, given that our ontology's scope is primarily focused on shape and appearance information, we restrict the stored data to the GlobalID and ObjectType. We then refer to the aforementioned ontologies for the serialization of detailed semantic building information. Notably, we define both URIs as string datatypes, akin to fog:hasIfcId-guid (linked through the owl:similarAs relationship), since our requirements for IFC functionality are limited. This approach contrasts with ifcowl:IfcGloballyUniqueId, which, as a full ObjectType, necessitates a significantly larger number of triples for description.

For image information, we incorporate elements from Capturing Reality's XCR ontology for photogrammetric data and the EXIF standard for image metadata. Terminology from GOM, specifically gom:hasCoordinateSystem, is utilized to establish connections between various coordinate systems from the EPSG ontology database and the Cartesian Transform.

3.2.2. Relationships

Both vertical and horizontal relationships are defined. The vertical relationships take two forms. First, there is the SetNode (Fig. 6), a direct subclass of Node, which serves as a collector for multiple data nodes. For instance, it can include all the imagery acquired in a single flight or all the measurements captured on a certain day. It has the same metadata definitions as the data nodes, allowing it to be queried in the same manner. However, its resource is the convex hull that spans all contained nodes. Each attribute of the SetNode, such as cartesianTransform, orientedBounds, etc., is a function of all its parts and is updated when any part is updated (Fig. 6). The relationship geomapi:hasPart (based on bot:hasSubElement and geo:GeometryCollection) governs the SetNode's links to the data nodes. SetNodes are especially useful for significantly reducing the combination complexity of a query. For example, to evaluate whether an object is observed in a flight, a preliminary check is run against the SetNode instead of against every part of the set. Initially, SetNode was designed to govern different

measurement epochs, but its functionality can be employed to define any grouping of nodes, including a multi-temporal collection from different measurement epochs.

Secondly, there is the geomapi:partOf relationship (based on omg:isPartOfGeometry and geo:partOf) that links the geomapi:AbsolutePart and geomapi:RelativePart classes to a parent data Node (Fig. 7). These nodes describe direct subsets of a data Node, for example, a portion of an image. A distinction is made between absolute and relative parts, which affects the inheritance of metadata. The former, the absolute part, is a deep copy of the section and thus has its own resource and metadata (Fig. 7). It is important to note that this duplicates a portion of the data and should only be used if mutations are to be conducted on the resource. In contrast, the relative part consists only of a metadata subset and inherits the parent's resource and metadata. For instance, to compute the absolute boundaries of a relative part, one would need to combine the parent's bounds with the relative bounds.

The horizontal relations describe the similarity between sibling Nodes. Specifically, we define geomapi:derivedFrom and geomapi:similarAs, based on the owl:similarAs definitions. However, we make these relationships more concrete by linking them to the resources and metadata of the Nodes. For instance, geomapi:similarAs indicates that two nodes describe similar content, such as two separately scanned point clouds of the same scene. This is not a strict relationship; it implies that some, but not all, metadata is shared between both Nodes and that for most applications, using either one will yield the intended result. This can also target Nodes that are direct mutations of each other, such as a point cloud converted to a meshNode or the point cloud colored by a different scalar field. This is a strict relationship, implying that the metadata is shared between both Nodes and that only one of the Nodes should be used at a time in an application. These relationships are crucial in reducing the combination complexity of the Graph querying and should be established upon the creation of the Graph.

Analogously, topological relations are defined for adjacency, overlap, intersection, containment, within, and disjoint of the Nodes (Fig. 8). We expand upon similar concepts defined in BOT (Building Topology Ontology) and GEO (Geospatial Ontology) and link them to specific functions. For example, the overlap is defined as the intersection of the geomapi:orientedBounds, while the intersection relationship requires the intersection of both resources in the Node. This concept is concretized in the geomapi:Method, an owl:DatatypeProperty, which defines which function is to be used to retrieve a certain relation (see Section 3.3). It is important to note that adjacency, overlap, intersection, and disjoint are defined as owl:SymmetricProperty relationships, meaning they are bidirectional. In contrast, the containment and within

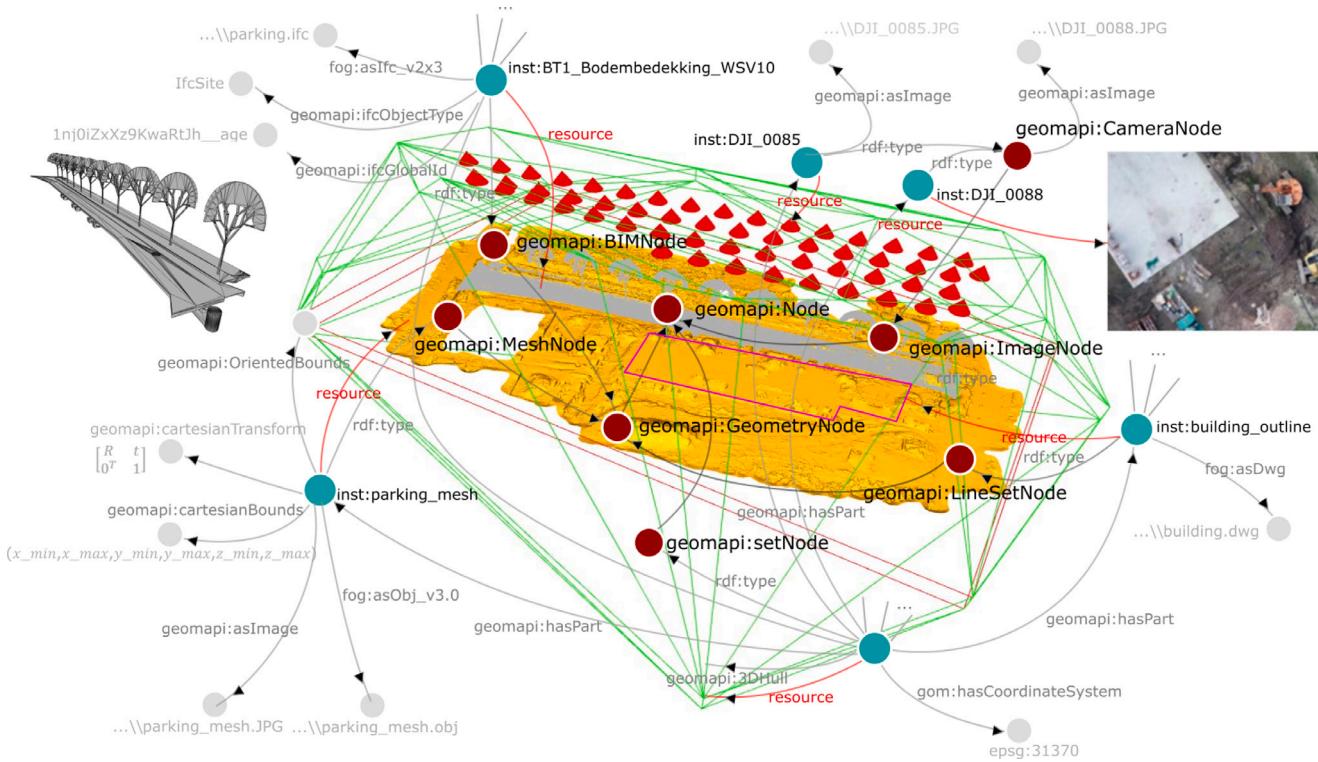


Fig. 6. Example of the GEOMAPI ontology nodes and metadata definitions visualized based on SPARQL-visualizer. The setNode's resource (green convex hull) spans over all the data nodes as parts. Note that the geometries and textures (red links) are not serialized but only called when a querying is ran against the graph.

relations are directional and are classified as owl:ObjectProperty. This distinction is crucial for accurately representing and querying the spatial relationships between Nodes in the graph.

3.2.3. Analyses

GEOMAPI defines analysis and result classes to store and reuse previous evaluations, thereby optimizing information processing (see Fig. 9). It is important to note that GEOMAPI does not aim to integrate every analysis ever run against the Graph, as this would expand the Graph beyond its useful limit, similar to a BIM model integrating every stakeholder's piece of information. Instead, we propose using a base Graph with resource Nodes and topological relations, along with a separate set of analysis Graphs that can be combined with the base Graph depending on the application's scope. For this purpose, an Analysis Node can consume data Nodes (including setNodes) as sources, along with a set of references, which include other data Nodes and previous analyses. From these analyses, a set of Result Nodes are computed, which can be linked to from the relevant data Nodes (geomapi:hasResult). Note that results do not necessarily have a one-to-one relationship with each Node. Instead, if results are shared across nodes, they can link to the same result, thereby reducing the number of triples needed to only contain unique values. The parameters of the analysis are currently defined as DataTypes instead of ObjectTypes. While it could be argued that, since parameters might be shared between analyses, they should be ObjectTypes, this approach clutters the Graph. The number of analyses is relatively limited compared to the results or data Nodes.

3.3. Querying

Utilizing resources and metadata serialized in an RDF Graph, we have developed a framework to query, evaluate, merge, and transform close-range sensing observations, a process we term the query-mutation-operation. Initially, we establish a method to query the Graph

without loading the data into memory, significantly enhancing efficiency and accessibility. Querying the Graph's triples is standardized and executable via SPARQL, facilitating RDF pattern matching. However, mathematical operations are also necessary for handling various geospatial information, as suggested in [74]. To address this, we enhance the GeoSPARQL extension functions as outlined in the RDFlib GeoSPARQL Functions for DGGS⁷ [75]. Specifically, we refine methods like sfEqual, sfWithin, sfContains, sfIntersects, sfTouches, sfDisjoint, and sfOverlaps, which rely on nested square DGGS grids, to work with our GEOMAPI matrices, including orientedBounds, 3DBOX, 3DHull, cartesianTransform, and others.

Furthermore, we optimize the queries, transforming them from list-based matchings with a $O(n^3)$ combination complexity to kd-trees, achieving an improved $O(n\log n)$ complexity. This optimization uses implementations from `scipy.spatial.KDTree`⁸ and Open3D kd-tree [67]. The specific methods utilized in these queries are incorporated into the GEOMAPI ontology, allowing for traceability of their implementation. The outcome is a suite of efficient geospatial query functions.

We have developed a series of mutation methods to transform our resources between different data types, tailored to the specific requirements of various queries. These methods include the projection of texture information, such as from pinhole cameras, panoramic, and orthomosaic sources, onto 3D geometries and the reverse process, leveraging ray tracing techniques as referenced in [76]. The transformation from point clouds to meshes is divided into two approaches: coarse modeling and fine modeling. Coarse modeling is achieved through voxel carving of octree leaf nodes, whereas fine modeling involves conventional ray marching and Poisson Meshing techniques [77].

Additionally, meshes can be reverted to point clouds by sampling their surface areas using barycentric coordinates. In a similar vein, BIM information in the IFC exchange format, alongside CAD objects,

⁷ <https://github.com/RDFLib/geosparql-dggs>.

⁸ <https://docs.scipy.org/>.

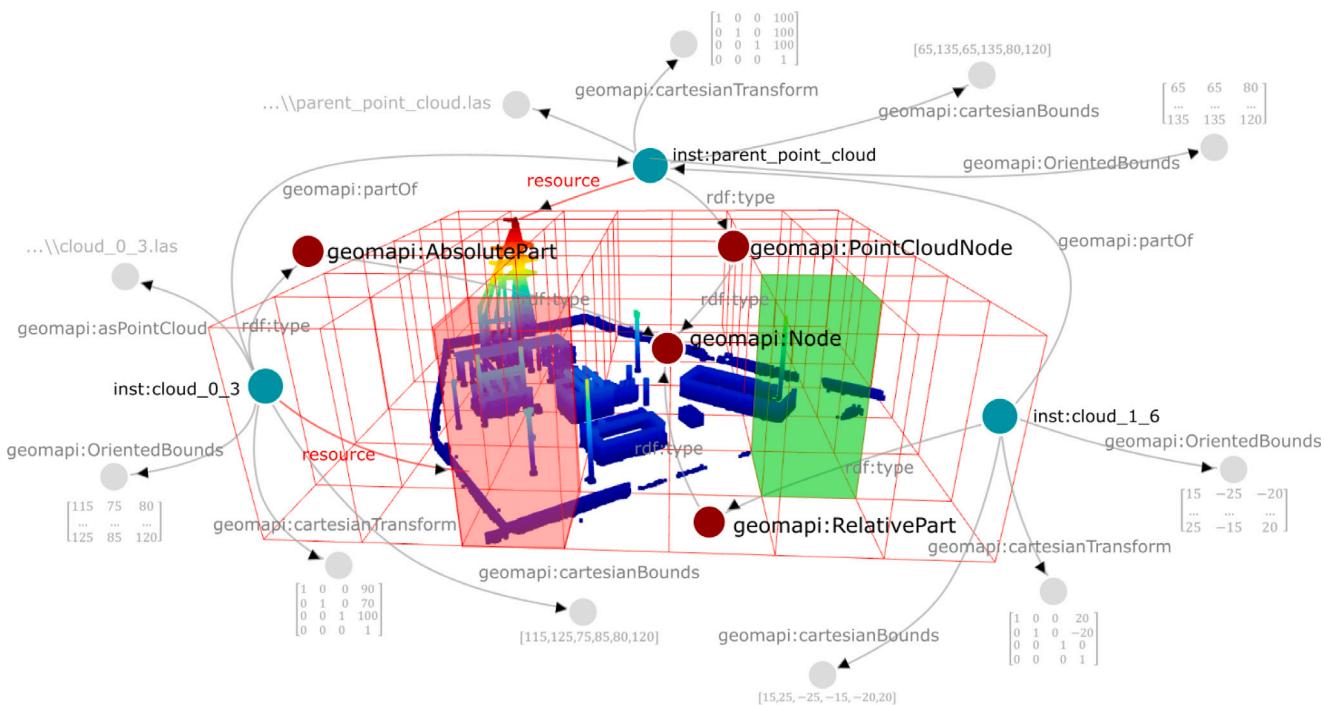


Fig. 7. Example of the GEOMAPI ontology Part Nodes and relationship definitions visualized based on SPARQL-visualizer. The AbsolutePart (red) node is independent and has its own geometry while the RelativePart (green) node is a subset only consisting of metadata relative to the parent node.

is interpreted as polygonal mesh geometries. These polygonal meshes can then be converted into either textures or point cloud geometries utilizing the aforementioned mutation methods. This comprehensive set of transformation capabilities enhances the versatility and applicability of our framework in handling various data types and formats within the realm of geospatial information processing.

We have developed geospatial operations that process geometries and textures in response to extended GeoSPARQL queries. These methods encompass set theory operations, such as intersection, union, and subsets, as well as manipulation techniques like cropping and selection. They also delve into proximity and topological relationships. These operations are seamlessly integrated with the previously mentioned RDF definitions. Typically, a GeoSPARQL query initially targets the metadata of resources to narrow down the search space, as depicted in Fig. 10. Following this, the selected resources are loaded into memory and manipulated to compute the desired results. The decision regarding whether to target geometry or texture depends on the leading data set or the modality most suited to the specific query. Notably, most queries can be executed using only metadata from references, which allows for complex geospatial operations on source geometries or textures to be carried out with significantly enhanced efficiency.

4. Testcases

Three testcases are presented to evaluate the proposed framework's capacity to leverage close-range sensing for construction applications. Each testcase focuses on a different construction domain at various stages of an asset's life cycle. Specifically, we evaluate

1. *Metadata extraction and data cube formulation.* This testcase focuses on acute progress monitoring challenges in the conventional building construction industry.
2. *Management and combination of RDF geospatial Graphs.* This testcase focuses on the operational stage of a construction, where facility management requires up to date information of constructions.

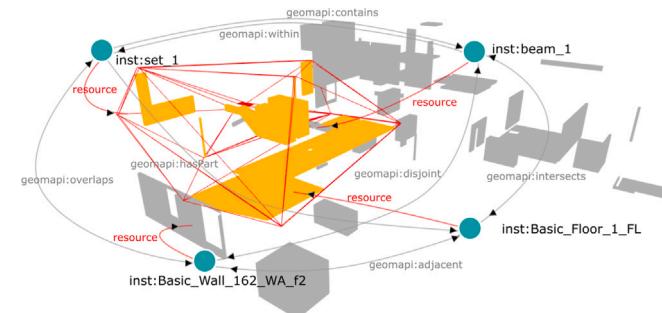


Fig. 8. Example of the GEOMAPI ontology topology definitions visualized based on SPARQL-visualizer. The setNode (red) includes the floor BIMNode which is adjacent to the wall BIMNode. The SetNode contains the beam BIMNode although its not part of the set. Yet, its `geomapi:within` can be derived through inverse reasoning of the containment triple.

3. *Subselection and tracking in multiple measurement epochs.* This testcase focus on the challenges of rapid changes and quantity take-offs in road construction.

Below, the summary of each testcase is briefly explained after which the process is optimized using the GEOMAPI framework. To maximize reproducibility, each testcase is also performed live in iterative notebooks in the documentation pages.⁹ We strongly advise readers to visit the documentation pages and follow the code examples for a better understanding of the GEOMAPI functionality.

4.1. Construction progress monitoring

In this test case, we evaluate the framework's capacity to extract the relevant metadata from close-range sensing observations and formulate

⁹ <https://ku-leuven-geomatics.github.io/geomapi/>.

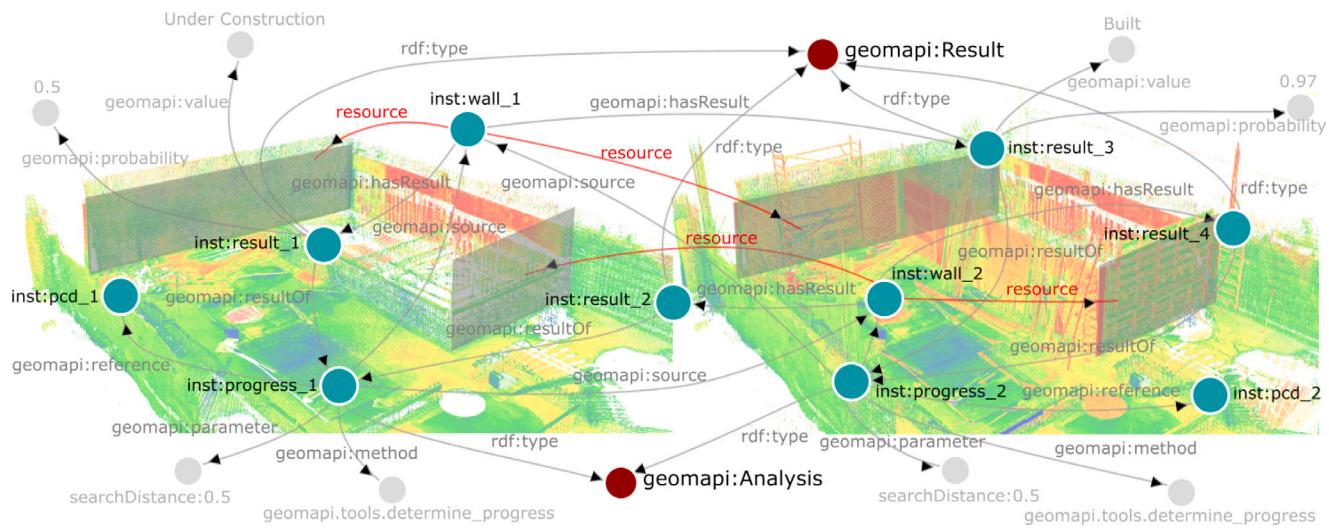


Fig. 9. Example of the GEOMAPI ontology analyses definitions visualized based on SPARQL-visualizer. Two consecutive analyses (progress 1,2) are conducted on two walls (wall 1,2) that are under construction. The results (result 1,2,3,4) show that wall 1 has progressed from “Under Construction” to “Built” using the geomapi.tools.determine_progress method with a set of parameters. Note that progress 2 uses result 1 and 2 to improve its detection rate.

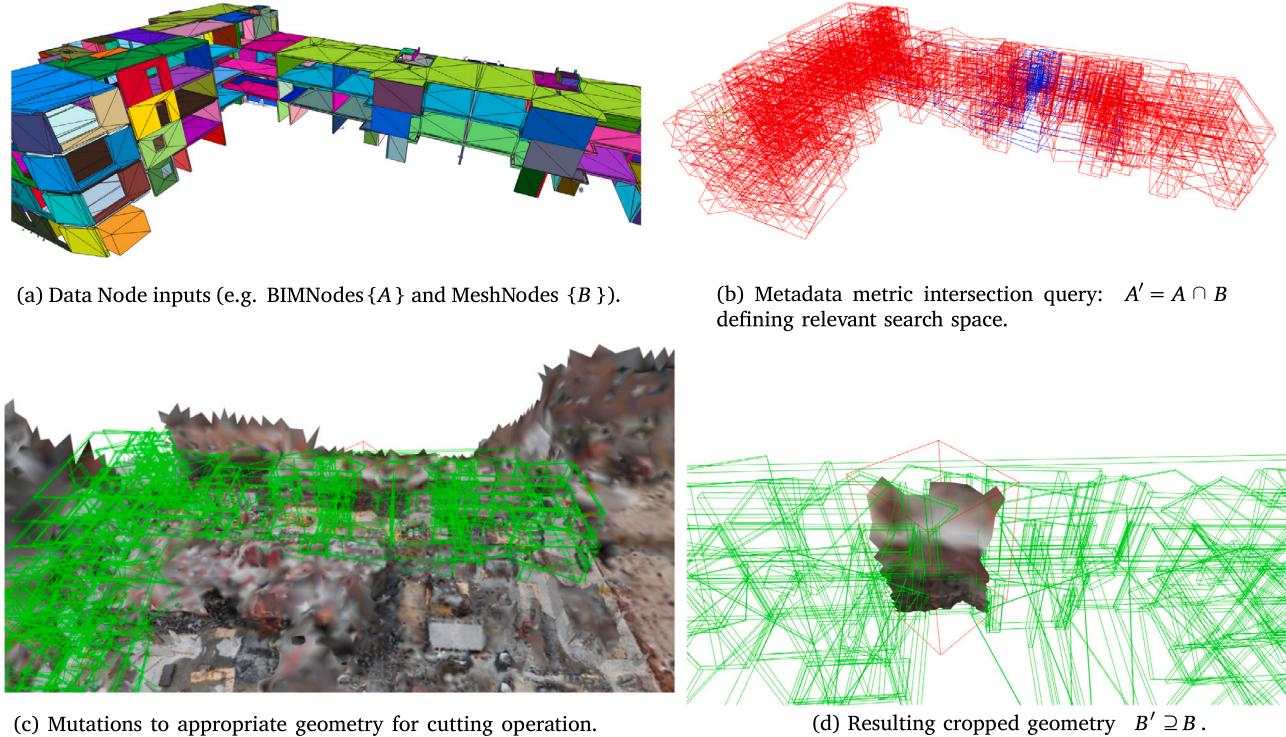


Fig. 10. Example of the three step query-mutation-operation procedure to compute the intersection between a series of meshes and BIM geometries.

the proposed data cubes. More specifically, the test includes the optimization of progress monitoring during the construction phase of an asset. This task is becoming increasingly popular in the construction industry and is performed using change detection between the BIM or CAD models and consecutive measurement epochs [8,78]. Popular close-range sensing observations in this context include imagery and point cloud data. These types of data rapidly lead to the creation of enormous datasets that must be processed, a challenge highlighted in [16]. A significant issue with current analysis methods is their failure to incorporate previous progress results, necessitating a recalibration of progress estimation over all BIM or CAD elements at each epoch. This not only increases the workload but also adds to the complexity of the

data processing task. Overall, the combination complexity is a major bottleneck in progress monitoring which can be ideally solved using the proposed Linked Data method.

4.1.1. Dataset

The dataset consists of a residential complex that includes three separate buildings and a common underground parking area, each defined in a separate IFC file. Fig. 11a shows Building 1, comprising 1192 elements, and the parking dataset, comprising 2705 elements. Both datasets depict the construction site during the structural phase, with the main classes being IfcWallStandardCase (649 elements), IfcSlab

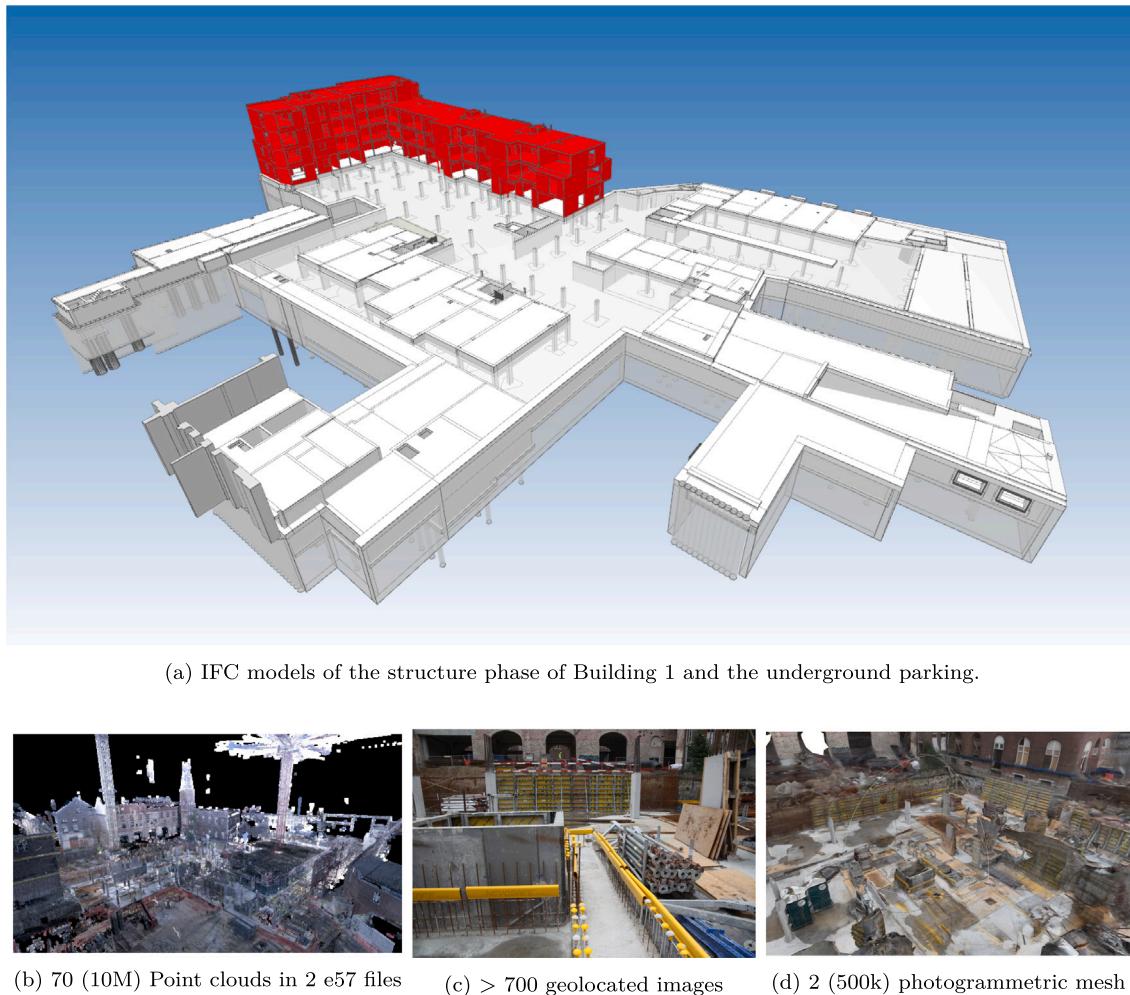


Fig. 11. Overview of week 22 and 34 close-range sensing data and BIM models.

(1387 elements), IfcBeam (425 elements), IfcColumn (220 elements), and IfcStairs (22 elements).

The progress is estimated in two consecutive datasets – week 22 and week 34 – with in total 70 point clouds in 2 e57 files (circa 700 million points), 2 photogrammetric meshes (500k vertices each) and 2600 geolocated images with an accompanying XML file with the exterior and interior camera parameters. In week 22, the site was captured with terrestrial laser scanners and hand-held imagery (Fig. 11b). During that period, the ground floor of the parking and structural columns were built, and the formwork of the first level was erected. Week 34 is similar. Documentation complexities are inherently present such as precipitation, occlusions, rebar, etc. (Fig. 11c). These objects cause significant amounts of noise and clutter and are expected to hinder the progress estimation.

4.1.2. Progress theorem

We formulate the progress per element as a binary query based on the Percentage-of-Completion (PoC) which is defined by the ratio of the observed boundary surface area of a geometry compared to the surface area that can theoretically be observed (Eq. (1)).

$$\text{PoC} = \frac{\text{Observed surface Area}}{\text{Theoretical visibility}} \quad (1)$$

where a threshold t_v can be used to state which objects are built or not (Eq. (2)).

$$\text{built state} = \begin{cases} 1, & \text{if PoC} \geq t_v \\ 0, & \text{else} \end{cases} \quad (2)$$

Both the numerator and denominator are defined as follows. The observed surface area of a geometry $n_i \in N$ (in this case a BIM element), which is represented as a polygonal mesh or BREP representation, is defined as the portion of the surface area that lies within a Euclidean distance of a collection of close-range sensing data points. For instance, the Lidar sensor or SfM pipeline that captured a portion of n_i will have points reconstructed on its surface. To this end, the Euclidean distance is observed between a fixed grid of points P_i sampled on n_i and nearby point clouds from Lidar or sampled meshes $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_j\}$ (Eq. (3)).

$$P_{o_i} = \left\{ p_i \mid p_i \in P_i, q_j \in \mathcal{Q} : \operatorname{argmin}_{q_j} \|p_i - q_j\| \leq t_d \right\} \quad (3)$$

where the distance threshold t_d is set in function of the spatial sampling resolution r of the fixed grid, e.g. 0.1 m, which can vary based on the size and type of n_i . Analogue, the theoretical visibility is defined as the exposed surface area given a BIM element's surrounding objects. For instance, a column which is partly obscured by a connected wall will have a reduced theoretical visibility. The theoretical visibility of a BIM element is established by evaluating the Euclidean distance between the same grid of points P_i and the sampled point clouds $\mathcal{P} = \{P_1, P_2, \dots, P_i, P_j\}$ on other nearby BIM Elements (Eq. (4)).

$$P_{v_i} = \left\{ p_i \mid p_i \in P_i, p_j \in \mathcal{P} \setminus P_i : \operatorname{argmin}_{p_j} \|p_i - p_j\| \geq t_d \right\} \quad (4)$$

The PoC is then defined by the ratio of the observed population over the theoretically visible population. Note that while the population P_i

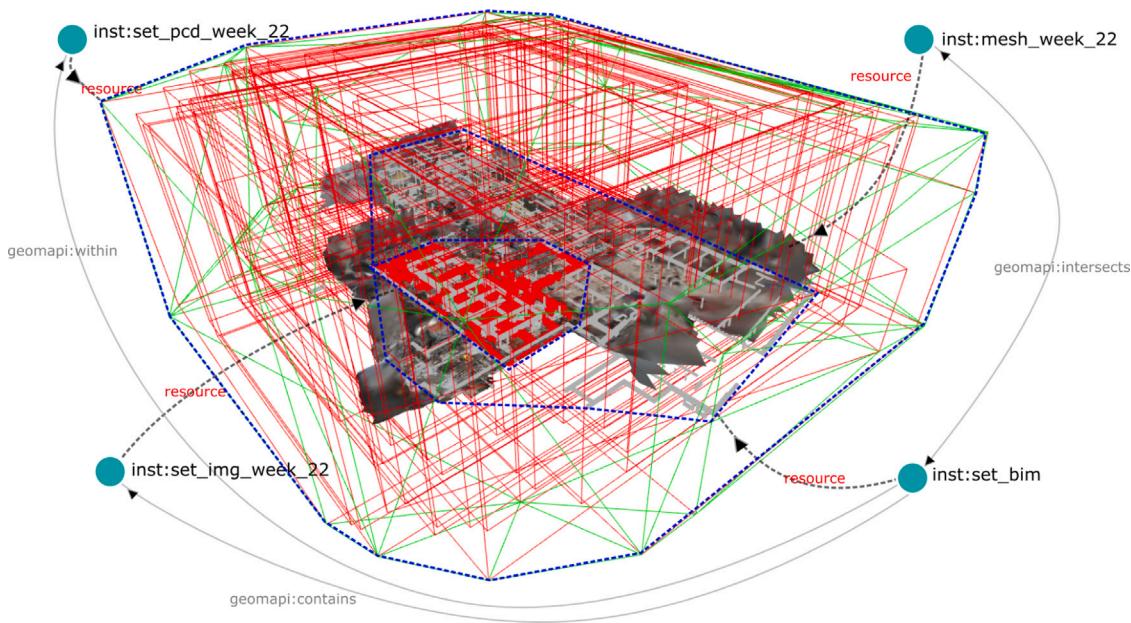


Fig. 12. Overview of the week 22 Nodes including the setNodes `inst:set_bim` with 645 IFCBeam and IFCColumn as part BIMNodes, the `inst:set_pcd_week_22` with 35 part PointCloudNodes, the `inst:set_img_week_22` with 700 part ImageNodes and the `inst:mesh_week_22` MeshNode. Note that each set's resource is the convexHull (blue dotted lines) over all its parts, while the MeshNode's resource is the polygonal mesh.

can be effaced from the PoC, this analysis is partially driven by the spatial resolution of the fixed grid and the thereof dependent distance thresholds (Eq. (5)).

$$\text{PoC} = \frac{|P_{o_i}|}{|P_{v_i}|} \quad (5)$$

So far, this analysis is straightforward with multiple variants being described both in our previous work [16] and the state of the art [8], although the use of a theoretical visibility parameter is not typically included. This method has several known drawbacks, such as false positives due to noise, the dependency on the spatial resolution, the point count being a too simplistic metric for PoC evaluation, occlusion causing false negatives and the theoretical visibility being negatively impacted the absence of constructed elements. Yet, it has an average detection rate of circa 80% which still makes it competitive.

4.1.3. Technological challenges and solutions

Combination complexity obstacle. The key issue is the combination complexity $O(n^3)$ of this 3D analysis. The resolution r is primary benefactor to the detection rate but exponentially affects the combination complexity. The close-range sensing data Q factors are especially problematic as week 22 and 34 have a combined collection of over 700 million points. So is iterating over this collection 645 times for every beam and column in the project. Only importing this data takes over 25 min with single core processing and already occupies 20–25 GB of RAM. If left unattended, this analysis would take hours and cause severe memory issues.

Solution. GEOMAPI tackles this by forming data cubes to better manage which information is fed to the analysis and serializing the results so they can be reused. This is concretized in four steps including the preprocessing, the data cube formulation, the multi-temporal detection and finally the results.

Preprocessing. First, both the BIM models and the close-range sensing repositories are preprocessed and their metadata serialized to RDF Graphs as described in Section 3.1. This is highly efficient as only the header of the e57 files and the accompanying xml of the images need to be accessed to build the necessary geometric metadata. The

photogrammetric meshes and IFC files have no such geometric information and thus need to be parsed. The result is a Graph for each week G_{22}, G_{34} , consisting of subgraphs PointCloudNodes G_P , ImageNodes G_I and MeshNodes G_M , and a Graph for the BIMNodes G_{BIM} . The combined Graphs contain in total 40,000 triples, which are efficiently stored using the Turtle¹⁰ syntax. For each week, a setNode is computed per datatype, save for the meshNodes, that only has one Node per week (Fig. 12). Note that the BIM Graph is only constructed once and then reused in subsequent analyses. To further optimize the workflow, the theoretical visibility per BIMNode is also determined in advance using the geomapi:Analysis class. The results (geomapi:Result) are stored as a separate lightweight Graph $G_{\text{BIM},V}$ so the base Graph can be reused while the new Graph can be consumed by the PoC analysis. Overall, the metadata extraction and graph creation of the in total 3317 Nodes only takes several tens of seconds and represents over 30 GB of close-range sensing data, with only a few MB of metadata.

Data cube formulation. Next, data cubes are created from G_{BIM} and the G_{22} and G_{34} Graphs to create a minimal search space using the proposed query-mutation-operation procedure (Section 3). First, the minimal number of data Nodes are retrieved for the IFCBeam and IFCColumn BIMNodes. The query is conditioned to only contain BIMNodes with a sufficient theoretical visibility t_v and that do not already have a geomapi:Result “Built” with a probability greater than t_{PoC} . The data Nodes are conditioned on the geomapi:overlaps Objectproperty that evaluates whether the orientedBoundingBox of a Node $\text{bbox}(n)$ overlaps with that of a BIMNode. Note that this relationship is symmetrical and thus is also used to exclude BIMNodes that do not overlap with any data Nodes. It therefore navigates the RDF graphs and conducts a lightweight spatial evaluation against the Nodes’ metadata to reduce the search space (Eq. (6)).

$$\begin{aligned} G' &= \left\{ n_i \mid n_i \in G_{\text{BIM},V} : v_{n_i} \geq t_v \wedge n_{i,(\text{PoC})} \leq t_{\text{PoC}} \right\} \\ G'' &= \left\{ n_i \mid n_i \in G', n_j \in G_{22} : \text{bbox}(n_i) \cap \omega\text{bbox}(n_j) \right\} \\ G'_{22} &= \left\{ n_i \mid n_i \in G_{22}, n_j \in G'' : \omega\text{bbox}(n_i) \cap \text{bbox}(n_j) \right\} \end{aligned} \quad (6)$$

¹⁰ <https://www.w3.org/TR/turtle/>.

where ω is a scale function $[0 : 1]$ for the search box to avoid including Nodes at the very edge of a resource. Following, the selected Nodes' resources are imported and mutated to a voxel octree with previously determined resolution r . This step is multi-processed as it is performed per Node. The resulting octree P is then fed to the above defined progress monitoring (Eq. (5)) function for both weeks.

Multi-temporal detection. Given the data cubes for each week, the PoC can be determined for the relevant BIMNodes. First, in week 22, the PoC is determined for all IfcBeam and IfcColumn elements that meet the theoretical visibility condition as described in Section 4.1.2. The process parameters (search distance and spatial resolution) are serialized in a separate Graph along with the PoC results for the selected BIMNodes as described in Section 3.2. In week 34, the intersection is computed between this analysis Graph, the original BIM Graph and the session 34 Graph to build the data cube for week 34. As described above, this heavily optimizes the search space in week 34. The PoC results of week 34 are then serialized in a separate analysis Graph with the same properties as the analysis Graph of week 22. As the weeks progress, these light-weight Graphs can be used to further optimize the search space i.e. by selecting certain time frames of sessions or areas of interest.

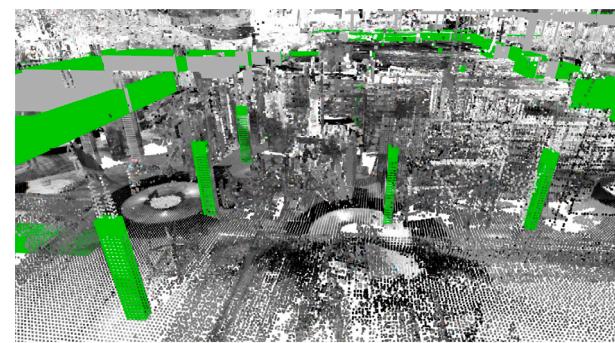
Results. The proposed method proves to be quite effective. For instance, the theoretical visibility condition reduces the number of BIMNodes by approximately 10% with the visibility threshold t_v set to 0.1 (Fig. 13a). The reduction of the spatial condition for the data Nodes respectively is 5% and 34% for week 22 and 34. The setNodes significantly speed up this process as geomapi:disjoint relationships can be deducted through every geomapi:hasPart relationship in the setNodes that fails the test (Fig. 13c). The limited effect in week 22 is due to the immense SfM mesh. More importantly is the reduction of redundant analyses. For instance, of the detected elements in both weeks, nearly 60% was already considered built in week 22, thus making a second evaluation unnecessary in week 34 (Fig. 13b). Overall, a data cube containing on average only 60% of the original Nodes could be constructed based on the above extremely fast metadata queries. In total, the preprocessing time was over 30 times faster than the SOTA analysis.

4.2. Dataset completion

In this test case, we evaluate the framework's capacity to manage and combine multi-temporal data through RDF Graphs. Specifically, the test involves completing an existing dataset of a facility with newly captured information. This increases the lifespan of the original dataset, which is a long-standing issue with close-range sensing observations. However, since there is no guarantee that the quality of the different scans will be similar, there is a need to evaluate at a point-based level to determine which parts of the datasets are the most reliable. The outdated points need to be identified and isolated, after which the zone should be updated with new recordings where needed. The parts that are unaltered should use the more detailed dataset wherever possible.

4.2.1. Datasets

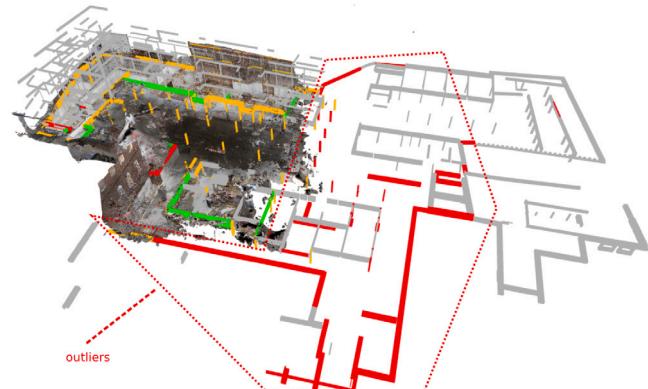
Two datasets of an educational facility, captured at different times and with different sensors, are combined. The original dataset was captured using a high-end iMMs (NavVis VLX), while the new dataset was obtained with a low-end LiDAR sensor (Microsoft Hololens). The datasets, captured a few months apart, reveal significant changes in the site. Certain objects that were previously on-site are no longer present, and new objects have appeared that were not there before. Additionally, some objects are still present but have moved from their original positions. Table 3 provides an overview of the two datasets. The comparison clearly indicates that the original dataset is of much higher overall quality, due to its higher density, coverage, and accuracy. Consequently, it is evident that the original dataset should be preserved as much as possible.



(a) BIMNodes with sufficient theoretical visibility.



(b) Already constructed elements (orange).



(c) BIMNode outliers that had sufficient theoretical visibility but had no nearby Data Nodes in week 34.

Fig. 13. Overview of the data cube formulation by the metadata analysis of the week 22 and 34 RDF graphs.

4.2.2. Combination theorem

The combination method consists of two phases: the removal phase and the addition phase. During the removal phase, outdated points are removed from the original dataset. In the addition phase, new points from the recently recorded data are added. Due to the difference in quality, the unaltered points should ideally come from the more detailed dataset. This is why it is important to identify the more detailed dataset before the combination process. In this case, the original dataset, captured by an iMMs, has much higher accuracy, making it the more accurate dataset. Additionally, a relevant sub-selection is made from the original point cloud that only contains the areas where changes have occurred. To this end, the convex hull of the new

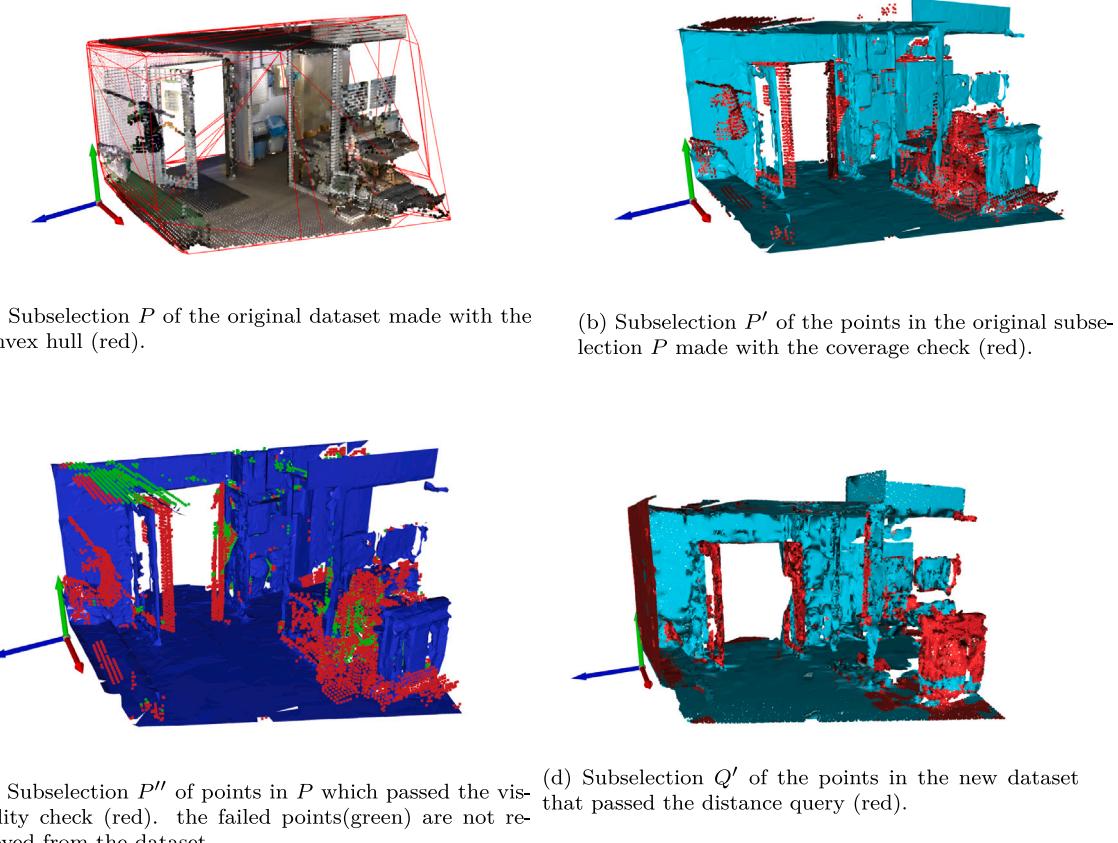


Fig. 14. Overview of the combination process including the removal and addition methods.

Table 3

Overview of the two used datasets, both taken from the same viewpoint. The VLX Dataset on the left covers a much larger area compared to the Hololens dataset on the right.

NavVis VLX	Microsoft Hololens
743,618 points	13,107 points
100 images	34 images
0.005 m density	0.05 m density
0.005 m accuracy	0.05 m accuracy

measurement epoch is used to segment the relevant area in the original point cloud (Fig. 14a).

Observation removal. First, the changes are removed from the original dataset by both evaluating the coverage and the line-of-sight (LoS) of the original data points to the sensor. The coverage is defined as the presence of a point from the new dataset near the original point (Fig. 14b). To this end, the Euclidean distances between both datasets are observed using the ktree of each dataset. The Euclidean distance is used over other algorithms because it allows us to rapidly compare observations from two separate data structures. By comparing point-to-sets we ensure that outliers do not affect the results. Each point p_i of the original dataset P that is further away from the new dataset Q than a threshold distance t_d is considered not covered and part of P' (Eq. (7)). When a point is not covered, it is either no longer present in

the new situation, or it was not captured during the new survey.

$$P' = \left\{ p_i \mid \forall p_i \in P, q_j \in Q : \operatorname{argmin}_{q_j} \|p_i - q_j\| \geq t_d \right\} \quad (7)$$

The LoS evaluation filters out the false positives in the uncovered points. Because the new dataset is smaller and less detailed, it is assumed that it has less coverage than the original dataset (Fig. 14c). The LoS is determined by evaluating whether each original point p_i is in front or behind the new dataset Q with respect to the sensor's location at the time of recording. If a point p_i is behind the new dataset Q , it could not have been visible to the new sensor. If the point is in front of the new dataset, the scanner should have been able to see the original point and can be considered out-of-date and should therefore be removed. To determine the side of the point, a raycasting scene is constructed. A ray $\overrightarrow{p_i q_j}$ is cast from each point p_i in P' to the n closest points q_j in the new dataset Q and the normals $\overrightarrow{n(q_j)}$ of the raycast hits are calculated. If the percentage of dot products $\overrightarrow{p_i q_j} \cdot \overrightarrow{n(q_j)}$ of the n closest points that is greater than t_n , is smaller than t_p , then the point p_i is considered visible and should be deleted (Eq. (8)).

$$\begin{aligned} Q_i &= \left\{ q_j \mid p_i \in P', q_j \in Q : \operatorname{argmin}_{q_j, n} \|p_i - q_j\| \right\} \\ Q'_i &= \left\{ q_j \mid \forall q_j \in Q_i, p_i \in P' : \overrightarrow{p_i q_j} \cdot \overrightarrow{n(q_j)} \geq t_n \right\} \\ P'' &= \left\{ p_i \mid \forall p_i \in P' : \frac{|Q'_i|}{|Q_i|} \leq t_p \right\} \end{aligned} \quad (8)$$

Observation addition. After the out-of-date points are removed, new points are added from the new dataset Q (Fig. 14d). To determine these points, a distance query is performed from the new points q_j to the not-removed points p_i from the original dataset $P \setminus P''$. By using the same

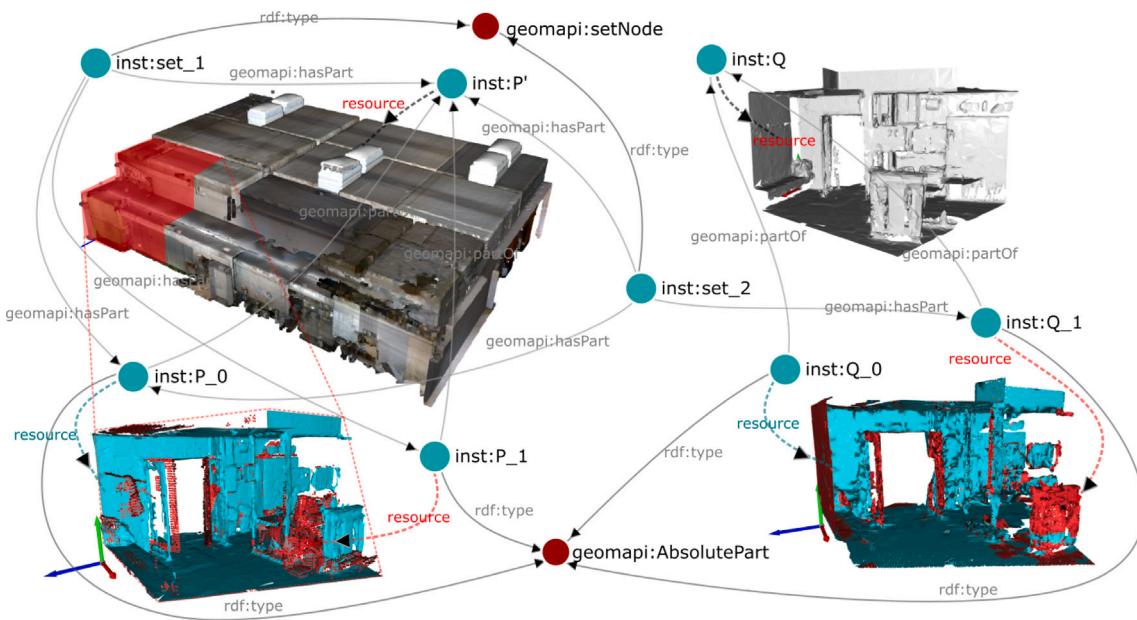


Fig. 15. Overview of the combination of PointCloudNode P and MeshNode Q . Each node is partitioned as geomapi:AbsolutePart and stored separately as the old (Cyan) and new (Red) parts of P and Q .

distance threshold t_d , we ensure a coherent combination of the datasets (Eq. (9)).

$$Q' = \left\{ q_j \mid \forall q_j \in Q, p_i \in \{P \setminus P''\} : \operatorname{argmin}_{p_i} \|q_j - p_i\| \leq t_d \right\} \quad (9)$$

The final dataset is then obtained by subtracting the points that are within LoS but have no nearby point, followed by the addition of points from the new dataset that are not covered by the original dataset. The result is a new up-to-date dataset where the original dataset is kept as much as possible and only the out-of-date points are replaced by the new points.

4.2.3. Technological challenges and solutions

Data duplication obstacle. The key obstacle in this task is the lack of a management system that prevents the duplication of information at each epoch. Current analyses typically integrate the datasets to formulate a single up-to-date dataset.

Solution. Instead, GEOMAPI depends on the RelativePart and AbsolutePart classes to facilitate the multi-temporal data management. Concretely, each dataset is defined as a data Node with the original point cloud as a PointCloudNode P and the new dataset Q as a MeshNode. Each geometry has their metadata serialized as explained in the previous test case.

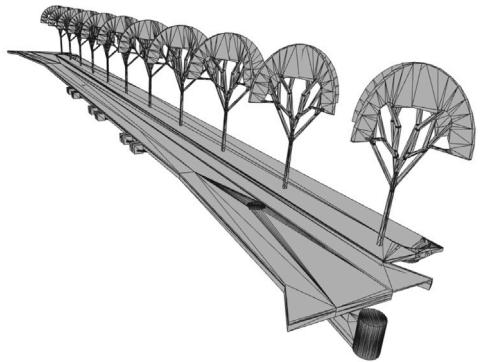
First, a data cube is constructed to only operate on the relevant data. To this end, the geomapi:convexHull of Q is used to segment P , resulting in P' . The opposite evaluation is not needed, because a simple topology evaluation yields that P fully contains Q . This filtering reduced the original dataset P by 95.5%, proving the effectiveness of the data cube methodology. For the observation removal, P' and Q are passed to the observation removal method as defined above. The result is a split of P' into the still relevant part of the dataset P_0 and the outdated part P_1 , which are defined as geomapi:AbsoluteParts (Fig. 15). During the removal phase, 36.2% of the original filtered points were not covered by the new dataset and were subjected to the LoS analysis. This flagged 56.6% of the uncovered points as being visible. As a result, P_1 consists of 15.7% of P' . Following, P_0 and Q are passed to the addition method to create the irrelevant Q_0 and new part Q_1 defined as geomapi:AbsoluteParts. Q_1 consisted of 26.4% of Q , i.e. points from the new dataset that had no existing counterpart in the

original dataset. The result is set of 6 data Nodes $\{P', P_0, P_1, Q, Q_0, Q_1\}$, each with its own metadata. Note that geomapi:AbsolutePart classes are used here instead of geomapi:RelativeParts as the source data is affected i.e. P and Q are mutated by the convex hull segmentation. Given these data Nodes, two setNodes are defined for each timeframe consisting of the original facility S_1 and the updated facility S_2 . S_1 consists of $\{P', P_0, P_1\}$ while S_2 consists of $\{P', P_0, Q_1\}$. Note that both datasets are over 95% identical which shows the high potential of increasing the lifespan of partially outdated datasets.

The proposed approach offers several significant advantages. Firstly, it avoids storing extra data, except for the lightweight Graphs that are utilized to compute data Nodes and setNodes. Secondly, additional information can be assigned to the nodes; for instance, rdfs:labels can be used to describe what each partition represents, while dcterms:created manages the temporal aspect of the datasets. Thirdly, the integration of RDF graphs with data cubes enables datasets to be processed both spatially and temporally. Moreover, the combined dataset also references the original dataset, allowing for the possibility of reverting certain changes or further refining the combination over time. This approach ensures that earlier metadata or linked analysis models can be easily connected to the most recent version of the environment.

4.3. Volume changes in road construction

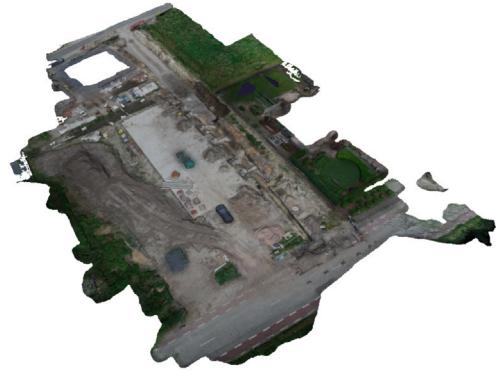
In this test case, we evaluate the framework's ability to create and track selections across multiple measurement epochs. Specifically, the test focuses on detecting volume changes in road construction earthworks. Similar to Section 4.1, this test is also a part of construction monitoring, emphasizing the tracking of material usage and displacement, such as soil displacement, grout, foundation layers, etc., between consecutive measurement epochs. Currently, the quantities of these materials are determined using a weighbridge, a method that is slow and highly susceptible to mass density changes depending on the moisture content. In this test case, we propose using volume changes derived from close-range sensing data. Both the flight-to-flight and flight-to-as-designed BIM comparisons are automated, leading to more comprehensive and faster volume calculations. Additionally, volume calculations can be determined for each element, which is essential for assessing progress and the as-built state.



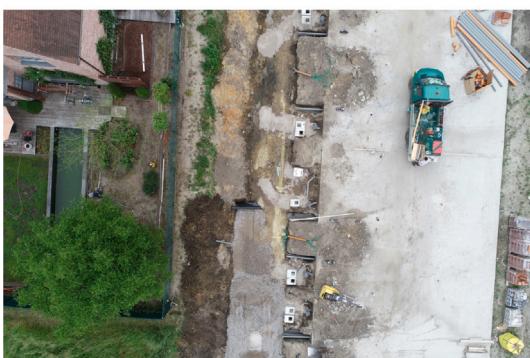
(a) The IFC-file from the testcase for volume changes in road construction.



(b) Point cloud from flight 101-0366



(c) Mesh from the flight 101-0367.



(d) An example of the occlusions in the datasets.

Fig. 16. Overview of the consecutive flights 101-0366 and 101-0367 and the BIM model.

4.3.1. Dataset

The volume changes are tracked in two consecutive UAV flights – 101-0366, captured at the beginning of June 2021, and 101-0367, captured the following week – of a new road section in an allotment between two existing roads. Fig. 16a shows the BIM model with the new roadway, pavement, sewage system, and an embankment. The corresponding IFC file contains 64 elements with different materials, including a sidewalk in clinker brick pavement with a porphyry pavement underneath on a sparse concrete foundation. Adjacent to the sidewalk is a road consisting of concrete pavement on a layer of sparse concrete. In parallel, a gravel lawn is found on an unbound crushed stone foundation with the presence of geotextile. Below the road, there are several utility lines with three inspection wells. Both datasets were collected with an RGB camera mounted on an RTK UAV. The captured images were photogrammetrically processed into one point cloud (Fig. 16b) and one mesh per epoch (Fig. 16c). Some complexities are inherently present, such as water accumulation due to precipitation, occlusions by machinery and materials, vegetation, etc. (Fig. 16d). Current methods partially solve this by semantically segmenting the ground points from the photogrammetric point clouds obtained by the RTK UAV, and we do the same. The starting point of the analysis is then the interpolated mesh so that a consistent model of the site is obtained.

4.3.2. Volume calculation theorem

Two types of volume changes are assessed. (1) A comparison between flights for the whole site in order to assess the general volume displacements and (2) a comparison of the volume changes per BIM object to assess the progress and quantity take-offs on site. Both methods rely on the distance calculation between the depthmaps of the assets.

Each depth map D is the matrix of which the depth assignment is equal to the distance calculation between a grid of 3D points $p \in P$ with a fixed resolution r and the corresponding intersection point $p(c)$ on the 3D mesh M of a resource. The intersections are determined by vertically raytracing each p per BIM object and flight geometry (Eq. (10)).

$$D = \left\{ d(x, y) \mid \forall p \in P, c \in M_{G_{BIM}} : d(x, y) = \frac{(p(c) - d(x, y)) \cdot \overline{n(c)}}{\hat{u}_z \cdot \overline{n(c)}} \right\} \quad (10)$$

where $p(c)$ and $\overline{n(c)}$ respectively are the center and normal of the triangle c in the Mesh M of the BIM object in G_{BIM} . \hat{u}_z is the unit vector in Z-direction. The two volume estimations are then derived by combining the depthmap D_{f_i} of the current flight with (1) the depthmap $D_{f_{i-1}}$ of the previous flight and (2) with the depthmaps $D_{b_1 \rightarrow b_n}$ for each BIM object in the scene. For the first assessment, The volume difference between flights is split in added and subtracted volume, and a gradient chart is created that marks the soil displacements (Eq. (12)).

$$V = \sum_{D_{f_i}, D_{f_{i-1}}} r^2 (d_{f_i} - d_{f_{i-1}}) \quad (11)$$

For the second assessment, the depth comparison between D_{f_i} and D_{b_n} is conditioned to be limited to $\{d_{b,min}, d_{b,max}\} \in D_{b_n}$ at $p(x, y)$. To this end, the depth is expressed as the height ratio given the lower and upper boundary of the BIM object at each point, (Eq. (12)) and (Fig. 17).

$$d(x, y) = \begin{cases} 0 & , d_f \geq (d_{b,max} - d_{b,min}) \\ (d_{b,max} - d_{b,min}) & , d_f \leq d_{b,max} \\ (d_f - d_{b,min}) & , \text{else} \end{cases} \quad (12)$$

Integrating this function over each BIM object's dimensions can be performed quite efficiently with only an $O(n^2)$ complexity for the object dimensions rather than for the dimensions of the entire project (Eq. (13)).

$$V = \iint_{X_{b_n}, Y_{b_n}} r^2 d(x, y) dx dy \quad (13)$$

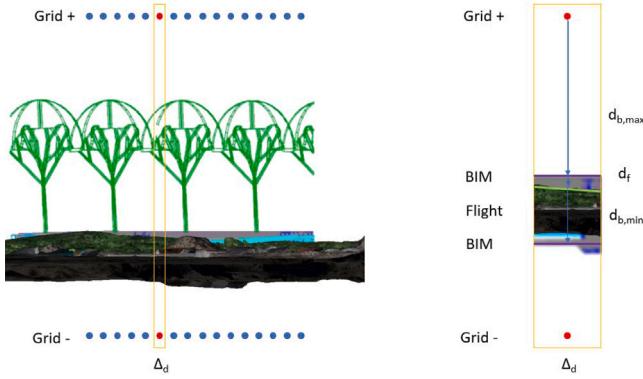


Fig. 17. The volume calculation between the BIM and one of the meshes.

So far, this analysis is a straightforward expansion of existing volume definitions to assign volume changes to a set of BIM elements. This method will yield slightly deviating results when there are more than two function returns for any $p(x, y)$ in objects with disproportionate geometries, but rarely occurs in road construction design. Overall, it is a very fast and fairly accurate method to assess volume changes on site.

4.3.3. Technological challenges and solutions

There are two primary challenges with the current methods that GEOMAPI looks to solve.

Scalability obstacle. The first obstacle concerns the scalability of the analysis. Similar to construction monitoring, there is a substantial amount of data involved due to frequent documentation of the site. In road construction, the necessity for localized data cubes is even more pronounced, as construction projects can extend over several kilometers. For example, in this relatively small project, both flights together already generate over 1000 resources with 50k meshes. This rapidly becomes problematic, especially considering the $O(n^3)$ computational complexity in relation to the XY dimensions multiplied by the number of BIM objects. Without optimization, even a relatively sparse grid sampling of 0.05 m results in a three-dimensional matrix of 64 million values, which is simply unscalable for larger projects.

Data duplication obstacle. Secondly, there is a lack of storage mechanisms for the analysis results without duplicating information, similar to the second test case. As such, current analyses typically store the absolute numerical value of changes between flights and BIM objects. However, this approach is not nuanced and does not capture the actual changes. For example, soil displacement within the site will not be reported, nor does it consider the accuracy of the flights. Similarly, changes to BIM objects must be reported using 3D geometries instead of numerical values, as occlusions, internal soil displacements, and the topology of the BIM will clutter the result.

Solution. GEOMAPI solves the first issue again through the formulation of a data cube for the relevant parts of the construction as in the previous test cases (Fig. 18). To this end, a joined meshNode $M = M_1 \cup M_2$ is created from the photogrammetric meshes (the UAV images are not needed here) and 64 BIMNodes $\{B_1, B_2, \dots, B_3\}$ are created from the IFC file, which are combined in a setNode S . The grid for the height difference estimation is determined from the M geomapi:convexhull for the flight-to-flight analysis. For the flight-to-as-designed BIM, the geomapi:3Dhull of S is projected above and below the BIMNodes using the geomapi:cartesianBounds. As such, the combination complexity is reduced from $O(n^3)$ to a much sparser $O(n^2)$ as overlaying BIMObjects do not generate extra grid points and no empty grid points are generated outside the bounds of the IFC file. Next,

M and the members of S are passed to the volume calculation analysis, of which the parameters are stored respectively in the geomapi:Analysis class and geomapi:parameters properties.

The results for the flight-to-flight analysis is a set of geomapi:RelativePart Nodes Q with the 3DHull DatatypeProperty, and the sign of the volume and the accuracy as additional metadata properties. These Nodes represent the disjoint parts of the enclosed volumes within M and are linked to these resources through the geomapi:partOf property. The sum of these parts yields the volume difference between both flights.

For the volume estimation in the BIMNodes, a topology query is first performed between the RelativeParts and the BIMNodes and also the BIMNodes internally. Following the first topology query, the intersection between both groups of data Nodes is computed to determine which parts of the intersecting BIM Nodes lie below and above the RelativePart. The volume for a BIMNode $B_i \in B$ that intersects with $\{M_1, M_2\}$ for the first and second flight is then computed as follows (Eq. (14))

$$\begin{cases} V_{B_i}(M_1) = V_{B_i} - V^+(M_1 \cap B_i) \\ V_{B_i}(M_2) = V_{B_i} - V^+(M_1 \cap B_i) + \sum_{Q_i \in Q} V(Q_i \cap B_i) \end{cases} \quad (14)$$

where $V^+(M_1 \cap B_i)$ is the difference between the volume of B_i and the volume within B_i that lies above M_1 . Note that Q_1 can be both negative and positive. For the $B_i \in B$ that do not intersect with either mesh, the volume can be deducted through the internal BIMNode topology query that yields which elements are adjacent. This adjacency is conditioned on the spatial positioning of the orientedBoundingBoxes of both nodes. For instance, all Nodes adjacent to an intersecting Node B_i that are positioned directly below B_i , are assumed to be filled, while those above are assumed empty. On average, by using the proposed method, the volume assignment to the different BIM objects and the site was accurate up to 3% when compared to a manual investigation.

While this similar to SOTA methods, there are several key differences. First, the accuracy of the evaluation can be traced as the result and analyses are linked to the initial Nodes. Second, GEOMAPI introduces the necessary sparsity by forming data cubes of only the relevant parts of the flight depthmap D_{f_i} for each BIM object. Dense matrix calculations are retained due to their superior optimization compared to sparse matrices for smaller dimensions. This reduces the computational complexity from quadratic propagation over the entire project to propagate only over each objects dimension. Additionally, depth maps of the BIMNodes only need to be computed once and are then buffered for the duration of the project. Finally, the RelativePart classes do not duplicate the data, making it very efficient to store and update volume calculations. The detected volume changes can also be confirmed in future flights by reevaluating previous volumes detected in the same area. This makes it possible to quantify whether changes have happened in a certain area and thus what the reliability is of the detected change.

5. Contributions & Limitations

In this section, we reflect upon GEOMAPI's capabilities including the processing of close-range sensing data, using multiple data types for construction applications, processing and analyzing spatio-temporal datasets and the Linked Data framework.

5.1. Contributions

5.1.1. Metadata

The first pillar of GEOMAPI is the use of metadata, which enables RDF and geospatial queries to operate with lightweight data instead of extensive datasets typically associated with close-range sensing information. This approach makes the overall framework significantly more efficient and faster compared to SOTA approaches, which typically

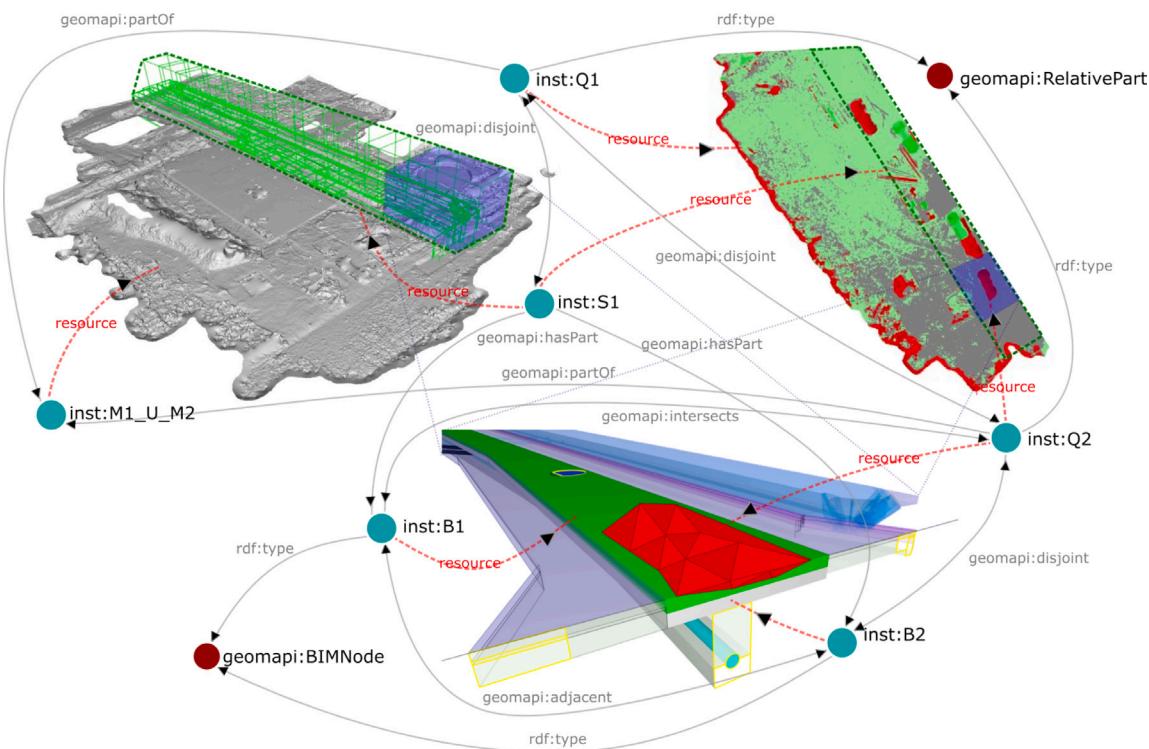


Fig. 18. Overview of the volume estimation from the differences in the set of the union of meshNodes $\{M_1, M_2\}$ and BIMNodes $\{B_1, B_2, \dots, B_3\}$. The volumes inst: Q_1 and inst: Q_2 are relative disjoint parts of inst: $M_1 \cup M_2$ and are defined by geomapi:3DHull DatatypeProperties. The volumes in the BIMNodes are derived from the intersection between inst: Q_2 that is disjoint with inst: B_2 but does intersect with the adjacent inst: B_1 . From this adjancency, the volume of inst: B_2 can be deducted.

start by importing all the data. This initial step often serves as a key bottleneck and hampers subsequent innovation.

For example, in the first test case, it was revealed that nearly 60% of the observed elements were documented in both weeks, indicating a substantial overlap between the measurement epochs. This overlap would impede traditional analysis processes, as the documentation is not only the most costly step but also places a computational burden on downstream analyses. While GEOMAPI does not enhance the site documentation process itself, it significantly reduces the computational burden associated with managing close-range sensing data. The analysis demonstrated that the search space could be effectively reduced by 40% by intelligently selecting which resource to use based on its metadata.

The storage of analysis results in RDF Graphs also substantially reduces the need to reprocess construction and close-range data. This approach improves the management of information fed into the analysis and generalizes the results for reuse. Additionally, it enables traceability of a result, a factor often overlooked in SOTA analyses. For instance, the accuracy transitivity in the third test case could be directly linked to the accuracy of the initial UAV flights. This connection highlights the effectiveness of using RDF Graphs for managing and understanding the lineage and quality of data in complex analyses.

5.1.2. Data cubes

Second, these queries facilitate the creation of data cubes that, with the implementation of k-d trees, reduce the combinatorial complexity of most operations from $O(n^3)$ to $O(n \log n)$. This is particularly crucial as construction documentation is still in its infancy, and a multitude of close-range sensing data will be captured shortly. This was demonstrated in the first test case, where processing times were over 30 times faster for a mid-sized project.

5.1.3. Nodes

Third, and central to the framework, is the node system that encompasses the mutation methods powering the data cube and query

formulation. This system not only facilitates the intuitive description of a broad spectrum of geometric and texture-based close-range observations, but also enables the assignment of additional information. For instance, timestamping allows for the homogeneous characterization of observations in both time and space. This feature is vital for multi-modal and multi-temporal applications, and each test case heavily depends on this assigned information. Furthermore, the Nodes compartmentalize much of the functionality, enabling the parallelization of various functions, including input/output operations and mutations. This parallelization significantly enhances the efficiency of processing close-range sensing data.

The use of setNodes, along with geospatial relations including disjoint, overlap, and others, further reduces the combinatorial complexity, as entire sets of Nodes can be dismissed based on their metadata. On the other hand, RelativeParts and AbsoluteParts enable dynamic fusion of datasets and are highly scalable over extended periods, in contrast to current storage approaches. This was demonstrated in the second test case, where 95% of the data could be retained, marking a significant reduction in data waste and the need to recapture construction environments.

5.1.4. RDF graphs

The utilization of RDF Graphs introduces much-needed standardization and modularity to the processing of close-range sensing data. As demonstrated in the test cases, operations can be swiftly reconfigured for entirely different construction applications without the necessity of reformulating the entire framework. This adaptability not only enhances the accessibility of the results but also significantly extends their longevity, addressing a critical issue in the current state of the art. Moreover, this approach fosters innovation by allowing users to interchange algorithms or Nodes without disrupting the overall pipeline. This flexibility is key in advancing the field, as it encourages experimentation and continuous improvement of data processing techniques.

5.2. Limitations

5.2.1. Scope of resources

A potential limitation of GEOMAPI is the standardized description of metadata for any geospatial resource, which is essential for efficiently constructing data cubes. As more data types are incorporated, maintaining this standardization becomes increasingly challenging, as each data modality has different definitions for its metadata. Currently, the metadata of the proposed Nodes can be standardized through the internal mutation methods in their respective Nodes. For example, any data Node can be converted to another Node type with the appropriate metadata. However, this process must be developed from scratch for every new data modality, which becomes increasingly difficult as the number of geospatial functionalities grows.

Additionally, some resource and function definitions are closely tied to popular APIs such as OpenCV and Open3D, which can limit their usability for specific applications. The same applies to the integration of machine learning methods. We are currently developing tools that leverage machine learning and deep learning concepts for various AEC tasks. However, we have chosen not to include them directly in GEOMAPI due to the significant hardware and software dependencies associated with different processing toolboxes like PyTorch and TensorFlow. Currently, we limit ourselves to CPU processing to reduce the number of dependencies, but GPU processing is necessary to handle geospatial inputs. While other packages may split their APIs into multiple packages to address this issue, as mentioned earlier, it can make the method less accessible.

5.2.2. Locality of information

A second limitation of GEOMAPI is related to the locality of the data. Close-range sensing information still constitutes big data, and despite the framework's efforts to reduce the combinatorial complexity, some data still needs to be imported and processed. This affects the online representation of the functionalities and the Node system that handles this data. While the metadata RDF can be hosted online, offline processing remains mandatory to fully leverage the framework's capabilities. Indeed, one could argue that structuring all close-range sensing data in a common database might offer even higher performance than the proposed method, but it would introduce yet another unique data format with reduced longevity. By focusing on standardizing the metadata and leaving the close-range sensing data in their original formats, GEOMAPI aims to strike a balance between performance and data longevity.

Furthermore, the current RDF serialization is local. Higher performance can be achieved by storing the Graphs in triple stores or using event streams, particularly for high-traffic applications like search engines or continuous sensor measurements. In contrast, close-range sensing data is only sporadically consulted, and the more frequently a dataset is consulted, the smaller it typically is. For example, weekly site documentations contain less data compared to sporadic complete site documentations for project planning. Storing RDF Graphs locally alongside the close-range sensing data also increases their longevity, as this serialization requires minimal to no maintenance.

6. Conclusions

In this paper, we tackle the construction industry's need for efficient processing and management of close-range sensing data. Specifically, we introduce a high-performing and accessible Linked Data Framework that bridges the gap between various types of close-range sensing data, including geolocated imagery, polygonal meshes, point clouds, and popular close-range sensing-driven construction applications such as construction monitoring, maintenance, and documentation. GEOMAPI is built on two key principles that push the boundaries of the state of the art: (1) Earth observation data cube principles are adopted for close-range sensing using an innovative Node system that describes an

asset's data and metadata and (2) All metadata and analysis results are managed using Linked Data techniques, enabling much-needed standardization that accelerates analyses and prolongs documentation longevity. This paper provides a comprehensive explanation of the GEOMAPI core, covering its utilities, node system, and Linked Data principles. Additionally, GEOMAPI is fully documented in a wiki¹¹ to enhance reproducibility and accessibility.

The test cases in this study evaluate the performance of the GEOMAPI framework in optimizing tasks throughout the entire lifecycle of a construction project. GEOMAPI is applied to three construction applications, including progress and volume estimations during construction, validation of as-built documentation upon delivery, and maintaining facilities up to date during the operational stage. The following aspects were thoroughly investigated: (1) The framework's ability to extract relevant metadata from close-range sensing observations and formulate the proposed data cubes, (2) its capacity to manage and combine multi-temporal data through RDF Graphs and (3) its capability to create and track selections in multiple measurement epochs. Overall, the combination of metadata extraction and the node system for formulating data cubes from close-range sensing data significantly reduces the computational workload of the operations. Furthermore, the analysis results can be easily utilized in multi-temporal tasks since the resulting Graphs can be queried in a standardized manner. Additionally, storing the process parameters enhances downstream processes, making them more nuanced and comprehensible.

The AEC industry will greatly benefit from GEOMAPI. The presented method offers stakeholders a wide range of tools to, for the first time, jointly process and manage close-range sensing observations. Images, meshes, point clouds, and other frequently used observations can be easily manipulated, interchanged, and linked to Building Information Modeling and CAD resources. Furthermore, as GEOMAPI incorporates linked data techniques, observations can also be linked to analysis results, which is crucial for multi-temporal analyses. GEOMAPI can be employed for a plethora of geospatial AEC tasks throughout a construction's lifecycle and lowers the threshold for doing so.

CRediT authorship contribution statement

Maarten Bassier: Conceptualization. **Jelle Vermandere:** Investigation. **Sam De Geyter:** Investigation. **Heinder De Winter:** Investigation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Maarten Bassier reports financial support was provided by Research Foundation Flanders. Sam de Geyter reports financial support was provided by Flanders Innovation & Entrepreneurship. Heinder de Winter reports financial support was provided by Flanders Innovation & Entrepreneurship. Jelle Vermandere reports financial support was provided by Research Foundation Flanders.

Data availability

The data that has been used is confidential.

Acknowledgments

This project has received funding from the VLAIO BAEKELAND programme (grant agreement HBC.2020.2819) together with MEET HET BV, the VLAIO COOCK project (grant agreement HBC.2019.2509), the VLAIO BAEKELAND programme (grant agreement HBC.2022.0153) together with BAUWENS NV, the FWO Postdoc grant (grant agreement 1251522N) and the Geomatics research group of the Department of Civil Engineering, TC Construction at the KU Leuven in Belgium.

¹¹ <https://ku-leuven-geomatics.github.io/geomapi/>.

- [46] S. Xu, G. Vosselman, S.O. Elberink, Detection and classification of changes in buildings from airborne laser scanning data, *Remote Sens.* 7 (2015) 17051–17076, <http://dx.doi.org/10.3390/RS71215867>.
- [47] S. Du, Y. Zhang, R. Qin, Z. Yang, Z. Zou, Y. Tang, C. Fan, Building change detection using old aerial images and new lidar data, *Remote Sens.* 8 (2016) 1–20, <http://dx.doi.org/10.3390/RS8121030>.
- [48] T. Meyer, A. Brunn, U. Stilla, Change detection for indoor construction progress monitoring based on bim, point clouds and uncertainties, *Autom. Constr.* 141 (2022) 1–20, <http://dx.doi.org/10.1016/j.autcon.2022.104442>.
- [49] S. Nikoohemmat, M. Koeva, S.J.O. Elberink, C.H. Lemmen, Change detection from point clouds to support indoor 3d cadastre, *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.* 42 (2018) 529–534, <http://dx.doi.org/10.5194/isprs-archives-XLII-4-451-2018>.
- [50] J. Beetz, J.V. Leeuwen, B.D. Vries, Ifcowl: A case of transforming express schemas into ontologies, *Artif. Intell. Eng. Des. Anal. Manuf.: AIEDAM* 23 (2009) 89–101, <http://dx.doi.org/10.1017/S0890060409000122>.
- [51] M.H. Rasmussen, M. Lefrançois, M. Bonduel, C.A. Hviid, J. Karlshø, Opm: an ontology for describing properties that evolve over time, in: *CEUR Workshop Proceedings*, Vol. 2159, 2018, pp. 23–33, URL <https://ceur-ws.org/Vol-2159/03paper.pdf>.
- [52] M. Lefrançois, J. Kalaoja, T. Ghariani, A. Zimmerman, [The seas knowledge model], *Smart Energy Aware Syst.* (2017) 1–18, URL <https://hal.science/hal-01885354/document>.
- [53] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, Brick : Metadata schema for portable smart building applications, *Appl. Energy* 226 (2018) 1273–1292, <http://dx.doi.org/10.1016/j.apenergy.2018.02.091>.
- [54] K. Janowicz, M.H. Rasmussen, M. Lefrançois, G.F. Schneider, P. Pauwels, Bot: The building topology ontology of the w3c linked building data group, *Semant. Web* 12 (2020) 143–161, <http://dx.doi.org/10.3233/SW-200385>.
- [55] A. Wagner, W. Sprenger, C. Maurer, T.E. Kuhn, U. Rüppel, Building product ontology: Core ontology for linked building product data, *Autom. Constr.* 133 (2022) 926–5805, <http://dx.doi.org/10.1016/j.autcon.2021.103927>.
- [56] M. Bonduel, A. Wagner, P. Pauwels, M. Vergauwen, R. Klein, Including widespread geometry formats in semantic graphs using rdf literals, 2019, pp. 341–350, <http://dx.doi.org/10.35490/EC3.2019.166>, <https://ec-3.org/publications/conference/paper/?id=EC32019166>.
- [57] A. Wagner, M. Bonduel, P. Pauwels, R. Uwe, Relating geometry descriptions to its derivatives on the web, 2019, pp. 304–313, <http://dx.doi.org/10.35490/EC3.2019.146>.
- [58] P. Bonsma, I. Bonsma, A.E. Ziri, E. Iadanza, F. Maietti, M. Medici, I. Bonsma, A.E. Ziri, E. Iadanza, F. Maietti, M. Medici, F. Ferrari, R. Sebastian, S. Bruinenberg, P. Lerones, Handling huge and complex 3d geometries with semantic web technology, *IOP Conf. Ser.: Mater. Sci. Eng.* (2018) (2018) 74–93, <http://dx.doi.org/10.1088/1757-899X/364/1/012041>.
- [59] L.F. Sikos, A novel ontology for 3d semantics: ontology-based 3d model indexing and content-based video retrieval applied to the medical domain, *Int. J. Metadata Semant. Ontol.* 12 (2017) 1–20, <http://dx.doi.org/10.1504/IJMSO.2017.087702>.
- [60] F. Wildgrube, A. Perzylo, M. Rickert, A. Knoll, Semantic mates: Intuitive geometric constraints for efficient assembly specifications, in: *IEEE International Conference on Intelligent Robots and Systems*, 2019, pp. 6180–6187, <http://dx.doi.org/10.1109/IROS40897.2019.8968041>.
- [61] R. Barbau, S. Krima, S. Rachuri, A. Narayanan, X. Fiorentini, Computer-aided design ontostep : Enriching product model data using ontologies, *Comput. Aided Des.* 44 (2012) 575–590, <http://dx.doi.org/10.1016/j.cad.2012.01.008>.
- [62] P. Pauwels, S. Zhang, Y.-C. Lee, Semantic web technologies in aec industry: A literature overview, *Autom. Constr.* 73 (2017) 145–165, <http://dx.doi.org/10.1016/j.autcon.2016.10.003>.
- [63] V. Stojanovic, B. Hagedorn, M. Trapp, J. Döllner, Ontology-driven analytics for indoor point clouds, in: *RE: Anthropocene, Design in the Age of Humans - Proceedings of the 25th International Conference on Computer-Aided Architectural Design Research in Asia, CAADRIA 2020*, Vol. 2, 2020, pp. 539–548, <http://dx.doi.org/10.52842/conf.caadria.2020.2.537>.
- [64] F. Poux, R. Neuville, G.A. Nys, R. Billen, 3D point cloud semantic modelling: Integrated framework for indoor spaces and furniture, *Remote Sens.* 10 (2018) 121–139, <http://dx.doi.org/10.3390/rs10091412>.
- [65] M.B. Ellefi, O. Papini, D. Merad, J.-M. Boi, J.-P. Royer, J. Pasquet, J.-C. Sourisseau, F. Castro, M.M. Nawaf, P. Drap, Cultural heritage resources profiling: Ontology-based approach, in: *Companion Proceedings of the the Web Conference, 2018*, pp. 1489–1496, <http://dx.doi.org/10.1145/3184558.3191598>.
- [66] P. Escobar, G. Candela, J. Trujillo, M. Marco-Such, J. Peral, Adding value to linked open data using a multidimensional model approach based on the rdf data cube vocabulary, *Comput. Stand. Interfaces* 68 (2020) 1–18, <http://dx.doi.org/10.1016/J.CSI.2019.103378>.
- [67] Q.-Y. Zhou, J. Park, V. Koltun, Open3d: A modern library for 3d data processing, 2018, pp. 1–20, <http://dx.doi.org/10.48550/arXiv.1801.09847>, arXiv:1801.09847v1, URL <http://arxiv.org/abs/1801.09847>. (Last Access 19 April 2024).
- [68] D. Caron, [Pye57], 2024, URL <https://github.com/davidcaron/pye57>. (Last Access 19 April 2024).
- [69] G. Brown, [Laspy], 2024, URL <https://github.com/laspy/laspy/tree/master>. (Last Access 19 April 2024).
- [70] O. TEAM, Opencv: Open source computer vision library, 2022, URL <https://opencv.org>. (Last Access 19 April 2024).
- [71] M. Dawson-Haggerty, [Trimesh], 2024, URL <https://trimesh.org/>. (Last Access 19 April 2024).
- [72] IfcOpenShell, [Ifcopenshell], 2024, URL <https://ifcopenshell.org/>. (Last Access 19 April 2024).
- [73] M. Moitzi, [Ezdxif], 2024, URL <https://pypi.org/project/ezdxif/>. (Last Access 19 April 2024).
- [74] X. Zhang, M. Zhang, P. Peng, J. Song, Z. Feng, L. Zou, Gsmat: A scalable sparse matrix-based join for sparql query processing, *arXiv* (2018) <http://dx.doi.org/10.48550/arXiv.1807.07691>, URL <http://arxiv.org/abs/1807.07691>.
- [75] David. Habgood, Rdflib geospqrql functions for dggs, 2019, URL <https://github.com/rdflib/geospqrql-dggs>, (Last access 2024).
- [76] M. Previtali, L. Diaz-Vilarino, M. Scaioni, Indoor building reconstruction from occluded point clouds using graph-cut and ray-tracing, *Appl. Sci. (Switzerland)* 2018 (2018) 827–841, <http://dx.doi.org/10.3390/app8091529>.
- [77] MeshLab, [Meshlab], 2014, URL <http://www.meshlab.net/%7D>. (Last access 2024).
- [78] Y. Xu, U. Stilla, Toward building and civil infrastructure reconstruction from point clouds: A review on data and key techniques, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 14 (2021) 2857–2885, <http://dx.doi.org/10.1109/JSTARS.2021.3060568>.