Co-funded by the European Union

ERASMUS MUNDUS JOINT MASTER DEGREE

MASTER DEGREE IN SUSTAINABLE DESIGN, CONSTRUCTION AND MANAGEMENT OF THE BUILT ENVIRONMENT

MBUILD

DEVELOPMENT OF A LIGHTWEIGHT END-TO-END SCAN-TO-BIM FRAMEWORK FOR CONSTRUCTION AND MODEL QUALITY ASSESSMENT

BASTIAN STEVANUS SITUMORANG

Supervision: Martins (UP), Sanhudo (BuiltCoLAB), Jacques (BuiltCoLAB)

Porto 2026

U.PORTO THM TECHNISCHE HOCHSCHULE MITTELHESSEN UC Universidad de Cantabria

MBUILD

ERASMUS MUNDUS JOINT MASTER IN SUSTAINABLE DESIGN, CONSTRUCTION AND MANAGEMENT OF THE BUILT ENVIRONMENT

✉ mbuild@fe.up.pt

✉ www.mbuild.eu

Edited by

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Rua Dr. Roberto Frias
4200-465 Porto
PORTUGAL
☎ +351-22-508-1400
✉ feup@fe.up.pt
▪ www.fe.up.pt

TECHNISCHE HOCHSCHULE MITTELHESSEN

Wiesenstraße 14
D – 35390 Gießen
DEUTSCHLAND
☎ +49-641-309-0
✉ info@thm.de
▪ www.thm.de

UNIVERSIDAD DE CANTABRIA

Avda. De los Castros, s/n
39005 Santander
ESPAÑA
☎ +34-942-20-10-18
✉ informacion.general@unican.es
▪ https://web.unican.es

Partial reproductions of this document will be authorized provided that the Author is mentioned and reference is made to MBUILD ERASMUS MUNDUS JOINT MASTER IN SUSTAINABLE DESIGN, CONSTRUCTION AND MANAGEMENT OF THE BUILT ENVIRONMENT, Portugal, Germany, Spain, 2026.

The opinions and information included in this document represent only the views of the Author and the Publisher cannot accept any legal or other responsibility in relation to errors or omissions that may exist.

This document was produced from an electronic version provided by the respective Author.

Dedicated to my beloved parents and family,
whose endless support, encouragement, and prayers have been my greatest source of strength
throughout this journey.

"I am always doing what I cannot do yet, in order to learn how to do it."
— Pablo Picasso

ACKNOWLEDGEMENTS

To begin with, I want to express my heartfelt thanks to my thesis supervisor, Professor Joao Pocas Martins, whose permanent guidance, precious counsel and comments on every single aspect of this thesis were indispensable to the success of it.

In addition, I would like to thank the members of the BuiltCoLab (Luis Sanhudo and Luis Jacques) who provided a lot of discussion time, technical assistance and access to research sources which supported the development of this work.

Also, I would like to express my sincere appreciation to the Erasmus Mundus Master's Program in Sustainable Design, Construction, and Management of the Built Environment (MBUILD), and all its partner Universities, as well as other institutions involved in the program, for having offered me an excellent international learning experience and academic opportunities during my two years of Master studies.

Moreover, I would like to thank all those people who are part of my group of friends and colleagues, for the support they gave me through our conversations and for encouraging me in different ways during the path followed by this research.

Lastly, I wish to extend my most sincere thanks to my dear parents and family, whose constant love, infinite support and confidence in me have represented the biggest incentive and inspiration throughout the course of this journey.

ABSTRACT

The current Scan-to-BIM process of assessing the geometrical quality of laser-scanned point clouds against Building Information Models has many manual steps, is time-consuming and depends upon high-end GPU infrastructure that is difficult to deploy in the field. Therefore, this research presents, demonstrates and assesses a lightweight end-to-end Scan-to-BIM system capable of running the entire as-built-to-report workflow on a standard workstation equipped with a consumer-grade CPU and GPU.

This research combines seven processing steps from data ingestion through to BIM model quality assessment into one automated pipeline. The seven steps are: data ingestion; BIM model preparation; preprocessing; coarse-to-fine registration with a multi-candidate yaw sweep; candidate-selected segmentation using a DL-First Hybrid Cascade; context-aware construction and BIM model quality assessment using type-specific tolerance standards (EN 13670 and EN 1992-3); and report export.

The framework was assessed using two case studies: a reinforced concrete hydraulic reservoir (106 IFC elements, terrestrial laser scanning) and a multi-storey university building (384 IFC elements, Matterport scanning). FGR (fast global registration) matched the RANSAC pipeline's final alignment quality on the reservoir and exceeded it on the university building (final fitness 0.490 versus 0.344), while completing registration approximately 1.21 – 1.27 times faster in both cases. No single segmentation operator consistently outperformed any of the others. The proposed DL-First Hybrid Cascade detected 91.1% of all elements in the university building and 59.4% in the reservoir.

Construction quality assessment classified deviations using context-aware, type-specific tolerance profiles for each element; because the achieved registration RMSE (76.8–104.3 mm) exceeds several of the applied tolerance thresholds, these conformity verdicts are reported as an indicative demonstration of the integrated workflow rather than as metrologically conclusive rulings. Model quality assessment (MQA) evaluated 350 assessable university-building elements against type-specific geometric tolerances (±10 mm for columns, mullions and curtain-wall panels; ±15 mm for walls, slabs, doors, windows and coverings). The mean quality score was 9.9/100 (8 A, 8 B, 9 C and 325 D). Mean-deviation severity comprised 49 Compliant (14.0%), 44 Minor (12.6%), 56 Moderate (16.0%) and 201 Critical (57.4%). Nevertheless, no element received a PASS verdict because its element-level RMSE exceeded the independent safety-net threshold of 1.5 times the applicable tolerance. Thus, Compliant denotes an acceptable mean-deviation band, not an overall PASS verdict. The mean absolute deviation was 67.1 mm against the 10–15 mm tolerance band. The MQA inputs were supplied pre-aligned; registration was evaluated separately using inputs expressed in different coordinate systems. Consequently, the MQA results quantify geometric agreement but cannot separate acquisition and segmentation effects from construction and model-geometry differences. For the reservoir, the entire pipeline took approximately 125.7 seconds with peak memory usage at approximately 305 MB (CPU host RAM) on a standard workstation equipped with consumer-grade CPU and GPU hardware and produced deliverables that could be imported into BIM applications including an annotated IFC4 model, IfcGroup for Revit scheduling, deviation-coloured E57 point cloud and CSV quality schedules.

The research provides both reusable methodology frameworks and empirical evidence regarding accuracy-efficiency trade-offs among classical and deep learning segmentation techniques for construction point clouds as well as integrated quality assessment methodologies combining both construction quality assessments and BIM model quality assessments within lightweight end-to-end Scan-to-BIM workflows.

Keywords: Scan-to-BIM; point cloud registration; construction quality assessment; building information modeling; lightweight processing; deep learning segmentation.

ABSTRAK

Proses Scan-to-BIM sekarang untuk menilai kualitas geometri point cloud hasil pemindaian laser terhadap Building Information Model (BIM) masih melibatkan banyak langkah manual. Proses Scan-to-BIM memakan waktu lama dan mengandalkan infrastruktur GPU kelas tinggi. Karena itu, penelitian ini memperkenalkan, menunjukkan, dan menguji sebuah sistem Scan-to-BIM ringan yang bersifat end-to-end. Sistem Scan-to-BIM dapat menjalankan seluruh alur kerja mulai dari kondisi as-built hingga menghasilkan laporan pada workstation standar yang dilengkapi CPU dan GPU kelas konsumen.

Penelitian ini menggabungkan tujuh tahapan pemrosesan yang berbeda menjadi satu pipeline otomatis. Tujuh tahapan itu meliputi pemasukan data (data ingestion); persiapan model BIM; prapemrosesan; registrasi bertahap (coarse-to-fine registration) dengan penyapuan beberapa kandidat sudut yaw (multi-candidate yaw sweep); segmentasi dengan pemilihan kandidat menggunakan DL-First Hybrid Cascade; penilaian kualitas konstruksi dan model BIM berbasis konteks dengan standar toleransi khusus untuk tiap jenis elemen (EN 13670 dan EN 1992-3); serta ekspor laporan.

Kerangka kerja ini dievaluasi menggunakan dua studi kasus, yaitu: sebuah reservoir hidrolik beton bertulang dengan 106 elemen IFC yang dipindai menggunakan terrestrial laser scanning; dan sebuah gedung universitas bertingkat dengan 384 elemen IFC yang dipindai menggunakan Matterport. Fast Global Registration (FGR) menghasilkan kualitas registrasi yang setara dengan RANSAC pada reservoir dan melampauinya pada gedung universitas (fitness akhir 0,490 berbanding 0,344). Namun, proses registrasi selesai sekitar 1,21–1,27 kali lebih cepat. Tidak ada operator segmentasi yang secara konsisten mengungguli operator lain. Metode DL-First Hybrid Cascade yang diusulkan berhasil mendeteksi 91,1 % semua elemen di gedung universitas dan 59,4 % di reservoir.

Penilaian kualitas konstruksi mengklasifikasikan deviasi memakai profil toleransi khusus per jenis elemen dengan memperhitungkan konteks; karena nilai RMSE registrasi yang dicapai (76,8–104,3 mm) melebihi beberapa ambang batas toleransi yang diterapkan, hasil kesesuaian ini dilaporkan sebagai demonstrasi indikatif dari alur kerja terintegrasi, bukan sebagai keputusan metrologis yang konklusif. Penilaian kualitas model BIM menilai setiap elemen terhadap toleransi geometrik khusus per jenis (±10 mm untuk kolom, mullion, dan panel dinding tirai; ±15 mm untuk dinding, pelat lantai, pintu, jendela, dan penutup). Dengan kriteria ini tidak ada elemen gedung universitas yang lulus: dari 350 elemen yang dinilai, skor kualitas rata-rata adalah 9,9 dari 100; diklasifikasikan berdasarkan tingkat keparahan deviasi, 350 elemen yang dinilai terdiri atas 49 Compliant (14,0 %), 44 Minor (12,6 %), 56 Moderate (16,0 %), dan 201 Critical (57,4 %), dan deviasi absolut rata-rata 67,1 mm terhadap rentang toleransi 10–15 mm menunjukkan adanya ketidaksesuaian; namun, kontribusi kesalahan dari akuisisi, registrasi, segmentasi, dan geometri model tidak dapat dipisahkan hanya berdasarkan hasil penelitian ini. Pipeline ini memerlukan waktu sekitar 125,7 detik dengan memori puncak sekitar 305 MB pada workstation. Pipeline ini juga menghasilkan output yang dapat diimpor ke aplikasi BIM meliputi model IFC4 beranotasi, IfcGroup di Revit, point cloud E57 berwarna berdasarkan deviasi, dan jadwal kualitas dalam format CSV.

Penelitian ini menyediakan kerangka metodologi yang dapat dipakai kembali. Penelitian ini menyajikan bukti empiris tentang trade-off antara akurasi dan efisiensi pada teknik segmentasi yang memakai metode klasik dan deep learning untuk point cloud konstruksi. Penelitian ini memperkenalkan metodologi penilaian kualitas untuk konstruksi dan model BIM yang terintegrasi dan ringan.

Kata kunci: Scan-to-BIM; registrasi *point cloud*; penilaian kualitas konstruksi; *Building Information Modeling* (BIM); pemrosesan ringan (*lightweight processing*); segmentasi berbasis *deep learning*.

TABLE OF CONTENTS

1 INTRODUCTION 1
1.1 BACKGROUND AND MOTIVATION 2
1.1.1 Limitations of Traditional Quality Assessment 2
1.1.2 The Scan-to-BIM Paradigm as a Quality Assessment Enabler 3
1.1.3 Geometric Complexity and the Challenge of Self-Similarity 3
1.1.4 Computational Burden in Point Cloud Registration and Processing 3
1.1.5 Automation of Tolerance-based Construction Quality Assessment 4
1.2 OBJECTIVES 4
1.2.1 Problem Statement 4
1.2.2 Research Question 4
1.2.3 Research Objectives 5
1.2.4 Research Scope and Delimitations 5
1.3 METHODOLOGY 6
1.3.1 Data Acquisition and Representation 7
1.3.2 Pre-processing & Down-sampling 7
1.3.3 Initial Alignment 7
1.3.4 Coarse-to-Fine Registration 7
1.3.5 Segmentation 8
1.3.6 Deviation Computation 8
1.3.7 Integrated Quality Assessment 9
1.4 ORGANIZATION 9
2 STATE OF THE ART 11
2.1 LASER SCANNING AND AS-IS GEOMETRIC DATA ACQUISITION 12
2.1.1 Digital Photogrammetry and Limitations in Construction 12
2.1.2 Light Detection and Ranging (LiDAR) Technology 13
2.1.3 Justification for Lightweight Algorithms Due to Inherent Data Flaws 14
2.1.4 Comparative Assessment: Accuracy, Coverage, and Practical Constraints 15
2.2 POINT CLOUD PRE-PROCESSING AND DOWN-SAMPLING STRATEGIES 18
2.2.1 Voxel Grid and Random Down-Sampling 18
2.2.2 Normal-Vector Estimation and Filtering 19
2.2.3 Noise Removal and Outlier Detection 20

2.3 POINT CLOUD REGISTRATION METHODS 20
2.3.1 Coarse Registration (RANSAC) 22
2.3.2 Limitation of RANSAC in Orthogonal and Self-Similar Structures 23
2.3.3 Fast Global Registration (FGR): Principles and Efficiency Gains 24
2.3.4 Iterative Closest Point (ICP) 25
2.3.5 Spatial Indexing Structures: Kd-Tree and Octree for registration Acceleration ... 26
2.4 POINT CLOUD SEGMENTATION APPROACHES 28
2.4.1 Geometric and Region-Growing Methods 28
2.4.2 Density-Based Spatial Clustering (DBSCAN) 29
2.4.3 Graph-based Segmentation 30
2.4.4 Capabilities and Computational Cost of Deep Learning-based Segmentation ... 31
2.5 COMPUTATIONAL EFFICIENCY IN POINT CLOUD WORKFLOWS 33
2.5.1 Benchmarking Approaches: Processing Time, Memory, and Accuracy Trade-offs 33
2.6 SCAN-TO-BIM WORKFLOWS AND AUTOMATED QUALITY ASSESSMENT 35
2.6.1 Deviation Analysis (As-Built vs. As-Designed Comparison) 35
2.6.2 Tolerance Standards and Construction Quality Metrics 36
2.6.3 BIM Model Validation (Geometric Consistency Checking) 38
2.6.4 Interoperability Formats and Revit API for Point Cloud-BIM Exchange 39
2.6.5 IFC-based Element Classification and Tolerance-based Quality Assessment 40
2.7 SUMMARY AND IDENTIFICATION OF RESEARCH GAPS 40
2.7.1 Synthesis of the Literature 40
2.7.2 Identified Research Gaps 42
2.7.3 Positioning of the Proposed Framework 43
3 METHODOLOGY **46**
3.1 FRAMEWORK ARCHITECTURE AND DESIGN PRINCIPLES 46
3.2 SYSTEM ARCHITECTURE AND THE OVERALL PIPELINE 48
3.3 INPUT DATA AND PRE-PROCESSING 50
3.4 BIM MODEL PROCESSING AND ARTIFICIAL POINT CLOUD GENERATION 51
3.5 REGISTRATION PIPELINE 53
3.5.1 Pre-processing and Feature Computation 54
3.5.2 Global and Local Operators 55

3.5.3 Candidate Strategy 55
3.6 ELEMENT SEGMENTATION PIPELINE 58
3.6.1 Shared Pre-processing and Candidate Generation 59
3.6.2 Classical Segmentation Operators 61
3.6.3 Cascaded Fallback and Fail Reason Registry 62
3.6.4 Deep Learning Architecture 63
3.6.5 Hybrid Cascade Architecture 66
3.7 DEVIATION MEASUREMENT AND CONTEXT-AWARE CLASSIFICATION (CQA) 67
3.7.1 Point-to-plane Deviation 67
3.7.2 Context Aware Tolerance Classification 68
3.8 MODEL QUALITY ASSESSMENT (MQA) 71
3.8.1 Geometric Deviation Assessment 71
3.8.2 Quality Scoring and Grade Assignment 71
3.9 VALIDATION STATUS SYSTEM 73
3.10 EXPORT PACKAGE AND BIM-INTEGRATED DELIVERABLES 74
3.11 EVALUATION METRICS AND COMPARATIVE PROTOCOL 75
3.12 METHODOLOGICAL LIMITATIONS AND ERROR BUDGET 77
3.13 SUMMARY 79
4 CASE STUDY AND EVALUATION 80
4.1 OVERVIEW 80
4.2 CASE STUDY DESCRIPTION 81
4.2.1 As-built Point Cloud 81
4.2.2 As-designed BIM Model 82
4.3 EXPERIMENTAL SETUP 83
4.3.1 Hardware and Software Environment 83
4.3.2 Parameter Configuration 84
4.4 REGISTRATION PERFORMANCE 84
4.4.1 Quantitative Comparison (RQ1) 85
4.4.2 Qualitative Alignment and Candidate Strategy 87
4.4.3 Summary 88
4.5 SEGMENTATION PERFORMANCE 88
4.5.1 Ground Truth and Evaluation Protocol 89

4.5.2 Comparison of Classical Operators (RQ2) 89
4.5.3 Deep Learning Segmentation Without BIM Priors (RQ 2) 93
4.5.4 Hybrid Segmentation Strategy (RQ 2) 98
4.5.5 Fallback Frequency and Validation Status (Robustness) 101
4.6 Construction Quality Assessment 102
4.6.1 Deviation Statistics by Element Type (RQ 3) 102
4.6.2 Severity Classification (RQ 3) 104
4.6.3 CQA Computational Cost 107
4.7 Model Quality Assessment 107
4.7.1 Element-specific Quality Scores and Grades (RQ 3) 107
4.8 END-TO-END COMPUTATIONAL FEASIBILITY 108
4.9 DISCUSSION 110
4.9.1 Threats to validity 111
4.9.2 Error Budget and Noise Sensitivity 112
4.10 SUMMARY 114
5 CONCLUSIONS **115**
5.1 SUMMARY OF WORK 115
5.2 ANSWER TO RESEARCH QUESTIONS 115
5.2.1 Primary Research Question - A lightweight End-to-end Feasibility 115
5.2.2 RQ 1 - Registration 116
5.2.3 RQ 2 - Segmentation 116
5.2.4 RQ 3 - Quality Assessment 117
5.3 RESEARCH CONTRIBUTIONS 117
5.3.1 Methodological Contributions 118
5.3.2 Technical Contributions 118
5.3.3 Practical Contributions 118
5.4 LIMITATIONS 118
5.5 FUTURE DEVELOPMENTS 120
5.6 FINAL REMARKS 121
6 REFERENCES **123**

LIST OF FIGURES

Figure 2.1 Challenges of photogrammetry in construction environments 13
Figure 2.2 Coarse-to-fine point cloud registration pipeline 22
Figure 2.3 RANSAC feature-based global registration pipeline 23
Figure 2.4 Illustration of kd-tree and octree spatial subdivision 27
Figure 2.5 Visualization of DBSCAN clustering algorithm 30
Figure 2.6 PointNet++ hierarchical set-abstraction and feature-propagation architecture 31
Figure 3.1 Overview of the proposed methodology lightweight Scan-to-BIM framework 48
Figure 3.2 Overall end-to-end pipeline of the framework 49
Figure 3.3 Scan and BIM model unaligned coordinate frames before registration 51
Figure 3.4 Back-face filtering of a sampled IFC wall and round column 53
Figure 3.5 Create artificial PCD: Converting IFC geometry into a reference point cloud 53
Figure 3.6 Registration strategy 56
Figure 3.7 Visual reference for the registration result 57
Figure 3.8 Segmentation pipeline 59
Figure 3.9 Segmentation applied to a planar wall and a non-planar column 60
Figure 3.10 Deep Learning Architecture Pipeline 65
Figure 3.11 Hybrid Cascade Architecture 67
Figure 3.12 CQA severity-classification logic 70
Figure 3.13 Jet-coloured deviation point cloud over the grey BIM reference (CQA viewer state) 70
Figure 3.14 Visual reference for the viewer and the BIM-integrated export deliverables 75
Figure 4.1 Evaluation roadmap mapping research questions to experiments, metrics, and results 81
Figure 4.2 The case-study hydraulic structure raw as-built point cloud 82
Figure 4.3 As-designed BIM model and the artificial reference point cloud derived from it 82
Figure 4.4 Final alignment fitness and inlier RMSE for FGR+ICP+Yaw-180 vs RANSAC+ICP+Yaw-180 87
Figure 4.5 Distribution of total registration wall-clock time over five repeated runs 87
Figure 4.6 Qualitative registration: centroid-seeded, coarse-to-fine and yaw-search result 88
Figure 4.7 Coarse-to-fine stage progression for Reservoir structure (a) and University building (b) 88
Figure 4.8 Classical operator accuracy (mIoU) on clean point clouds 90
Figure 4.9 Clean vs. noisy mIoU across datasets 91
Figure 4.10 Per-element segmentation speed on clean data (log scale) 92
Figure 4.11 Per-class IoU heatmap on ScanNet 94
Figure 4.12 Per-class IoU heatmap on S3DIS 95
Figure 4.13 Per-class IoU heatmap on Rohbau3D (cross-domain) 95
Figure 4.14 Deep learning mIoU-IFC across three datasets 96
Figure 4.15 Normalised multi-criteria radar plots by dataset 97
Figure 4.16 Accuracy vs. computation time on Rohbau3D 98
Figure 4.17 Element detection breakdown for the hybrid pipeline on two case studies 100
Figure 4.18 Severity distribution by IFC element type (Reservoir case study) 104
Figure 4.19 Top 35 elements ranked by mean point-to-plane deviation with tolerance thresholds 106
Figure 5.1 Summary of research questions and their evidenced answers 117

LIST OF TABLES

Table 2.1 Comparative assessment of as-is data acquisition technologies 15
Table 2.2 Comparison of point cloud registration methods 26
Table 2.3 Comparison of point cloud segmentation approaches 32
Table 2.4 Comparison of typical accuracy and computational cost 33
Table 2.5 Rigid versus adaptive tolerance judgment in automated quality assessment 37
Table 2.6 Mapping of identified research gaps to the proposed framework and thesis structure 45
Table 3.1 REST application-programming-interface endpoints exposed by the framework back end 48
Table 3.2 IFC element types and their comparison categories 52
Table 3.3 Registration parameters, expressed as multiples of the common voxel size v = 0.10 m 55
Table 3.4 The three registration candidates evaluated 56
Table 3.5 The segmentation pipeline 60
Table 3.6 Cascading-fallback fail reason registry 63
Table 3.7 Context-aware tolerance table for the HydraulicStructure domain 69
Table 3.8 Severity levels assigned from the mean deviation and the type tolerance thresholds 69
Table 3.9 Geometric tolerance profile by IFC type 72
Table 3.10 Quality-score calibration (tolerance = 15 mm) and letter-grade thresholds 72
Table 3.11 Validation status system 73
Table 3.12 Contents of the CQA and MQA export packages 74
Table 3.13 The four states of the Three.js interactive viewer 75
Table 3.14 Error budget for a reported element deviation 78
Table 3.15 Summary mapping of methodological stages to the research questions (RQ) 79
Table 4.1 Characteristics of As-built Scan dataset 82
Table 4.2 Characteristics of As-designed BIM dataset 83
Table 4.3 Hardware and software environment 84
Table 4.4 Registration performance: FGR + ICP versus RANSAC + ICP on both case studies 85
Table 4.5 Accuracy of the classical operators on clean point clouds 90
Table 4.6 Robustness Analysis 91
Table 4.7 Per-element CPU segmentation speed (ms/elem) on clean point clouds 92
Table 4.8 Per-class IoU on ScanNet validation (50 scenes, IFC-mapped classes) 93
Table 4.9 Per-class IoU on S3DIS Area 5 (68 rooms, IFC-mapped classes) 94
Table 4.10 Per-class IoU on Rohbau3D (3 construction-site scenes, cross-domain transfer) 95
Table 4.11 Cross-dataset summary: mIoU-IFC, consistency, inference time, and multi-criteria ranking 96
Table 4.12 DL-First Hybrid Cascade results on the UC Building and Reservoir case studies 99
Table 4.13 Distribution of validation statuses across both case studies (DL-First Hybrid Cascade) 101
Table 4.14 Point-to-plane deviation statistics aggregated by IFC element type (mm) 103
Table 4.15 Summary of measured geometric deviations and average tolerance by storey 103

SYMBOLS, ACRONYMS AND ABBREVIATIONS

MATHEMATICAL SYMBOLS

d – Point-to-plane deviation distance [mm]

n – Surface normal vector

R – Rotation matrix, $R \in SO(3)$

RMSE – Root Mean Square Error [mm]

s – Composite registration score, s = fitness − 0.35 × inlier RMSE (Equation 3.5)

t – Translation vector, $t \in \mathbb{R}^3$

T – Homogeneous transformation matrix [4×4]

v – Voxel edge length [m]

Δ – Relative accuracy decline between clean and degraded conditions [%] (Equation 4.1)

ε – Neighbourhood radius parameter (DBSCAN) [m]

σ – Standard deviation of the deviation distribution [mm]

τ – Evaluation distance threshold, $\tau = 1.5v$ [m]

ACRONYMS AND ABBREVIATIONS

B-Rep – Boundary Representation

BIM – Building Information Modelling

C2C – Cloud-to-Cloud (deviation analysis method)

C2M – Cloud-to-Model (deviation analysis method)

CPU – Central Processing Unit

CQA – Construction Quality Assessment

CQC – Construction Quality Control

CSV – Comma-Separated Values

CUDA – Compute Unified Device Architecture

CV – Coefficient of Variation

DBSCAN – Density-Based Spatial Clustering of Applications with Noise

DL – Deep Learning

E57 – ASTM E2807 point cloud exchange format

F1 – Harmonic mean of precision and recall

FGR – Fast Global Registration

FPFH – Fast Point Feature Histogram

GCP – Ground Control Point

GPU – Graphics Processing Unit

ICP – Iterative Closest Point

IFC – Industry Foundation Classes

IMU – Inertial Measurement Unit

IoU – Intersection over Union

KNN – k-Nearest Neighbor

LiDAR – Light Detection and Ranging

MEP – Mechanical, Electrical and Plumbing

mIoU – Mean Intersection over Union

mIoU-IFC – Mean Intersection over Union computed over IFC-mapped classes

ML – Machine Learning

MLS – Mobile Laser Scanning

MQA – Model Quality Assessment

OA – Overall Accuracy

OBB – Oriented Bounding Box

PCA – Principal Component Analysis

Pset – IFC Property Set

PTv3 – Point Transformer V3

QA – Quality Assurance

QC – Quality Control

RAM – Random-Access Memory

RANSAC – Random Sample Consensus

RGB-D – Red–Green–Blue plus Depth

S3DIS – Stanford Large-Scale 3D Indoor Spaces dataset

SfM – Structure from Motion

SLAM – Simultaneous Localisation and Mapping

SpU-Net – Sparse Convolutional U-Net

TLS – Terrestrial Laser Scanning

UGV – Unmanned Ground Vehicle

VRAM – Video Random-Access Memory

1 INTRODUCTION

There are currently significant disparities at a structural level within the construction sector regarding the very fast rate of technological development related to the use of computers to create models for building design, and the large manual nature of determining whether the building created from those designs matches the models. According to the National Institute of Building Sciences, Building Information Modelling (BIM) is a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life cycle; defined as existing from earliest conception to demolition. Although BIM presents a tremendous conceptual advancement on how data is represented and exchanged among stakeholders, ensuring the integrity of that data throughout the entire life cycle of a project remains a significant challenge. When comparing the designed intent of a building to its expected execution on-site, and ultimately to the "as-built" condition of that same building (Park et al., 2026) there will inevitably be some difference. Those discrepancies could lead to reduced levels of confidence in the model's ability to represent reality accurately and subsequently affect potential uses of the model for building performance modelling and facilities assessments.

Historically, geometric compliance checking was completed manually via comparison of the original construction drawings against the physical dimensions measured in the field (on-site) (B. Wang et al., 2025). This manual process is time consuming, labour intensive, and open to subjective interpretation and/or human error (Tan et al., 2025). Furthermore, traditional workflows often result in a fractured set of documents that do not represent a single integrated digital document set. As construction projects grow larger and more complex, these limitations can adversely affect cost and schedule performance, especially for large building elements like walls, slabs, columns, and MEP systems (Bassier et al., 2020).

As a response to many of these challenges associated with documenting changes on site during construction, Three-Dimensional Laser Scanning (LiDAR) technology has rapidly emerged as a viable tool for capturing a detailed description of both interior and exterior surfaces of structures and buildings. The results of LiDAR surveys are presented as a collection of discrete points called a point cloud. The resulting point cloud is then compared to BIM models using Scan-vs-BIM methods to identify geometric deviations (Chen & Cho, 2018). Spatial registration is required to perform this comparison since the point cloud and BIM model need to exist within the same coordinate reference system. Currently however, spatial registration is primarily completed manually. Therefore, much time and resources are lost due to inefficiency and operational bottlenecking (Hajian & Becerik-Gerber, 2010).

Many automated registration techniques currently utilize the Iterative Closest Point (ICP) algorithm. While ICP has proven itself capable of performing fine registrations at very high levels of accuracy, it

also tends to be computationally expensive and sensitive to poorly selected initial parameters and limited overlap conditions (Zou et al., 2023). More recently developed deep learning methods based upon PointNet++, SpU-Net, and PTv3 have improved the robustness to variations in point cloud content but they require high-performance computing capabilities (e.g., GPUs). Thus, they cannot be easily deployed on typical construction-site hardware (P. Li et al., 2020).

Automatic construction quality assessment is a practical application of Scan-to-BIM. Geometric differences alone are insufficient: construction quality must be assessed against project-specific tolerances. Accurate registration, robust segmentation and element-level deviation analysis are therefore needed to reduce the effects of scanner noise, registration error and temporary occlusions. Although many studies address individual stages, integrated and computationally efficient workflows remain an important research need.

Thus, this research presents a lightweight end-to-end Scan-to-BIM framework for assessing the quality of both construction work and digital models. The term "lightweight" is defined as having an operational definition with limited memory requirements and practical execution times. An example of a lightweight framework is when it meets the following three criteria: (1) The entire geometric processing pipeline can run within five minutes or less on a commodity workstation without requiring high-end GPU infrastructure; (2) Peak resident memory usage will be under one gigabyte; GPU VRAM required by the optional learned component will be reported separately and (3) Any deep learning module utilized will serve as an optional filter which will not prevent the generation of output by the remaining processing pipe. This last criterion represents the most significant aspect of the framework. It allows a pre-trained neural network to potentially enhance segmentation results when a GPU is available, while allowing geometric processing to continue without the learned filter. As such, these three criteria represent the actual hardware available on construction sites and are reported upon in detail relative to their measurements in Section 4.8.

1.1 BACKGROUND AND MOTIVATION

The motivation behind conducting this research lies in inefficient practices associated with typical quality assessment in construction and issues with Scan-to-BIM processes, specifically transitioning from point cloud collection to BIM based model evaluation. The purpose of this research will be to address both industry and academia identified issues related to enhancing safety, decreasing costs and improving sustainability.

1.1.1 Limitations of Traditional Quality Assessment

There are many factors involved in construction projects. Thus, they are characterized by high levels of uncertainty and complexity in field conditions (Tsige et al., 2025). Therefore, quality control during construction projects is inherently difficult. Since many of the measurements are taken manually, for example the wall inclination check is subject to potential inconsistencies and inaccuracies.

Since early identification of deviation is crucial in order to maximize corrective actions while minimizing costly rework, it is imperative that quality assurance (QA) processes identify and report deviation immediately upon occurrence. However, the amount of time required to perform frequent manual inspections in large scale construction projects limits the feasibility of performing such inspections (Samsami, 2024). Scan-vs-BIM addresses this by comparing point clouds against BIM models under specified tolerance levels (Erdélyi et al., 2022).

Nevertheless, the ability of scan-vs-BIM to accurately evaluate quality is contingent upon the degree of accurate registration of point clouds. It should be noted that even slight errors may result in false positives and/or false negatives. Because most construction specifications call for millimetre-level

precision (Mora et al., 2021), registration accuracy is critical. Although various tools are currently available to automate Registration, complete automation has yet to occur (Sheik et al., 2022).

1.1.2 The Scan-to-BIM Paradigm as a Quality Assessment Enabler

Scan-to-BIM is a method used to transform a quality assessment from a sample-based inspection to a spatial assessment using laser scans. Using laser scanning technology, construction projects produce massive amounts of point cloud data. Point cloud data allows for element level assessments versus sampling-based assessments (Do et al., 2023). In addition to allowing element level assessments, point cloud data can be reused for other uses including as built documentation and spatial analysis (Escudero, 2023). Reusing point cloud data creates a single digital representation connecting design intent and actual constructed physical space (L. Yang et al., 2024). Therefore, Scan-to-BIM is a paradigm that supports a transition from quality control processes to quality assessment processes.

The paradigm facilitates the transformation of quality assessments from traditional quality control processes to structured quality assessment processes. Structured quality assessment processes allow for quality deviations to be measured and quantitatively evaluated and then aggregated into measurable indicators. Various studies have previously developed and applied automated systems to generate aggregated quality scores from BIM based evaluation frameworks (Valinejadshoubi et al., 2024). This thesis builds upon previous work by integrating both construction quality assessment and model quality assessment within a singular framework.

1.1.3 Geometric Complexity and the Challenge of Self-Similarity

Registration methods can be broadly categorized into feature-based methods, point-based methods and hybrid methods (Mei et al., 2023). Feature-based registration relies on geometric features; however, feature-based registration methods experience difficulties in environments with repeated architectural features. Self-similarity occurs when two different objects appear identical (Tsige et al., 2025) or when two similar structures have similar normals. Both self-similarity and repetition can cause incorrect correspondence assignments and visually plausible but incorrect Registration outcomes (Zou et al., 2023). To mitigate these risks, the proposed framework does not rely on a single feature-correspondence solution. Instead, it seeds an alignment at the centroid of both point cloud datasets and then evaluates three candidate transformations as potential matches; it selects based upon a combination of a "fitness" measure and the best inlier root mean square error (RMSE). In cases where there are multiple valid solutions due to in-plane symmetries, the solution that has the highest value will be selected.

1.1.4 Computational Burden in Point Cloud Registration and Processing

Due to the large amount of data contained in point cloud files, point cloud file processing is computationally intensive (Qin et al., 2022). Many registration algorithms such as ICP and RANSAC use near neighbour search. Optimization techniques such as kd-trees are commonly used to optimize the computational burden associated with such operations. Even though optimization techniques have improved the performance of optimized versions of algorithms, such as optimized RANSAC and optimized ICP; performance remains highly dependent on noise content in the data set (K. Yang et al., 2024). Similarly, deep learning algorithms offer enhanced robustness relative to traditional algorithms. However, deep learning algorithms require GPU resources and therefore limit their applicability in field environments (L. Qi et al., 2022). Therefore, computationally efficient approaches that balance registration accuracy, segmentation robustness, and deployment feasibility on commonly available computing platforms are warranted. Rather than relying exclusively on either traditional geometric algorithms or deep learning approaches, hybrid solutions that selectively combine both paradigms offer a promising direction for practical Scan-to-BIM applications.

1.1.5 Automation of Tolerance-based Construction Quality Assessment

The geometric basis that allows for comparison of the as-built condition with the designed model is provided by Scan-to-BIM. Nevertheless, a correct construction quality assessment is obtained from the interpretation of measured deviations from the established acceptance limits and not from the geometric differences between the as-built condition and the design model (Sheik et al., 2022). Since the acceptable tolerances depend on the construction context, the mere numerical value of a deviation is often insufficient. As an example, a difference of 50 mm is generally acceptable for earthworks; however, it would likely be unacceptable for structural steel. Therefore, tolerance-based BIM evaluation tools will be needed to convert measurement variability in building geometry to useful quality conclusions. In this context, the USIBD developed a "Level of Accuracy" (LOA), which identifies acceptable variations among the physical building elements and their corresponding digital representations. Ultimately, comparisons made between measured deformation and established LOA values provide objective ways to evaluate construction quality for each individual building component. However, these assessments are also based on the quality of all previous processing steps as well, that is, robust pre-processing techniques, accurate registration methods, reliable object extraction/segmentation processes, and error-robust algorithms for measuring element-level deviations to minimize registration and/or measurement noise effects.

1.2 OBJECTIVES

1.2.1 Problem Statement

As digital construction has been developed through advances made in both 3-Dimensional Laser Scanning (3DLS) and BIM, there still remain many inefficiencies and issues with how these technologies are being utilized in regard to the integration between design models and the as-built environment. The current Scan-to-BIM process requires significant amounts of manual or semi-manual effort in terms of the registration and quality assessment phases (Chen & Cho, 2018; Park et al., 2026).

The segmentation stage provides one of the major bottlenecks in this process because it is responsible for differentiating individual structural components of an object so that semantic interpretation can occur. Deep learning methods are able to obtain high levels of accuracy, but they often need access to large data sets and require very powerful computing resources (GPUs). This limits their use for applications requiring field deployment (Guo et al., 2021; L. Qi et al., 2022).

Sensitivity to noise, ambiguity due to repetitive structural elements, and overall lack of reliability in real world environments are some of the challenges faced by traditional registration methods including ICP, RANSAC, feature based approaches (Tsige et al., 2025; Zou et al., 2023).

Given the previous information, the primary problem examined in this research is that there remains a need for integrated Scan-to-BIM workflows that can be efficiently deployed on limited-resource hardware as part of a complete solution to the Scan-to-BIM process. The major components of the Scan-to-BIM process include initial registration (matching scan data to existing building models), segmentation of objects from scans, measurement of deviations at an individual object level, and evaluation of quality of both physical construction and generated BIM models based on tolerances.

1.2.2 Research Question

To fill the gap between the increasing automation needs of the construction industry and the computing limitations of this sector, this study presents the primary research question: How can a computationally lightweight, end-to-end Scan-to-BIM framework be designed, implemented, and validated for construction and BIM model quality assessment? The following sub-questions are defined below:

1. Registration: How do FGR and RANSAC compare in terms of registration fitness and computational efficiency when applied to building-scale point cloud alignment in the context of the proposed framework?
2. Segmentation: How should classical geometry-driven operators and pretrained deep learning architectures be combined to maximise element detection and segmentation quality, and to what extent does each approach remain sufficient for downstream quality assessment?
3. Quality Assessment: How efficient and reliable is the proposed framework in terms of accuracy, computational performance, and robustness for construction and model quality assessment?

RQ2 involves two related examinations. The first evaluates four traditional segmentation algorithms (normal filtered-graph based segmentation, adaptive growing bounding boxes, RANSAC plane extraction, and DBSCAN) in terms of their performance on CPU. A second analysis evaluates three deep learning models (PointNet++, SpU-Net and PTv3), again comparing their performance but this time on a GPU because all of these models have been trained to be used on GPUs. Likewise, Research Question (RQ 3) evaluates the classification of deviation severity for each measurement compared to the specified tolerances for the respective part types.

1.2.3 Research Objectives

The primary purpose of this research is to develop, implement, and test a lightweight end-to-end Scan-to-BIM system for assessing building geometric quality through comparison of point clouds with BIM models of those buildings using consumer-grade hardware (CPU and GPU). More specifically, the goals are:

1. To create a modular seven-step pipeline integrating ingestion and transformation of data into coordinates, BIM processing, pre-processing, coarse-to-fine registration, segmentation, quality assessments (model and construction), and exporting and reporting into one fully automated workflow.
2. To evaluate and test two different registration techniques (FGR versus RANSAC as global operators within the candidate sweep method) and four traditional segmentation techniques and three deep-learning based segmentation techniques to determine the best possible technique for each step given the limitations of being lightweight.
3. To develop a contextualized tolerance classifier that can map geometric errors to an engineer's rating of engineering severity based upon element type and applicable industry standards (ISO 4463-1, EN 13670, EN 1992-3) and to assess the impact of tolerance profiles on verdicts.
4. To demonstrate full end-to-end computing capability against the constraints set forth in the opening of Chapter 1 and to generate integrated BIM outputs - PSet annotated IFC4 model, IfcGroup for scheduling, E57 point cloud with deviation-coloured points, and CSV schedule files that report back assessment.

1.2.4 Research Scope and Delimitations

The objective of this study is the creation of a lightweight Scan-to-BIM system to perform automatic geometric registration and assess the quality of construction through the use of point clouds, BIMs, etc. Four areas define the technical scope of this study: data processing, registration, segmentation, and point-based geometric deviations analysis. The overall scope of this study involves:

1. Designing and testing a coarse-to-fine registration pipeline based on a combination of the Fast Global Registration (FGR) algorithm, ICP together with geometry-based structural constraints improving registration robustness.
2. A comparative analysis of classical segmentation techniques (RANSAC, DBSCAN, Graph-Based, Adaptive Bounding Box) and deep learning architectures (PointNet++, SpU-Net, PTv3) using three

publicly annotated benchmark datasets (S3DIS, ScanNet, and Rohbau3D) to assess segmentation accuracy, robustness, and computational performance.
3. Developing a context-aware deviation classification system to assess construction quality at the element level against type-specific tolerance standards.
4. Emphasizing the computational feasibility of these solutions to be executed on commercially available hardware, specifically consumer-grade CPUs and, where required for deep-learning inference, a mid-range GPU.

The boundaries of this study can be defined by the following limitations:

a) Hardware. The entire geometric pipeline - ingestion, preprocessing, registration, traditional segmentation, deviation analysis, quality assessment and export—operates solely on a CPU. Benchmarking of the deep learning models presented in section 4.5.3 and the inference stage of the SpU-Net portion of the hybrid cascade were executed using an NVIDIA GeForce RTX 3070 Ti with CUDA. If the implementation cannot access a CUDA-enabled GPU (it will select the available hardware automatically), then it will fall back to using the PyTorch CPU backend. However, since this study did not measure CPU-based inference times, it is important to note that all deep learning network evaluations are based upon their performance relative to one another.
b) The learned components. For comparison purposes, deep learning models have been used as comparative benchmarks and in addition the S3DIS trained SpU-Net has been incorporated into the hybrid cascade for clutter removal and semantic conflict resolution. Therefore, they are not stand-alone alternatives to the traditional segmentation pipeline. In addition, all networks have been used in a zero-shot capacity, i.e., with no fine-tuning or additional training beyond the original pre-training. Fine-tuning to construction domain data has not been performed as there does not exist a sufficiently large construction annotated dataset; it is suggested as future work in section 5.5
c) Ground Truth. Manual segmentations of neither of the two case-study point clouds have been done. Segmentation ground truth is derived from the three public benchmark datasets that were independently created by others prior to this study. As such, case-study segmentation performance results are reported as element detection rates (i.e., the percentage of modelled elements with an extracted segment) instead of as per-element intersection over union (IoU).
d) Sensitivity to Parameters. One voxel resolution and one neighbourhood configuration have been used for each of the methods under test. This limits incidental parameter tuning, although operator-specific sensitivities and implementation differences may still affect performance. Automatic adjustment of parameters is outside of the scope of this project.
e) Validation by Experts. There has been no investigation of inter-rater reliability. The consistency of the framework's verdicts has been assessed through the validation status system (Section 3.9); run-to-run repeatability testing of element-level verdicts is identified as future work in Section 5.5.

1.3 METHODOLOGY

This study utilized a hybrid empirical-computational approach to provide both complete and validated responses to the research inquiry. Specifically, this research was completed using a quantitative approach in geomatics and computer science, that being, big data spatial analysis. A multi-stage computational pipeline called coarse-to-fine registration was applied and adapted with reference to geometric abstraction and segmentation methods discussed by Lin and Golparvar-Fard (2020).

Coarse-to-fine registration is well-recognized as scalable and reliable when working with large amounts of LiDAR data. It allows for reducing the complexity associated with large-scale data sets while increasing the scalability and reliability of the registration process. At a high level, the framework consists of data acquisition and representation, preprocessing, coarse-to-fine registration, element

segmentation, element-level deviation analysis, construction quality assessment, BIM model quality assessment, and automated reporting. The implementation details of each stage are presented in Chapter 3. The methodology integrates principles from civil engineering, geomatics, computer vision, and computationally efficient point cloud processing.

1.3.1 Data Acquisition and Representation

Data collection represents two condition models. The first model is an as-designed BIM model represented in IFC. An IFC format provides the BIM model's three-dimensional geometry, as well as its semantic metadata, such as coordinates and orientation. Geometry is first identified by each element type in the IFC, then converted using an explicit mesh generation process to create a boundary representation (B-Rep) representation of the BIM as a point cloud: First, vertices (x, y, z) and triangular faces are created, and these are combined into a single triangle mesh. The resulting mesh is cleaned (normals are calculated for the mesh and any geometric error is removed), and the surface is evenly spaced with 200,000 sampling points to create an artificially constructed BIM point cloud. This point cloud is subsequently referenced for comparison against the original scan (Chen & Cho, 2018).

The second model consists of obtaining point cloud data collected during the survey scan. Unlike many other forms of sensors, laser scans do not rely on a single sensor tool. They may utilize multiple types of sensor tools depending on their application, such as Terrestrial Laser Scanning (TLS) which uses static sensors providing highly resolved data, to Mobile Laser Scanning (MLS) which utilizes mobile sensors which provide low-resolution data and may be coupled with a Simultaneous Localization and Mapping (SLAM) system mounted on Unmanned Ground Vehicles (UGVs), or quadruped robots (Stop-and-Go Scanning) (Park et al., 2026). In this study, the point cloud used is the survey results that have been provided by BuiltCoLAB and the author.

1.3.2 Pre-processing & Down-sampling

The survey scan produces an enormous sized file that contains millions of points. These large files require a huge amount of computing power. In addition, the surface reflection and other non-structural objects like excavators, rock piles and people can introduce significant amounts of noise into the point cloud. Therefore, it is necessary to implement a preprocessing phase to decrease the amount of data generated while cleaning up the data prior to the next steps (Dong et al., 2025; Mengiste et al., 2024). First, the point cloud's density is standardized through voxel-based subsampling or voxel grid filtering. Through this process, all of the points in each of the cubic cells are replaced with a single representative point and the number of points in the cloud is decreased from millions to an appropriate number for use with lightweight algorithms. Afterward, per-point geometric features are calculated for both clouds as Fast Point Feature Histograms (FPFH). These geometric features will be used to facilitate registration in subsequent processes. The particular filter and feature parameter values are documented in chapter 3.

1.3.3 Initial Alignment

In order to establish a basis from which to register the scanned data with the BIM, it is important that both datasets have some form of common reference. If there exists an inconsistency regarding their respective references, then a preliminary coarser scale registration will be performed so as to position the two sets of data within each other’s proximity. This provides the registration process with a well-defined starting point for registration. See chapter 3 for further details on how this registration occurs.

1.3.4 Coarse-to-Fine Registration

The initial coarse alignment process uses a candidate set for global registration, primarily FGR and a RANSAC based registration on the generated features. As opposed to registering each point

individually, this method registers representative feature points and therefore allows for an efficient extraction of significant features like wall corner locations and column centroids (Zou et al., 2023).

Subsequently the most effective transformation from the candidate set is selected and iteratively refined through fine registration with ICP. ICP was accelerated utilizing a data-structure that enables fast nearest neighbour searches such as a k-d tree (Fu et al., 2021; Park et al., 2026).

In terms of the formulation of the ICP, a point-to-point ICP formulation is preferably utilized compared to a point-to-plane ICP formulation due to its increased robustness against the large variations in density of points and remaining noise typically present in site-scans. In contrast, a point-to-plane ICP formulation requires reliable normal estimates of surfaces.

The candidate set, the fitness function used to determine if a transform can be accepted and the function that determines the best candidate are all further explained within Chapter 3.

1.3.5 Segmentation

The segmentation process is critical in the Scan-to-BIM process because it enables the separation of raw unprocessed scan data into the architectural and engineering component parts that make up a building structure. The primary purpose of segmentation in this project was to remove unwanted material from the scan data (noise, clutter, etc.), identify regions with insufficient scan coverage (occlusions) and assign each part of the building to a category so that it could be compared against its original design.

As opposed to segmenting images of scenes in nature, there exist many obvious patterns in architectural scans such as flat planes, similar structures, and perpendicular relationships. Despite having many of these common characteristics, architectural scans also have problems inherent in them such as noise, missing pieces (occlusions), and similarities to other elements in the scan, all of which can limit how well cluster-based algorithms can perform on them (Grilli et al., 2017; Perez-Perez et al., 2021; Zoumpekas et al., 2021).

Due to these limitations, a hybrid strategy is applied; geometric based grouping of point cloud data by vector normal consistency is used to identify planes (walls, floors, etc.) with DBSCAN and Graph-based utilized for outlier identification as a way to assess an alternative approach where no prior knowledge of how many clusters exist (Hahsler et al., 2019). Additionally, a deep learning segmentation Model was evaluated to examine its ability to accurately segment point clouds without reference to BIM and whether this technique could serve as an initial screening process prior to conducting a BIM guided Quality Assessment.

1.3.6 Deviation Computation

Following a successful registration of the point cloud, geometric quality analysis can begin. Geometric quality analysis consists of examining the deviations of all elements in the point cloud relative to their counterparts in the BIM model. When evaluating deviations, the points defined within the point cloud are compared and evaluated against surfaces in the BIM model (Anil et al., 2011). The distance between each point in the point cloud and each corresponding artificial point on the surface of the BIM model is computed (Chen & Cho, 2018).

Measured deviations combine the effects of acquisition noise, residual misregistration, sampling and segmentation with any actual difference between construction and model geometry. The framework does not independently estimate these contributions. It reports the mean, maximum, standard deviation and RMSE of the deviation distribution, together with a status indicating whether segmentation required a fallback.

1.3.7 Integrated Quality Assessment

The pipeline has performed two quality assessments. These have been based upon two different assessment objects, which correspond with the two objects identified in the title of this thesis.

Construction Quality Assessment (CQA) is used to assess the physical construction represented by the scan. For each segment in the pipeline CQA uses a point-to-plane measurement to determine how far each of the as-built scan data points were from their reference BIM surfaces. Each segment is classified as either compliant, minor, moderate, or major depending on how much variation existed between the average deviation for each segment and the established threshold limits for the specific type of segment being evaluated under the governing execution standard for that particular project.

Model Quality Assessment (MQA) is used to evaluate the digital models of the pipeline. MQA provides a rating system on a 0-100 scale for each individual element based on the relationship between the measured deviation for each element versus an element-specific tolerance. In addition to the rating system provided by MQA for each element, MQA also assigns a letter grade to each element and includes a separate engineering pass/fail criteria for each element. This engineering pass/fail criteria will fail any element that exhibits a maximum local deviation greater than the element-specific tolerance, or a root mean square error greater than 1.5 times its tolerance, regardless of whether its mean deviation meets the tolerance criteria.

In general terms, these two modules differ in that CQA treats the model as the reference and determines if the physical structure conforms to it. Conversely, MQA assumes the alignment exists and determines if the model represents it accurately at the intended level of accuracy.

Additional information regarding all elements listed in the tolerance tables and associated classification logic can be found in Chapter 3.

1.4 ORGANIZATION

In order to organize the structure for this research, this thesis report has been organized into five chapters as follows:

Chapter 1 - Introduction contains the context which leads to the motivation and formulation of the research question along with the related investigations, defines the scope and methodology that will be used.

Chapter 2 - State-of-the-art reviews of laser scanning and geometrical data acquisition, point cloud preprocessing, registration, classical and machine learning-based segmentations, and computational efficiency are covered as well as a comparison to established tolerance limits for deviation analysis. The review concludes with a summary of findings from existing literature; identifies four research gaps that together form the basis of the developed framework; and places the framework in relation to these identified gaps.

Chapter 3 - The seven steps of the Lightweight Scan-to-BIM pipeline are defined as follows: Data Ingestion and Coordinate Reconciliation (stage 1); BIM processing and artificial reference cloud generation (Stage 2); Pre-processing (Stage 3); Coarse-to-fine registration with candidate selection (Stage 4); best-candidate element segmentation using a DL-first hybrid cascade (Stage 5); construction and model quality assessment (Stage 6); Export Package (Stage 7).

Chapter 4 - Case study and evaluation presents the application of the framework to two case studies of real world buildings (a reinforced concrete hydraulic reservoir and a multi-storey university building); it shows the quantitative results obtained from each stage of the process (registration, segmentation, assessment of construction quality and quality assessment of BIM model) and analyses how errors

propagate and how sensitive the entire process is to noise. Finally, it evaluates if the entire process can be carried out on standard commodity hardware.

Chapter 5 - Conclusion summarizes the main conclusions reached during the development of this thesis; answers the research question formulated and all associated sub-investigations; lists all contributions made by this work; makes honest reference to all limitations of this work (noise sensitivity, spatial resolution limits and potential error propagation due to registration) and finally proposes possible directions for subsequent research.

2
STATE OF THE ART

The advancement of digital technology has transformed the way that quality inspections and construction are conducted. The volume of data generated by these digital technologies has grown exponentially, exposing the limitations of traditional inspection procedures, which are time-consuming and prone to inaccuracy (J. Kim et al., 2025). New inspection processes have been developed to resolve these problems and to create opportunities for decreasing human error in the quality control inspection processes at construction sites. Laser scanning is one of the possible new inspection processes for reducing human error in quality control inspections at construction sites. However, laser scanning generates a considerable quantity of information, therefore creating a corresponding large requirement for computing power. As a result, numerous complicated tasks must be completed when transferring this generated data, which allows for a considerable opportunity for both human error and the time required to complete the inspection (Perez-Perez et al., 2021).

Additionally, as a result of the vast volume and unstructured format of point cloud data, the requirements for storing, processing and utilizing computing power represent a number of substantial technical barriers to producing high-quality Scan-to-BIM models. The workflow includes several phases: pre-processing, registration, segmentation, and extracting geometric features. Most of these phases currently require the manual involvement of skilled inspectors. Therefore, the greater complexity of the workflow represents an increased probability for human error in addition to delaying the finalization of the inspection (Rashdi et al., 2022). In addition to these factors, the quality of the point cloud data, specifically the point density, completeness and level of noise in the point cloud data directly affects the accuracy of subsequent analysis of the point cloud data, such as detecting deviations, evaluating flatness or assessing structural integrity. A lack of complete or noisy point clouds may produce inconsistent or incorrect assessments of the construction quality that highlight an essential void in the present workflows (Rashdi et al., 2022).

Therefore, researchers are increasingly focusing on developing lightweight Scan-to-BIM workflows capable of handling large volumes of point cloud data while providing accurate enough assessments for confident determination of construction quality and making informed decisions regarding projects (Abreu et al., 2023; Rashdi et al., 2022).

This chapter provides an overview of the current developments in all the phases of a Scan-to-BIM Quality-Assessment Workflow. Section 2.1 describes how laser scanning and photogrammetry can be used for the acquisition of as is data. Section 2.2 presents methods for processing and downsampling of point clouds. The section 2.3 explains how different registration methods can be applied to the point clouds. In section 2.4 segmentation methods are reviewed. A discussion about computational efficiency is presented in section 2.5. A review of existing Scan-to-BIM Workflows and Automated Quality

Assessment is given in section 2.6. Finally, a synthesis of the literature is provided in section 2.7 and the identified research gaps are motivated as the main purpose of this research.

2.1 LASER SCANNING AND AS-IS GEOMETRIC DATA ACQUISITION

The point cloud representing an as-is geometric condition for a construction project, represents the beginning point in this process flow and its quality directly affects the subsequent steps (registration, segmentation) of the entire workflow. This part of the paper describes the two families of technologies used to create these point clouds: active laser scanning and passive photogrammetry. The data produced from collecting spatial data using laser scanning is referred to as a point cloud. The point cloud includes x, y, z coordinates and may include other forms of attribute data such as laser reflection intensity values and colour information in the RGB spectrum (red, green, blue) when captured through photogrammetry or other methods (Z. Liu et al., 2024). Factors that impact the quality of point cloud data include density distribution, accuracy and the level of noise present within the point cloud. As previously mentioned, these factors directly affect the registration process. Overall, field geometry acquisition can be completed using one of two distinct methodologies; they are either through passive photogrammetry or through an active laser scanning methodology.

2.1.1 Digital Photogrammetry and Limitations in Construction

Photogrammetry is a method used to create three-dimensional representations from multiple images with overlapping views utilizing algorithms such as Structure from Motion (SfM), and Multi-View Stereo (MVS). A major advantage of photogrammetry is its cost-effective equipment requirements and flexibility for use in the field through traditional camera systems, Unmanned Aerial Vehicles (UAVs) and smartphone systems. This makes photogrammetry particularly well-suited for documenting and preserving cultural heritage sites and large-scale outdoor environments (Remondino et al., 2014).

Although photogrammetry has significant advantages over other methods for capturing a scene, there are also several disadvantages when applying photogrammetry to the task of measuring construction quality. As previously mentioned, SfM relies heavily on detecting and matching visual characteristics across image pairs with descriptors such as Scale-Invariant Feature Transform (SIFT) and Speeded-Up Robust Features (SURF). In most cases, construction sites have numerous surfaces that provide very limited visual information due to their homogeneity. Examples include texture-less, flat surfaces composed of pre-cast concrete panels, uniform interior surface finishes, and the large number of straight-line steel structural members. As a result, construction sites generally provide poor sources of corresponding visual characteristics for SfM-based reconstruction algorithms which can lead to reconstructed models that are either under-defined or distorted (Pepe et al., 2026). Furthermore, in order to produce accurate scale measurements from a photogrammetric model, some form of absolute reference system needs to be applied through ground control points (GCP). Photogrammetric models are susceptible to illumination dependent errors and generally do not offer the same level of geometric accuracy as laser scanning systems, especially at the millimetre tolerance levels expected for construction quality assessments (Anil et al., 2013; Bosché, 2012).

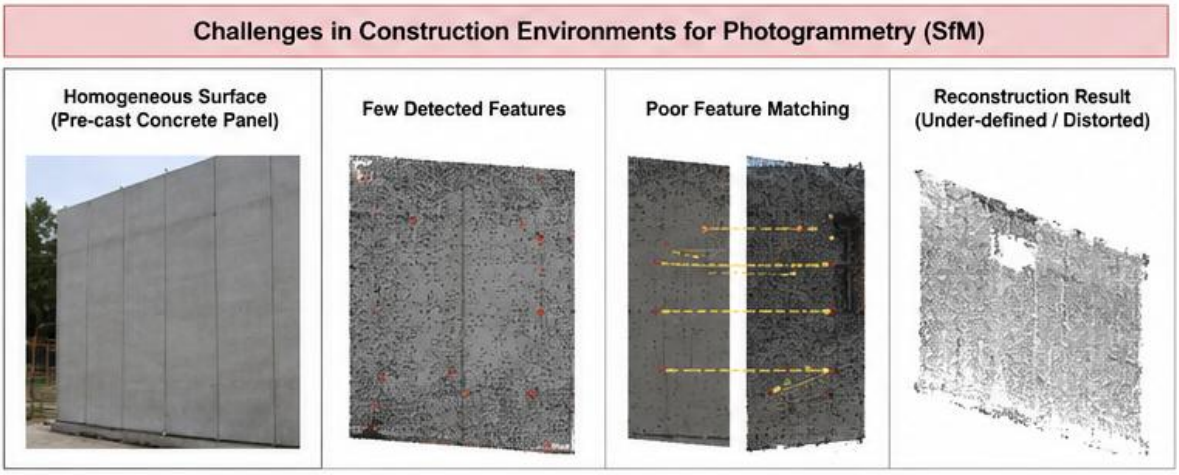


Figure 2.1 Challenges of photogrammetry in construction environments

Therefore, this dissertation will utilize LiDAR derived point cloud data as the primary data source. Due to the fact that laser scanning provides highly reliable geometry independent of lighting conditions, laser scanning is considered to be the best suited technology for the precise deviation analyses that are central to the proposed framework.

2.1.2 Light Detection and Ranging (LiDAR) Technology

The preceding discussion highlighted the limitations of photogrammetry based on passive sensors. Active sensing technologies represent an alternative for overcoming the limitations. The most common active sensing technology used in three-dimensional modelling is light detection and ranging (LiDAR). A LiDAR unit uses a time-of-flight (ToF) or phase shift technique to measure the time delay caused by returning laser pulse energy reflected back from an object's surface to calculate absolute distances. Most commercial LiDAR units also collect the intensity of returned pulse energies along with RGB colour information (Pedro & Sanhudo, 2021).

Because LiDAR devices generate their own electromagnetic illumination sources, they function independently of environmental lighting conditions. Therefore, LiDAR devices operate reliably in low-light or total darkness. Additionally, LiDAR devices are capable of detecting surfaces having very small amounts of visual texture, including concrete walls. These properties enable LiDAR to serve as the best option for gathering spatial data in construction settings (Pedro & Sanhudo, 2021).

LiDAR technology is classified into two types based on their operation: terrestrial laser scanning (TLS) and mobile laser scanning (MLS).

2.1.2.1 *Terrestrial Laser Scanning (TLS)*

Terrestrial laser scanning (TLS) represents a type of static scanning device that operates while stationary. The scanning device is usually mounted on a tripod for support. Many construction companies employ TLS tools such as the Leica ScanStation, RTC360, or the FARO Focus Series to acquire precise geometric data (Rashdi et al., 2022). TLS has established itself as the gold-standard for data acquisition via its ability to produce highly accurate measurements (submillimeter to centimetre-scale) at hundreds of feet range with relatively low noise levels.

However, despite TLS's exceptionally high geometric accuracy, there are significant limitations associated with its operational efficiencies. TLS is characterized by a slow rate of data collection due to the sequential nature of its acquisition process. To perform scanning properly, the operator must physically relocate the scanning instrument multiple times, calibrate its new location, and establish overlapping scans to facilitate subsequent registration. Unfortunately, construction sites often feature

large pieces of machinery and scaffolding obstructing the field of view, creating substantial occlusions which result in areas with no data being present in the final point cloud (Abreu et al., 2023). Along with longer than desired scan durations, this impacts both construction activities and cost. Thus, TLS will be much less useful for daily progress monitoring and/or quality assurance requirements.

2.1.2.2 *Mobile Laser Scanning (MLS)*

Static TLS has some major drawbacks, including the need for time-consuming setups at multiple stationary positions to perform complete scanning. Therefore, it is reasonable to introduce Mobile Laser Scanning (MLS), which performs active scanning. This can be achieved through a number of different configurations. For instance, there are small handheld scanners similar to the GeoSLAM ZEB Horizon and Leica BLK2GO, lightweight backpack-mounted scanners, and even LiDAR modules that have been mounted on UGVs (robotic ground vehicles), UAVs (drones), and quadruped robots.

MLS works using a combination of the SLAM algorithm (Tan et al., 2025), a method that enables the scanner to identify both its location in space at all times while creating a map of the surrounding environment. The SLAM system generates the trajectory in real-time using IMUs (inertial measurement units) and LiDAR measurements. It accomplishes this by utilizing data from the IMU to track movement and orientation, and then by matching the LiDAR measurements over time (a technique called scan matching). Therefore, there is no need to create maps using external reference points, unlike in TLS; instead, the scanner identifies its own location during the mapping process.

When using an automated quadruped robot equipped with LiDAR and SLAM, scanning time can be reduced by 42% compared to traditional manual scanning methods (Tan et al., 2025). As a result, due to the ability to move about freely, MLS is typically easier to utilize within narrow corridors and/or other restricted access locations, such as utility installation corridors and confined spaces where the utilization of a tripod is impractical.

2.1.3 Justification for Lightweight Algorithms Due to Inherent Data Flaws

While increased speed of acquisition provided by SLAM-based MLS systems provides several benefits, they also provide a drawback: decreased geometric quality (data fidelity) (Pepe et al., 2026). When compared to TLS, which produces highly accurate data that is highly consistent across datasets, MLS produced data tends to produce lower-quality point clouds and present many obstacles. Literature indicates that MLS data presents greater amounts of noise than TLS, i.e. apparent thickness in thin features such as walls, among others. Additionally, MLS is prone to accumulating positional error (trajectory drift), particularly when utilizing these systems along long paths with minimal geometric variability, e.g. straight corridors. Finally, MLS produced point clouds exhibit varying degrees of non-uniformity of point densities, depending upon the operator's movement speed while scanning (Pepe et al., 2026).

Due to these inherent issues, such as noise, occlusions, and resolution inconsistencies, advanced algorithms are necessary to ensure successful spatial registration and construction quality control (CQC) for processing MLS-produced data, as this data may not always be clean or optimal. However, if an overly sensitive registration algorithm requires high computational power just to handle noisy data while maintaining MLS's primary advantage as a rapid acquisition system, it would undermine one of its main benefits.

As a result, the development of computational approaches to algorithms will require that they are not only accurate but also robust - robust against noise and/or data imperfections - and computationally efficient. This results in a need to develop lightweight algorithms that are capable of minimizing memory

requirements so that they may be implemented on equipment with limited capabilities (Abreu et al., 2023).

2.1.4 Comparative Assessment: Accuracy, Coverage, and Practical Constraints

Geometric accuracy, spatial coverage, acquisition speed, processing power and practicability influence the selection of three-dimensional data acquisition methods in construction. As a result, digital photogrammetry and LiDAR have been the most commonly employed methods for Scan-to-BIM workflows, construction monitoring and geometric quality assessments. Both technologies aim at creating virtual representations of real-world environment based on their respective sensing principles. However, there are many differences between them concerning accuracy, robustness under various field conditions, computing effort and costs.

Table 2.1 Comparative assessment of as-is data acquisition technologies

Technology	Geometric Accuracy	Speed & Coverage	Cost & Robustness	Suitability For This Framework
Photogrammetry (SfM / MVS)	Moderate accuracy; dependent on lighting, texture, and calibration	Fast, low-cost, and suitable for large-area coverage	Inexpensive but sensitive to lighting and low-texture surfaces	Suitable for large-scale mapping, but not for millimetre-level construction QA
Terrestrial laser scanning (TLS)	High accuracy with low measurement noise	Slow acquisition requiring multiple scan stations	Expensive and requires skilled operators; lighting-independent and highly consistent	Ideal for precise deviation analysis, but less practical for frequent monitoring
Mobile laser scanning (MLS)	Lower accuracy than TLS due to drift and noise	Fast SLAM-based acquisition with wide coverage	Moderate-to-high cost; robust but noisier than TLS	Matches this framework by enabling rapid acquisition despite lower data quality

Digital photogrammetry generates three-dimensional geometry from multiple overlapping two-dimensional photographs using feature matching and SfM algorithms. Advantages of digital photogrammetry include a lower cost of the necessary equipment and a higher degree of freedom regarding the type of cameras used (e.g. drones, smartphones) or the way in which data is captured. Therefore, it is especially advantageous for large-scale outdoor documentation and heritage preservation applications where cost efficiency plays an essential role (Remondino et al., 2014). Image-based reconstruction provides dense spatial data in combination with additional attributes like texture and colour that facilitate visual inspections as well as semantic interpretations.

Although digital photogrammetry has some advantages, it is limited in the context of construction quality control and Scan-to-BIM workflows. For example, the accuracy of the reconstructed geometry primarily depends on the quality of the photographs taken, the illumination conditions during the capture process, the correct calibration of the cameras, the amount of overlap between individual photographs and the amount of texture found on surfaces (Fassi et al., 2013). Surfaces made out of

shiny material, with repetitive textures or other elements like transparent objects or low-contrast areas tend to generate unreliable feature correspondences resulting in missing parts of a complete model. Furthermore, photogrammetric point clouds typically show less geometric precision than laser-scanned point clouds when it comes to dimensional verifications on a millimetre level for quality control purposes in construction (Bosché, 2012). In addition to this, construction sites present changing environmental conditions such as different light intensities, shadowing effects, moving machinery or partially occluded objects causing unreliability in image-based reconstruction processes.

On the contrary, LiDAR technology directly measures distances by determining the round-trip time or phase shift of emitted laser pulses. Since LiDAR technology does not depend as much on texture or ambient illumination conditions during the acquisition process as photogrammetry does, LiDAR technology enables more consistent geometric measurement results across diverse construction environments (Tang et al., 2010). Due to their ability to create highly dense point clouds with an accuracy ranging from millimetres to centimetres, terrestrial laser scanning systems (TLS) are particularly suitable for dimensional quality control, deformation analyses as well as as-built documentation (Bosché et al., 2015). Due to this high geometric accuracy LiDAR technology has established itself as one of the leading sensing technologies for modern Scan-to-BIM workflows.

The accuracy advantage of LiDAR technology is mainly relevant in terms of applications including deviation analyses between as-designed BIM models and as-built site conditions. Registration errors on a small scale may lead to misleading deviation measurements during construction quality assessment. Bosché (2012) showed that the use of laser-scanned point clouds is possible due to their sufficient precision for automated dimensional compliance checks in construction environments. Anil et al. (2013) also pointed out that reliable geometric quality control demands precise measurements of distance from points to surfaces. These types of measurements are typically more accurately obtained through the application of laser scanning technologies.

Furthermore, LiDAR systems have some disadvantages. Firstly, terrestrial laser scanners are more expensive than photogrammetric systems and usually need specific hardware and skilled users. Secondly, laser-scanned point clouds contain very large amounts of points per building project (up to hundreds of million samples), thereby increasing the computational complexity for storing, pre-processing, registering and segmenting the large number of points generated (Tang et al., 2010). This increased computational complexity can be further exacerbated if iterative algorithms such as ICP or feature extraction are applied to resource-constrained hardware.

As well as the theoretical limitations mentioned above there are some very significant practical limitations related to the "visibility" of features captured by TLS systems. TLS systems are essentially static sensors which will only measure the surface that is exposed to from the scanning station, i.e. surfaces that are in line-of-sight of each scanning position. As such, it is possible that a variety of construction site layouts (e.g. narrow corridors, scaffolding, temporary items of plant/equipment etc.) may necessitate use of several scanning stations to ensure full coverage of the area of interest. Although increasing the number of scanning stations will provide better overall coverage of the area of interest, it will equally increase the total time taken to collect all the data, reduce the ease with which this data can be registered together and produce redundant data. This problem has been partly mitigated using Mobile Laser Scanners (MLSs) and SLAM based systems where improved mobility and faster collection times can be achieved but usually at a loss of geometric precision compared with TLS systems.

There are also distinct differences in terms of coverage between photogrammetry and LiDAR technologies. In general, photogrammetry allows for wider spatial coverage at lower costs, especially when used in conjunction with unmanned aerial vehicle (UAV) platforms for larger outdoor areas.

However, due to various factors including environmental conditions and specific imaging techniques employed during collection, photogrammetric results can exhibit varying degrees of geometrical quality. Conversely, LiDAR produces consistently stable and precise geometric measurements but usually at longer collection periods and narrower coverage areas per scanning location. Therefore, the choice of sensor type is generally determined by the specific requirements of the project, e.g. required degree of accuracy, available access to the target area, available computing resources and potential operational constraints.

From a Scan-to-BIM perspective, the practical limitation of acquisition accuracy is extended to include the additional computational implications arising from processing the acquired point cloud data. The increased density of LiDAR derived point clouds improves the geometrical fidelity of the models produced but greatly increases the processing time required for tasks such as registering the point clouds together, segmenting out individual components from the combined point cloud, finding neighboring points in the combined point cloud and performing deviation analysis. Qin et al. (2022) found that the computational complexity associated with processing point clouds grows rapidly with increasing dataset size, thus requiring efficient spatial indexing structures (such as k-d trees or octrees) for fast querying capabilities. Additionally, lightweight processing frameworks are becoming increasingly important to many construction professionals who rely on conventional CPUs located on portable field devices and do not possess GPU equipped high-performance workstations.

Therefore, the equilibrium between acquisition accuracy and computational efficiency represents one of the most pressing issues currently being investigated in modern Scan-to-BIM research. While highly accurate sensor systems are advantageous only if the resultant point cloud datasets can be processed in a timely manner consistent with typical construction workflows, deep learning-based approaches have shown great promise for registration and segmentation tasks but are limited in practice by reliance on GPU intensive computation and availability of large amounts of training data (C. R. Qi et al., 2017). Thus, much recent research emphasizes lightweight geometry driven methods that aim to achieve acceptable accuracy while minimizing computational overhead.

Another practical consideration relates to compatibility between LiDAR generated point clouds and BIM environments. Due to preservation of accurate spatial coordinates, LiDAR generated point clouds typically exhibit greater compatibility with geometric quality assessment workflows as they allow for direct point-to-surface comparisons against BIM elements. Conversely, photogrammetric reconstructions, although visually appealing, may contain non-uniform distribution of points and/or noise characteristics that make subsequent registration and semantic segmentation tasks difficult. Hence, despite its higher cost and computational burden, LiDAR remains the technology of preference for precision-oriented construction quality control applications.

A second area of debate involves whether the difference in accuracy among low-cost acquisition technologies and those used with surveys will continue to be significant enough to affect how well buildings are constructed. Some believe the differences are diminishing to the point where they become irrelevant in practice. For example, (Pepe et al., 2026) have demonstrated that the use of low-cost photogrammetry and SLAM pipelines, which have been compared to terrestrial laser scanning (TLS), can produce sufficient point clouds for many Scan-to-BIM applications. In addition, (Tan et al., 2025) found that the time required to collect data was approximately 42% less when using a mobile platform equipped with SLAM technology; however, there were no detectable losses in terms of the quality of the collected data. Therefore, some feel the advantages associated with increased speed, increased mobility, and reduced costs from acquiring data at lower resolutions far exceed what would be considered an acceptable loss in resolution for typical monitoring activities. On the other hand, others feel that any tolerance-based deviations must be based upon very high precision, since any decrease in

precision is likely to result in the incorrect classification of true construction problems as measurement artifacts. Rather than addressing this issue, the current research views it as a limitation: therefore, it takes imperfect and rapidly collected data as input and processes them robustly while explicitly acknowledging that processing cannot guarantee compensation for acquisition errors or justify relaxed acceptance thresholds.

In summary, comparative evaluation between photogrammetry and LiDAR illustrates that no single acquisition technology is optimally suited for all applications in construction industry. Photogrammetry provides relatively low-cost flexibility in terms of coverage but lacks geometrical precision and robustness under adverse environmental conditions. On the other hand, LiDAR provides superior geometrical accuracy and stability for construction quality assessment but imposes significant hardware and computational burdens. The trade-off associated with choosing between photogrammetry and LiDAR technology clearly motivates continued development of lightweight Scan-to-BIM frameworks capable of quickly processing large density point cloud data while maintaining sufficient geometrical fidelity to support reliable quality assessments. Accordingly, this study places emphasis on developing computationally efficient processing strategies that are capable of operating on consumer-grade hardware with modest CPU and GPU requirements while utilizing the geometrical reliability provided by LiDAR-derived point clouds.

2.2 POINT CLOUD PRE-PROCESSING AND DOWN-SAMPLING STRATEGIES

Pre-processing is a key component of the Scan-to-BIM workflow. It ensures that the raw point cloud data generated through scanning is available for segmentation, feature extraction, registration, and quality evaluation processes. Typically, the raw point cloud contains hundreds of thousands to tens of millions of points. These points may have varying spatial distributions and contain high levels of noise. Outliers may arise due to the physical limitations of sensors or other environmental factors encountered during field data collection. As a result, these complexities may increase the computational burden associated with performing geometric analyses.

To mitigate these issues, a pre-processing strategy is required. A pre-processing strategy would ideally eliminate redundant data, preserve important geometric characteristics, and enhance the quality of spatial representations while preserving all relevant information. Pre-processing is particularly important within a lightweight Scan-to-BIM framework since the objective of the system is to provide efficient computation while ensuring adequate accuracy for validation purposes and assessing construction quality.

Typical techniques described within the literature include downsampling, estimating normal vectors, filtering based upon surface orientation, and removing noise and outliers (Lyu et al., 2024; Y. Wang et al., 2013).

2.2.1 Voxel Grid and Random Down-Sampling

Downsampling involves reducing the total number of points contained within a point cloud to facilitate faster processing times and lower memory usage. Two popular downsampling methods include Voxel Grid downsampling and random downsampling.

Voxel Grid methods involve partitioning the three-dimensional space into small discrete cubes called voxels. Then each voxel is replaced by a single representative point, usually the centroid or the point closest to the centre of the voxel. This technique creates a more uniformly distributed sampling pattern and helps to preserve the overall spatial organization of objects. (Lyu et al., 2024) describe voxel-based methods as being highly effective at reducing the complexity of large point clouds as they allow conversion of irregularly sampled data to a regularized grid format which facilitates efficient processing.

Although voxel-based methods offer several advantages including creating a more uniform sample pattern, preserving the global spatial structure of objects, and facilitating efficient processing there are some disadvantages associated with these techniques. Specifically, conventional voxel-based methods tend to lose local detail at edges and sharp features. If the voxel size is too large, significant amounts of geometric information are lost. On the other hand, if the voxel size is too small, suboptimal reductions in data are achieved. Consequently, determining the appropriate voxel size is a critical factor in implementing these types of systems in practice (Lyu et al., 2024).

An alternative to voxel-based methods is random downsampling where a random selection of points is selected from throughout the dataset. The benefits of using random downsampling include its relatively simple implementation and fast execution time. Unfortunately, random downsampling does not take into consideration either the spatial distribution of points or the relative importance of local details, resulting in a disproportionate elimination of detailed regions. Additionally, (Lyu et al., 2024) demonstrated that random downsampling produces fewer stable simplifications than those produced through voxel-based methods, particularly when dealing with datasets that exhibit large densities of variation in point clouds.

Both methods are anchored within an even broader discussion regarding how many degrees of abstraction a quality-oriented pipeline can support. Efficiency based approaches to work often advocate for the need to aggressively reduce data sets as being both necessary and relatively safe when considering semantic tasks, as long as some form of global structure has been maintained in order to make large scan data sets processible; adaptive methods have recently emerged to demonstrate that sample rate can be varied with respect to local complexity to maintain key geometric information at significantly lower cost (Dong et al., 2025). An opposing approach, based upon the control of geometric quality, suggests that any fixed rate reduction in data will necessarily discard the critical microscopic geometry (edges, sharp corners, and thin members) on which measuring deviation from centimetre level measurements is dependent. Therefore, a cloud optimized for fast segmentation may already be insufficient for accurate compliance evaluation (Z. Liu et al., 2024). Thus, the argument is not one of which method is better in theory as it pertains to downsampling, but rather what resolution floor does the specific downstream task impose. In other words, the current framework uses voxel-grid downsampling for computational tractability and explicitly reports the resulting resolution floor. At v = 0.10 m, it cannot resolve deviations below approximately 50 mm; therefore, millimetre-level tolerance verdicts are indicative rather than conclusive.

2.2.2 Normal-Vector Estimation and Filtering

Normal vectors represent the local surface orientation of points contained in a point cloud. Local surface orientations play critical roles in segmenting points into different components, classifying shapes based on their topological properties, reconstructing surface geometries, identifying planar elements, and analysing planarity in building models.

One common method for estimating normal vectors utilizes Principal Component Analysis (PCA) on the local neighbourhood surrounding a point. PCA analyses the neighbouring points of interest to determine which plane fits best to the local surface. The eigenvector having the smallest eigenvalue is subsequently utilized as an estimate for the normal direction (Y. Wang et al., 2013).

The primary shortcoming of traditional methods occurs in areas containing sharp features, corners or boundary transitions between surfaces. When points from both surfaces are included in the analysis region used for PCA, then normal estimates will likely yield inaccurate results. To remedy this situation, (Y. Wang et al., 2013) proposed an adaptive method for estimating normals that automatically identifies

the optimal regional neighbourhood for each point to utilize for accurate normal estimations in point clouds exhibiting sharp features.

This adaptive approach yields improved performance under noisy conditions or outlier contamination and maintains accurate surface orientations near edges. Within the context of construction applications this functionality is critical since many structural elements such as columns, beams, walls and structural connections possess clear geometric distinctions.

Normal vectors are employed in conjunction with orientation estimation in order to filter out points with significantly deviant orientations from the predominant normal direction. Thus, normal vector filters serve as an additional cleansing operation, enhancing quality prior to extracting BIM elements.

2.2.3 Noise Removal and Outlier Detection

In terms of noise and outliers in a point cloud, they refer to the distortion of point locations from an object's actual surface and the points that don't relate to a valid object and are located away from the general distribution of the point cloud. These two issues may develop due to sensor failures; the reflection of materials or the occlusion of surfaces by other parts of the object. Noise and outliers will damage the process of estimating the geometry of an object, affect the quality of segmenting the object into parts, and reduce the reliability of measuring the distance between a BIM object and its measured counterpart. As such, noise elimination and outlier identification are both critical processes in pre-processing.

Statistical outlier removal compares each point’s mean distance to its nearest neighbours with the distribution of those mean distances. Radius-based methods count neighbours within a specified radius and identify points with insufficient local support.

Recently, researchers have started applying deep learning to clean up point clouds. In particular, (Y. Li & Sheng, 2023) described one type of single stage point cloud cleaning model. The primary advantage of this method is that it can remove outliers and perform denoising using a unified algorithm. (Y. Li & Sheng, 2023)'s model has two additional capabilities beyond traditional algorithms: it uses adaptive downsampling, allowing it to remove irrelevant points and maintain relevant geometric information, and it employs a feature compensation module to compensate for lost information when removing noisy points.

In summary, the study by Y. Li and Sheng (2023) demonstrated that these new deep learning-based methods are able to generate significantly cleaner and more uniformly distributed point clouds compared to traditional methods and improve on existing computer vision tasks, including surface reconstruction and object classification.

One major limitation of deep learning-based methods is that their computational complexity restricts their applicability in real-time settings and in environments with limited memory resources. For instance, this limitation becomes critical in scenarios where thousands of scans must be converted into BIM models in a short time frame without access to substantial computational power.

2.3 POINT CLOUD REGISTRATION METHODS

Core registration techniques are critical analytical processes required after obtaining point cloud raw data from the site to align multiple datasets into a common coordinate system. The registration process involves converting all data sets from the various sensors' local coordinates to a global coordinate system (Sheik et al., 2022), usually an IFC-based BIM design model. Without the registration process, the scanned data could never be compared to the designed model since they would reside in different coordinate systems.

Mathematically, the registration process attempts to find the rigid body transformation parameters. This is done via a transformation matrix composed of two primary elements; rotation (pitch, roll and yaw) and translation (x, y and z axis shift) (Sheik et al., 2022). Therefore, the registration process seeks to establish how much a point cloud needs to be rotated and shifted so that it will properly align with the reference model.

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \tag{2.1}$$

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

Where:

T = A 4 × 4 homogeneous transformation matrix

R = A rotation matrix representing orientation, where R ∈ *SO*(3)

t = A translation vector representing position displacement, where t ∈ $\mathbb{R}^3$

0 = A 1 × 3 zero vector

r_ij = Elements of the 3 × 3 rotation matrix R, representing the directional cosines

t_x, t_y, t_z = Elements of the 3 × 1 translation vector t, representing displacements along the X, Y, and Z axes, respectively.

Due to the precision of establishing this transformation being extremely important, errors in determination can significantly affect further analysis. If an as-built scan is slightly off or skewed relative to the as-designed model, a detection system might interpret the slight deviation as a construction defect rather than simply a misalignment (false positive).

Prior to this development in technology, operators were required to perform manual registrations. They selected reference points through subjective means (via software) between the point cloud and 3D models. Surveyors installed target devices in the field (checkerboard or reflective spheres) prior to scanning (Hajian & Becerik-Gerber, 2010). Both methods provided limited efficiency due to the requirement of operator involvement, contradicting the objectives of automation in Industry 4.0.

As a result of these technological constraints, researchers have developed algorithms for automated registration methods in geomatics. Typically, registration methods follow a two-step (coarse-to-fine) procedure. Coarse registration to provide an initial estimation of the dataset alignment, followed by fine registration to refine the dataset alignment in greater detail (Lin & Golparvar-Fard, 2020). The overall two-step coarse-to-fine registration paradigm followed in this review is summarised in the figure below.

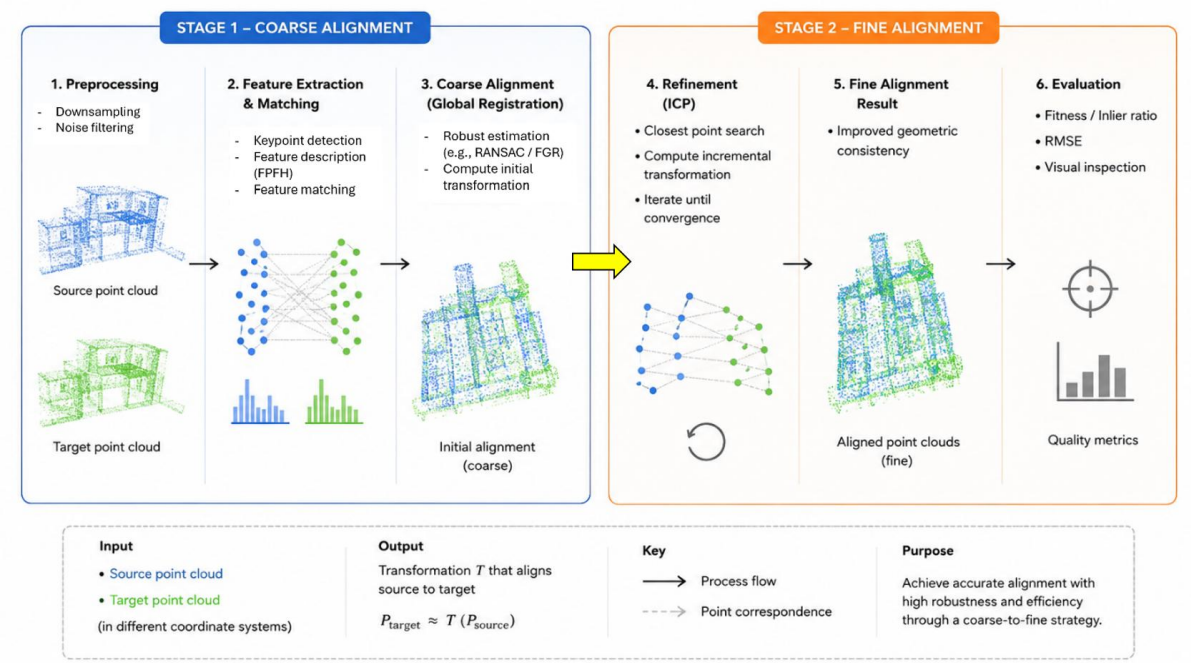


Figure 2.2 Coarse-to-fine point cloud registration pipeline

Source and target point clouds (input)

↓

Pre-processing: voxel down-sampling ; normal estimation ; FPFH feature extraction

↓

Coarse (global) registration: feature-based operator (RANSAC or FGR) → initial alignment

↓

Fine (local) registration: ICP refinement (point-to-point / point-to-plane)

↓

Refined rigid transformation aligning source to target (output)

2.3.1 Coarse Registration (RANSAC)

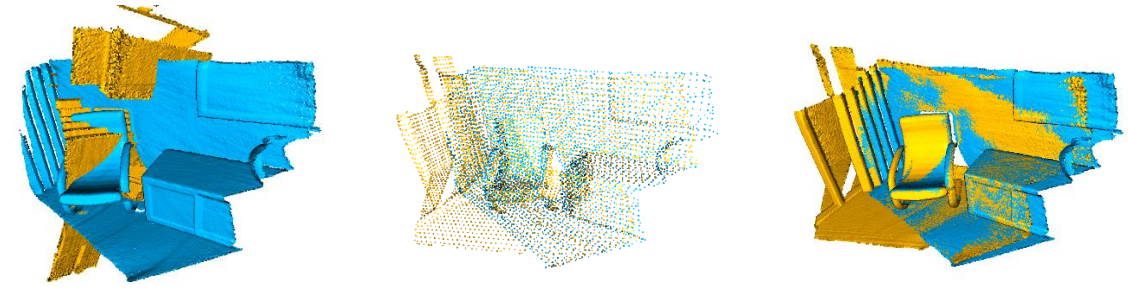
The purpose of coarse registration is to develop an initial guess of the position and orientation between two non-aligned data sets. At this point, the system must be capable of identifying an initial spatial relationship between two-point clouds with no assistance from user input (initialization free) (Lin & Golparvar-Fard, 2020). Therefore, the algorithm must identify how the two data sets can be aligned based on their geometric forms alone.

One common technique used for global registration is based on extracting local features. A descriptive representation of each point in the point cloud is produced by transforming each point into a numerical description of its surrounding geometry including surface shape and relationships between neighbouring points. A commonly used descriptive representation is fast point feature histograms (FPFH), which produce a 33-dimensional histogram describing angle and surface normal relations and shape descriptions (Rusu et al., 2009). Following this step, once FPFH representations are generated for

both data sets (for example: a BIM model and a scan), the next task is to locate corresponding points having similar features.

Similarity in point pairs is determined using feature matching, to which random sample consensus (RANSAC) is applied as a statistical method for robust estimation (Fischler & Bolles, 1981; Rusu et al., 2009). RANSAC iteratively selects a few random point pairs (typically three) and applies them to calculate a temporary transformation that aligns the selected pairs. The selected transformation is then applied to the entire point cloud. The number of points agreeing with the transformation within a defined threshold (inliers) is calculated. This iterative process occurs numerous times (thousands) until the transformation producing the highest number of matches is identified.

Although RANSAC is widely used, there are several inherent limitations associated with RANSAC-based coarse registration methods. As RANSAC relies upon random sampling, repeated iterative processing may lead to excessive computation time, particularly when noise levels are high or overlap between datasets is low. With increased noise or reduced overlap between data sets, the probability of selecting accurate point pairs decreases dramatically resulting in either failure to produce a consistent solution or inefficient processing (Mahmood et al., 2020). These limitations have promoted research efforts into developing rapid, reliable alternatives to RANSAC-based coarse registration methods. The overall feature-based coarse-registration pipeline is summarised in Figure 2.3.

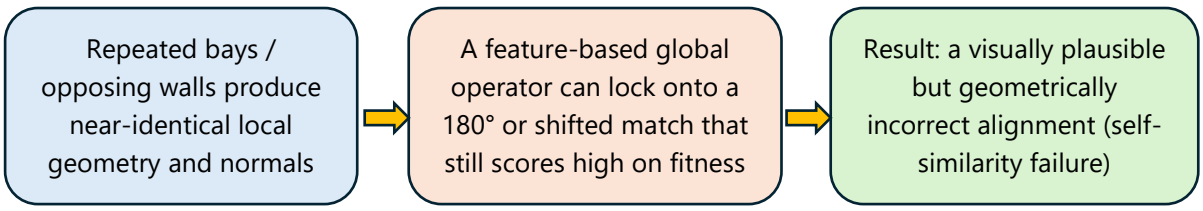


(a) Initial point cloud misalignment (b) FPFH feature correspondence (c) RANSAC alignment

Figure 2.3 RANSAC feature-based global registration pipeline

2.3.2 Limitation of RANSAC in Orthogonal and Self-Similar Structures

In addition to the previously discussed limitations, another area where RANSAC performs poorly includes orthogonal structures. The main reason for this poor performance is that RANSAC relies on random sampling, which becomes inefficient when the inlier ratio is low as in real-world scenes with high noise or outliers. One example of a scene where RANSAC would fail is a scene consisting of a number of buildings. Many of the structural elements of a building such as walls, windows, doors etc. are very similar at various locations throughout the building (Tsige et al., 2025). The self-similarity failure mode is illustrated schematically in the figure below.



Therefore, while the local descriptor for each feature is unique to that location, the value produced by the local descriptor could potentially be very similar for several other locations in the building. Due to this property, there are potential ambiguities in determining what constitutes correspondence in the

image registration problem. Since RANSAC does not account for this type of information, it is possible for RANSAC to select point pairs that appear to be corresponding based solely upon their local properties, whereas they may not actually be corresponding based on the structure of the environment (Quan et al., 2022).

A second limitation associated with using RANSAC is its reliance on randomization. When RANSAC is run repeatedly on a particular set of data, the results obtained from one iteration will almost certainly differ from those obtained from another iteration. The ability to repeat the same experiment over and over again is fundamental to scientific inquiry (Fischler & Bolles, 1981).

RANSAC was developed for robust parameter estimation in computer vision, including applications on real image data beyond just small-scale experiments and simulations. At larger scales, RANSAC becomes impractically slow. For example, the use of RANSAC requires a number of iterations that grows approximately logarithmically with desired confidence relative to the inlier ratio and minimal subset size. Given a dataset size of millions of points and high levels of noise, this can cause a significant increase in the number of iterations. Consequently, as the scale of experimentation increases, the probability that RANSAC will become computationally impractical for resource-constrained systems increases (Hu et al., 2017).

2.3.3 Fast Global Registration (FGR): Principles and Efficiency Gains

Another alternative to RANSAC that has been proposed to alleviate some of its shortcomings is Fast Global Registration (FGR). Instead of relying on repeated random sampling like RANSAC, FGR uses deterministic methods. Specifically, FGR uses all available feature pairs simultaneously regardless of how many false positives exist. Once all feature pairs are established, FGR uses direct minimization of a single objective function to find the best transformation to apply to the data (Zhou et al., 2016).

FGR differs from RANSAC by avoiding iterative sampling and instead using all correspondences with a robust penalty function called the Geman–McClure function in a single global optimization. The purpose of the penalty function is to minimize the effects that outliers have on the overall fit of the model. If a given feature pair contains a large error (indicative of an outlier), its contribution to the overall fit is minimized (Zhou et al., 2016).

$$\min_{R,t} \sum_{i=1}^{n} \frac{|Rp_i + t - q_i|^2}{|Rp_i + t - q_i|^2 + \mu^2} \tag{2.3}$$

Where:

p_i = The i-th source point in the initial coordinate system

q_i = The i-th target point corresponding to p_i

R = The rotation matrix applied to the source points

t = The translation vector applied to the source points

$||R\ p_i + t - q_i||^2$ = The squared Euclidean distance (residual error) between the transformed source point and the target point

μ = A scale parameter that controls the robustness of the estimator against outliers.

2.3.4 Iterative Closest Point (ICP)

The next step after coarse registration is usually fine registration or refinement. Fine registration represents the final stages of registration and aims to further refine the registration results. It involves registering the previously coarsely registered data sets into an even more precise fit. In general, it is desirable to achieve accuracy in the order of millimetres for a meaningful construction quality assessment (Bosché, 2010; Sheik et al., 2022). Fine registration is addressed via iterative optimization-based techniques, among which the ICP algorithm is probably the best known since its publication by Besl and McKay in 1992 (P. Li et al., 2020; Rusinkiewicz & Levoy, 2001).

In the classical ICP formulation, the so-called point-to-point strategy is employed. Hereby, nearest neighbours are identified between two-point sets, namely the source point cloud and the target or reference model. After establishment of correspondence between the points, the algorithm iteratively computes rigid body transformations that aim to minimize the Euclidean distances between the paired points (Cheng et al., 2018; Rusinkiewicz & Levoy, 2001). Based on this concept, nearest-neighbour correspondences are established; these need not be one-to-one, thus allowing direct geometric minimization of point distances.

This assumption is criticized as being unrealistic for real world scenarios, especially in construction sites where data acquisition is subject to sensor noise, occlusion and uneven spatial distribution. For example, Lidar based point clouds are characterized by non-uniform distributions while BIM models represent idealized versions of the built environment. This discrepancy between both types of data can lead to lower chances of correct correspondences and therefore can increase the chance of ambiguities during the pairing process and also influence the stability of the convergence (Chetverikov et al., 2005; Tam et al., 2013). Additionally, some researchers demonstrated that point-to-point ICP can get stuck in local minimums depending on how well aligned the initial position is (Pomerleau et al., 2013).

To overcome these restrictions other formulations were developed. These include variants of ICP with point-to-plane as opposed to point-to-point. By utilizing this strategy instead of computing the Euclidean distances between corresponding points, the distances from each scan point to the local tangent plane of the reference geometry are computed. Therefore, the optimization takes into account the geometric relation between a point and a tangent plane located at the closest surface area of the BIM model. Consequently, the dependency on exactly fitting pairs of points is reduced and the criteria for matching is shifted towards surface congruency.

A methodological perspective shows that this could be especially beneficial for environments with high amounts of planar components like walls, slabs and ceilings. In this type of environment, points will slide along the surface tangents when moving and this will have little impact on the error metrics. This property was shown to improve the convergence behaviour of point-to-plane ICP against point-to-point formulations under specific circumstances (Rusinkiewicz & Levoy, 2001; Segal et al., 2009).

However, it cannot be said that point-to-plane ICP is always better than point-to-point ICP. Some studies show better convergence speeds and higher robustness to noise in comparison to point-to-point ICP. On the contrary, other studies show that its performance heavily depends on the precision of surface normal estimations and whether locally planar geometry exists (Pomerleau et al., 2013; Tam et al., 2013). Additionally, when dealing with non-planar geometries or complex geometries, its benefits may be lost.

Beginning with these classic methodologies, there exists an emerging methodology based on a larger-scale philosophical argument regarding whether iterative geometric registration will remain in some form, or if it would be replaced by alternative methods based upon machine learning. Proponents of machine-learning based registration claim that data-driven registration remedies two major deficiencies

of ICP: sensitivity to initial conditions and propensity for local minima; (Mei et al., 2023), through the use of optimized learning of features, improves performance when noisy input data is provided, and (L. Qi et al., 2022), demonstrate that a low-weight learned descriptor can provide results equivalent to those achieved via traditional methods while greatly reducing the training time required relative to previously developed neural networks. Conversely, proponents of traditional geometric methods assert that the improvements associated with machine-learning based registration are dependent upon having representative data available during the training process and, due to dependence on high-performance computing resources (GPU), may suffer from degradation of accuracy when presented with out-of-distribution scene types including but not limited to cluttered or partially constructed site conditions. While ICP and its variants do not require training and operate in a predictable manner once an acceptable coarse seed is identified (P. Li et al., 2020); both families have their own unique applications. Machine-learning based methods offer the greatest improvement where significant amounts of representative data and access to high-performance computing hardware are available. Traditional refinement offers greater portability and interpretability within the CPU-bound environment with data limitations that exist during most on-site quality assessments.

Table 2.2 Comparison of point cloud registration methods

Method	Stage	Principle & initialization	Key strengths	Key limitations
RANSAC	Coarse (global)	Randomly samples feature correspondences to estimate the transformation	No initial guess required; robust and widely adopted	Slow for large datasets; inconsistent due to random sampling
FGR	Coarse (global)	Optimizes all feature correspondences simultaneously	Fast optimisation; repeatability depends on implementation	Depends on feature quality; a weak seed still requires fine refinement
ICP (point-to-point)	Fine (refinement)	Minimizes Euclidean distances between corresponding points	High registration accuracy after coarse alignment	Sensitive to initialization; may converge to local minima
ICP (point-to-plane)	Fine (refinement)	Minimizes point-to-surface distances using surface normals	Faster convergence on planar structures	Requires reliable normals; less effective on complex geometry

2.3.5 Spatial Indexing Structures: Kd-Tree and Octree for registration Acceleration

While the ICP algorithm achieves accurate registrations up to the submillimeter level, it has a significant drawback: it requires a lot of computational power (Zou et al., 2023). Research results show that at least 90 % of the ICP execution time is spent on finding the best matching pair of neighboring points (nearest neighbour search) (Zou et al., 2023), which is much less efficient than optimizing the transformation (Zou et al., 2023).

When nearest neighbours are searched without additional support and thus in naive linear scan mode, the computational complexity increases dramatically with each increase in the size of the point cloud. The application of such an approach may therefore be too costly in terms of both computation and memory usage for many large-scale LiDAR datasets (containing up to millions of points). In extreme cases, this could lead to system failure through freezing or crashes.

To counteract these problems, several research efforts have proposed the pre-processing of point clouds by means of spatial index structures (Park et al., 2024). These provide a topologically organized hierarchical representation of the point cloud by recursively dividing the 3D space according to spatial criteria (e.g., k-d trees and octrees) (Park et al., 2024). The spatial subdivision mechanisms of k-d trees and octrees are illustrated in figure below.

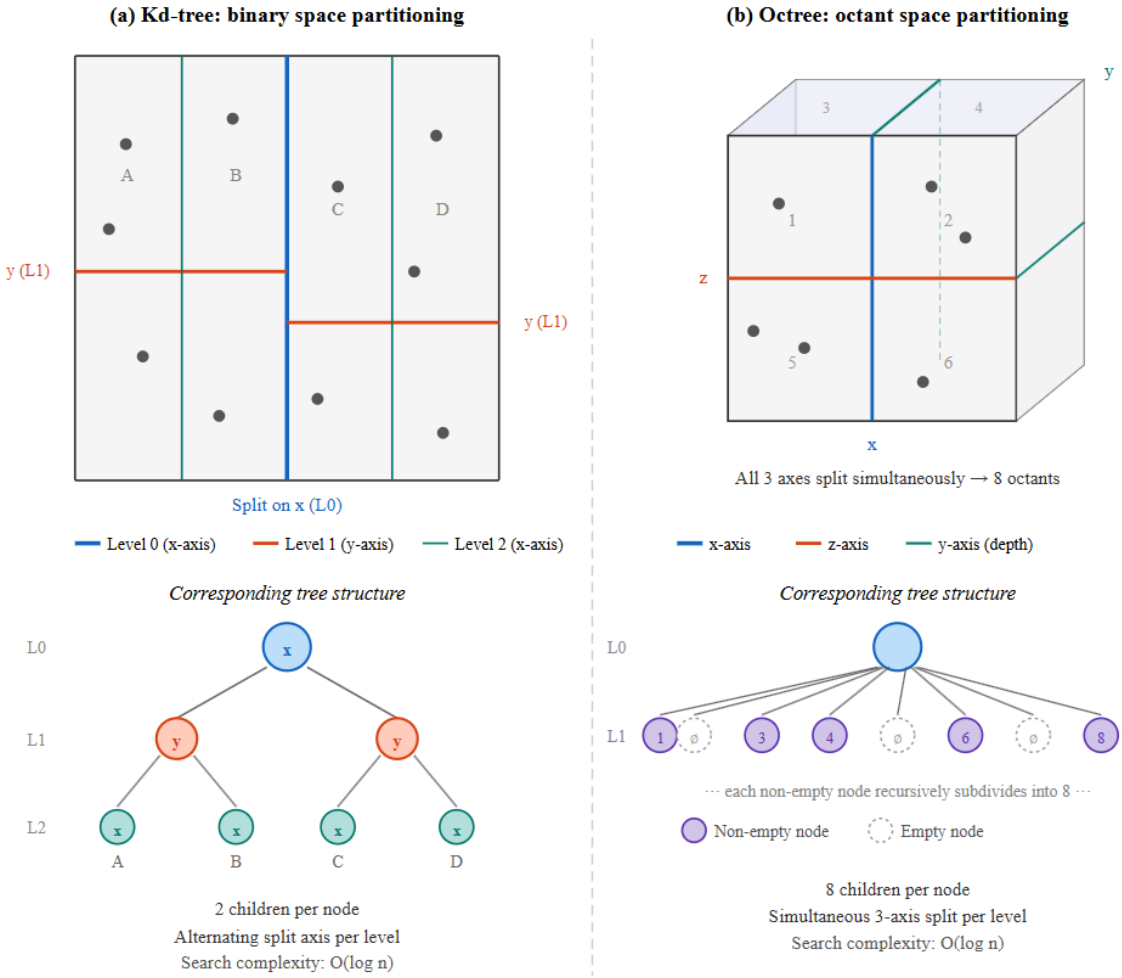


Figure 2.4 Illustration of kd-tree and octree spatial subdivision

For instance, the k-d tree divides the area of research by each axis as per established rules. While an octree divides the area of research into eight hierarchical sections when subdivided. Using both data structure types to reduce neighbourhood searches results in only searching through those data elements that have been identified as part of their respective neighbourhoods. This significantly reduces the size of the search space resulting in improved performance (i.e., faster execution times).

Spatial indexing reduces the cost of nearest-neighbour queries. For suitably distributed three-dimensional data, a k-d tree often gives approximately O(log N) query time, compared with O(N) for a linear scan; a full pass over N query points therefore remains more expensive than a single query. Actual speed depends on the data and implementation. Reported improvements in fine-registration time include those discussed by Lv et al. (2025).

Thus, combining ICP with spatial index structures such as kd-trees or octrees represents an important strategy for establishing an optimal trade-off between achievable accuracy and computational performance. This combination is especially interesting for implementing lightweight computations,

e.g., in mobile field applications with restricted hardware resources, such as in Scan-to-BIM synchronization or online digital twin processing.

2.4 POINT CLOUD SEGMENTATION APPROACHES

Point-cloud segmentation groups scan points into geometrically or semantically meaningful regions. In the framework developed here, element segmentation follows registration and assigns scan evidence to BIM elements for deviation measurement. The following subsections review region-growing, density-based, graph-based and learned segmentation approaches (Ntiyakunze & Inoue, 2023). Semantic clutter filtering before registration is a separate operation, described in Chapter 3.

2.4.1 Geometric and Region-Growing Methods

Geometric processing of images has traditionally been a widely used method of separating the "macro-structure" of a building in the construction sector. This type of geometric processing utilizes regional growing algorithms. The majority of regional growing algorithms are called "bottom-up," or "seed-based." Bottom-up algorithms start by selecting a seed point, then expand outward from this point as long as possible. The local expansion process is controlled through the verification of two properties of the neighboring points. Each property is verified separately before the neighboring points are added to the cluster. The first property checked involves verifying that the normal vectors of neighbouring points have sufficiently similar orientations. If the normal vectors satisfy the angular-consistency criterion, the neighboring points are added to the cluster. The second property checked is that all neighbours meet a certain criterion (distance less than a specified value). If a neighboring point satisfies both the directional property and the proximity property, the neighboring point will be assigned to the same group (Kavaliauskas et al., 2022).

Region-growing algorithms have been used to perform tasks in a variety of fields, but particularly in medical imaging and general image-segmentation applications. These algorithms were first introduced to the field of construction as an early implementation of geometry-based methods for segmenting point-clouds into BIM.

Spatial indexing supports efficient neighbourhood queries during region growing. Query cost depends on the structure, spatial distribution and number of returned neighbours; it should be distinguished from the cost of processing an entire point cloud.

There are some situations where trade-offs occur between the two goals of accuracy and performance. In many cases it is useful to improve data quality before attempting to segment very large collections of point-clouds. For example, improving the quality of a BIM generated from a point cloud through the application of preprocessing steps such as down-sampling of voxels or the removal of noise/outliers will improve the final product.

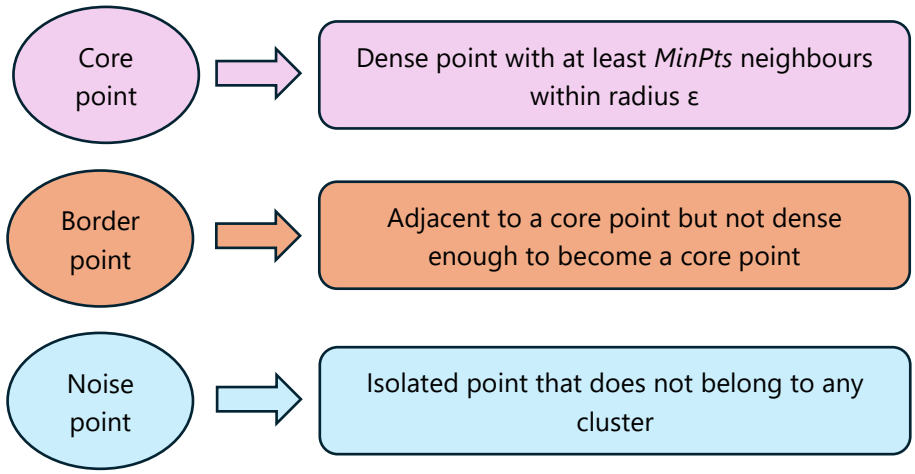
The three main factors that contribute to the widespread acceptance of geometry-based methods are as follows. First, geometry-based methods produce easily interpretable BIM models. Since region growing produces BIM models utilizing explicitly defined geometric parameters, users know exactly what each element in the final product represents (e.g., floor, ceiling, wall). Second, since geometry-based methods do not rely heavily on manually annotated data for training machine learning algorithms, they offer alternative options that are lower-cost compared to typical machine learning alternatives. Third, although geometry-based methods may not be as effective as some machine learning alternatives under actual working field conditions, they remain essential tools for producing BIMs. Specifically, geometry-based methods are especially useful as preprocessing tools. By significantly reducing the amount of unneeded information present in a larger point cloud prior to inputting a smaller point cloud into more sophisticated segmentation algorithms, geometry-based methods can

effectively reduce the size of large point clouds. Even though there are limitations to utilizing geometry-based methods independently in complex site conditions without employing other segmentation technologies, they serve as a necessary precursor step in developing BIMs.

While geometry-based methods have an advantage over machine learning methods in terms of ease-of-use and reduced costs associated with labelling training data, they also have several drawbacks related to performance. One such limitation is that edge-based methods generally resist noise worse than region-growing methods (Ghule AG & Deshmukh PR, 2012). Noise relates to any deviation from reality contained within the point-cloud data. There are many different sources of noise contained within point clouds, including: errors created by scanning equipment (e.g., laser ranging error), motion-induced errors (e.g., movement of scanner during scanning), and environmental factors (e.g., rain, snow etc.) (Yue et al., 2024). Another example of incomplete point clouds is missing data. Missing data occurs when elements within a scene restrict accessibility to a scanning device thus excluding those points from being recorded in a point cloud. As mentioned earlier, another limitation is sensitivity to differences in point density. Differences in point density arise when multiple scanners are employed or when a single scanner is placed at varying distances from objects during scanning. In cases where segments have similar densities along a scan-line or strip, segmentation results in either too much separation or too much coalescence of features along said line/strip.

2.4.2 Density-Based Spatial Clustering (DBSCAN)

To begin, DBSCAN works differently than Region Growing. Instead of defining and extracting geometric areas using similarities in surfaces, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm provides a totally unique methodology for analyzing point clouds, namely, spatially dense clustering. Generally, DBSCAN does not act as a primary semantic extractor in standard Scan-to-BIM methodologies. As opposed to this, DBSCAN is used during the process of instance segmentation once classes of objects have been somewhat separated (Z. Wang et al., 2022). The core, border, and noise point categories used by DBSCAN are summarized as follows.



DBSCAN determines whether a set of points forms a densely populated contiguous area (core points), and how many other points can be reached within a given neighbourhood size (ε) as determined by the minimum number of points required within that radius (MinPts) (Hahsler et al., 2019). One of the main theoretical benefits of DBSCAN over K-Means is that it can recognize clusters with shapes of any geometry without needing to know beforehand how many clusters will exist. It also has a built-in filter function that automatically rejects "noise" based upon the spatial location of isolated features. Therefore, DBSCAN is commonly recommended when separating multiple parallel MEP piping systems and/or when isolating individual structural columns from surrounding open space.

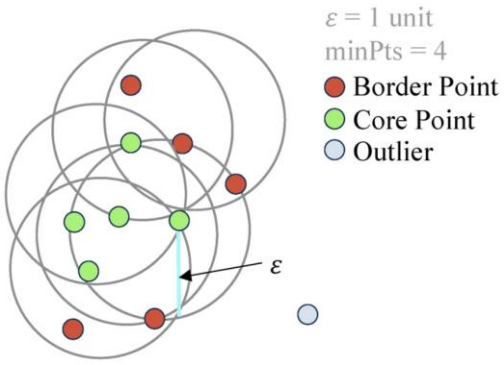


Figure 2.5 Visualization of DBSCAN clustering algorithm

Although DBSCAN was developed on solid mathematical foundations, DBSCAN suffers from a serious practical problem due to the way it fails to adapt to changing densities in construction point clouds. The most significant challenge posed to the academic community relates to the algorithms' inability to deal with varying levels of density. Since TLS and drone sensors measure decreasing amounts of point density as they get farther away from their central locations, it follows logically that applying a fixed radius parameter ε across the entire database would result in classifying all elements located at a greater distance from the sensor centre as noise and discarding them. At the same time, DBSCAN would fail to classify objects of differing classes as separate entities if they were to physically contact one another in high-density areas (K. Liu et al., 2024).

When it comes to dealing with complex as-built environments, there is no longer the possibility of utilizing DBSCAN directly because it treats intersecting 3-D planes as being part of a single cluster unit. A comparative study demonstrated empirically that a multi-view fusion-based method produces dimensional precision errors of less than 1.3% and F1 scores better than 99.5% in a precast element extraction task, far exceeding what would be obtainable through traditional methods like DBSCAN (Hua-Ping et al., 2025). These studies indicate that DBSCAN suffers from low accuracies and poor robustness in complex scene scenarios with occlusion (Hua-Ping et al., 2025).

2.4.3 Graph-based Segmentation

A key feature of graph-based segmentation techniques is that these represent point clouds as undirected graphs G = (V, E), where vertices V denote the points (or point groups) of the point cloud and edges indicate the spatial proximity of points (with associated edge weights describing how similar points are in terms of position, surface normal direction, colour etc.) (Felzenszwalb & Huttenlocher, 2004).

One of the most important benefits of graph-based methods over purely density-based methods like DBSCAN is that the spatial connectivity between the points is explicitly encoded; i.e., if two points are spatially close but belong to different planar surfaces due to e.g. a wall-floor junction then it is possible to assign them to different segments. A benefit of this property in building environments is that there exist many planar surfaces with sharp intersection angles.

Typical algorithmic methodologies applied to graph-based methods include spectral clustering, normalized cut and connected component analysis on k-nearest neighbour (KNN) graphs (Vicente et al., 2008).

In terms of lightweight Scan-to-BIM applications, one major trade off exists; namely, segmentation quality vs. computation time. Spectral methods provide theoretically optimal partitions of the graph; however, they require performing the eigen decomposition of the graph Laplacian, which has a poor

scaling with respect to the number of points. On the other hand, connected component analysis on a KNN filtered graph allows us to obtain a solution with a better scalability than the spectral methods and preserves the connectivity property with an acceptable computational load suitable for field deployment. This trade-off between segmentation quality and computation time motivates the use of graph-based operators that combine a normal-consistency filter with connected-component extraction on a filtered KNN graph, an approach revisited in the methodology of this dissertation.

2.4.4 Capabilities and Computational Cost of Deep Learning-based Segmentation

In comparison to classic BIM-guided segmentation pipelines using prior knowledge that are explicitly based on the geometry (for example, surface normal thresholds, bounding box heuristics, and curvature constraints), deep learning-based architecture uses prior knowledge derived directly from the raw point cloud data for extracting semantic objects. A representative example is the hierarchical encoder–decoder architecture of PointNet++, shown in Figure 2.6, which learns local-to-global features through successive set-abstraction and feature-propagation layers. Although the heavy reliance on explicit prior knowledge in classic approaches makes them very sensitive to parameter-tuning, their deterministic geometric boundary definition provides strong geometric boundaries. On the other hand, the potential disadvantage of deep learning-based architectures is that since there are no hard geometric constraints, they could have trouble maintaining the strict spatial fidelity required for downstream quality assessment (especially in terms of quality control at varying point-density levels under hardware constraints).

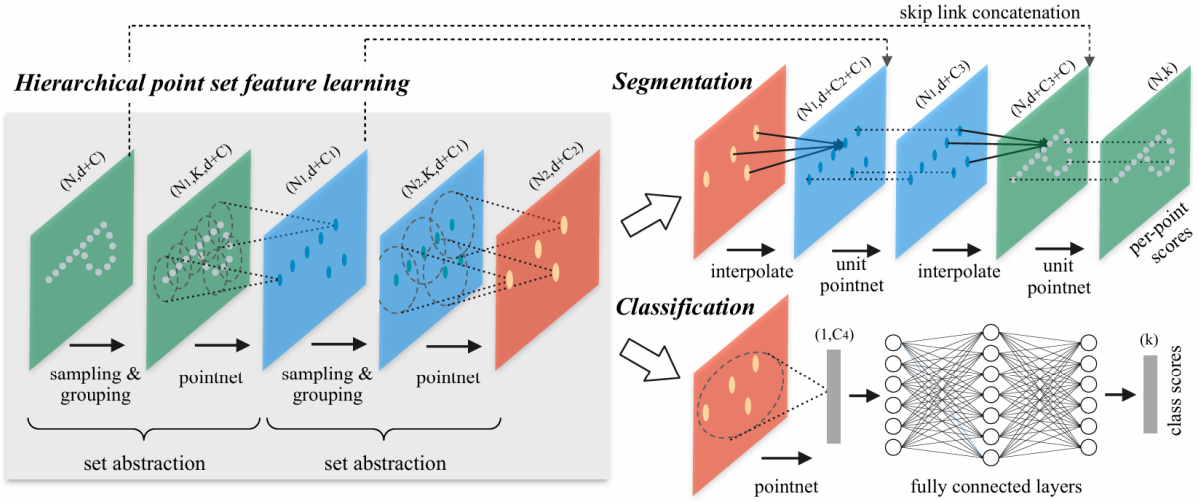


Figure 2.6 PointNet++ hierarchical set-abstraction (encoder) and feature-propagation (decoder) architecture for point-cloud segmentation

Deep learning's development in terms of processing point clouds started off with PointNet, which is a framework that processes point clouds without first converting them to voxels (Zoumpekas et al., 2021). Although PointNet was foundational, it did not effectively capture local geometric relationships. As a result, PointNet++ was created; providing hierarchical feature extraction through farthest-point-sampling (fps) (C. R. Qi et al., 2017). Supporters of PointNet++ state that hierarchical grouping enables effective identification of micro-geometric features and therefore local relationships found within buildings. On the other hand, opponents such as (Aditya et al., 2024) point out that fps is an enormous bottleneck for computing resources when working with large scale construction data sets. Therefore, when running on CPU-based devices where low overheads are crucial for lightweight programming, FPS scales poorly, rendering it impractical for use at the level of deployment in the field.

As a direct response to the limitations of processing points directly, sparse three-dimensional convolution neural networks (sparse u-net/SpU-Net), specifically those implemented via Minkowski Engine, present a new paradigm. SpU-Net uses a sparsity approach to convert the input point cloud into a sparse occupancy grid. In contrast to PointNet++, SpU-Net applies convolution operations exclusively to the occupied voxels. As far as efficiency goes, proponents such as (Choy et al., 2019) state that SpU-Net dramatically reduces both memory usage and computational loads relative to a dense three-dimensional grid. Nonetheless, from a viewpoint of construction quality assessment, voxelization has many detractors. Detractors argue that voxelization creates discretization errors inherently related to grids; thereby reducing spatial accuracy (Y. Wang et al., 2019). For example, if a planar wall is converted into a voxel grid, sub-millimetre geometric fidelity required to accurately compute deviations against a BIM model is lost. This has led to continued discussions concerning trade-offs between SpU-Nets efficiency and loss of geometric integrity.

Point Transformer V3 (PTv3) uses efficient serialised point ordering and patch-based attention to capture contextual information in point clouds (Wu et al., 2024). Its computational cost should not be described as dense all-pairs $O(N^2)$ attention over an entire scene. Runtime and memory depend on patch size, implementation and hardware, and are measured for the selected models in Chapter 4.

A second complication in evaluating architecture is the type of dataset being utilized to evaluate. In the past, models were evaluated on idealized indoor datasets such as S3DIS and ScanNet. Although SpU-Net and PTv3 demonstrate exceptionally high mIoU-scores on ScanNet, many researchers believe that these scores are misleading. As demonstrated by (Bai et al., 2023; Rauch & Braml, 2025), models that are trained on clean, densely populated data exhibit poor performance when they encounter the domain-gaps associated with construction site-data, i.e., data that includes large amounts of noise, extensive occlusion caused by heavy equipment, and wide variations in point-density, as seen in the Rohbau3D dataset. (Xiao et al., 2024) note that, without geometry-based priors to help guide them, many deep-learning models experience extreme difficulty distinguishing between a structural column and adjacent scaffolding material under chaotic environmental conditions.

In conclusion, although achieving excellent segmentation results may be sufficient for quality-assessment purposes, the true measure of success for Scan-to-BIM application should be determined based upon how well the segmentation maintains accuracy throughout the entire process. Incorrect classification can lead to disastrous consequences. For example, if a segmenter incorrectly identifies a group of points on a wall as a structural beam, then the calculated centroid of the extracted element will move. Since this will invalidate the comparison of the deviation against the planned BIM model, the segmenter will produce false positive results during model checking, thus destroying confidence in the automated system (Aditya et al., 2024). Thus, defining a "lightweight" framework must go beyond simply comparing model size and/or CPU-inference speeds; it must compare how accurately the model can extract geometric feature information without relying on heuristics found in traditional pipelines.

Table 2.3 Comparison of point cloud segmentation approaches

Approach	Principle	Key strengths	Key limitations / lightweight suitability
Region-growing (geometric)	Seed-based bottom-up expansion adding neighbours that satisfy normal-consistency and proximity criteria	Interpretable geometry; no labelled data; low cost; effective as a pre-processing / clutter-reduction step	Sensitive to noise, missing data, and point-density variation; limited on its own in complex scenes

Approach	Principle	Key strengths	Key limitations / lightweight suitability
DBSCAN (density-based)	Groups core points by neighbourhood radius (ε) and MinPts; isolated points are rejected as noise	Finds arbitrarily shaped clusters without a preset count; built-in noise filtering; good for instance separation (e.g., MEP, columns)	Fixed ε fails under the varying density of TLS/drone scans; merges intersecting planes; poor robustness under occlusion
Graph-based	Represents the cloud as a graph G = (V, E) with edges weighted by proximity, normal, and colour similarity	Explicitly encodes connectivity; separates surfaces at sharp intersections (e.g., wall–floor junctions)	Spectral partitioning scales poorly (eigen decomposition); connected component on a filtered KNN graph is the scalable compromise adopted here
Deep Learning (PointNet++, SpU-Net, PTv3)	Learns local-to-global semantic features directly from raw points, without explicit geometric heuristics	High benchmark accuracy (mIoU); captures semantic context in cluttered scenes	FPS / attention / voxelisation costs; slow CPU inference; domain gap on noisy site data; no hard geometric constraints for sub-mm fidelity

2.5 COMPUTATIONAL EFFICIENCY IN POINT CLOUD WORKFLOWS

2.5.1 Benchmarking Approaches: Processing Time, Memory, and Accuracy Trade-offs

Benchmarking and performance analysis of point cloud processing algorithms in construction IT must be strictly defined according to the objective: exact geometric deviation measurement. As point clouds are unstructured, have an extremely irregular structure, and are usually very large, determining whether lightweight DL models could substitute the parametric sensitivities of classical pipelines, there needs to be an evaluation of the relationship among the three main inter-related factors: processing time, memory use, and accuracy. The trade-offs among the main point cloud processing method families in terms of typical accuracy and computational cost are summarized in table below.

Table 2.4 Comparison of typical accuracy and computational cost among major point cloud processing method families

Method family	Typical accuracy	Computational cost
Deep-learning operators (e.g., PTv3, SpU-Net)	High	High - GPU, long inference
Classical geometric operators (region-growing, DBSCAN, FGR + ICP)	Moderate	Low - efficient on conventional CPUs
This framework - lightweight hybrid (target)	Competitive with deep learning	Target: low cost on consumer-grade CPU and GPU

Processing time includes all of the time needed to obtain semantic and geometric data: including pre-processing, inference, and post-processing. Classical BIM-guided processes have traditionally employed registration techniques (such as FGR) based on deterministic speed (Zhou et al., 2016). The process of transforming coordinates is resolved quickly with the minimal number of iterations possible for other older methods. Supporters of classical BIM-generating pipelines claim that once the parameters of the classical pipeline are optimized, heuristic segmentation will run relatively quickly on typical PCs. On the other hand, critics indicate that the manual time expended in optimizing distance measures and angle measurements for each new scan eliminates the computational savings realized through the classical pipeline's faster processing time (Gomaa et al., 2021). Conversely, although deep-learning models eliminate the need for handcrafted features, and their manual optimization, their processing time may be limited when running on hardware-restricted PC environments. (Yin et al., 2019) also emphasize that much criticism exists regarding the benchmarks currently being reported in literature for deep-learning approaches. Specifically, most authors report only the idealized GPU inference times for idealized datasets when they evaluate their approach's computational performance. These idealized results do not reflect what occurs when lightweight DL algorithms are deployed on typical construction computing equipment.

Memory use represents the amount of RAM consumed by the algorithm during execution. Large memory consumption comes from both high-precision floating-point arrays and complicated spatial structures such as k-NN graphs (Besl & Mckay, 1992). In addition, memory-efficient architecture designs such as SpU-Net have been often cited as examples of memory-efficient architectures. (Choy et al., 2019) also note that sparsity allows sparse tensor representation to significantly reduce the memory footprint when compared to either PointNet++ or classical graph-based representations of memory-intensive dynamic edge networks. On the other hand, proponents such as (Y. Wang et al., 2019) express concern about memory savings achieved through voxelization being a false economy if it comes at the cost of geometric integrity required for structural assessments.

Accuracy has two separate meanings when evaluated in terms of Scan-to-BIM quality assurance. First, Semantic Accuracy addresses element classification through metrics such as OA and mIoU (Weinmann et al., 2015). Second, Geometric Accuracy represents the spatial faithfulness of segmented points relative to the actual condition using RMSE and HD (Javadnia et al., 2017). The controversy surrounding academic research has now reached its peak here. Advocates of lightweight down-sampling (including voxel-grid down-sampling utilized before SpU-Net inference) suggest that preserving every single point is computationally expensive and unnecessary for general-purpose semantic classification (IEEE Staff, 2011). From the standpoint of geometric quality control however, many researchers disagree, indicating that aggressive down-sampling leads to permanent destruction of micro-geometry (Besl & Mckay, 1992). Therefore, if the spatial resolution falls below the minimum allowable tolerance threshold (i.e. 5 mm to 10 mm for construction deviation), then no matter how accurately semantically classified the final segmentation may be it is utterly inadequate for subsequent quality assessment of models.

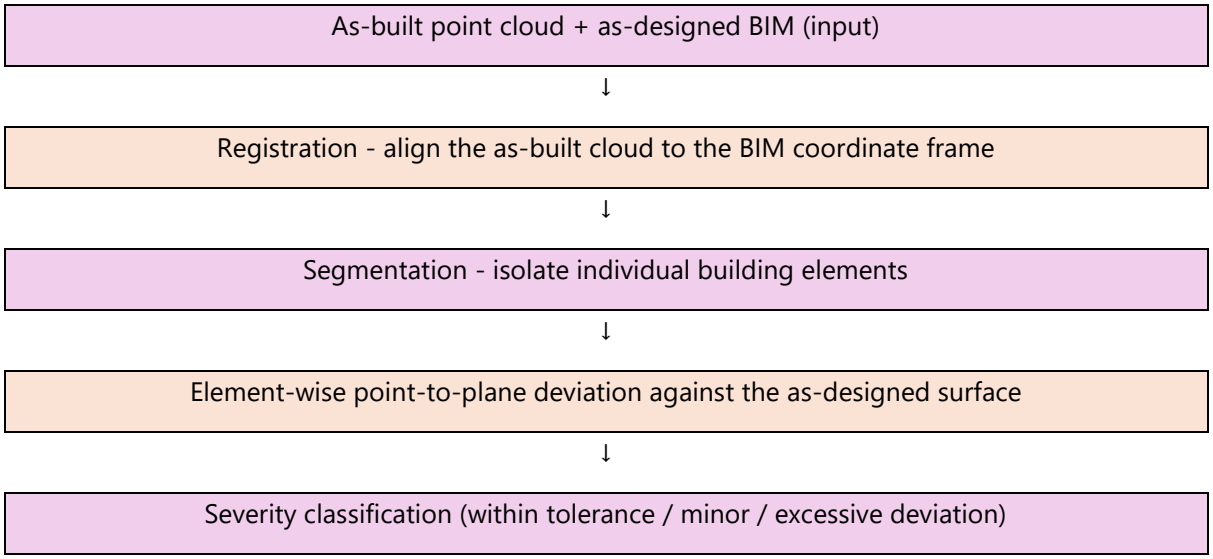
Additionally, existing metric benchmarks are inherited from the 2-D computer vision area and do not take into consideration the topology of BIM (Yin et al., 2019). For instance, an algorithm may achieve a high mIoU on S3DIS dataset by correctly classifying large flat surface areas like floor surfaces while incorrectly classifying complex MEP connection joints that are essential for detecting clashes and quality control.

Ultimately, assessing how well models such as PointNet++, SpU-Net and PTv3 can replace the parametric heuristics of classical pipelines requires a stringent benchmark. This benchmark should punish DL models that achieve rapid computation speeds at the cost of geometric accuracy necessary for reliable estimation of construction deviations.

2.6 SCAN-TO-BIM WORKFLOWS AND AUTOMATED QUALITY ASSESSMENT

2.6.1 Deviation Analysis (As-Built vs. As-Designed Comparison)

The core purpose of the concept "deviation analysis" is to assess the spatial differences between the as-designed BIM model, which is the conceptual basis upon which a building was designed, and as-built point clouds real-world representations of the completed building (Bosché et al., 2015). A shift away from small-scale sample-based measurement inspections and toward full-scale geometric inspections using millions of verification points across the complete building envelope (Bassier et al., 2020) is proposed as a method that could help define how successful deviation analysis may be in assessing construction quality (Bonduel et al., 2017). The end-to-end deviation-analysis workflow is summarised below.



Deviation analysis methods fall under two primary categories, those that compare two-point clouds, known as Cloud-to-Cloud (C2C) and those that compare point clouds against CAD/BIM models, known as Cloud-to-Model (C2M). Each type of comparison method operates differently, varies greatly in terms of computational resources required, and varies in terms of their accuracy.

For example, C2C methods first perform spatial registration between two-point clouds and then calculate the Euclidean distance from each as-built point to the nearest point in the as-designed point cloud (Bonduel et al., 2017). These types of methods are very efficient computationally and relatively easy to implement especially if used in conjunction with kd-tree data structures to speed up nearest neighbour searches. On the other hand, there exists a significant drawback to this method in that once the CAD/BIM model is converted into a point cloud representation, surface topology is no longer represented explicitly in the point set; semantic labels can be retained separately.

On the other hand, Cloud-to-Model (C2M) methods preserve a parametric representation or B-Rep of the CAD/BIM model. For example, each as-built point can be projected onto the nearest BIM surface and then orthogonal to it to find the distance between them. Because the object remains intact geometrically and semantically, this type of method provides a more accurate means of evaluating the surface quality of buildings. Moreover, additional methods, such as multiscale model-to-model cloud comparison (m3c2), improve robustness by computing deviations along local normal vectors, thereby yielding better results on rough or curved surfaces.

In general terms, C2C is superior in terms of its computational efficiency, whereas C2M is superior in terms of its representation accuracy. However, C2M requires more complex geometric processing, particularly when dealing with models containing a large number of polygons that can place a significant load on CPUs. As such, C2M presents a classic trade-off between speed and geometric fidelity.

While some researchers may question whether merely reporting mathematical deviation values constitutes a meaningful measure of construction quality, researchers also acknowledge that many times reported deviation values lack context and therefore result in misleading conclusions. Two primary limitations of current deviation analysis may exist: First, most deviation analysis algorithms could be highly sensitive to noise in the scans (Kavaliauskas et al., 2022) that temporarily occupy space near the building during construction, potentially resulting in false positive reports, as noted in construction research.

Secondly, since most scan vs. BIM frameworks utilize iterative closest point (ICP) to register the coordinates of the point clouds with the coordinates of the BIM model, problems may arise when ICP fails to accurately register the two coordinate systems due to drift from insufficient overlap or uneven features, which could accumulate and be misinterpreted as construction errors (Bassier et al., 2020), according to relevant studies.

Overall, studies such as (Bassier et al., 2020) identify one research gap specifically relating to an analytical methodology that can automatically distinguish between genuine deviations resulting from construction defects and spurious deviations generated by registration errors, scanner artifacts or temporary items on-site, as part of broader literature.

Therefore, future studies involving lightweight frameworks need to be developed so as to minimize their reliance on globally insensitive and contextually insensitive deviation calculations. Future studies need to develop localized assessment methodologies through limiting C2M evaluations to particular element areas via oriented bounding boxes (obb) (C. Kim et al., 2013). OBBs enable localized assessments by limiting the number of points that require calculation while reducing the amount of impact from clutter, as used in point cloud processing techniques. Additionally, incorporating localized feature-matching methods or drift-invariant metrics would further reduce dependence on computationally intensive global ICP registrations.

2.6.2 Tolerance Standards and Construction Quality Metrics

The problem of establishing universally applicable tolerance standards has led to considerable academic debate. The main argument against using standard tolerance values was that they assume that the same threshold would apply equally well to every type of project. It is generally acknowledged that tolerance levels are heavily influenced by the production environment, building techniques, and the materials being used. Standardized tolerance levels cannot account for the diversity of site practices.

An example of where this applies is in the case of modular fabrication facilities. In these cases, certain pre-cast concrete components are manufactured under extremely tight tolerances; e.g., some types of components may have tolerances as small as 6 mm for specific alignment purposes. However, when these components undergo transport, lifting, and placement, their original manufacturing tolerances will almost inevitably deteriorate. When these increased tolerances are large enough to be considered "Excessive Geometric Deviation" (EGD), even though the components were originally designed to meet the desired performance criteria, there is a high probability that on-site adjustments to the components will occur. These on-site modifications allow workers to take advantage of the inherent flexibility in building materials, manipulate joints, and/or force-fit the components together so that the overall structure is functionally complete. In most cases, geometric deviations measured digitally do not equate to failures in construction. An Automated Quality Assessment system that uses Euclidean deviation as

its sole metric to measure geometric compliance can generate an excessive number of false-positive alerts about elements that would otherwise qualify as compliant. These excessive false positives can lead to the rejection of good quality products, create conflicts among contractors, and cause delays in projects due to reliance upon digital metrics alone.

In addition, this debate exposes an important limitation of traditional BIM. Traditional BIM systems regardless of what CAD/BIM software is used are primarily structured around defining geometry as fixed entities with sharp boundaries. They are not specifically designed to accommodate tolerance as a flexible variable that takes into consideration error propagation, variability based upon probability, or visually represents Geometric Dimensioning and Tolerancing (GD&T) associated with inspection methods. Therefore, BIM is better described as a spatial database than a comprehensive tolerance management tool.

Analytically speaking, today's quality assessments remain segmented and focus on each individual part/element independently. Practically speaking, defects in one component can affect the correct placement of adjacent parts/components through tolerance accumulation/stack up. For example, minor deviations in the positioning of columns can affect both the proper placement of beams, facade panels or prefabricated modules whose positions rely upon identical reference points (Dong et al., 2023). This systemic view is rare in LiDAR- or BIM-based inspection frameworks.

Therefore, the identified research gap exists because there currently remains limited integration between scan-based deviation analysis, BIM's ability to perform parametric modelling and probabilistic simulations of tolerance chains (Rausch et al., 2017). Most scanning/BIM-based systems operate on a basis where each part/component is evaluated separately, without considering how those parts/components will impact subsequent assembly operations.

Consequently, developing a lightweight framework for quality assessment requires going beyond simple binary pass/fail judgments. Rather than simply providing pass/fail judgments, an adaptive judgment mechanism should utilize Dimensional Chain Theory to evaluate how each deviation impacts the entire assembly system. Within this approach, a deviated part/component is not automatically disqualified. It is evaluated to determine if the deviation can still be absorbed through locally adjusting nearby parts/components (buffer adaptations) (Y. Zhang et al., 2026).

Through use of this method of evaluation, the system provides recommendations for making localized updates to the as-designed BIM model so that any necessary changes are computed and accounted for without requiring disassembly. Ultimately, this allows the decision-making process to be more practical, efficient and consistent with real-world construction practices instead of merely being determined by rigid numeric standards. The present framework takes a first step toward this vision through the context-aware, type-specific tolerance classification developed in Chapter 3 (Section 3.7.2); full dimensional-chain modelling with automated buffer adaptation lies beyond its scope and is identified as future work.

Table 2.5 Rigid versus adaptive tolerance judgment in automated quality assessment

Aspect	Traditional rigid pass/fail	Adaptive tolerance judgement (future direction)
Decision basis	Compares Euclidean deviation against a fixed tolerance threshold	Evaluates whether the deviation can be absorbed within the surrounding assembly through tolerance stack-up

Aspect	Traditional rigid pass/fail	Adaptive tolerance judgement (future direction)
Treatment of deviated elements	Automatically classified as non-compliant once the threshold is exceeded	Assesses whether local buffer adaptations or neighbouring components can compensate before rejection
Tolerance accumulation	Neglected; each element is evaluated independently	Explicitly considers error propagation across connected elements using Dimensional Chain Theory
Quality assessment outcome	May generate false positives, leading to unnecessary rejection, contractor disputes, and project delays	Reduces false positives by recommending localized updates to the as-designed BIM model instead of unnecessary disassembly
Relation to BIM	BIM represents fixed geometry and provides limited support for tolerance reasoning	Supports context-aware and probabilistic reasoning over tolerance chains

2.6.3 BIM Model Validation (Geometric Consistency Checking)

The validation of a BIM model serves as a gatekeeper to ensure that models are developed to the highest quality and that important information is not lost. Furthermore, the model's quality ensures compliance with applicable engineering and design rules prior to approval for physical construction or energy simulation (Solihin et al., 2015). Geometric consistency checking is one key component of this validation process. It performs topological checks on the model, including detecting failures such as overlapping materials, duplicate elements within the same object, boundary gaps at the edges of defined spaces, and elements without structural support (Jansen et al., 2025). Historically, methods used to validate BIM models have been either rule-based geometric checking or automated knowledge acquisition based upon the Semantic Web (SW).

Both rule-based geometric validation of BIM models and automated semantic web validation are paradigmatic methods for assessing model integrity, but they differ in their focus.

Rule-based geometric validation represents a fundamentally different paradigm in which validations occur by analyzing the spatial relationships (intersections, distances, etc.) of objects represented as either part of a B-Rep or as swept solids (Wu & Zhang, 2019). While geometric validation has been shown to be an efficient means for detecting geometric errors, such as hard-clashes (physical overlap), and soft-clashes (tolerance violation, etc.), it does not provide a comparable level of context to validate compliant behaviour, since the method cannot interpret the intent of the designer.

The relative advantages of the two paradigms make them complementary rather than competitive technologies. Geometric validation will continue to play a critical role in detecting spatial conflicts among building components, while semantic validation will become increasingly important for ensuring compliance with regulations, and validating compliance with industry wide practices. As such, development of hybrid validation methodologies that combine the strengths of both paradigms would be beneficial.

A subset of researchers argues that geometric validation utilizing spatial representation only cannot be relied upon to confirm that a BIM model is "valid" or "ready to build." For example, if a tool identifies a void meeting area between the roof and walls of a building with no detected collisions, it may indicate a successful outcome (validation passed), however failing to identify such missing elements could

impact fire zoning (fire ratings) or negatively affect thermal energy load analysis. Further, without semantic interpretation, seemingly consistent geometry represents merely an empty polygonal mesh (Sacks et al., 2018).

Conversely, some proponents of semantic-based rule validation seek to translate human-generated building code requirements into machine-executable formats to facilitate autonomous execution of normative validation tasks (J. Zhang & El-Gohary, 2017). Semantically based validation tools have demonstrated an average success rate of validating 66% of all task requirements autonomously without human assistance (Pauwels et al., 2024). However, semantic-based validation tools may struggle computationally when required to represent complex spatial relationships (i.e., intersection of slopes) and/or geometric ambiguities (Pauwels et al., 2024).

The compromise position is thus that geometry lacks contextual significance, whereas semantics lacks geometric relevance. Thus, there exists a clear research gap where current software platforms have failed to sufficiently integrate semantic reasoning with in depth geometric mathematical examinations. Moreover, software platforms fail to avoid error-prone data transfers, e.g., IFC format translations and associated potential data loss objects (Wu & Zhang, 2019).

Therefore, new lightweight frameworks need to abandon file-conversion-based validation processes (export/import files) and instead use an Early-Stage Topologic Validation methodology that leverages a lightweight knowledge graph to efficiently abstract relevant semantic metadata for a Geometric Consistency Check. The hybrid approach divides the calculation for determining geometric overlap into a separate engine layer than the querying of semantic parameters. Thus, the application will not crash during large-scale integrative analysis of large numbers of complex infrastructure assets. Within this current frame of reference "Model Quality" is strictly defined as that per-element assessment of Model Quality described in Section 3.8 - The Model Quality Assessment based upon deviations (how well does the as-is BIM model represent what was measured) and not those assessments of clash detection and semantic rule compliance evaluated herein.

2.6.4 Interoperability Formats and Revit API for Point Cloud-BIM Exchange

A fully functioning Scan-to-BIM workflow will depend completely upon the efficient flow of both geometric and semantic data between reality capture equipment, processing applications, and BIM authoring platforms. Construction projects now increasingly require greater computing speed, therefore, it has become extremely important to avoid vendor-specific lock-ins and utilize the use of lightweight, open-source formats for data exchange. Reality capture generates very large amounts of data that are generally stored in formats specific to vendors (.rcp; .fls), limiting automation of subsequent processing and cross-platform interoperability. A solution to these problems has been provided through the development of the ASTM E57 (ASTM E2807-11(2019)), a neutral, hardware independent standard for three-dimensional image data (Huber, 2011). The ASTM E57 format consists of two components; first, an XML based header containing human readable information regarding camera type, orientation, location etc., secondly a binary component storing the actual dense point cloud data. The ability to process data quickly and perform lightweight validation and initial alignment checks utilizing the XML header reduces the processing time required for heavy-duty point cloud processing, thus enabling faster processing times for web-based or edge-computer platforms.

At the same time as developing spatial data standards, the Industry Foundation Classes (IFC), managed by buildingSMART and standardized as ISO 16739, serves as the global open standard for exchanging BIM data. While IFC does encode both 3D geometric representations of building objects and relationships between those objects, IFC encodes the properties and semantics associated with each object in addition to their geometric representation. The EXPRESS schema is utilized to encode the

semantics of objects. More recently there have been efforts to support encoding in XML and JSON. When performing quality assessments for Scan-to-BIM workflows, the flexibility of IFC to embed validation results directly within the BIM model through the use of the EXPRESS schema - specifically through the use of "IfcPropertySet" plays a significant role. Following the completion of point cloud segmentation and deviation analysis, various types of quality metrics can be encoded and stored in the form of property sets on corresponding IfcBuildingElements. For example, if one were to calculate millimetre level deviations from target tolerances or determine whether certain parts of a project were outside acceptable limits, those calculated values could be added as new properties to corresponding IfcBuildingElements. By doing so, the IFC file would provide a single repository for capturing design intent and as built conditions providing more utility than merely being a means of representing geometry. Within this framework the next step is to integrate these results into a common BIM authoring tool like Autodesk Revit. The integration of the ifc based results with a common BIM tool like Autodesk Revit is done using ifc. Deviation and quality results are added to the model as ifc property sets (such as a per-element deviation schedule) and then are exported as an IfcGroup which will be openable in Autodesk Revit for use in scheduling.

2.6.5 IFC-based Element Classification and Tolerance-based Quality Assessment

While geometric alignment and segmentation provide solutions to the quantitative aspect of the Scan-to-BIM process, identification of IFC elements is required to determine whether measured deviations represent acceptable construction tolerances or actual defects. A numerical difference does not convey sufficient meaning since acceptable tolerances depend upon context; e.g. a 50 mm difference could be acceptable in earthworks and unacceptable in structural steel. Therefore, tolerance-based assessment frameworks are required to translate geometric deviations into quality decisions. The United States Institute of Building Documentation (USIBD) created the Level of Accuracy (LOA) specifications, which provide tolerance-based accuracy levels such as LOA 10 to LOA 50 based on the difference between what was physically recorded through scanning technology and digital representation. LOA limits can be used as comparators when evaluating point cloud differences relative to defined tolerances within Scan-to-BIM workflows, thereby assessing compliance of building elements with quality standards.

2.7 SUMMARY AND IDENTIFICATION OF RESEARCH GAPS

2.7.1 Synthesis of the Literature

A summary of the previous sections, and identification of research gaps for the transition from traditional methods of inspection to digital methods of inspection in the form of Scan-to-BIM workflows. In conclusion, the body of literature reviewed has provided the technological basis and methodology for transitioning to new methods of inspection. The themes outlined below provide an overview of the existing state of knowledge and also indicate areas where further investigation is required.

In terms of data acquisition, the review compared two paradigms: passive photogrammetry and active LiDAR-based laser scanning. While photogrammetry offers advantages in cost-effectiveness and flexibility in construction environments, it has been shown to be ineffective in construction environments due to the predominance of texture less and homogeneous surfaces (such as those found on precast concrete panel walls and uniform interior finishes) that hinder successful tie-point matching using SfM algorithms (Pepe et al., 2026).

In addition, photogrammetry lacks the ability to establish an absolute scale unless one uses external reference points to provide scale information, adding further complexity to procedures. In contrast, laser technology (particularly terrestrial laser scanning) has been established as the current standard for high levels of geometric accuracy, achieving levels of precision on the order of submillimeters to centimetres

over long distances. However, terrestrial laser scanning creates serious restrictions in operational terms, including slow sequential data collection, the requirement to move/re-position instruments repeatedly, and the likelihood of occlusions caused by construction equipment and/or scaffolding. Mobile Laser Scanning (MLS), facilitated by SLAM algorithms was identified as an attractive alternative to reduce acquisition time by up to 42 percent when compared to manual techniques (Tan et al., 2025). Although mobile laser scanning provides a number of specific degradation characteristics in terms of data quality, including drift in location tracking, increased noise, and varying point density, these effects must be directly addressed through subsequent processing.

Preprocessing was illustrated to be critical to the success of any Scan-to-BIM workflow. Standardized preprocessing techniques, such as voxel grid down-sampling, PCA-based normal vector determination, and statistically/radius-based outlier detection are well-documented within the body of existing literature and will continue to play a role in ensuring sufficient structural detail while minimizing computational costs associated with subsequent BIM-related processes. It was also indicated that voxel-based down-sampling is preferable to random down-sampling for lightweight structures due to its better preservation of overall spatial structure (Lyu et al., 2024). Additionally, emerging deep-learning-based denoising techniques were found to be demonstrably more effective than their statistical counterparts, but at the cost of significantly increased memory and computational requirements that prohibit their deployment in resource-constrained environments (Y. Li & Sheng, 2023).

Registration, which aligns scan data to a reference BIM coordinate system, has been studied at both the coarse registration stage (aligning scans globally to a reference coordinate system without an initial guess), and the finer registration stage (aligning two point clouds together using a previously registered pair of scans). Although the use of FPFH with RANSAC-based feature matching still dominates as an initialization-free method for global coarse registration, it does have documented problems in orthogonal and self-similar building environments. These types of buildings can contain repeated structural elements, which will produce similar features at different locations, causing ambiguity when selecting features for correspondence. Furthermore, RANSAC uses random sampling. Therefore, the output of this method may be non-deterministic, and because each sample requires computation time, the cost of RANSAC grows with increasing amounts of noise and larger datasets (Mahmood et al., 2020). Fast Global Registration is an alternate method to RANSAC that does away with the randomness of the sampling process; instead, every possible relationship among features is considered at once by optimizing every relationship simultaneously using a Geman–McClure robust cost function, and this deterministic nature is a characteristic of the algorithm itself. However, its deterministic behaviour is conditional on the implementation, and multi-threaded feature matching can still introduce run-to-run variability in the resulting correspondences (Zhou et al., 2016).

Point-to-Plane versions of ICP, while being generally faster than point-to-point formulations for applications with large amounts of planar geometry like many man-made objects, are highly dependent upon the data quality including the estimation of surface normals and whether or not there are good planar features available (Pomerleau et al., 2013). Thus, when the input data has significant errors in surface normal estimates then the point-to-plane ICP version will be less robust. This could cause an unstable convergence or even completely wrong alignments. On the other hand, the point-to-point version of ICP provides a much more general and less assumption-based model that makes it more robust in cases where the surface characteristics are unknown. However, the increased robustness of the point-to-point model comes at the cost of lower speeds and increased computational requirements. Additionally, similar to point-to-plane ICP models, the point-to-point ICP models suffer from decreased performance when exposed to extreme noise levels. The most common way to improve the efficiency of an ICP-based registration process from a computational standpoint involves the utilization of spatial

indexing structures. Spatial indexing structures such as k-d trees and octrees provide a means to accelerate nearest neighbour searches. Exhaustive pairwise distance computations can be avoided when using these structures to organize the point clouds in hierarchical space partitions thereby reducing search complexities per query. This provides significant reductions in overall computation times for each iteration of alignment, but results vary based on both the size of the point clouds and specific implementation detail (Pomerleau et al., 2013; Rusinkiewicz & Levoy, 2001).

As discussed below, a wide variety of techniques exist in the literature for segmenting point clouds into sets of points representing individual objects or entities. This includes geometric region growing methods, density-based clustering (DBSCAN), graph-based segmentation, and deep learning architectures spanning point-based (PointNet, PointNet++), sparse-convolutional (SpU-Net) and attention-based (PTv3) paradigms. While the geometric methods were very interpretable, required less labelling effort than other segmentation techniques, and could serve as useful pre-processing tools to reduce the dimensionality of input data prior to downstream processing steps, they were also susceptible to degradation in cases where there existed significant amounts of noise or varying densities throughout the point cloud. Similar to the geometric region growing methods, while DBSCAN is theoretically robust to noise, it has been observed to fail in environments where multiple planes are present or where there exists variable point densities across different parts of the point cloud (K. Liu et al., 2024). Although deep learning-based architectures have achieved state-of-the-art semantic accuracy for object detection tasks on point clouds, these architectures require considerable amounts of annotated data to train accurately, often do not generalize well to real-world construction environments versus synthetic training environments and in the case of attention-based architectures, exhibit computational complexity quadratic in neighbourhood size which limits their practicality for resource-constrained deployment scenarios. Overall, the literature suggests that none of these three categories of segmentation algorithms satisfy all three criteria (accuracy, robustness to real-world conditions, and computational efficiency) simultaneously; accuracy and computational efficiency remain competing objectives.

Deviation analyses performed between as-built point clouds and as-designed BIM models represent another important aspect of quality control. Two general approaches used to perform these analyses include C2C and C2M. C2M has been shown to be geometrically superior to C2C but is computationally expensive for polygon-dense models. An additional issue common to both approaches is that they are highly dependent upon accurate registration. Misregistration resulting from ICP drift could result in a false positive quality assessment if misinterpreted as actual defects in construction (Bassier et al., 2020). Finally, tolerance decisions based upon fixed Euclidean distance thresholds have been shown to ignore tolerance accumulation between related items and the flexibility allowed in many field-assembled systems. Concurrently reviewing BIM model validation indicated that current implementations have insufficiently merged two paradigms, geometric consistency checking and semantic web-based validation both provide useful information regarding the validity of BIM models but lack a unified architecture to efficiently combine these paradigms.

2.7.2 Identified Research Gaps

In addition to the many gaps identified in previous research, there are four major areas of identified research gaps in this study that support the need for developing a Scan-to-BIM integrated lightweight framework.

2.7.2.1 *Lack of a Lightweight Framework That Integrates All Stages of Processing*

The literature review demonstrates gaps in fully lightweight frameworks that integrate all stages of processing (i.e., preprocessing, registration, segmentation, etc.) into process flows suitable for resource-constrained computing platforms in on-site or near-real-time processing environments. The lack of an

integrated framework that provides a full spectrum view of the entire processing chain is the first area of needed research (Abreu et al., 2023; Rashdi et al., 2022).

2.7.2.2 *Trade-offs Between Accuracy and Computational Cost In Segmentation Methods*

Researchers agree that geometry-based segmentation approaches are fast but limited due to their sensitivity to noise and density variations. Conversely, researchers also agree that deep learning Architectures are probably more accurate but slow and expensive in terms of memory usage and require large amounts of domain-matched annotation training data. There thus exists an open need to develop a segmentation strategy where there is an equilibrium between how quickly data can be processed through the system, versus the level of detail in those results. Additionally, such a solution needs to operate at a cost and with a functionality compatible with consumer-grade CPU and GPU hardware, rather than requiring high-end server-class GPU infrastructure. This would allow the segmentation process to degrade instead of fail if the required hardware cannot be located or is otherwise unavailable (Singh & Yadav, 2024; Xiao et al., 2024).

2.7.2.3 *Robustness of Contextual Deviation Analysis Methods*

The bulk of deviation analysis methodologies currently employed by researchers involve comparing measured data against model data on a global basis. Consequently, researchers have reported a number of cases where measurement discrepancies arose not due to actual construction errors but rather from inaccuracies in registration artifact representation, scanner noise and/or temporary obstructions at the site. The inability to employ local evaluation strategies at the element level and assess deviations on a per-element basis reduces the reliability of automatic quality assessments (Bassier et al., 2020; Kavaliauskas et al., 2022).

2.7.2.4 *Integration Between Geometric Consistency Checking and BIM-Derived Structural Information*

Although BIM has been applied to geometric check extensively, most studies use BIM-derived structural data as static references to perform deviation analysis at the global scope. This decoupling will limit the contextuality of quality assessment, especially at the element scope. Moreover, although advanced methodological approaches (such as semantic validation or knowledge-based validation) can provide higher levels of accuracy than traditional checks, they tend to be highly computational and thus cannot be easily implemented into an on-site application. Thus, there is a gap to develop a low-weighted framework which will integrate the BIM-based structural segmentation with geometric registration and element-level deviation analysis in one workflow.

2.7.3 Positioning of the Proposed Framework

Section 2.7.2 described the shortcomings of the current Scan-to-BIM landscape. Those shortcomings all stem from an overarching limitation: There is no one unified framework that is both able to be processed efficiently on resource-constrained computing platforms and is also able to handle imperfect point clouds in a manner that allows element level construction quality assessments.

This proposed lightweight Scan-to-BIM framework will serve as a solution to the limitations of the existing frameworks, which will achieve this through the creation of a cohesive workflow that centres itself upon three fundamental ideas, end-to-end processing, computational efficiency, and element level geometric quality assessment.

Concerning the idea of End-to-End Processing, the framework is designed to integrate every part of the entire pipeline into a single workflow that begins at point cloud capture/preprocessing and continues on to include voxel based down-sampling/normal estimation, global registration using FPFH descriptors & FGR, refinement using ICP and lastly, element level deviation analysis versus IFC derived BIM geometry. By integrating each step of the workflow in such a way that each step is dependent upon the

previous steps, there is less separation between processing stages and thus each stage provides direct support to the subsequent stages in regard to the quality assessment process.

In terms of the amount of computation required (computational efficiency), the approach uses lightweight geometric operations throughout. Additionally, the cost associated with the one learned operator is limited. The use of voxel grids minimizes the number of points in comparison to traditional methods that preserve structural geometry. Matching based on features occurs using FPFH descriptors during the global alignment stage. Fast Global Registration is used as opposed to random sampling consensus due to fast consistent performance. The best candidate is finally refined using point-to-point ICP.

Four classical operators are compared: adaptive bounding-box growth, graph-based clustering with normal filtering, DBSCAN with normal filtering, and RANSAC plane fitting. The deployed pipeline selects per-element candidates from the first three operators using coverage and compactness; RANSAC plane fitting is retained only as a comparative baseline. Proximity-based recovery is available when the primary segmentation does not provide sufficient support.

Prior to applying the geometric segmentation, an optional auxiliary processing step may first apply a previously trained S3DIS SpU-Net. This neural network does not replace the segmentation step but rather provides a pre-processing step that eliminates non-structural clutter and minimizes confusion between similar geometric candidates.

For quality assessment purposes, the framework conducts localized deviation analyses at the element level by calculating point-to-plane distances between segmented scan points and BIM surface elements generated from IFC Geometry. In contrast to other methods that rely upon global binary thresholds for determining if a particular segment has deviated from design specifications or not, this method assesses deviation at each individual element level from filtered and geometrically consistent point subsets. This helps minimize the effects of misregistration noise and/or scan related artifacts on overall deviation calculations.

Finally, the framework addresses the connection between geometric alignment and BIM-based validation by linking as-built point cloud regions segmented from scanned images with their respective IFC-derived element geometries. Ultimately this linkage enables element-wise comparisons between actual and designed state conditions.

Therefore, the proposed framework constitutes more than an aggregation of independent methods. It provides a structured and computationally efficient end-to-end pipeline that integrates feature-based registration, deterministic refinement, and geometry-driven segmentation to enable practical and scalable Scan-to-BIM quality assessment in real-world construction environments. The results of the framework are reported on both quality aspects of construction (construction quality assessment or CQA) that evaluates whether the constructed building meets with specifications regarding the tolerance of each element (as defined in section 3.7), and the quality aspect of the model (model quality assessment or MQA) which assesses how closely the as-is BIM model is able to represent the physical facility as it has been measured (section 3.8).

Table 2.6 Mapping of identified research gaps to the proposed framework and thesis structure

Identified research gap (§2.7.2)	How the proposed framework addresses it	Developed in
Gap 1 No lightweight framework integrating all processing stages (§2.7.2.1)	A single end-to-end pipeline from capture and voxel down-sampling through FPFH + FGR registration, ICP refinement, and element-level deviation analysis	Ch. 3 (§3.2); Ch. 4 (§4.8); Primary RQ
Gap 2 Accuracy vs. computational-cost trade-off in segmentation (§2.7.2.2)	A best-candidate hybrid cascade of geometric operators with a single lightweight learned operator (SpU-Net) as an auxiliary clutter remover, targeting consumer-grade hardware	Ch. 3 (§3.6, esp. §3.6.5); Ch. 4 (§4.5); RQ2
Gap 3 Global deviation analysis is not robust to registration / scan artefacts (§2.7.2.3)	Element-level point-to-plane deviation computed on filtered, geometrically consistent point subsets rather than global thresholds	Ch. 3 (§3.7); Ch. 4 (§4.6); RQ3
Gap 4 Geometric checking decoupled from BIM-derived structural information (§2.7.2.4)	Segmented as-built regions are linked to their IFC-derived element geometry, enabling element-wise as-built vs. as-designed comparison	Ch. 3 (§3.7–§3.8); Ch. 4 (§4.6–§4.7); RQ3

3
METHODOLOGY

3.1 FRAMEWORK ARCHITECTURE AND DESIGN PRINCIPLES

A single linear Scan-to-BIM pipeline was chosen for this study. Each stage has been designed to be modular. This modularity allows validation, benchmarking and replacement of each stage independently. Unlike many other Scan-to-BIM pipelines which depend on high-end GPU infrastructure for Deep Neural Inference (DNI), this framework primarily relies on classical geometry-driven operators supplemented by GPU-accelerated inference on consumer-grade hardware, namely feature based global registration, local refinement, graph connectivity, plane detection and point-to-plane comparison. All these operators have good cost vs. accuracy characteristics for use on commodity hardware. The input artifacts to this framework are the laser scan of the building under consideration, and a corresponding BIM model. Three types of output artifact are produced. These are an annotated IFC model representing the scanned data; a deviation colour coded point cloud showing the deviations from nominal, and tables detailing the quality of the modelled building parts.

The output will allow users to understand how well the model represents reality. They also provide a means of assessing whether discrepancies between reality and model are acceptable or need to be corrected. The framework addresses the research question outlined in section 1.2.2 by breaking the overall problem into seven distinct stages. These stages are summarized in Figure 3.1.

Stage one is the data ingestion and coordinate alignment process. At this stage, the laser scan and BIM model are read into memory. The coordinate systems of both models are compared. If they do not match exactly then a preliminary transformation of the scan is carried out. Stage two is concerned with preparing the BIM model. Target IFC elements are triangulated and transformed into a synthetic reference point cloud. The geometry and volume of these elements are determined. An identifier for each element is assigned and remains constant throughout processing.

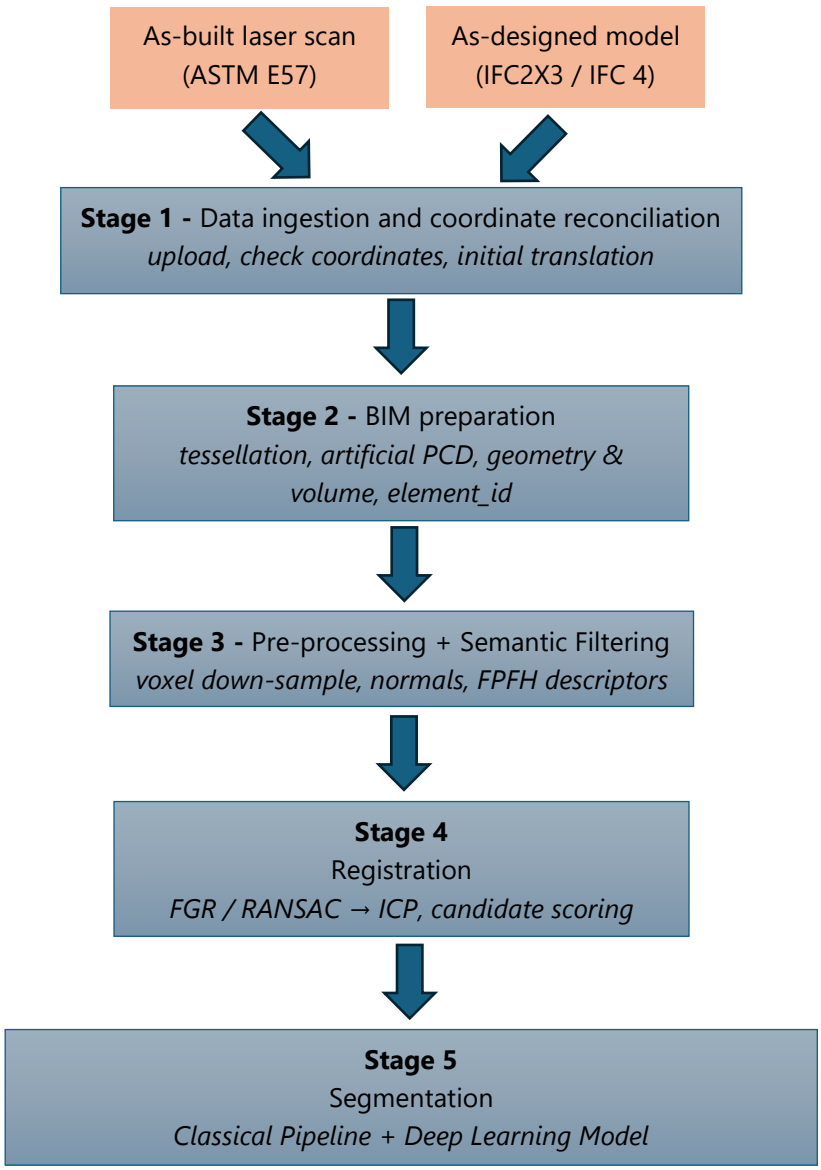
At stage three, preprocessing is performed using voxel downsampling, surface normal calculation and the determination of FPFH descriptors. The goal of the FPFH descriptors is to capture key features of the surface of interest. For example, the shape of a corner may be identified. As mentioned above, it may be difficult to identify certain features if too much noise exists in the original scan. Following preprocessing, the final stage of processing is the semantic inference step where a trained S3DIS model called SpU-Net is used on all data points in the working cloud for classification. Any data points classified as non-structural elements will be removed from the registration cloud. The removal of non-structural points (such as furniture or other objects that could cause inaccuracies) from the registration cloud will

reduce the amount of information processed by the registration algorithms and prevent them from attempting to match these points to BIM elements.

Stage four attempts to register the purged scan onto the BIM model by using a coarse-to-fine search method. Because the non-structural points have already been removed, the registration operates on a cleaner, smaller cloud and converges faster than it would on the unfiltered data.

In addition to providing information about how accurately the scan matches the model, this registration also provides a means of determining which part of the scan corresponds to each modelled element. To accomplish this task, the fifth stage utilizes the DL-First Hybrid Cascade method of segmentation by combining BIM-driven geometric processing methods (the adaptive bounding box and surface normal conformity check) and the SpU-Net classification results from Stage 3 as a tiebreaker or fall back to create individual segments for each model component that were created from the registered point cloud.

Once all elements have been segmented from the scan, stage six determines how large each deviation is relative to an allowable tolerance and produces both a CQA score and a MQA score. Finally, in stage seven, all of the information collected during previous stages is used to produce a series of reports including CSV schedules and an annotated IFC file and an interactive HTML report.



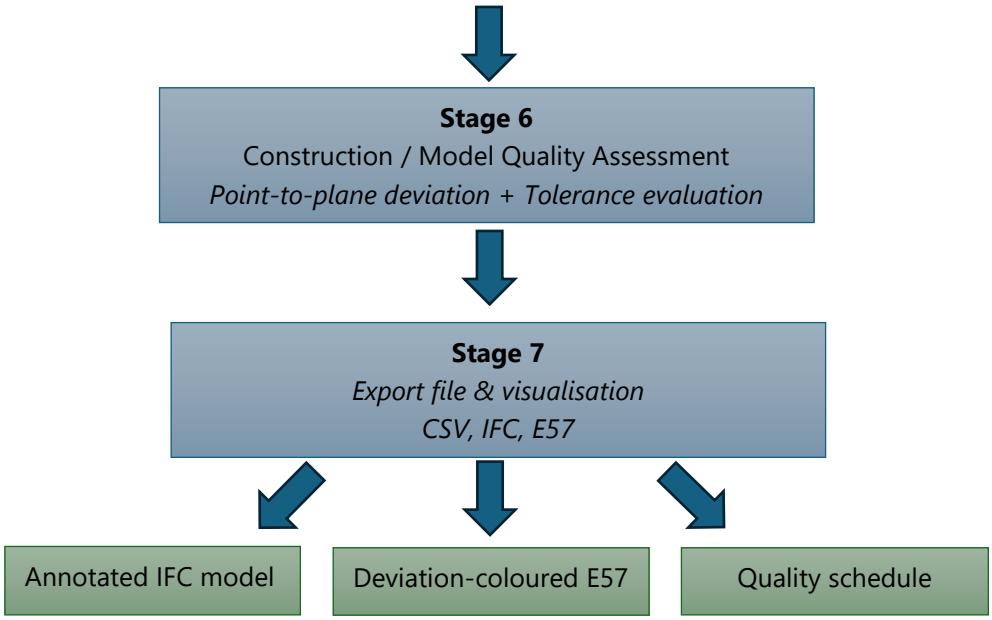


Figure 3.1 Overview of the proposed methodology lightweight Scan-to-BIM framework

3.2 SYSTEM ARCHITECTURE AND THE OVERALL PIPELINE

The Framework has been realized as a lightweight client-server-based application, designed to run the entire pipeline reliably on standard commodity hardware, utilizing a consumer-grade GPU for deep-learning inference while keeping all geometric operations on CPU, and without consuming all of the memory of a typical desktop. In order to achieve this requirement, the backend was developed in Python using the FastAPI framework. The frontend is a client based in a web browser created using Three.js for 3D visualization and Bootstrap 5.3 for layout. High-compute jobs will be submitted to a single-threaded worker thread-pool (ThreadPoolExecutor with max_workers = 1) and were deliberately made sequential to ensure that high-memory use activities would be serialized instead of maximizing raw compute performance. As such, this represents the central premise of the project: That a well-ordered collection of conventional processing elements can provide quality assessment of sufficient quality to be useful in the field (on-site), utilizing existing hardware with GPU acceleration limited to the learned component.

Communication between Client and Server will take place via REST (Representational State Transfer) over HTTP. All stages of the pipeline are represented as separate endpoints. A "status" endpoint will be polled by the client periodically to determine the current stage in the workflow, what percentage of completion has occurred (as a normalized floating-point value within the range [0,1]) and a brief human-readable description of the current phase. Polling provides for short-term connection lifetimes, regardless of how long the processing time may be, thus improving the apparent responsiveness of the client application. Endpoints are described in Table 3.1.

Table 3.1 REST application-programming-interface endpoints exposed by the framework back end

Endpoint	Method	Function within the pipeline
/api/upload	POST	Ingests the E57 scan and the IFC model; performs coordinate reconciliation, tessellation, artificial point-cloud generation, geometry and volume extraction, and element identification

Endpoint	Method	Function within the pipeline
/api/register	POST	Executes pre-processing and the coarse-to-fine registration strategy, returning the best rigid transform together with its fitness and inlier RMSE
/api/cqa	POST	Runs element-wise segmentation, point-to-plane deviation measurement, and context-aware tolerance classification for construction quality assessment
/api/mqa	POST	Runs IFC-based element classification and the deviation tolerance assessment for BIM model quality assessment
/api/export	GET	Assembles and returns the export package (annotated IFC, deviation-coloured E57, CSV schedules, interactive HTML report)
/api/status	GET	Reports the current step, normalized progress in [0, 1], and a status message for real-time progress polling

All endpoints follow sequentially according to the same linear ordering defined in the Research Framework's Process Model. Following both models being loaded onto the server, the server performs two preliminary operations. If the scans and model have been registered into the same coordinate frame, then registration can be performed using an identity transformation and if they have different coordinate frames then the server will perform an initial centroid translation to begin aligning them prior to applying pre-registration and registration. Upon completion of either the registration, segmentation, deviation measurement, or selected quality assessment (either CQA or MQA) the final output package is generated and rendered. The complete end-to-end pipeline is depicted in Figure 3.2, showing the conditional decision point for determining coordinate equality.

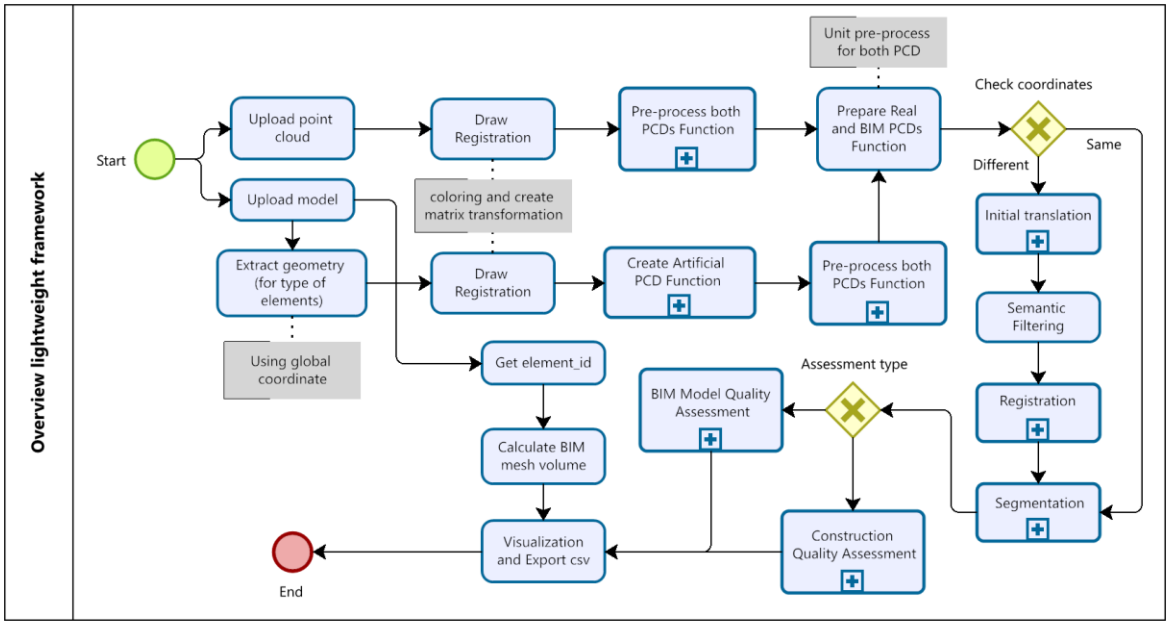


Figure 3.2 Overall end-to-end pipeline of the framework

Two observations about Figure 3.2 suggest future discussions in later chapters. Firstly, "Prepare Real and BIM PCDs", is a composite function that internally generates a synthetic reference cloud from the

BIM mesh; it is discussed further in section 3.4. Secondly, although there is only one “Registration” activity shown in Figure 3.2, it conceals much of the candidate search logic discussed in detail in section 3.5; the deployed web application differs significantly from the Research Framework in its method for searching through potential candidates.

3.3 INPUT DATA AND PRE-PROCESSING

As a requirement for any operation that follows it, the system must reconcile two different types of heterogeneous artifacts before they can be processed: a discrete, unorganized point cloud representing an existing structure ("as built") and a structured, semantically enriched representation of a new structure ("as designed"). The as-built geometry is delivered as one or more scans in the binary ASTM E57 exchange format and is read using the pye57 library. The E57 file contains the positional and orientational information for each individual scan station. In addition to scan data, an E57 file stores a rigid transformation for each scan that defines the location and orientation of the scanning station relative to other stations. When importing scans from an E57 file into a project, if transform = True then the rigid transformations are used to convert the coordinates of each point from the local space of each individual scan to the global (project/world) space. If colour is available, the red-green-blue (RGB) triplet values, represented as 8-bit unsigned integers, are normalized to the unit interval [0,1] to provide a standard method for colouring and downstream colour mapping. A global limit of 1,500,000 points is imposed to restrict both memory usage and computation time. Once the global limit is reached, each scan is stride subsampled according to its size so that spatial coverage is preserved while adhering to the constraints of a lightweight design.

The as-designed geometry is provided as an IFC model in either the IFC2X3 or IFC4 schema, and is interpreted using ifcopenshell with world coordinates (USE_WORLD_COORDS=True), so that element placements are resolved in the model’s world coordinate system. Due to the fact that the two input artifacts are created through separate processes (a field survey vs. a design office), their coordinate frames often do not agree. Therefore, after importing both artifacts, the first processing step is to determine whether the coordinate frames have been established to be equivalent (see Figure 3.2). When they do not agree, the initial alignment will begin by translating the source artifact (scan) so that its centroid aligns with the centroid of the target artifact (model), where this process is formally described in Section 3.5. This simple translation eliminates the substantial translational offset that exists and would otherwise prevent feature-based matching from being successful.

Once both inputs have been translated so that their centroids match, "pre-processing" refers to the composite "Preprocess both PCDs" function. Each input is treated equally, as indicated above, and consists of three sequential steps, described in detail in Section 3.5.1: voxel downsampling to a uniform spatial resolution for both inputs, normal estimation to calculate a surface direction at each point for each input, and calculation of a 33 dimensional Fast Point Feature Histogram (FPFH) descriptor that captures the geometric features surrounding each point. Voxel downsampling for both inputs results in homogeneous point densities and speeds up all subsequent processing steps. Normal estimation serves as a necessary condition for both FPFH and for future normal consistency filtering during segmentation. The calculated FPFH descriptors serve as the correspondence measures used in global registration.

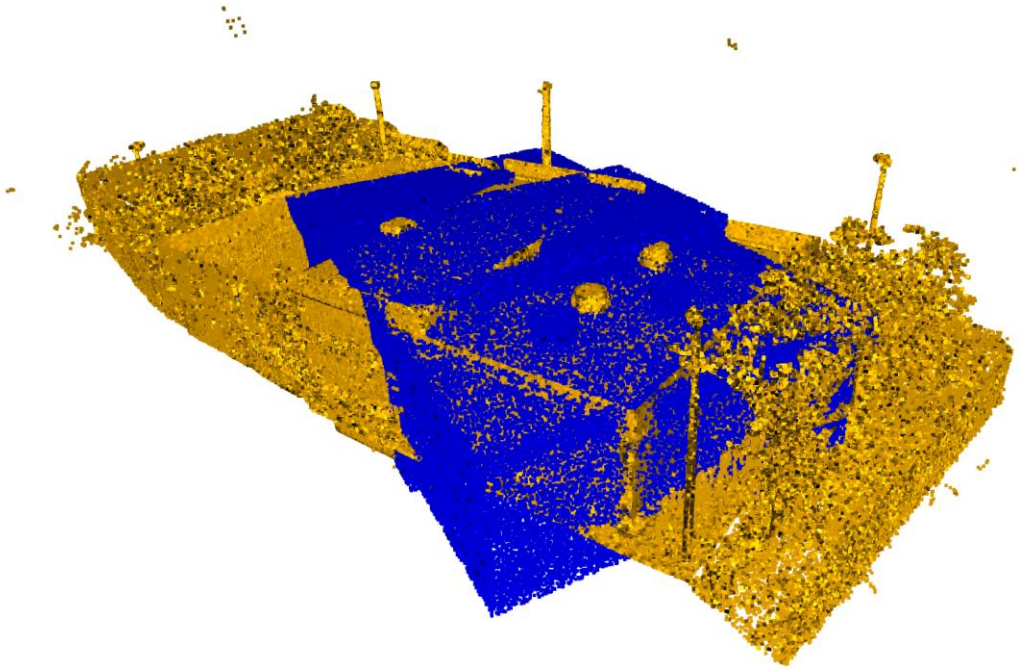


Figure 3.3 Scan and BIM model unaligned coordinate frames before registration

Decisions made prior to ingestion such as loading in world coordinates, RGB normalization, establishing a hard upper bound on total number of points globally, and selecting a common voxel spacing, together ensure that a reasonable and computationally feasible comparison may be performed between an existing structure and a new proposed structure. Preparation of the BIM artifact, however, requires more than simply parsing: the structured model must be transformed into an analogous point set and assigned persistent identities/metadata as discussed in the following section.

3.4 BIM MODEL PROCESSING AND ARTIFICIAL POINT CLOUD GENERATION

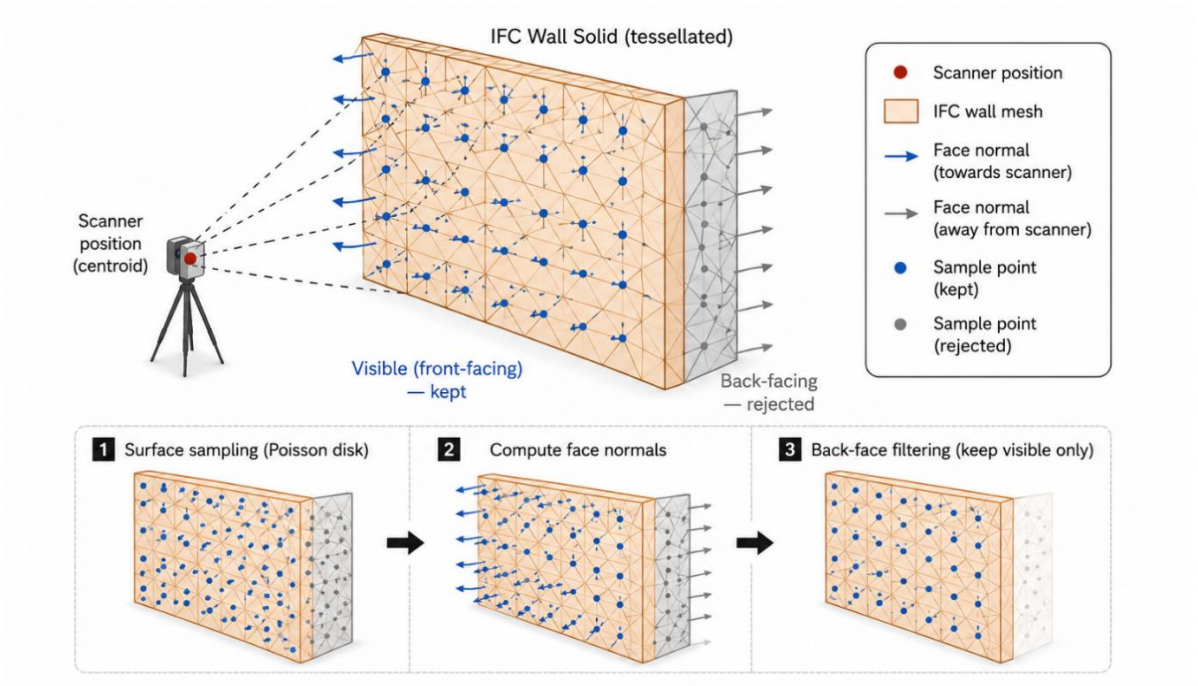
In order to compare an as-built scan with an as-designed model, these two models need to exist in the same conceptual space. Since a scan exists as a collection of points and an IFC model represents a collection of parametric solids as a collection of boundaries, the system transforms the parametric solids into a synthetic point cloud representing what the scanner would have seen of the designer's intent. This transformation is referred to as the "Create Artificial PCD" operation, and it serves as the BIM-side equivalent of the field survey.

In addition to creating an artificial point cloud from all of the parametric geometry associated with all elements of the nineteen IFC element types shown in Table 3.2 that collectively encompass all structural, envelope, and principal service components of a building, this function also limits itself to the consideration of only those elements. The parametric geometry for each individual element is converted into a triangle mesh using ifcopenshell.geom.create_shape() for each of those target types. The resulting mesh surface is then sampled into points.

Table 3.2 IFC element types and their comparison categories

Category	IFC types
Vertical / horizontal structure	IfcWall, IfcSlab, IfcColumn, IfcBeam, IfcRoof, and IfcCurtainWall
Envelope & openings	IfcDoor, IfcWindow, IfcCovering
Circulation	IfcStair, IfcRamp
Secondary structure	IfcMember, IfcPlate, IfcFooting, IfcPile
Building services (MEP)	IfcPipeSegment, IfcDuctSegment, IfcFlowTerminal, IfcFlowFitting

Sampling of the mesh surface occurs at an area proportional rate: large structural surfaces are sampled at a much higher frequency than smaller items such that the artificial point cloud will mimic the density distribution that a real field survey would display. Therefore, subsequent statistics generated based on a comparison of points to planes will not be adversely affected by very small elements. Two sampling budgets are applied to generate clouds - a combined BIM cloud of 200 000 points is created to support both registration and visualization, while a per-element cloud of 20 000 points is created to support measurements of deviations among the various elements. Following sampling, a back-face filter is applied to retain points with normals pointing towards the scan centroid and remove back-facing points. Only points corresponding to the scanner facing surface of each element remain, since a real terrestrial scan will observe only the exposed surface of walls or slabs. Comparisons between the real scan and the design solid would be inflated due to comparisons being made against both sides of each wall or slab.



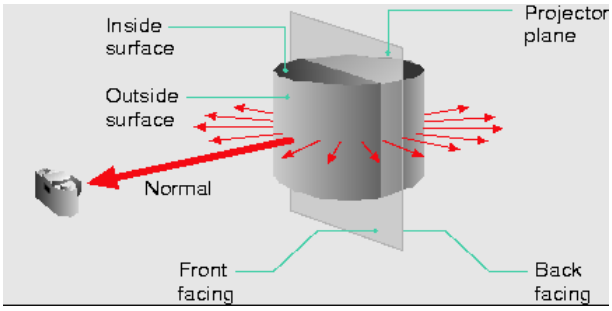


Figure 3.4 Back-face filtering: retaining only the scanner-facing surface of a sampled IFC wall and round column

The procedure for generating the artificial point cloud is identical to the "Create Artificial PCD" process utilized within the research framework illustrated in Figure 3.5. Beginning with the elements identified by their respective types, the function creates an empty container, constructs vertices (verts) containing (x,y,z) coordinate data and triangular faces, creates and aggregates per-element meshes, cleans up the aggregated mesh, and ultimately translates the mesh into the artificial point cloud that is transmitted to registration. As previously mentioned, since the generator reads geometry directly from tessellations provided by the authoring tool's IFC exporter, there are limitations imposed upon geometry that may be read by the generator. This limitation is similar to other tools capable of converting scans into BIM and is addressed in detail in Chapter 5.

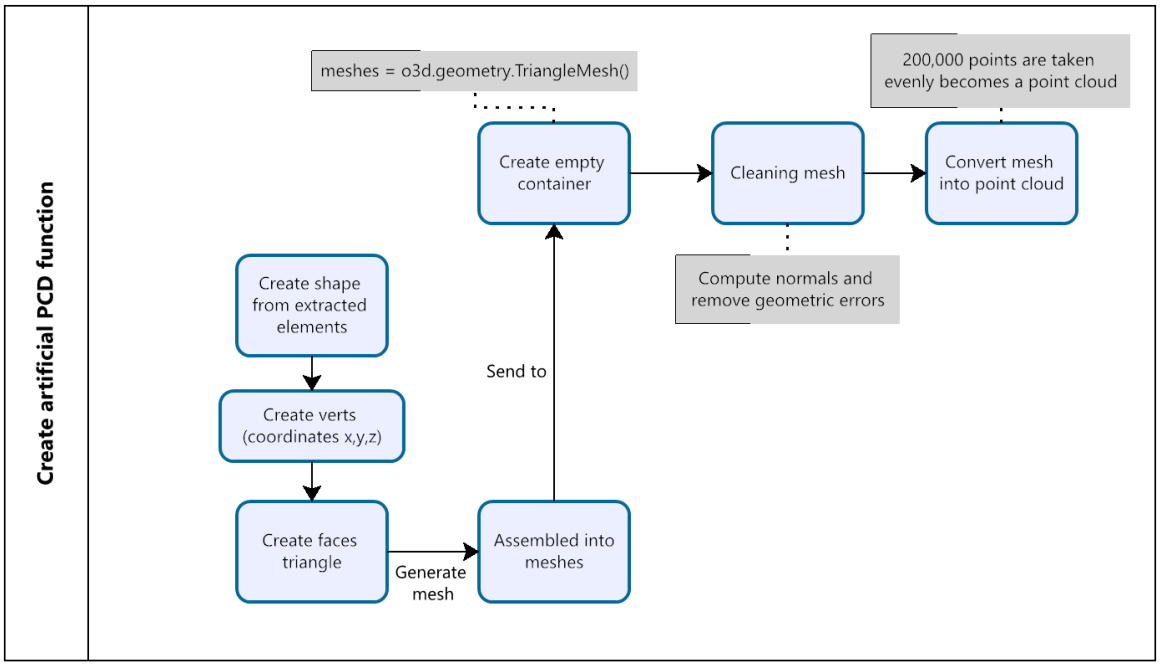


Figure 3.5 Create artificial PCD: Converting IFC geometry into a reference point cloud

Together with the extracted geometry, per-element volumes, and the storey map, the artificial point cloud provides a comprehensive BIM-side surrogate that can be registered and compared with the real scan.

3.5 REGISTRATION PIPELINE

The structural points were determined via a single semantic inference pass prior to registration. The S3DIS-trained SpU-Net classified each of the voxelized scan's points as either structural or non-structural (clutter, furniture, etc.), after which all of those points that were designated as non-structural

were eliminated from consideration for use during alignment. There was a reason this order occurred. In particular, the presence of a piece of furniture at a point in the voxelized scan provides no counterpart within the BIM model and therefore cannot be involved in an inlier correspondence; however, it will always be included in every FPFH descriptor, every nearest neighbour search, and the denominator of the fitness function. Thus, removing these pieces of furniture before performing the alignment results in a reduction in the computational cost of aligning the two models, while also eliminating none of the possible evidence upon which the alignment could rely. At the time of processing, the removal of the furniture reduced the number of points remaining to 140,213 from 209,093 on the reservoir.

A note about the module borders will have to be included prior to the registration stage being detailed. In the case of deployment, the registration stage precedes the segmentation and quality assessment stages, and the transform generated at this stage is applied once to the scanned image. However, for evaluation purposes, the registration module has been evaluated as a stand-alone process; the experiments described within section 4.4 intentionally use the local coordinate so that the ability of the registration module can be measured in recovering the alignment from any random starting position. Therefore, the Fitness and Inlier RMSE values reported in Section 4.4 represent how well the registration algorithm performs when initialized with adverse initial conditions. These values cannot be interpreted as the remaining misalignment that would propagate through the Model Quality Assessment Modules, since these modules receive their input images aligned.

3.5.1 Pre-processing and Feature Computation

All data preprocessing was done using a voxel size of v = 0.10 m. This voxel size was determined based on a balance between the need for sufficient spatial detail to allow for adequate resolution of building components versus the need to minimize the computational load required for point cloud processing. All spatial parameters used in the registration process were normalized relative to this voxel size. The normals for each surface element were calculated by conducting a hybrid k-d tree search around the normal estimation location that had a radius of 2 * v = 0.20 m and a maximum of 30 neighbours. The FPFH descriptor for each surface element was then calculated from a k-d tree search within a radius of 5 * v = 0.50 m and having up to 100 neighbours. As illustrated in Equation (3.1) and summarized in Table 3.3, these two searches represent different radii for the calculation of descriptors.

$$r_{normal} = 2v = 0.20\,m, \quad r_{FPFH} = 5v = 0.50\,m, \quad v = 0.10\,m \tag{3.1}$$

Prior to any feature-based matching being conducted between the two clouds, the significant translational offset existing between them was removed by establishing a seed centroid translation within the transform defined by equation (3.2). In particular, this seed translation establishes the upper right quadrant of the initial homogeneous transform as equal to the vector representing the differences between the centroids of both the source (scan) and target (model) clouds. Therefore, this step represents the "initial translation" portion of the research framework, converting what may have been an unbounded global search into a local well-defined search problem.

$$T_0[0{:}\,3,\,3] = \overline{c_{target}} - \overline{c_{source}}, \quad \bar{c} = \frac{1}{N}\sum_{i=1}^{N} p_i \tag{3.2}$$

Table 3.3 Registration parameters, expressed as multiples of the common voxel size v = 0.10 m

Parameter	Value	Role
Voxel size, *v*	0.10 m	Common down-sampling resolution for both clouds.
Normal radius / max neighbours	2*v* = 0.20 m / 30	Hybrid k-d-tree normal estimation.
FPFH radius / max neighbours	5*v* = 0.50 m / 100	33-dimensional local-geometry descriptor.
FGR max correspondence distance	3*v* = 0.30 m	Inlier gate for Fast Global Registration.
FGR iterations / max tuple count	32 / 300	Optimization budget for FGR.
ICP max distance	8*v* = 0.80 m	Correspondence radius for point-to-point ICP.
ICP max iterations	300 (ICP fallback) / 200 (yaw)	Local refinement budget.
Evaluation distance	*1.5v* = 0.15 m	Distance at which fitness and inlier RMSE are evaluated.
Scoring weight	0.35	Penalty on inlier RMSE in candidate scoring, Equation (3.5).

3.5.2 Global and Local Operators

Fast Global Registration uses the Geman–McClure type scaled loss function to gradually narrow down the region of interest for the set of true inliers. Because this process does not require an iterative sampling loop like RANSAC, the processing time for the registration will be significantly less than a traditional feature-based RANSAC algorithm. However, because the fast global registration method will also generate some false positives as part of its output due to the nature of the FPFH matching results, they can potentially affect the subsequent refinement process. As such, after the initial coarse alignment produced by fast global registration is determined, point to point ICP will be used to refine the coarse alignment.

$$E_{FGR}(T) = \sum_{(p,q)\in\mathcal{K}} \rho(\|p - Tq\|), \quad \rho(x) = \frac{\mu x^2}{\mu + x^2} \tag{3.3}$$

$$E_{ICP}(T) = \sum_{(p_i,q_i)\in\mathcal{C}} \|p_i - Tq_i\|^2 \tag{3.4}$$

3.5.3 Candidate Strategy

When there is a large in-plane (or "yaw") rotation between the two clouds, a single global-local pass may be unsuccessful, because the centroid seed accounts for translations but not orientations.

Therefore, the framework uses a family of three Candidate transforms to find the optimal solution. However, the two implementations of the framework will search over this family of candidates differently and the differences have a significant impact on methodology.

Table 3.4 The three registration candidates evaluated

Candidate	Construction	ICP iterations
1	FGR coarse alignment + ICP refinement	300
2	ICP directly from the centroid seed	300
3	Yaw pre-rotation of 180°+ ICP	200

Each Candidate was evaluated based upon the exact same criteria, equation (3.5), i.e., the alignment fitness (i.e., the ratio of the number of source points that had a model correspondence within the evaluation distance of 1.5v = 0.15 m) with respect to a penalty term due to the weighted inlier RMSE. The Candidate that resulted in the highest score was used as the selected transform. In other words, a higher score indicates that the alignment has explained as much of the source point cloud as possible with residuals that were minimized.

$$\text{score} = \text{fitness} - 0.35 \times \text{RMSE}_{\text{inlier}}, \quad T^{\star} = \underset{k}{arg\,max}\ \text{score}_{k} \tag{3.5}$$

The comparative experiments vary the global operator between FGR and RANSAC while retaining the same candidate-generation and scoring protocol. Each candidate is refined and scored using Equation (3.5), and the highest-scoring transform is selected. The reported protocol applies the full three-candidate sweep to every scene (Section 4.4).

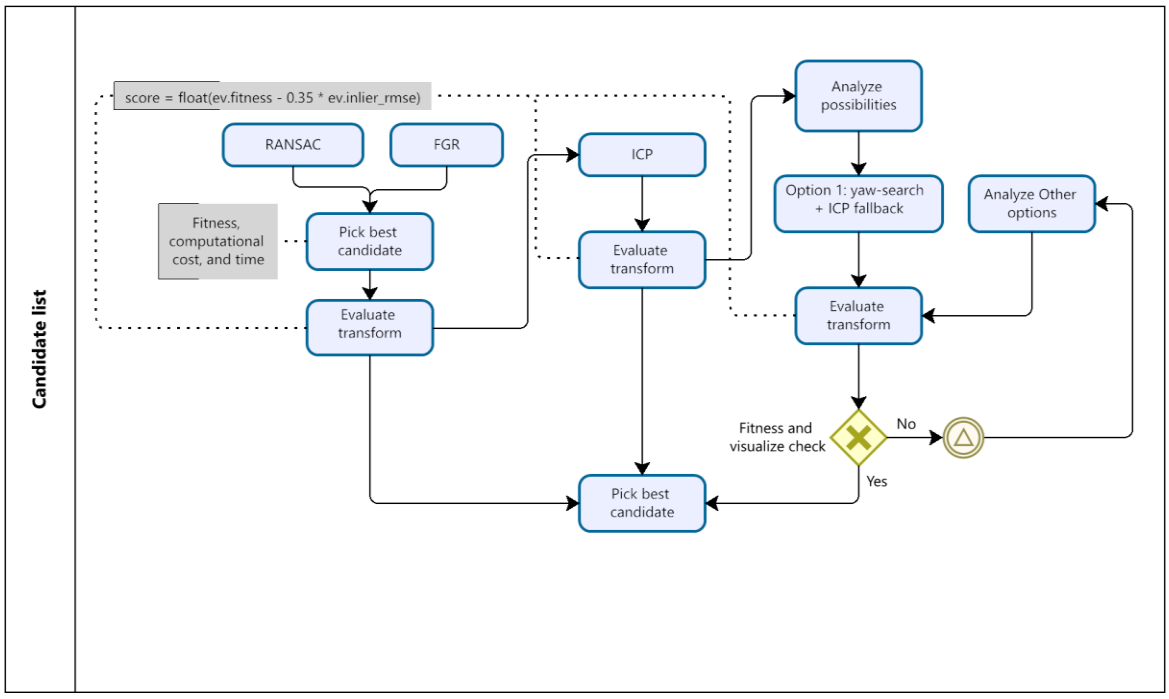


Figure 3.6 Registration strategy

The overall coarse-to-fine procedure for processing input scans includes preprocessing steps, the use of centroids seeds, identifying a family of Candidate solutions via sweeps and scoring. All aspects of this procedure are presented in Algorithm 3.1.

Algorithm 3.1 — FPFH + (FGR | RANSAC) + ICP coarse-to-fine registration

```
# Input : source scan S, target BIM cloud B, voxel v = 0.10 m
# Output: best rigid transform T* with fitness and inlier RMSE

function register(S, B, v):
    Sd, Sf = preprocess(S, v)   # voxel down-sample, normals, FPFH
    Bd, Bf = preprocess(B, v)
    T0     = I4;  T0[0:3,3] = centroid(Bd) - centroid(Sd)   # Eq. (3.2)
    Sd     = transform(Sd, T0)

    cands = []
    # --- global + ICP ---
    Tg = FGR(Sd, Bd, Sf, Bf, max_corr=3v) or RANSAC_FPFH(Sd, Bd, Sf, Bf)
    cands.add(ICP(Sd, Bd, init=Tg, max_dist=8v, iters=300))
    cands.add(ICP(Sd, Bd, init=I4, max_dist=8v, iters=300))   # from seed
    # --- yaw candidates ---
    for yaw in [45,90,135,180,225,270,315]:
        Ry = Rz(deg2rad(yaw))                          # Z rotation, 4x4
        cands.add(ICP(Sd, Bd, init=Ry, max_dist=8v, iters=200))
    # --- score and select (Eq. 3.5) ---
    for T in cands:
        f, e   = evaluate(Sd, Bd, T, dist=3v)    # fitness, inlier RMSE
        T.score = f - 0.35 * e
    return argmax(cands, key=score)
```

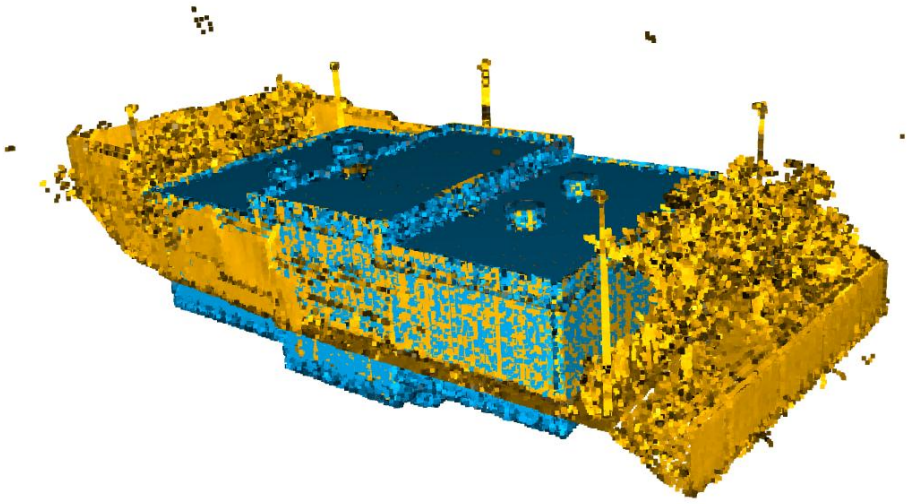


Figure 3.7 Visual reference for the registration result

The comparison between FGR and RANSAC based upon registration fitness and computational complexity, research question 1 is accomplished utilizing the same overall procedure as described above with the addition of swapping out the global operator. These results are discussed in detail in Chapter 4. With a reliable rigid transform in hand then the framework can attribute evidence of the scan to individual modelled elements. The problem of segmentation is addressed next, where robustness to

occlusion and partial coverage is obtained by generating geometric candidates for every element, selecting by coverage and compactness, and applying explicit fallbacks.

3.6 ELEMENT SEGMENTATION PIPELINE

For every modelled element in the point cloud the element segmentation pipeline answers a seemingly straightforward question. Which of the scanned points have been assigned to this particular element? There are numerous challenges to answering this question. These include incomplete scan data. Overlapping neighboring surfaces cause leakage into each other, and clutter exists around all walls and columns. Therefore, in addition to applying at least one geometric operator to every element's geometric data to create candidate segmentations, the processing framework applies all of the remaining geometric operators to each element. Therefore, there is no need for a separate planar vs. nonplanar routing process. Furthermore, no single-plane-fit process is needed. The best performing candidate from the total number of candidates generated during this candidate generation and selection process, therefore, becomes the final segmentation for each element. In other words, based solely on the performance metrics established within the flow, the optimal operator is chosen for each element.

The basic methodology developed throughout this research effort has been implemented through the use of a candidate-generation-and-selection approach. Additionally, this exact same methodology will be utilized to implement the two studies described within Research Question 2 (RQ 2). First, the initial study provides an understanding of how much of a role that independent operation can play when utilizing geometry-based classical algorithms included in this processing framework regarding element detection and segmentation. More specifically, these classical algorithms include adaptive (growing) bounding boxes with normal consistency filtering, graph connectivity with normal consistency filtering and DBSCAN. RANSAC plane fitting remains as a comparative baseline to the first study only. It should be noted that RANSAC plane fitting does not form part of the candidate set that will be deployed. Second, this second study compared the previously mentioned classical algorithms to those of a lightweight pointwise segmenter that had been trained using learning-based models. The second study was conducted in order to assess to what extent training a data-driven model improves segmentation performance, while simultaneously minimizing dependency on high-end GPU hardware. The complete segmentation pipeline is represented in Figure 3.8.

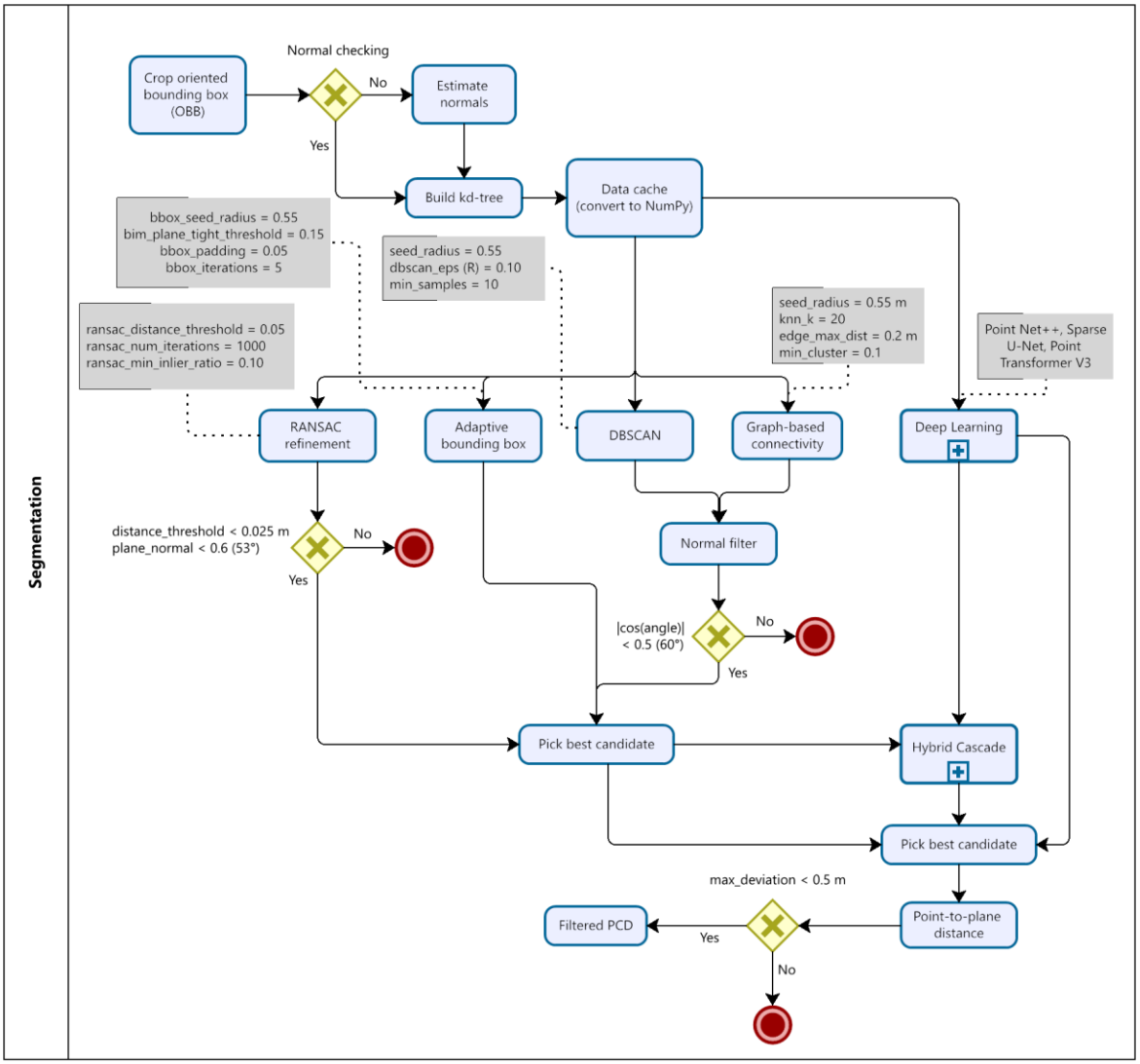


Figure 3.8 Segmentation pipeline

3.6.1 Shared Pre-processing and Candidate Generation

The four preprocessing operations are performed on every element prior to invoking any segmentation operator:

1. Bounding box cropping & padding
2. Normal estimation
3. K-D Tree construction
4. Data Caching

First, the preprocessing operation is cropping and padding. Using the registered scan data against the model, the system identifies the seed radius of the current element, and uses these values to crop the scan data, and construct an Axis Aligned Bounding Box (AABB) or an Oriented Bounding Box (OBB) for the research platform. The OBB provides a better filter than the AABB when dealing with diagonal features (walls). Second, is estimating the normals. Normals are required so that downstream operators can utilize them. If no surface normals have been provided for the cropped points, then surface normals are estimated from the local neighbourhood of each point. Third, is constructing a k-d tree over the remaining points so that fast spatial lookups can be made. Fourth, is caching the data as contiguous NumPy arrays to prevent repeated format conversions during the per-element loop. Following all four

processing stages, the exact same group of candidate operators are applied to every component, no matter what the IFC Type, so that each candidate segmentation can be generated. Rather than apply Routing criteria based on Planarity to each component, the primary innovation at this stage is selecting the best candidate from the candidates for Coverage and Compactness.

Each modelled component (nominally planar types; e.g. IfcWall (IfcWallStandardCase and IfcWallElementedCase); IfcSlab; IfcRoof; IfcCovering; IfcPlate; IfcFooting; and IfcCurtainWall; or non-planar types; e.g. IfcColumn; IfcBeam; IfcMember; IfcDoor; IfcWindow; IfcPipeSegment; IfcDuctSegment; IfcFlowFitting; and IfcFlowTerminal), will go through the same process with respect to the candidate operators. The IFC type of a component is only to provide a prior BIM (bounding box and expected surface normal) for that component; and not to direct it to one particular operator. Each Operator will produce a candidate segment, and the best candidate is selected at the selection stage on the basis of coverage and compactness. RANSAC plane fitting is only considered as a baseline comparison in RQ 2 and is therefore not included within the candidates.

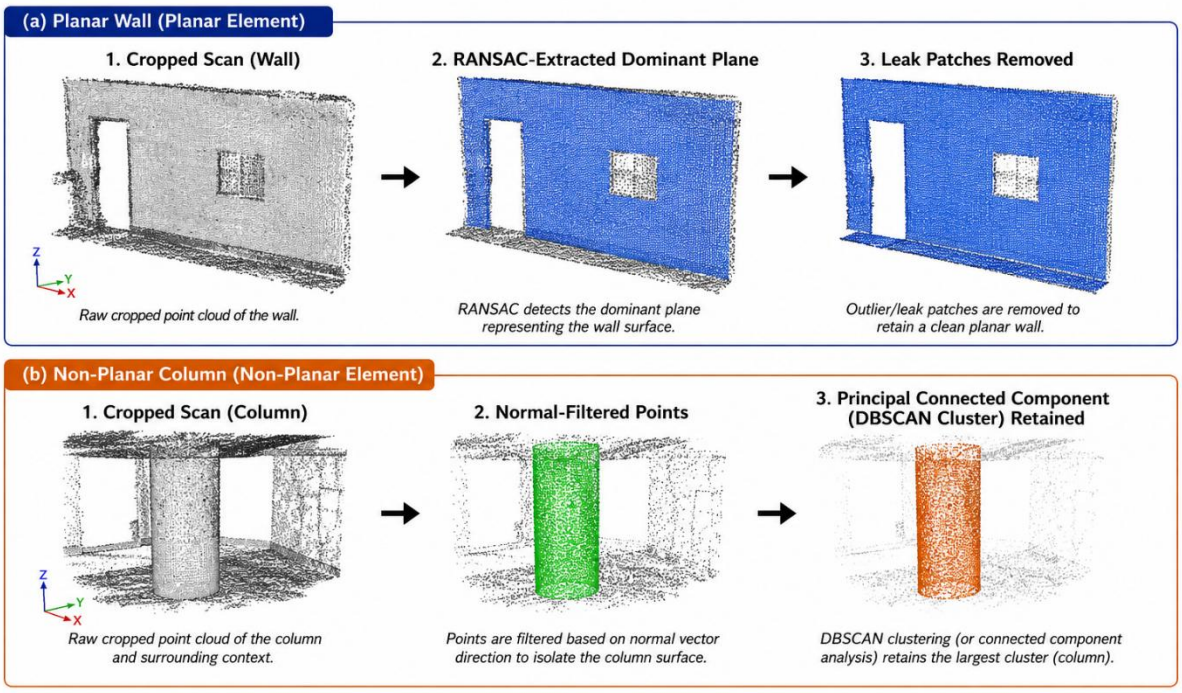


Figure 3.9 Segmentation applied to a planar wall and a non-planar column

A summary of the pre-processing operations applied to all components, and the candidate operators applied to them along with the default parameter settings for each candidate operator are summarized in Table 3.5.

Table 3.5 The segmentation pipeline

Phase	Operation	Description and parameters
Shared pre-processing	Bounding-box crop	Crop registered scan to the element box padded by *seed_radius*. AABB (application) / OBB (research framework). Coarse spatial pre-filter.
	Normal check / estimate	If normals are missing, estimate them from the local neighbourhood; required by every downstream operator.

Phase	Operation	Description and parameters
	Build k-d-tree	Spatial index over the cropped points for nearest-neighbour and radius queries.
	Data cache	Convert to contiguous NumPy arrays; avoid repeated conversion inside the per-element loop.
Candidate generation	Generate candidates	For every element, run the geometric candidate operators below; no separate planar/non-planar routing is applied.
Considered baseline (RQ 2)	RANSAC plane fitting (comparison only)	Evaluated only as an RQ 2 comparison operator, not part of the deployed candidate set. Peel up to max_planes; accept a plane if $\|\cos\theta\| \geq$ plane_normal_thr against the element mean normal (research framework: distance < 0.025 m, plane-normal similarity ≥ 0.866, i.e. 30°).
Candidate operators (RQ 2)	Adaptive growing bounding box	Element box dilated in fixed voxel increments until enclosed point count stabilizes; fallback where neighbourhood relations are too sparse.
	Graph-based + normal vector	Normal-consistency filter (Equation 3.6, research framework $\|\cos\theta\| \geq 0.5$, i.e. 60°) then KNN-graph connected components, retaining components ≥ *min_cluster_ratio* × largest (default 0.10).
	DBSCAN + normal filter	Normal-filtered points clustered by density (DBSCAN); the principal cluster is retained as the element segment.
	Deep learning	Learned pointwise segmenter (RQ 2 benchmark) producing a per-element mask, compared on equal footing with the classical operators.
Selection	Pick best candidate	Among all candidates, retain the segment with the best coverage/compactness score before measurement.

3.6.2 Classical Segmentation Operators

The accuracy of the segmented elements relies on eliminating points which are part of other neighboring surfaces. As such, all classical operators rely upon the same normal-consistency criterion. A scan-point will be assigned to a particular BIM element only if the local surface normal matches the expected normal of that BIM element. In addition, we have implemented and tested the RANSAC plane-fitting algorithm as a comparative operator (RQ 2), however when used it only accepts planes whose normal vectors fall within +\- 30 degrees of the average normal vector of the BIM object; however, it was not considered as part of the candidate set used in our deployment.

The Plane-Normal (normal-consistency) criterion uses the cosine similarity metric as defined by equation (3.6) to evaluate the difference between the candidate surface normal extracted from the point cloud and the mean-normal vector for each respective BIM element. This criterion is utilized by both the graph-based operator and the RANSAC comparative operator; while in conjunction with a point-to-plane distance of 25 mm as part of the research framework.

$$\left|(\hat{n})_{(scan)} \cdot (\hat{n})_{(BIM)}\right| = |cos\,\theta| \geq \tau_{(normal)} \quad (3.6)$$

$(\hat{n})_{(scan)}$ represents the unit normal vector of the candidate plane determined from the scanned point cloud data; $(\hat{n})_{(BIM)}$ is the mean unit normal vector of the corresponding BIM element; θ is the angle between $(\hat{n})_{(scan)}$and $(\hat{n})_{(BIM)}$; and $\tau_{(normal)}$ is the cosine similarity threshold, which corresponds to an allowable angular deviation of 30 degrees ($\tau_{(normal)} = \cos 30° \approx 0.866$) since it is based on the cosine function.

The use of this criterion eliminates unwanted cross-surface contamination (typically neighboring wall leak patches with dimensions of roughly 50-60 cm), thereby allowing accurate statistical evaluation of deviations and ensuring that the retained segment is oriented similarly to its corresponding BIM element.

For each element, a set of complementary geometric operators is applied to generate candidates for segmentation: an adaptive growing bounding box; a graph-based method; and a density-based operator (DBSCAN), as well as a learned deep-network operator evaluated under RQ 2. The adaptive growing bounding box grows each element box outward by fixed-sized voxels until a stable number of enclosed points is achieved. It proves useful when neighbourhood relationships are poorly formed (i.e., sparse). Each operator produces a candidate, and all candidates are rated on both coverage and compactness. The candidate with the highest combined coverage and compactness rating is chosen and passed along to the next processing stage ("pick best candidate"), ensuring fair comparison and optimal segmentation for deviation measurement.

1. Adaptive Bounding Box Method:
   The element bounding box is dilated in fixed voxel increments until the enclosed point count stabilizes. This method serves as a fallback where neighbourhood relationships are too sparse for graph connectivity or density clustering to be reliable.
2. Graph-Based Method:
   The first method consists of combining normal vector filtering and graph-based connectivity. Points whose normals disagree with the closest BIM normal by more than the threshold specified in Equation (3.6) are excluded. Remaining points are linked into a k-nearest neighbour graph. The largest connected component is identified (Algorithm 3.2).
3. DBSCAN Method:
   Points filtered by normals are grouped into clusters using density-based clustering, DBSCAN. Clusters are isolated based on the density of points rather than proximity. It is particularly well-suited to consider tubes/columns/MEP piping etc.
4. Learned Network Method:
   Additionally, a learned network-based operator is used to create a per-element point mask; it represents a data-driven counterpart to compare with those produced by classical operators under equivalent conditions (RQ 2).

3.6.3 Cascaded Fallback and Fail Reason Registry

Instead of simply failing if a filter would clear the candidate set, each filter tracks an associated failure reason and goes back to the previous intermediate result where there were still candidates in it so that some sort of segmentation is always created and will be identified appropriately. All of the various failure reasons for which this could occur are listed in Table 3.6. These failure reasons are also utilized by the validation-status system in section 3.9 to inform the user regarding the quality of the segmentation that was responsible for generating each reported variation.

Table 3.6 Cascading-fallback fail reason registry

Fail_Reason	Meaning and Fallback Behaviour
"" (empty)	Full pipeline succeeded - result is *Valid*.
crop_fallback	No candidates within *seed_radius*; AABB-cropped points used.
normal_filter_fallback	Normal filter removed all points; proximity candidates used.
graph_fallback	KNN graph has no edges; normal-filtered candidates used.
candidate_relaxed	Selected candidate removed too many points; graph-component result used instead.
no_nearby_points	AABB crop contains zero scan points (no scan coverage).
low_coverage	Final result below the *min_points* threshold.
less_than_min_points	Fewer points than the minimum required for reliable measurement.

Algorithm 3.2 — Graph connectivity component extraction

```
# Input : normal-filtered candidate points P, edge_max_dist e, min_cluster_ratio m
# Output: retained points belonging to large connected components

function graph_components(P, e, m):
    tree   = KDTree(P)
    edges  = []
    for i in range(|P|):                        # build KNN graph
        for j in tree.knn(P[i], k):
            if dist(P[i], P[j]) <= e:
                edges.add((i, j))
    if edges is empty: return P, "graph_fallback"
    G       = sparse_graph(|P|, edges)
    n, lab  = connected_components(G)      # scipy.sparse.csgraph
    sizes   = bincount(lab)
    big     = max(sizes)
    keep    = { c : sizes[c] >= m * big }      # default m = 0.10
    return { P[i] : keep[lab[i]] }, ""
```

All of the above-mentioned operators are completely traditional in nature; they depend upon hand-crafted geometry-based prior knowledge and do not utilize semantic knowledge obtained through learning. Therefore, section 3.6.4 discusses an additional method for segmenting elements utilizing learned semantics and section 3.6.5 presents how both methods can be combined into a single hybrid cascade. Once every element has received a clean set of scan points, the remaining task is purely geometric: assessing how much the as-built surface differs from its planned location (signed distance) which is categorized based on context-specific tolerance ranges (section 3.7).

3.6.4 Deep Learning Architecture

Sections 3.6.1 – 3.6.3 describe how classical geometric encoding operators (based on bounding boxes and normal-consistency gates) are used to encode the geometric information of the environment. As stated in section 3.6.3, there is a second component that is learned based on deep learning that is based on a lightweight semantic segmentation algorithm that classifies each scan point into a structural

category like wall, floor, ceiling, column, beam, clutter, etc. It does this based on the geometry of the scan point itself and requires no hand defined BIM. Since research question 2 states that this module cannot depend on high-end GPU infrastructure to deploy, the methodology will focus on using publicly accessible architectures that are trained from pre-trained weights and run in inference mode on consumer-grade hardware. Therefore, the following sections are architecture and dataset independent, defining the workflow, data preprocessing and preparation, initialization/fine tuning strategies, and evaluation methods that apply to all candidate architectures. The specifics of realizing the architectures and performing empirical evaluations among them will be presented in chapter 4.

Candidate Architectures

Three types of neural network architectures are considered to represent the three major approaches to point-cloud semantic segmentation. These include PointNet++, representing the hierarchical Multi-Layer Perceptron paradigm; SpU-Net, representing the sparse-voxel convolutional paradigm based on a U-Net encoder-decoder architecture (specifically a variant of the Minkowski/MinkUNet family); and PTv3, representing the attention-based transformer paradigm. By evaluating all three architectures under an identical protocol, the impact of architectural inductive bias, rather than implementation details, can be isolated when comparing the models in Chapter 4. All architectures receive the same per-point feature representation and undergo identical preprocessing, thereby ensuring a fair comparison.

Data Preparation and Preprocessing

After converting each dataset into a single format, whether it is for training or inference purposes, each dataset is identical in its representation. Each scene is represented by an unordered list of points with a predetermined number of feature channels (Cartesian coordinates, colour, and (when available) locally estimated surface normals). All architectures are given the exact same input. The coordinates are centred around and scaled relative to each sampling unit. Color values are normalized over a unit range. Due to memory limitations, and to keep this lightweight architecture goal intact, a fixed grid resolution was chosen to downsample densely populated scenes. Scenes are segmented into spatially localized regions; specifically cube shaped regions for point-based and transformer models and sparse voxel grids for convolutional models where overlap exists to account for points close to unit boundaries being captured in multiple contexts. Additionally, because different datasets contain label taxonomies that do not match (wall vs. floor vs. ceiling vs. column vs. beam etc.), a structural class mapping (e.g. wall, floor, ceiling, column, beam and clutter/other) was created for comparison of semantic labels across source and target domains using a mapping function.

Pretrained Initialisation and Scope of Adaptation

The experiments reported in this dissertation use publicly released pretrained weights without additional training or fine-tuning, as specified in Section 1.2.4. Cross-domain evaluation applies those weights to a different dataset without updating model parameters. Fine-tuning on construction data is a proposed future extension, not an experiment demonstrated here.

Training and Evaluation Provenance

Training provenance must be distinguished from the evaluation performed in this study. The models were evaluated using the subsets reported in Section 4.5.1; their weights were obtained from existing checkpoints. No new six-fold cross-validation, construction-domain training or common optimiser schedule is established by the experiments reported here. Checkpoint training splits and any use of evaluation data for model selection must be documented to assess leakage and reproducibility.

Inference Pipeline

When running inference, the same process that occurred during training occurs for every input scene. Subsequent to being processed, the input scene is split into overlapping units. Each unit has the output of the network, or per-point class logit values, generated based upon processing the unit. The overlapping predictions that occur due to this processing are then resolved using aggregation of the per-point logits. This allows the points near the boundaries of different units to use the contextual information from the units next to them. Therefore, for each point within the input scan, the network generates a predicted structural class for each point as well as a normalized soft-confidence value representing that class. Due to voxel down-sampling reducing memory requirements during inference time, the network may be run on CPU-based systems as well as GPU-based systems, matching the objective of deploying the framework. The per-point predictions produced at the down-sampled resolution are then propagated back to the original resolution via nearest neighbour's assignments.

Hybrid Framework Integration

The output of this module (a per-point structural class label with an associated confidence) provides exactly the type of semantic priors that traditional operators are missing. It is not intended as a standalone segmentation result; instead, it is sent downstream as part of the hybrid cascade described in Section 3.6.5. Within this cascade, high-confidence non-structural predictions are utilized to remove clutter from the working cloud prior to applying traditional segmentation methods, and class labels are used to resolve geometric ambiguities between competing segments generated by traditional methods. By framing the learned segmenter as a semantic filter rather than as a substitute for traditional geometry-based operators, both paradigms can be combined without introducing either high-end GPU dependence or the domain shift fragility inherent to completely learned pipelines; this is formalized in the next section.

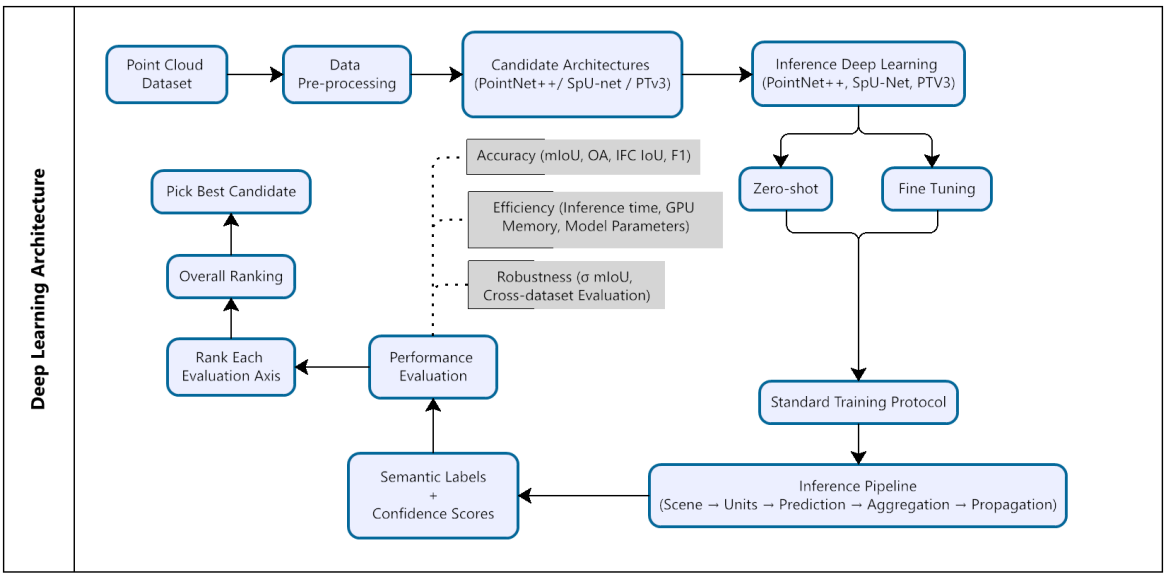


Figure 3.10 Deep Learning Architecture Pipeline

3.6.5 Hybrid Cascade Architecture

The proposed hybrid cascade DL first is an architectural combination of the previous two generations of operators, which have been shown to exhibit complementary properties:

Deep Learning Architectures

Deep learning architectures can be trained using 3D point clouds to find features from data. Since these models are able to determine geometric relationships without prior knowledge of BIM-priors, they require no BIM-priors. However, deep learning architectures generally consume large amounts of GPU memory, and it has not yet been determined whether they generalize well across all previously unseen construction environments - a topic investigated in empirical form in Section 4.5.3.

Classical Segmentation Operators

In contrast, the classical segmentation operators that were based on BIM-priors and were presented in Sections 3.6.1–3.6.3 provided geometric priors, e.g. bounding boxes, surface normal orientations, for guiding segmentation. As stated above, the classical segmentation operators are very good at imposing geometric constraints; however, they are heavily reliant upon their parameters. Additionally, when operating in noisy environments, they tend to degrade, a behaviour that will be characterized in Section 4.5.2. Therefore, there is a complementary property associated with combining both DL-based and BIM-guided segmentation approaches to produce a hybrid DL-BIM cascade architecture that allows classical segmentation operators to use BIM-priors to guide segmentation while simultaneously providing a semantic filter by way of a deep learning algorithm to clean up the input point cloud so as to provide a cleaner input for the subsequent classical segmentation operator.

The architecture will consist of five stages:

Stage 1 - Global DL Inference

This step conducts one pass through a pre-trained point-wise semantic segmentation model (here instantiated with SpU-Net weights that had been pre-trained on S3DIS), which produces a global set of points classified into structural element types (e.g. walls, floors, columns etc.), along with confidence values associated with those classifications.

Stage 2 - Semantic Clutter Purge

Semantic predictions were calculated prior to registration (§3.5) and these were applied after scanning using the scan to provide the same information for the entire geometric processing sequence. Therefore, there was a single forward pass of SpU-Net to accomplish two objectives: removing clutter from each element prior to registration and determining which object class has precedence when objects or surfaces intersected at walls and slabs during per-element segmentation. This is what the term DL-first signifies in the context of this model - it is not that machine learning is superior to other processes; rather, the one learned process is executed first and provides all necessary information for every subsequent geometric processing task.

Stage 3 - Classical BIM-Guided Segmentation on Cleaned Cloud

The point cloud produced in Stage 2 will be subjected to the same best-candidate classical segmentation pipeline that was presented in Sections 3.6.1–3.6.3. Wherever there may be multiple segmentations in regions of relatively equal coverage, the classification labels generated by the DL inference model (Stage 1) will serve as a tiebreaker.

Stage 4 - Proximal Recovery

Elements that failed in Stage 3 (therefore, no valid segment was generated) will have another opportunity to find an element by searching in the vicinity of the previously defined coordinates using the full original scan and its associated raw data. This gives elements affected by the DL purge another opportunity for recovery; it does not guarantee recovery.

Stage 5 - Coverage Limitation Reporting

All remaining elements that still have not been identified after processing all of the elements in Stage 4 will be assigned to one of several predefined failure categories (scanner coverage limits, algorithmic failure, insufficient IFC geometry). An audit log of all detected elements and/or failure reasons will be maintained and provided to the user.

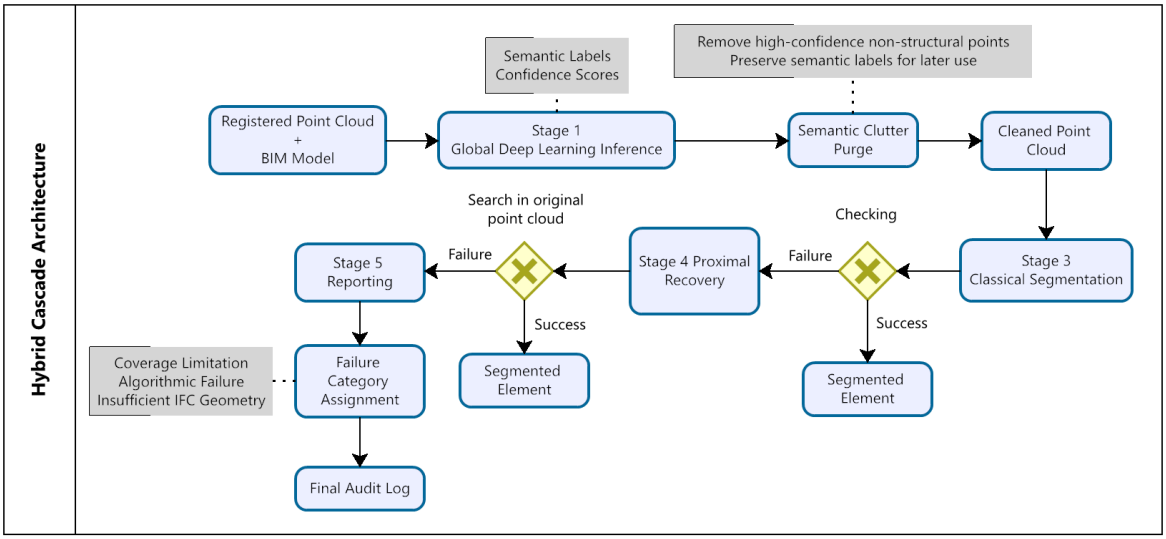


Figure 3.11 Hybrid Cascade Architecture

3.7 DEVIATION MEASUREMENT AND CONTEXT-AWARE CLASSIFICATION (CQA)

In essence, quality assessments of the construction elements are represented by one physical quantity (the distance between the constructed and planned surface) and one method of evaluation (whether or not the deviation will affect the functionality of the element). This framework calculates the size of the deviations using the magnitude of the point-to-plane deviation, whereas it decides using a tolerance classification that takes into account all relevant factors. At the end of this process, each element receives an assigned "severity" value which reflects what most engineers would write down as the reason for the defects noted in their defect register.

3.7.1 Point-to-plane Deviation

To determine a point-to-plane deviation for all points p on a segmented element, the framework first uses an Open3D KDTreeFlann to find the closest sampled BIM reference point q and then projects the residual onto the unit BIM surface normal at q, takes the absolute value, see equation (3.7). In cases where the BIM mesh has no normals, they are calculated beforehand by performing a hybrid k-d-tree search over a radius of 0.20 metres and up to twenty neighbours. By projecting the residual onto the surface normal (as opposed to calculating the distance from p to q), the framework provides a local point-to-plane approximation of surface deviation, one that is not dependent upon how far p may slide along the surface relative to q. From the point-by-point deviations computed by the framework, for each element, the framework computes the mean, maximum and standard deviation of these point-by-

point deviations. The framework also computes the RMS Error of Equation (3.8) as its head-line measure of accuracy for use in subsequent evaluations.

$$d(p) = \left|(p - q) \cdot \widehat{n_q}\right|, \quad q = arg\ \min_{q' \in \mathcal{M}} \|p - q'\| \tag{3.7}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d(p_i)^2} \tag{3.8}$$

In order to provide support for visualization, the registered scan is coloured with a jet colour map such that deviations equal to zero are represented as blue, and those representing the greatest deviation recorded for each element is represented as red. A transition occurs when moving from blue toward red via green and yellow. To maintain responsiveness in the web-based browser environment, the viewer will both subsample the scan down to 200 000 points and limit the number of triangles displayed in the IFC mesh to 400 000.

Algorithm 3.3 — Point-to-plane deviation computation

```
# Input : segmented scan points P_e for element e; BIM mesh M_e (with normals)
# Output: mean, max, std, RMSE of signed point-to-plane deviation

function deviation(P_e, M_e):
    if M_e.normals is empty:
        M_e.estimate_normals(radius=0.20, max_nn=20)
    tree = KDTreeFlann(M_e.points)
    d    = []
    for p in P_e:
        q   = tree.nearest(p)              # closest BIM surface point
        n   = M_e.normal_at(q)               # unit surface normal at q
        d.add( abs( dot(p - q, n) ) )       # Eq. (3.7)
    return mean(d), max(d), std(d), sqrt(mean(d^2))   # Eq. (3.8)
```

3.7.2 Context Aware Tolerance Classification

While a five-millimetre deviation may be inconsequential in regard to a door, it represents a significant deviation in terms of a structural column. Furthermore, tolerance levels vary across different types of buildings. Therefore, the framework will automatically detect the project context based on specific IFC metadata keywords (i.e., storey and space names and/or project name) and select from the following categories: hydraulic structures, bridges, tunnels, industrial, education, medical, residential, commercial. If none of these categories apply, it will default to a clearly labelled general profile rather than defaulting silently to any specific profile. For the hydraulic structure case study of Chapter 4, the HydraulicStructure profile is used deliberately. Once selected, the context category will determine which tolerance table is to be applied. Tolerances applicable to hydraulic structures were determined by referencing the execution tolerances defined in EN 13670 (Execution of Concrete Structures - Class 2) and EN 1992-3 (Liquid-Retaining and Containment Structures). The tolerance tables for hydraulic structures are listed in Table 3.7. Each individual element is subsequently evaluated according to its mean deviation and assigned a severity rating based upon whether its mean deviation exceeds (or falls below) three threshold values as defined in Table 3.8 and illustrated in Figure 3.12.

Table 3.7 Context-aware tolerance table for the HydraulicStructure domain (basis: EN 13670 Class 2 and EN 1992-3; circulation and openings per ISO 4463-1)

IFC type	Importance	Compliant	Minor	Moderate	Standard basis
IfcWall	High	≤5 mm	≤15 mm	≤30 mm	EN 13670 Cl.2 + EN 1992-3
IfcSlab	High	≤5 mm	≤15 mm	≤30 mm	EN 13670 Cl.2 + EN 1992-3
IfcColumn	High	≤5 mm	≤10 mm	≤20 mm	EN 13670 Cl.2
IfcBeam	High	≤5 mm	≤10 mm	≤20 mm	EN 13670 Cl.2
IfcMember	High	≤5 mm	≤10 mm	≤20 mm	EN 13670 Cl.2
IfcFooting	High	≤8 mm	≤20 mm	≤40 mm	EN 13670 Cl.2 + EN 1992-3
IfcPile	High	≤8 mm	≤20 mm	≤40 mm	EN 13670 Cl.2 + EN 1992-3
IfcRoof	Medium	≤8 mm	≤20 mm	≤40 mm	EN 13670 Cl.2
IfcStair	Medium	≤8 mm	≤20 mm	≤40 mm	ISO 4463-1
IfcPipeSegment	Medium	≤10 mm	≤25 mm	≤50 mm	ISO 4463-1
IfcDoor	Low	≤15 mm	≤30 mm	≤50 mm	ISO 4463-1
IfcWindow	Low	≤15 mm	≤30 mm	≤50 mm	ISO 4463-1

Table 3.8 Severity levels assigned from the mean deviation and the type tolerance thresholds

Severity	Condition on mean deviation
Compliant	$\leq t_{compliant}$
Minor	$> t_{compliant}$ and $\leq t_{minor}$
Moderate	$> t_{minor}$ and $\leq t_{moderate}$
Major / Critical	$> t_{moderate}$

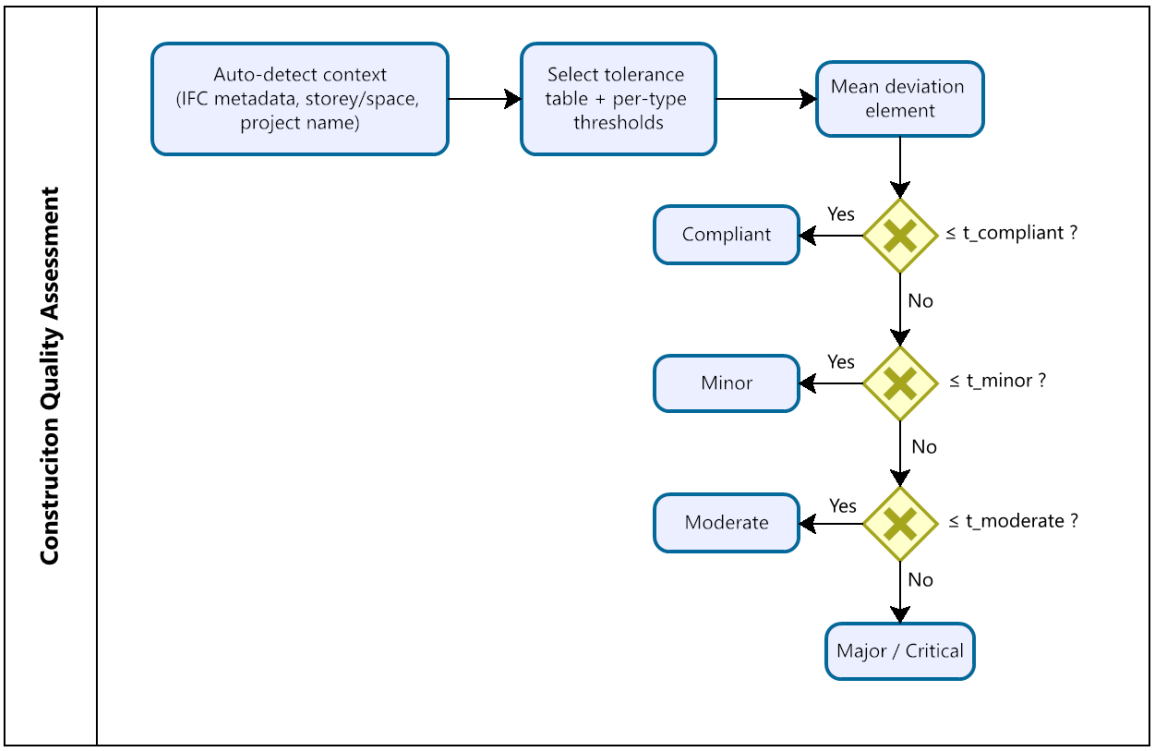


Figure 3.12 CQA severity-classification logic

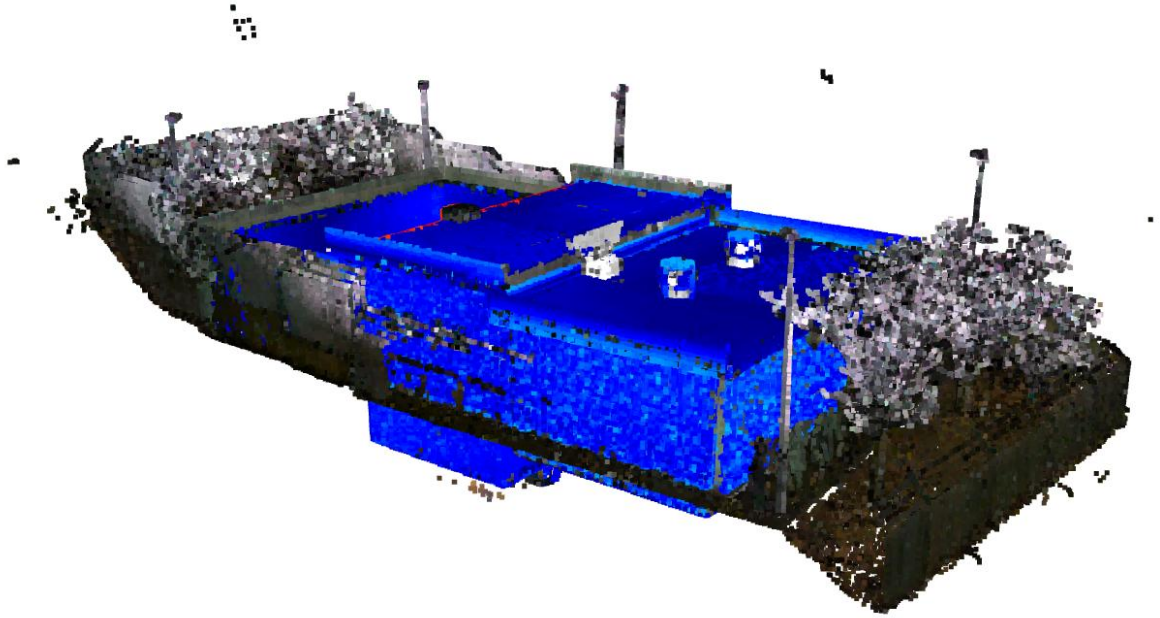


Figure 3.13 Jet-coloured deviation point cloud over the grey BIM reference (CQA viewer state)

The severity ratings developed during CQA will contribute to the export schedules (Section 3.10). However, quality assessments of construction presume that there is confidence in the BIM model against which the scan was compared. As such, the framework includes another assessment related to the quality of the digital model itself, described in the following section.

3.8 MODEL QUALITY ASSESSMENT (MQA)

Model Quality Assessment evaluates how well a BIM model represents the physical characteristics of the real-world built facility without regard to whether that actual facility meets the specifications contained in the digital model. As noted earlier, while Construction Quality Assessment fixes the model in place and examines the properties of the physical structure, MQA assumes that the physical structure has been accurately aligned with the digital model (in other words, the "registration" process has produced accurate spatial reconciliation); hence, each is taken into consideration by MQA, but neither performs registration, nor is the alignment error of Section 4.4 considered as an input.

For each item within the model, a quality rating is assigned on a scale of 0–100 based upon the ratio of its mean deviation to an item-specific tolerance (see Equation 3.9), a letter grade that will be determined using the criteria outlined in Table 3.10 and a separate pass/fail determination made by applying the engineering safety-net check of Equation 3.10. The safety net prevents the favorable mean deviation from masking significant localized errors through failure when the root-mean-square error (RMSE) is greater than 1.5 times the tolerance value regardless of the mean.

3.8.1 Geometric Deviation Assessment

While all IFC entities exist within a typical IFC-based model, not all represent entities that have assessable geometric data (e.g., IfcProject, IfcSite, IfcBuilding, IfcBuildingStorey, IfcSpace). Moreover, there are IFC entities referred to herein as container entities (e.g., IfcCurtainWall) where the geometry associated with the entity exists elsewhere in the model (as part of the decomposed components of the container e.g., IfcPlate panel members and IfcMember mullion members) rather than being defined directly by the container. Evaluating both a container and its constituent parts together could cause duplicate counting. Therefore, MQA focuses on a filtered subset of entities that meet three conditions:

(i) Belong to one of the entity types identified in Table 3.2.
(ii) Contain valid B-Rep data from which a reference surface may be generated.
(iii) Are not decomposition containers for elements already in the set.

3.8.2 Quality Scoring and Grade Assignment

Each component type is allocated to a predefined geometric tolerance according to its corresponding IFC element type, shown in Table 3.9; e.g., structural elements (columns, mullions and curtain-wall panels) are assigned a tolerance of ±10 mm, whereas architectural and general elements (walls, slabs, doors, windows and coverings) are assigned a tolerance of ±15 mm. An equation (Equation 3.9), as well as a calibration table (Table 3.10), are used to convert the mean deviation of each component into a numerical quality score ranging from 0 to 100. The calibration table shows how mean deviations of 0, 1.5, 3.75, 6 and ≥15 mm correspond to quality ratings of 100, 90, 75, 60 and 0 respectively. The quality rating is then translated into a letter grade according to Table 3.10. Additionally, an engineering-related safety net that operates independently of the score (Equation 3.10) fails any component if RMSE is greater than one-and-a-half times the tolerance. The reason for this is that while a low mean deviation can hide a significant local deviation, the latter represents a potentially serious deficiency. The entire process for evaluating the quality of components is presented in Algorithm 3.4.

Algorithm 3.4 — Component quality scoring with engineering safety net

```
# Input : element e with mean, max, RMSE (mm); IFC type t
# Output: quality score [0..100], letter grade, PASS/FAIL

function component_quality(e, t):
    tol   = TOLERANCE_TABLE[t]          # Table 3.9
    score = interp(e.mean, CALIB)       # Eq. (3.9), Table 3.10
    grade = "A" if score>=90 else "B" if score>=75 else "C" if score>=60 else "D"
    fail  = (e.max > tol) or (e.rmse >= 1.5 * tol)  # Eq. (3.10)
    return score, grade, ("FAIL" if fail else "PASS")
```

$$Score = 100\left(1 - min\left(\frac{Mean\ Deviation}{Tolerance}, 1\right)\right) \tag{3.9}$$

$$verdict = \begin{cases} \boldsymbol{FAIL} & if\ max\ deviation > tolerance\ or\ RMSE > 1.5 \cdot tolerance \\ \boldsymbol{PASS} & otherwise \end{cases} \tag{3.10}$$

Table 3.9 Geometric tolerance profile by IFC type

Geometric Tolerance	IFC types
±10 mm	IfcBeam, IfcColumn, IfcMember, IfcPlate, IfcPipeSegment, IfcDuctSegment, IfcFlowFitting, IfcFlowTerminal
±15 mm	IfcWall, IfcSlab, IfcRoof, IfcStair, IfcRamp, IfcDoor, IfcWindow, IfcCovering, IfcFooting, IfcPile

Table 3.10 Quality-score calibration (tolerance = 15 mm) and letter-grade thresholds

Mean deviation	Quality score	Grade	Threshold
≤ 1.5 mm	90–100	A (Excellent)	score ≥ 90
3.75 mm	75	B (Good)	75 ≤ score < 90
6 mm	60	C (Fair)	60 ≤ score < 75
> 6 mm (score = 0 at ≥ 15 mm)	0–59	D (Poor)	score < 60

This section defines a quality assessment that is relationship-based. It provides links from the elements of a building to its levels/floors, associated tolerance profiles and applicable standards. This relationship-based information provides the context needed to assess the geometric deviation in individual building elements and to ascertain if these elements meet their respective quality requirements. The results of this quality assessment will be used as the foundation upon which to evaluate and report on the quality of the models produced by the proposed Scan-to-BIM process.

3.9 VALIDATION STATUS SYSTEM

The size of the deviation value does not measure how well an element has been segmented. Therefore, the framework assigns to each element a Deviation_status that indicates the quality of the segmentation (not the magnitude of the deviation) by converting the internal failure causes from section 3.6 into readable statuses, shown in Table 3.11. An element will be valid if it completes the entire process of being processed through all stages of the process without having to go into a fallback that requires hand inspection for validation. The need validation status indicates the point in the process where the cascade failed, giving the engineer time to view the item before using its measured deviation. The separation of confidence in measurements from the actual values of those measurements is an intentional decision based upon methodology that would otherwise allow false precision to occur due to total automation of quality assessments.

Table 3.11 Validation status system

Deviation_Status	Triggering condition
Valid	Full pipeline succeeded.
Review	The element's mean deviation lies within 2 mm of a severity-band boundary. The measurement is valid, but the classification may change because of minor numerical variation and should therefore be confirmed by visual inspection.
Need Validation (No Nearby Points)	Bounding-box crop contains zero scan points.
Need Validation (Crop Fallback)	No seed-proximity candidates; cropped points used.
Need Validation (Normal Filter Fallback)	Normal filter rejected all points.
Need Validation (No Graph Connectivity)	KNN graph has no edges.
Need Validation (Candidate Relaxed)	Selected candidate removed too many points; graph result used.
Need Validation (Low Coverage)	Final cluster below the *min_points* threshold.
Need Validation (Less Than Min Points)	Below the minimum required for reliable measurement.
Need Validation (Not Segmented - No Scan Coverage)	Element present in the model but missed by the scan.
Need Validation (No Geometry / Tessellation Failed)	IFC element has no computable mesh.

The review status is different from all need validation statuses. Need Validation shows errors with the input data or the segmentation process (such as insufficient scanner coverage, no geometry, poor segmentation) that will result in an inability to generate a totally reliable measurement for the framework. The Review shows that the actual measurement was valid; however, it is within two millimetres of a severity band limit and therefore is classified into one severity category while a slight numerical difference would place it into another. Distinguishing between elements that could not have

been measured and elements that can be measured but may require confirmation of their assigned severity class through engineering judgment is shown by reporting both types of elements.

Now that every element carries both a deviation and a severity/quality grade along with true validation status, the framework combines the output together to create two outputs that fit within the broader BIM-ecosystem-the export package and interactive viewer described below.

3.10 EXPORT PACKAGE AND BIM-INTEGRATED DELIVERABLES

The real value of an assessment can only be realized when the assessment has left the tool and entered into the engineer's normal work environment. Thus, the framework exports a complete package, separate for CQA and MQA, but structurally parallel, whose content is shown in Table 3.12. Each package includes a machine-readable, by element, CSV file, a Pset_DeviationSchedule property set-annotated IFC4 model for use with BIM tools, an IfcGroup to support Autodesk Revit scheduling, a jet-coloured registered scan in E57 format, and an interactive HTML schedule report. The annotated IFC is key to integrating the deviation results into the BIM workflow. It allows the deviation results to be opened and scheduled within Revit and/or interrogated using IFC viewers like BIMcollab Zoom and Solibri, thus closing the loop back to the authoring environment.

Table 3.12 Contents of the CQA and MQA export packages

	CQA package	MQA package
1	Per-element deviation CSV (CustomName, ElementID, GlobalId, FamilyName, point counts, Mean / Max / Std / RMSE, Importance, Severity, Tolerance, Standard, Deviation_Status, BIM volume)	Per-element deviation CSV (CustomName, ElementID, GlobalId, FamilyName, Method, Severity_Status, Severity and Deviation_Status)
2	Construction-quality CSV	BIM-quality CSV
3	Annotated IFC4 (Pset_DeviationSchedule + IfcGroup for Revit schedules)	Annotated IFC4 (per-element jet colours + Pset_DeviationSchedule)
4	Jet-coloured registered scan (.e57)	Jet-coloured IFC per element
5	Interactive HTML schedule report	Interactive HTML schedule report

An interactive viewer was developed using Three.js (Release 152), which was imported via an ES-module import map. The viewer displays four distinct views that correspond to the flow through the process outlined in Table 3.13. The scan positions were encoded as base64-encoded 32-bit float buffers that were subsampled down to 200,000 points. The colours were also encoded as 32-bit floats and converted to base64 encodings. The IFC mesh was also encoded as a non-indexed (i.e., vertex expanded) triangle buffer that was capped at 400,000 triangles due to stride-based subsampling. On the client-side, the centre of the scene is calculated from the position buffer centroid. Orbit controls are used for navigation. The same jet colour map, blue through cyan, green, and yellow to red, is used to represent increasing levels of deviation in both the CQA and MQA states. This provides the user with a consistent visual representation between the two assessments.

Table 3.13 The four states of the Three.js interactive viewer

State	Scene composition
Upload	Scan (blue) + BIM (orange), in original frames
Registration	Scan (blue) + BIM (orange) + fitness / RMSE overlay
CQA	Jet-coloured scan + grey BIM point cloud
MQA	Grey scan + jet-coloured IFC mesh

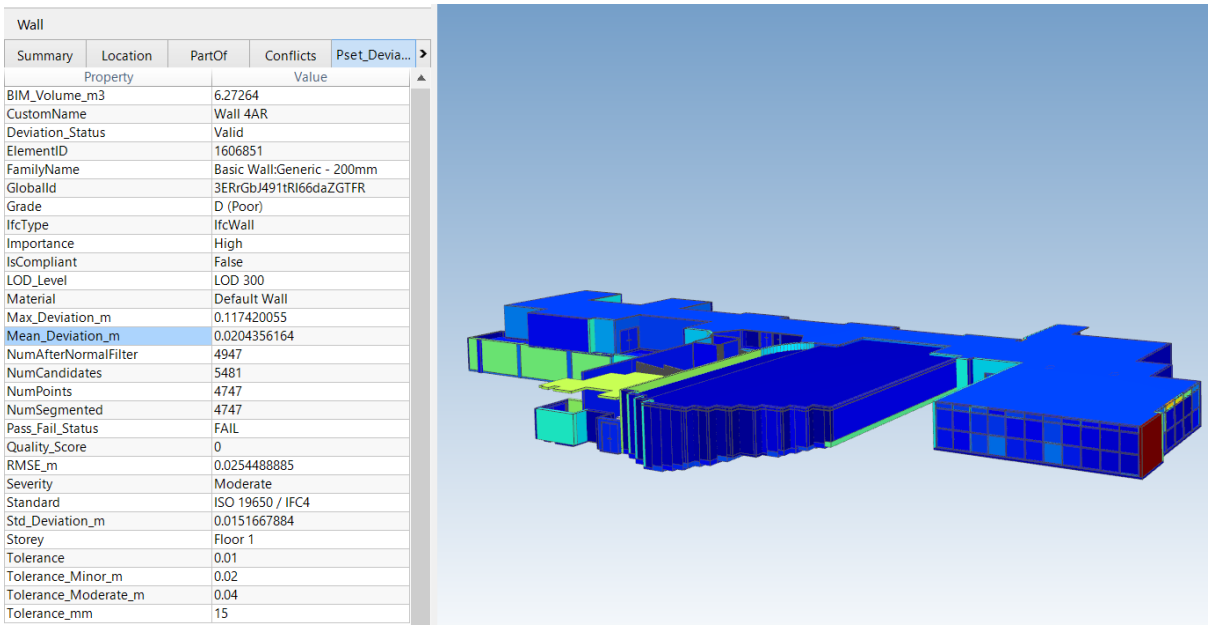


Figure 3.14 Visual reference for the viewer and the BIM-integrated export deliverables

The deliverables presented above describe the tangible outputs of the framework. However, the scientific merit of these deliverables will depend upon how thoroughly the framework is tested. Therefore, the final section describes the test protocols and procedures that connect the methodological framework back to the research objectives.

3.11 EVALUATION METRICS AND COMPARATIVE PROTOCOL

The framework will be compared to the three research questions outlined in section 1.2.2 through use of a common set of evaluation measures - accuracy, speed and robustness. This allows for each part of the process to be individually assessed, but also maintains the relationship between accuracy and practicality of the whole pipeline throughout.

Registration (RQ1). Alignment quality is reported using fitness (the fraction of source points with a target correspondence within the evaluation distance) and the inlier RMSE. Candidate selection uses Equation (3.5). Computational cost is measured by wall-clock time. The global operator is varied while the remaining stated parameters are held constant.

Segmentation (RQ2). Semantic segmentation performance is traditionally represented through the following metrics: per class Intersection-over-union (IoU), average IoU ($mIoU$) across all structural classes, per class F1 score, overall accuracy (OA), and average class accuracy (mAcc). For each class c,

TP_c represents the true positives, FP_c the false positives and FN_c the false negatives. Where C represents the total number of classes:

$$IoU_c = \frac{TP_c}{TP_c + FP_c + FN_c} \quad (3.11)$$

$$F1_c = \frac{2TP_c}{2TP_c + FP_c + FN_c} \quad (3.12)$$

$$mIoU = \frac{1}{C}\sum_{c=1}^{C} I\,oU_c \quad (3.13)$$

$$OA = \frac{\sum_{c=1}^{C} T\,P_c}{N_{total}} \quad (3.14)$$

Equation 3.13 calculates the average of equation 3.11 for each of the C classes, while equation 3.14 shows the percentage of correctly identified points out of a total of N_{total} points within the cloud.

Segmentation accuracy has been tested in two ways. Performance in-domain was tested by using a separate portion of the training data, whereas cross-domain performance was tested by evaluating models trained on one dataset with another dataset. The difference between these represents the amount of domain drift and provides evidence for why a hybrid cascade approach was selected as shown in section 3.6.

Efficiency: computational cost is reported as wall-clock time and measured memory use, with CPU host RAM distinguished from GPU VRAM. Geometric operations run on CPU, whereas the reported deep-learning inference measurements use CUDA. CPU inference times were not recorded, so CPU-only performance of the learned component is not established.

Robustness: Stability has been assessed on two dimensions. Firstly, run-to-run stability was evaluated through repeated runs of each stochastic operator (notably RANSAC) with a fixed seeds and by calculating the coefficient of variation for each metric.

$$CV(m) = \frac{\sigma(m)}{\mu(m)} \quad (3.15)$$

Secondly, the interdomain generalization gap mentioned previously represents how sensitive the methods are to varying in scene type, sensor, acquisition method etc.

The robustness of the operators to data degradation was assessed because as-built scans taken on active sites have been influenced by the inherent noise associated with sensors, occlusions and clutter, every classical operator has also been stress tested against controlled degradation of the input instead of only evaluating them on clean data. Three degradations were added to each scene to simulate range measurement error, occlusion and the rough dusty environment found at active construction sites; these included random removal of 20% of points from the scan data, additive Gaussian noise (σ = 0.02 m) to simulate the errors created when measuring ranges and displacements of surface normal vectors. The loss in mIoU due to the introduction of degradations was used as a measure for how robust the

operators were, and those operators where their mIoU fell by greater than 15% from clean to degraded state were considered sensitive to noise, while operators with less than a 15% fall in mIoU could be directly applied to raw scan data. The results of the stress testing of the operators are presented in Section 4.5.2.

Comparison protocol: In order to compare fairly, each competing algorithm attempting to solve the problem of finding non planar elements would need to operate at an identical voxel resolution and neighbourhood configuration. All algorithms were therefore constrained to perform at an equivalent voxel resolution and neighbourhood configuration when comparing them (as defined in section 1.2.4).

Under similar constraints, the simple pipeline was compared with three deep learning approaches for segmenting point clouds - PointNet++, SpU-Net, and PTv3, where the algorithms were compared on a per class basis for accuracy and GPU inference time under limited resource allocation. A full description of both in-domain vs cross-domain behaviour along with the resultant model selection is given in chapter four.

Assessment quality (RQ3): The assessment stage was evaluated against two criteria: (i) Deviation RMSE relative to reference data which had been created under controlled conditions; (ii) Robustness indicators representing the proportion of elements tagged as needing manual verification. These criteria were applied directly to both the per-element csv reports and the jet-coloured deviations export referred to in section 3.10 to ensure that reported defects could be traced back to specific elements as opposed to aggregated scene-based statistics.

In combination, this procedure maintains both accuracy and computational feasibility for each stage as illustrated in section 3.7. The experimental design, dataset properties and results achieved under this procedure are presented in chapter 4.

3.12 METHODOLOGICAL LIMITATIONS AND ERROR BUDGET

The proposed Scan-to-BIM Quality Assessment Framework has similar practical constraints to any other Scan-to-BIM quality assessment methodology. The accuracy of point cloud measurements depends upon the data quality of the original scan. This includes sensor noise, the number of points scanned per unit area (point density), potential for occlusion, and how well the points are registered to their correct location. Sensor noise, and surface roughness will also potentially introduce random error into the measurement of point-to-plane distances. Incomplete scans, and incorrect classification of objects during semantic segmentation can both cause propagated errors through to the final quality assessment.

Additionally, the voxel resolution chosen as v = 0.10 m provides a practical lower bound on the spatial resolution of deviation measurement. Using Equation 3.7, when computing the point-to-plane distance the spatially averaged geometry inside a single voxel (a "cell") is treated as if it were an average value over space, which acts as a spatial low pass filter. Therefore, deviations from nominal form that are less than about half the voxel dimension (approximately 50 mm) cannot be differentiated from sampling/voxelization artifact.

Since section 3.7.2 uses millimetre thresholds (i.e., 5 mm for wall, slab, column, beam) for determining compliance verdicts, then it is necessary to identify the measurement uncertainty. The total measurement uncertainty of a reported element deviation can be expressed as a root-sum-of-squares combination of four error terms, if they can be treated as independent standard uncertainties:

Table 3.14 Error budget for a reported element deviation

Source	Symbol	Origin and order of magnitude
Sensor noise	σ_sensor	Range noise and beam divergence of the terrestrial scanner; typically 1–3 mm at working range.
Registration error	σ_registration	Residual inlier RMSE of the rigid transform selected by Equation (3.5); reported per case study in Chapter 4.
Discretisation	σ_discretisation	Voxel down-sampling at v = 0.10 m acts as a spatial low-pass filter; the sampling uncertainty within a voxel is v/√12 ≈ 29 mm, and features below v/2 = 50 mm cannot be separated from voxelisation artefact.
Segmentation leakage	σ_segmentation	Residual points from adjacent surfaces retained after the normal-consistency filter; bounded by the coverage/compactness score and flagged through the validation status of Table 3.11.

$$\sigma_{total} = \sqrt{\sigma_{sensor}^2 + \sigma_{registration}^2 + \sigma_{discretisation}^2 + \sigma_{segmentation}^2} \quad (3.16)$$

As stated previously, the registration benchmark residuals (76.8–104.3 mm) are larger than the illustrative discretisation term (approximately 29 mm). Therefore, the values defined as compliant thresholds in table 3.7 (5 mm for walls, slabs, columns and beams) fall one order of magnitude below the resolution limit of the pipeline at v = 0.10 m and the registration residuals reported in chapter 4 are above these compliant thresholds. Based upon this, three conclusions have been made.

Firstly, the framework will report all of the standard-based thresholds shown in table 3.7 because these are the thresholds that an engineer must use when evaluating a building. However, when an engineer reads a verdict of compliant, it should be interpreted as "the element deviation cannot be distinguished from compliant with current resolution". Secondly, moderate or major/critical classifications should be treated as indicative measurements pending independent validation, because registration residuals exceed the relevant tolerance thresholds. Finally, if the engineer wants to determine whether or not an element deviates beyond the minor boundary then it must be possible to resolve this boundary with a smaller voxel size than used currently and a registration residual less than the threshold. A voxel-size sensitivity study is proposed as future work in Chapter 5.

Many of the thresholds listed in Table 3.7 have values smaller than what can be measured accurately using voxels with a resolution of v = 0.10 m. While the framework correctly reports all thresholds set forth in Table 3.7, there is no doubt that any judgments regarding compliance at scales smaller than one-half voxel diameter will include greater uncertainties. This section defines the error budget symbolically; the limitations of separating these contributions are discussed in Section 4.9.2.

3.13 SUMMARY

This chapter described an entirely automatic, lightweight computational Scan-to-BIM process for assessing quality of BIM models of construction projects. This procedure accepts as-built laser scanning data in E57 format and as-designed BIM data in IFC format. It then aligns these two models by converting the BIM geometry into an artificial reference cloud with stable element identities and metadata, while retaining the scan as a separate point cloud. Next it registers each of the clouds by using a centroid seeded coarse/fine candidate registration algorithm. Then it assigns evidence from the scans to each element within the artificial reference cloud. After that, it calculates point to plane deviations between the scan points and planes in the artificial reference cloud. Finally, it provides a construction verdict based upon the deviation values (severity based upon context) and a model verdict in the form of a geometric quality rating based on element-specific tolerance. In addition, this application will produce output files (a BIM integrated export package with an interactive viewer) containing the results. Each module was selected for computational tractability on commodity CPUs, with the stated accuracy limitations. Methodologically, all modules were developed according to principles found in the geometric processing literature. Furthermore, the development of the modules followed established practices for executing construction projects, such as those found in EN 13670 and EN 1992-3.

Table 3.15 Summary mapping of methodological stages to the research questions (RQ)

Methodological stage	Section	RQ addressed	Principal contribution
Architecture & overall pipeline	3.1 – 3.2	RQ [Primary]	CPU-bound, single-worker, REST-based lightweight design.
Input & pre-processing, BIM preparation	3.3 – 3.4	RQ [Primary]	World coordinate ingestion, artificial PCD, back-face filtering.
Registration pipeline	3.5	RQ 1	FGR vs RANSAC + ICP, multi-candidate scoring.
Segmentation pipeline	3.6	RQ 2	Classical pipeline → RANSAC plane, growing method, graph/normal, DBSCAN Deep Learning → PointNet++, SpU-Net, PTv3
Deviation & CQA	3.7	RQ 3	Point-to-plane deviation, context-aware severity.
MQA	3.8	RQ 3	Element-specific tolerance assessment.
Validation status	3.9	RQ 3	Separation of measurement confidence from measurement value.
Export and viewer	3.10	RQ [Primary]	Annotated IFC, deviation E57, Three.js viewer.
Evaluation protocol	3.11	RQ [Primary]	Accuracy, efficiency and robustness; CPU geometry and GPU inference.
Limitations & error budget	3.12	RQ [Primary]	Quantified resolution floor and its consequences for compliance verdicts.

4
CASE STUDY AND EVALUATION

4.1 OVERVIEW

The previous chapter outlined a step-by-step lightweight end-to-end Scan-to-BIM approach designed to operate on consumer-grade hardware with lightweight CPU and GPU requirements. The current chapter will examine this lightweight end-to-end Scan-to-BIM approach empirically using two real-world case studies and, thus, provide the data-based answers to the primary research question and three supporting research questions in Chapter 1, Section 1.2.2. Each of the research questions will be addressed through a distinct experiment with its own specific protocol, metrics and discussion.

This chapter follows the comparative methodology established in Section 3.11. Section 4.2 outlines the case study structure and the two input artifacts used to evaluate the as-built E57 scan and the as-designed IFC model. In addition, Section 4.3 presents information related to the experimental setup (i.e., the parameter settings, evaluation metrics, etc.) with respect to the evaluation of each metric stated in such detail as to ensure clarity of interpretation in subsequent results. The registration process is evaluated in Section 4.4 by varying the global registration method between FGR and RANSAC while keeping all other parameters constant (RQ 1). Next, Section 4.5 assesses the segmentation stage by evaluating the four traditional methods relative to one another, followed by testing the new geometry driven pipeline against three typical deep learning architectures operating under resource-constrained conditions with consumer-grade GPU acceleration (RQ2). Finally, Sections 4.6 and 4.7 report on both CQA and MQA. Section 4.8 reports on a consistent comparison of timing and memory usage for each of the stages required to complete the entire end-to-end workflow in order to validate end-to-end computational feasibility on commodity hardware (primary RQ). Lastly, Section 4.9 provides a summary of the findings from all experiments performed in this chapter along with an outline of potential threats to validity and Section 4.10 summarizes this chapter.

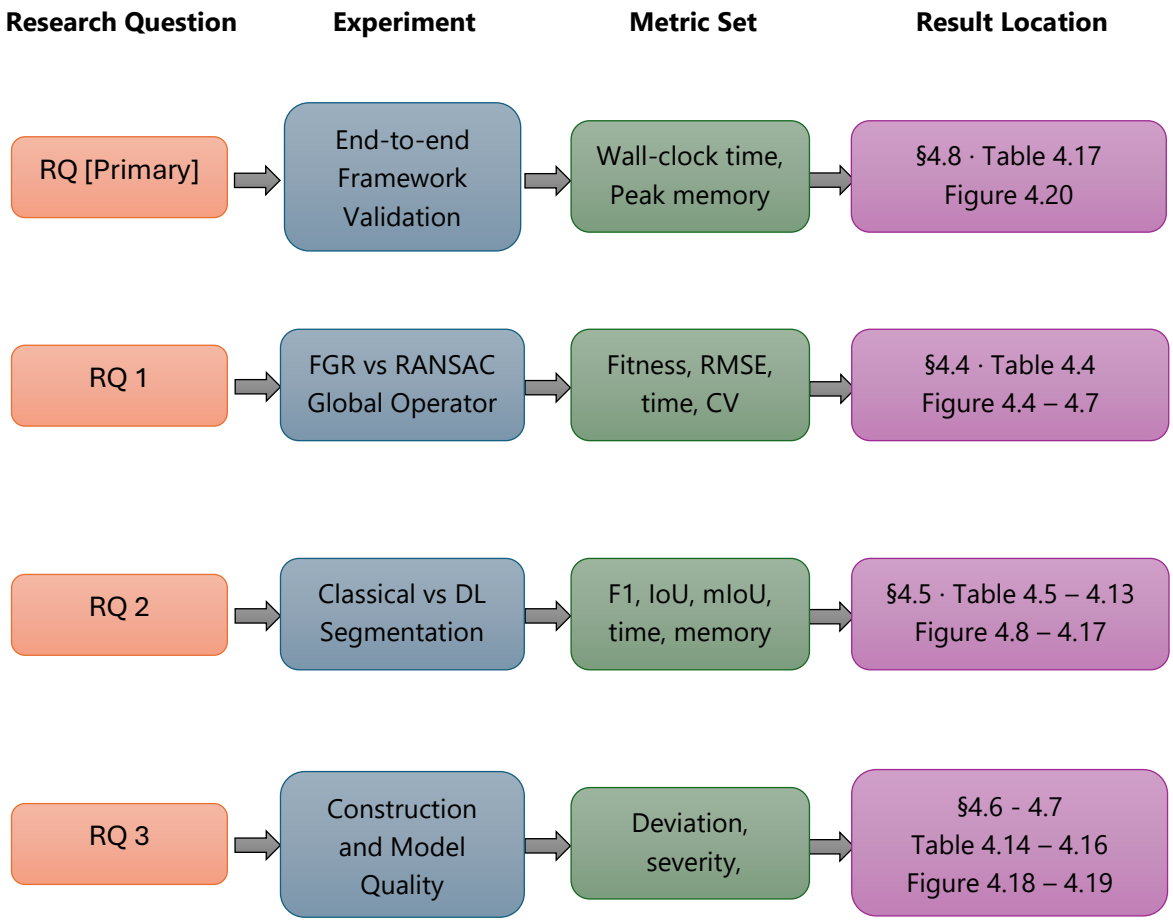


Figure 4.1 Evaluation roadmap mapping research questions to experiments, metrics, and results

4.2 CASE STUDY DESCRIPTION

The Scan-to-BIM framework was tested using a reinforced concrete hydraulic structure. It should be noted that reinforced concrete hydraulic structures were specifically chosen due to their very restrictive execution tolerances within the European Standard Framework (EN 1992-3 in combination with Class 2 Execution Tolerances of EN 13670) making them a difficult test case for an automated, tolerance driven quality assessment pipeline. Therefore, the Hydraulic Structure tolerance profile presented in Section 3.7.2 (Table 3.7) will be utilized during the evaluation process. The second is an example of a multistorey building at a university ("the UC Building"). The reservoir case study evaluates construction quality while the UC Building evaluates model quality. The first was to report on construction quality in terms of how well the constructed reservoir met its design specifications; there was an established tolerance for the execution, and it was the as-built condition that was being assessed. The second was to report on model quality using the University's rich and complex IFC model (as opposed to the simpler reservoir model with five modelled storeys), which tested the element specific tolerance logic outlined in Section 3.8.

4.2.1 As-built Point Cloud

The as-built geometry was delivered as one or multiple scans in the ASTM E57 exchange format and loaded using the pye57 library with station pose data baked into project/world coordinates as detailed in Section 3.3. The acquisition system's resolution, coverage and frequency of sampling determine both the noise characteristics, point density and occlusion patterns encountered by downstream operators and are documented below in full detail.

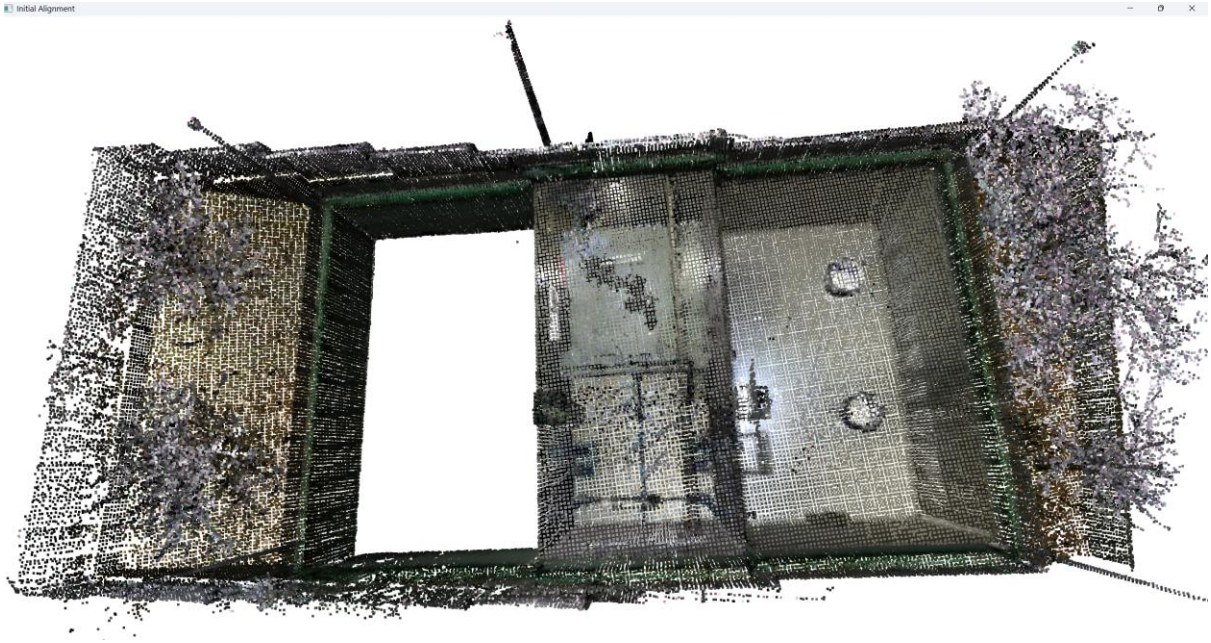


Figure 4.2 The case-study hydraulic structure raw as-built point cloud

4.2.2 As-designed BIM Model

The as-designed geometry was provided as an IFC model (either IFC2X3 or IFC4 schema) and parsed with ifcopenshell using world coordinates, ensuring that both scans share the same reference frame after reconciliation of coordinates. The distribution of elements across comparison families determines the balance of planar and non-planar geometry evaluated by the common candidate-selection pipeline.

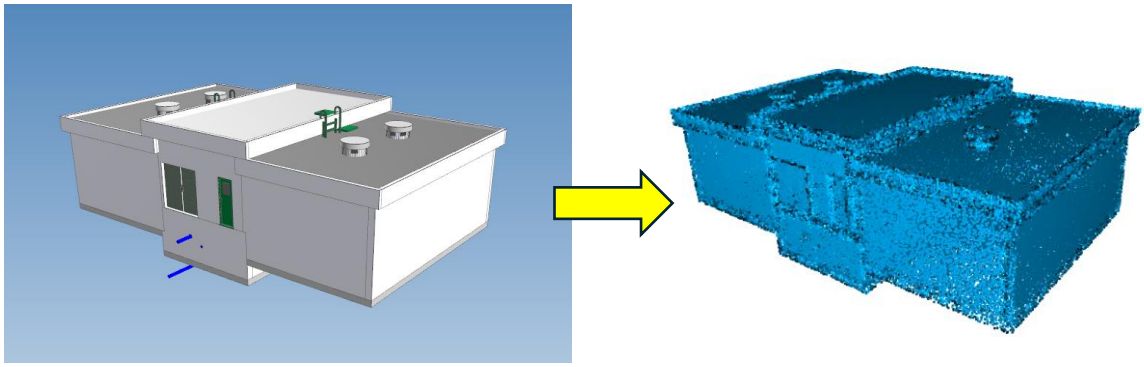


Figure 4.3 As-designed BIM model and the artificial reference point cloud derived from it

Table 4.1 Characteristics of As-built Scan dataset

Property	Reservoir	University of Cantabria
Format	.e57	.ply
Acquisition / authoring tool	Terrestrial Laser Scan	Matterport
Raw point count	11,145,392	104,062,558

Property	Reservoir	University of Cantabria
Points after global 1.5 M cap / voxel 0.10 m	209,093	433,327
Working scan points at segmentation (Table 4.12)	126,959	431,657
Centroid on import	[90.76871489, 75.20853443, 1.10497749]	[30.83108253, -3.92123192, 1.77784398]
File size (MB)	207,698	2,756,035

Table 4.2 Characteristics of As-designed BIM dataset

Property	Reservoir	University of Cantabria
Format	IFC (2X3 / 4)	IFC (2X3 / 4)
Sampled BIM reference point count	200,000	200,000
BIM reference points after voxel downsampling at 0.10 m	120,007	167,367
Coordinate frame on import	[-12937.70225762, 135823.50838365, 643.85503046]	[29.56653121, -2.77425667, 1.73561929]
Number of modelled elements	106	384
Element types present	{'IfcDoor', 'IfcRoof', 'IfcWindow', 'IfcBeam', 'IfcWall', 'IfcSlab'}	{'IfcWall', 'IfcSlab', 'IfcColumn', 'IfcDoor', 'IfcWindow', 'IfcCovering', 'IfcMember', 'IfcPlate', 'IfcStair'}
File size	10.552	1.691
Storeys	5	8

4.3 EXPERIMENTAL SETUP

4.3.1 Hardware and Software Environment

All of the geometric experiments were conducted on one computer workstation (that uses a commodity processor) that used just the CPU as all of the geometric experimentation described in this thesis is based upon the idea that a well-organized set of standard geometric operations can provide the level of quality assessment for field use on equipment that has been available to most field engineers for

years. The deep learning benchmarking described in Section 4.5.3, and the SpU-Net inference phase of the hybrid cascade of Section 4.5.4, were executed on an NVIDIA GeForce RTX 3070 Ti GPU by way of CUDA. In order to be able to determine which device to utilize at runtime, the program selects its device at runtime and if it is unable to find a CUDA-capable device it will fall back to the PyTorch CPU backend for execution; CPU inference times were not recorded in this study.

Table 4.3 Hardware and software environment

Component	Specification
CPU	AMD Ryzen™ 9 6900HX Processor (8 cores / 16 threads, up to 4.9 GHz)
System memory (RAM)	16 GB DDR5
GPU (deep learning only)	NVIDIA GeForce RTX 3070 Ti, 8 GB VRAM, CUDA
Operating system	Windows 11 (build 10.0.26200, 64-bit)
Python	3.11.0
Core libraries	Open3D 0.19.0, pye57 0.4.19, IfcOpenShell 0.8.4.post1, NumPy 2.4.2, SciPy, scikit-learn 1.8.0
Deep-learning runtime (benchmarks and hybrid inference)	PyTorch 2.10.0 (automatic device selection: CUDA if available, else CPU)
Worker model	Single-worker ThreadPoolExecutor (max_workers = 1), sequential stages

4.3.2 Parameter Configuration

All experiments utilize the parameters listed in Table 3.3 unless otherwise specified. Thus, a constant voxel size of 0.10 m, a normal estimation radius of 2 * voxel size = 0.20 m (up to 30 neighbours), FPFH radius of 5 * voxel size = 0.50 m (up to 100 neighbours), an outlier rejection (FGR) inlier gate of 3 * voxel size = 0.30 m, an ICP correspondence radius of 8 * voxel size = 0.80 m and an evaluation distance of 1.5 * voxel size = 0.15 m at which fitness and inlier RMSE are calculated. Since these parameters are held constant across all registration methods, this enables attribution of the differences due to the registration method (e.g., FGR vs. RANSAC), or segmenter comparison (e.g., segmenter A vs. segmenter B) to the respective method and not to incidental tuning. This corresponds to the fairness constraint from Section 1.2.4.

4.4 REGISTRATION PERFORMANCE

RQ 1 asks "how do fast global registration (FGR) and RANSAC perform in terms of registration fitness and computational efficiency when each is used as a global operator in the coarse-to-fine strategy described in Section 3.5?" The method of experimentation follows the coarse-to-fine strategy precisely - centroid seeding, candidate transforms (family of three); see table 3.4, candidate scoring via equation

(3.5), point-to-point ICP refinement, and full evaluation of all three candidate transforms with the highest-scoring alignment selected, and varies only the global operator. Due to stochastic correspondence sampling, RANSAC will have each sample repeated r = 5 times. FGR is deterministic based upon the use of fixed FPFH features, however like RANSAC, will also be repeated to account for variations due to multi-threading of feature matching. Other parameters remain at the values listed in table 3.3. Comparison will be made using both case studies: the reservoir (a nearly planar hydraulic structure) and the University of Cantabria (UC) building (a multi-room building scale scene) to test the two operators under substantially different geometric conditions.

4.4.1 Quantitative Comparison (RQ1)

The results of the registration process for FGR and RANSAC using identical parameters, separated into the global stage, the refinement/fallback steps, and the final selected alignment (mean ± standard deviation over r = 5 runs). Arrows indicate preferred direction.

Table 4.4 Registration performance: FGR + ICP versus RANSAC + ICP on both case studies

Metric	Reservoir RANSAC+ICP	Reservoir FGR+ICP	UC building RANSAC+ICP	UC building FGR+ICP
Global registration stage				
Stage fitness (mean ± SD)	0.180 ± 0.000	0.094 ± 0.030	0.344 ± 0.001	0.074 ± 0.029
Stage fitness CV (Eq. 3.15)	0.0 %	32.1 %	0.3 %	39.7 %
Stage time (s)	296.2 ± 36.8	216.0 ± 17.2	1255.1 ± 101.4	1013.2 ± 130.9
Stage time CV	12.4 %	8.0 %	8.1 %	12.9 %
Peak memory - median [range] (MB)	367 [347–495]	367 [334 – 1297]	570 [398–3031]	731 [478 – 2853]
Refinement and fallback				
ICP refinement time (s)	25.3 ± 4.5	28.7 ± 6.4	0.73 ± 0.12	26.0 ± 3.4
Yaw-search time (s)	23.9 ± 4.6	27.7 ± 3.1	91.8 ± 13.2	79.2 ± 13.0
Selected stage (all 5 runs)	Yaw	Yaw	Global/ICP	ICP
Final selected alignment				
Final fitness (at 0.15 m) ↑	0.520 ± 0.000	0.520 ± 0.000	0.344 ± 0.001	0.490 ± 0.001
Final inlier RMSE (registration correspondences) ↓ (mm)	76.8	76.8	97.8	104.3
Final composite score (Eq. 3.5)	0.493	0.493	0.309	0.453
Total registration time (s)	345.4 ± 38.7	272.4 ± 24.1	1347.6 ± 96.6	1118.3 ± 144.7
Total time CV	11.2 %	8.9 %	7.2 %	12.9 %
Runs reaching fitness > 0.50	5 / 5	5 / 5	0 / 5	0 / 5

There are three notable findings and one that reframes the original research question. First, the fitness of the global operator itself is a poor predictor of final registration quality. In every cell the raw global fitness is low which RANSAC achieves 0.180 (reservoir) and 0.344 (UC), while FGR achieves only 0.094

and 0.074, yet refinement and candidate selection improve the final fitness in three of the four configurations; UC RANSAC remains at 0.344. Therefore, RQ 1 must compare not only the seed fitness but also the final selected alignment produced by each operator within the full coarse-to-fine pipeline.

On the reservoir both operators are essentially equal in terms of accuracy: they converge to an exact same final fitness of 0.520 and an exact same inlier RMSE of 76.8 mm because the 180° yaw candidate produced the highest-scoring alignment in all five runs of both pipelines (Figure 4.7(a)). As a result, on this data set the choice of global operator is ultimately irrelevant to where the pipeline ends up, so the candidate sweep does the decisive work and thus reduces the comparison to cost. FGR completed the global stage in 216 s against RANSAC's 296 s and the entire registration in 272 s against 345 s, which is a 1.27x reduction in total time for exactly the same level of accuracy.

On the UC building the two operators diverged. However, the result was counter intuitive. RANSAC produces the higher global stage fitness (0.344 versus FGR's 0.074), yet it produces the worse final alignment. Its ICP step terminates almost instantly (0.73 s) without increasing the fitness value which indicates that the RANSAC seed lies in a local minimum from which point-to-point ICP cannot escape; the pipeline is left selecting that 0.344 result. FGR's much weaker seed lies in the basin of the correct alignment, so ICP climbs to 0.490 (see Figure 4.7(b)). FGR therefore provides a +0.146 absolute (~ 43% relative) gain in final fitness while also finishing faster (1118s versus 1348s total, a 1.21x reduction). FGR's slightly greater final RMSE (104.3 mm against 97.8 mm) reflects that the higher-fitness alignment admits more, and potentially more distant, inliers than RANSAC's partial locally trapped fit; so RMSE should be read together with fitness rather than alone.

A determinism caveat refines the textbook expectation stated in Section 2.3.3. In these experiments RANSAC's fitness was essentially constant across runs (cv ~ 0-0.3%, consistent with a fixed sampling seed); whereas FGR's fitness varied run-to-run (cv 32-40%) owing to non-deterministic threaded feature matching. Importantly, this dispersion did not propagate to the final result: the final variability was much smaller than the global-stage variability (final fitness SD 0.000 for the reservoir and 0.001 for the UC building) because ICP and the candidate sweep absorb the seed variability. Instead, RANSAC's instability appears in its runtime (total-time cv 11.2% on the reservoir), while FGR's runtime is both lower and steadier (8.9%). Peak memory is noisy - occasional spikes near 3gb occur for both operators on the 8gb-vram laptop and should best be read from the medians (367-731mb), which show no decisive separation.

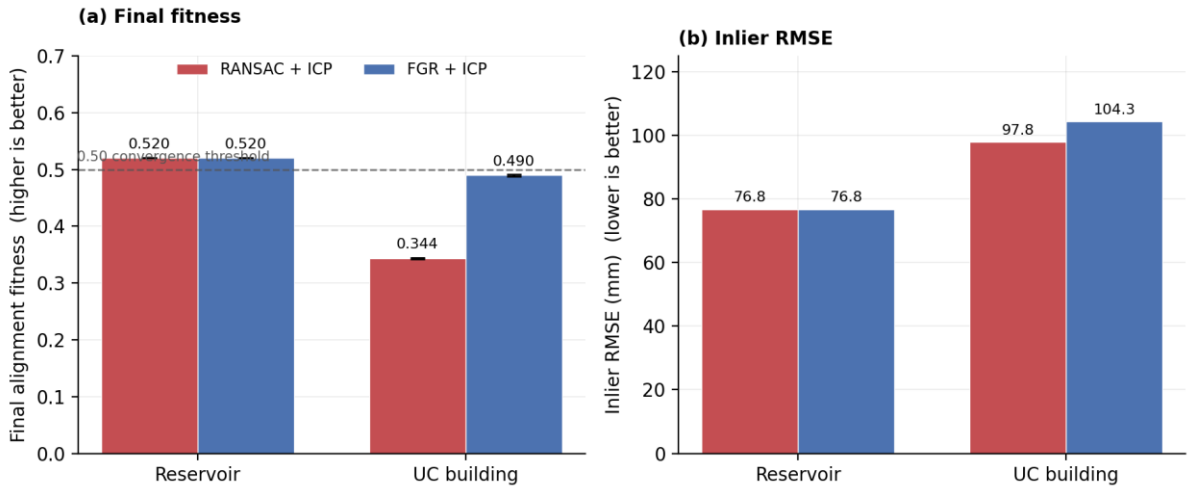


Figure 4.4 Final alignment fitness and inlier RMSE for FGR+ICP+Yaw-180 vs RANSAC+ICP+Yaw-180 on both case studies

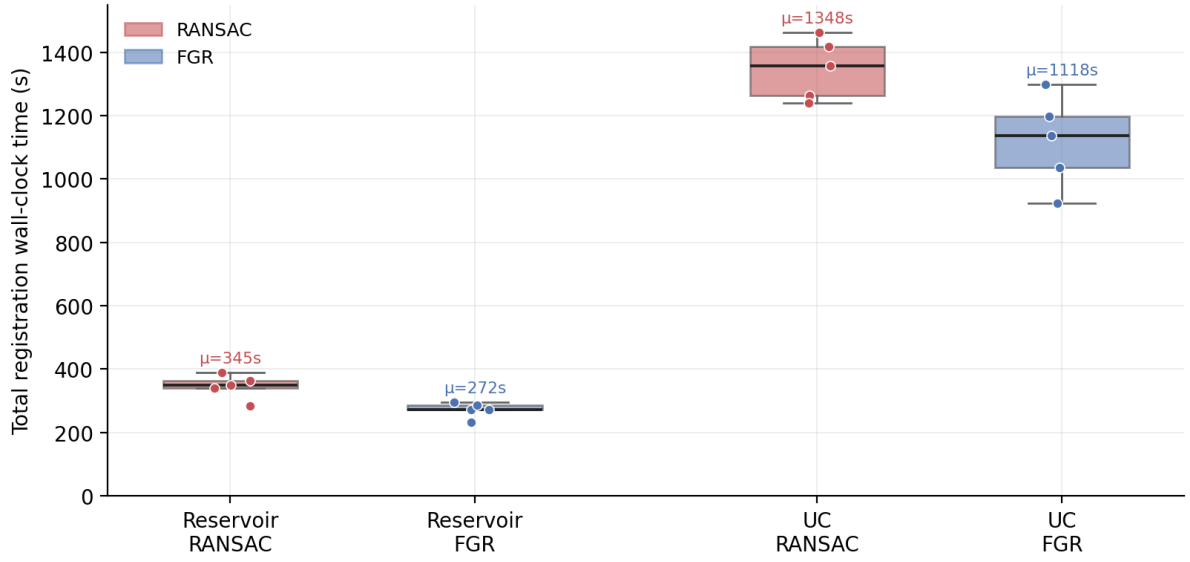


Figure 4.5 Distribution of total registration wall-clock time over five repeated runs (global stage + ICP + candidate sweep)

4.4.2 Qualitative Alignment and Candidate Strategy

The number of registered points alone is insufficient to determine whether an appropriate amount of similarity exists. A visual inspection of the registration must take place to confirm that the degree of fitness observed results from a good alignment as opposed to some other alignment that could also have resulted in a similar degree of fitness - this is a form of self-similarity failure mode as described in Sections 1.1.3 & 2.3.2. Each case depicted in figure 4.6 represents the unaligned point cloud, the centroid seeded coarse pose, and the final coarse to fine result. The last image in the series illustrates how the structural edges in the final panel align with each other (i.e., they appear to match) which verifies a geometrically correct registration has been achieved versus merely a random or coincidental "fitness" maximum.

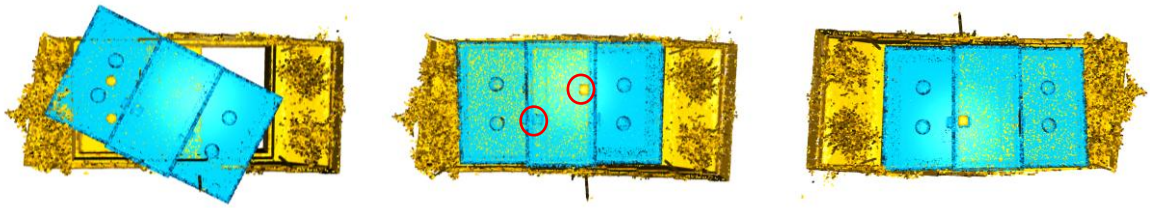


Figure 4.6 Qualitative registration: centroid-seeded, coarse-to-fine and yaw-search result

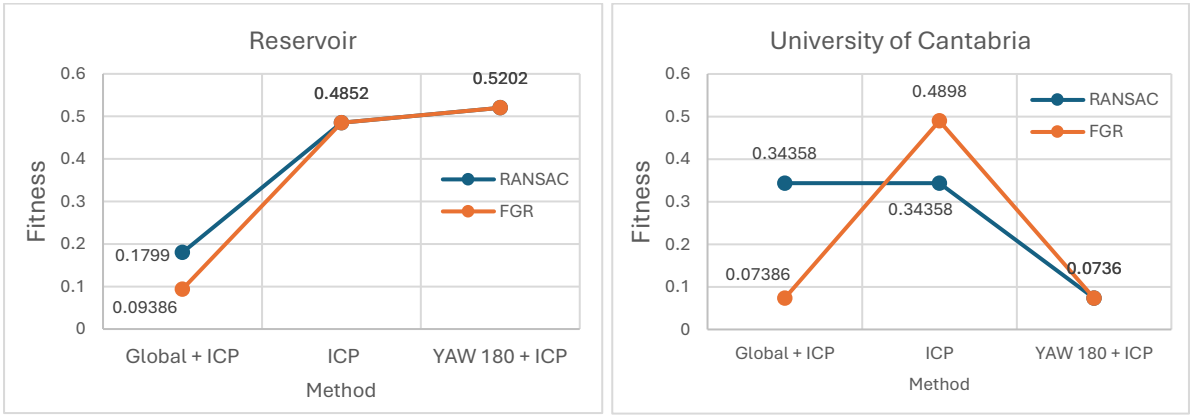


Figure 4.7 Coarse-to-fine stage progression for Reservoir structure (a) and University building (b)

Fig. 4.7 illustrates the mechanics involved in achieving a correct registration through the candidate strategy employed (as opposed to selecting a global operator). It explicitly addresses RQ 1. In cases where the geometry of a given scene exhibits significant amounts of self-similarity (such as the concrete surfaces of the reservoir), a single global-local iteration will stall at about 0.49 for either global operator. However, once the rotated yaw candidate breaks the in-plane ambiguity present within these images, the resulting registrations improve significantly such that both operators now achieve approximately 0.52 registrations. This same yawed candidate is used for every run made upon the reservoir. Conversely, upon the UC building, the scenario presents itself differently. Upon inspection, it appears the un-rotated candidate was correct. However, once transformed into the yawed candidate, the resultant registration improves very little if at all (a degradation to approximately 0.074) and is therefore rejected by the scoring rule. Therefore, there is no need for manual intervention based upon geometry characteristics of each scene to allow the pipeline to adapt accordingly.

4.4.3 Summary

The FGR-based pipeline is the preferred configuration for the two evaluated cases. It matches the RANSAC-based pipeline on the reservoir (fitness 0.520; inlier RMSE 76.8 mm) and gives higher final fitness on the UC building (0.490 versus 0.344), although its UC inlier RMSE is also higher (104.3 versus 97.8 mm). Total registration is approximately 1.21–1.27 times faster. Runtime variability is lower for FGR on the reservoir but higher on the UC building (Table 4.4). The yaw candidate improves reservoir fitness from approximately 0.485 to 0.520 for both configurations. These findings concern the complete candidate-selection pipelines and do not establish that one global operator is universally more accurate or stable.

4.5 SEGMENTATION PERFORMANCE

The segmentation process is key in determining which registered scan points are assigned to individual elements modelled from the data and is consequently fundamental to all future measurements of deviation. The focus of this chapter will be a full analysis of the segmentation process that is comprised

of two distinct but related pipelines. The approach used to establish ground truth segmentation, as well as the evaluation methods, will be detailed in Section 4.5.1. Section 4.5.2 will compare the results of the four traditional geometric operators with respect to the three independent datasets provided under both ideal and degraded noisy environments. Section 4.5.3 will determine if lightweight deep learning-based models can successfully segment construction point clouds using only point cloud geometry, rather than requiring BIM-derived geometric priors; and evaluate the overall quality of such segmented models for use in downstream quality assessments. Finally, Section 4.5.4 will synthesize these results into an overarching view of the complementary nature of segmentation approaches and present the detection performance of the hybrid pipeline over two real-world construction case studies. In Section 4.5.5 fallback rates and validation flags will be provided as quantifiable measures of robustness.

4.5.1 Ground Truth and Evaluation Protocol

A total of three publicly available data sets, annotated independently of the method, will be used to evaluate segmentation with independently annotated ground truth. There are thirteen semantic classes for which there exist human labelled ground-truths across all 271 rooms scanned with Matterport from Stanford's Large-Scale 3D Indoor Spaces (S3DIS); the standard Area 5 Test Split containing 68 rooms will be utilized. ScanNet consists of 1513 reconstructed RGB-D scenes that have been annotated with twenty NYU40 semantic classes; a 50-scene subset of the validation data will be utilized. Rohbau3D is an actively constructed terrestrial laser scanning data set that has been annotated into eighteen semantic classes by construction professionals; all three test scenes will be evaluated. Each of these annotations were created independently of any BIM model; thus, avoiding the circular dependency issue that could occur if distance-based pseudo-labels were utilized as ground truth.

For IFC-mapped evaluation, native semantic labels are mapped to their corresponding IFC structural classes. ScanNet contributes four mapped classes (IfcWall, IfcSlab, IfcDoor and IfcWindow), while S3DIS and Rohbau3D each contribute six (adding IfcBeam and IfcColumn). The reported mIoU-IFC is the unweighted arithmetic mean of the displayed per-class IoUs for each dataset, including zero-valued mapped classes. Per-class F1 score, overall accuracy (OA), and the coefficient of variation of F1 across repeat runs (F1 CV) are reported as supplementary measures. The evaluation metrics are defined in Chapter 3.

For the classical operators, noise robustness is evaluated by applying synthetic degradation to the Data Sets: positional gaussian noise (σ = 0.02 m), random dropout of twenty percent of points, normal vector perturbations.

4.5.2 Comparison of Classical Operators (RQ2)

The question of how four different types of segmentation (normal vector filtered graph-based segmentation, adaptive bounding boxes, RANSAC plane extraction, and DBSCAN) compare on their segmentation accuracy, stability and speed for extracting structural elements from construction point clouds is the subject of RQ2. Each method is tested using a constant voxel size and neighbourhood structure on all three test databases. The evaluation of each method was done along three dimensions: (i) Per class segmentation accuracy with and without noise, (ii) Run-to-run determinacy as measured by the F1 coefficient of variation, and (iii) Processing speed per structural element.

Table 4.5 Accuracy of the classical operators on clean point clouds (mIoU-IFC, OA, per-element CPU time)

Operator	S3DIS	ScanNet	Rohbau3D	OA (range)	ms/elem	F1 CV
Adaptive BBox	0.691	0.978	0.532	0.96 - 0.99	6.5 - 12.9	0.000
RANSAC plane	0.650	0.671	0.494	0.86 - 0.93	1.3 - 1.6	0.000–0.005
Graph + normal	0.610	0.592	0.453	0.77 - 0.87	24.7 - 69.1	0.000
DBSCAN + normal	0.590	0.575	0.382	0.62 - 0.84	31.7 - 89.0	0.000

Values are mIoU per dataset; OA and ms/elem are reported as the range across the three datasets. The coefficient of variation (CV) of the F1 score over repeated runs is reported as a determinism measure.

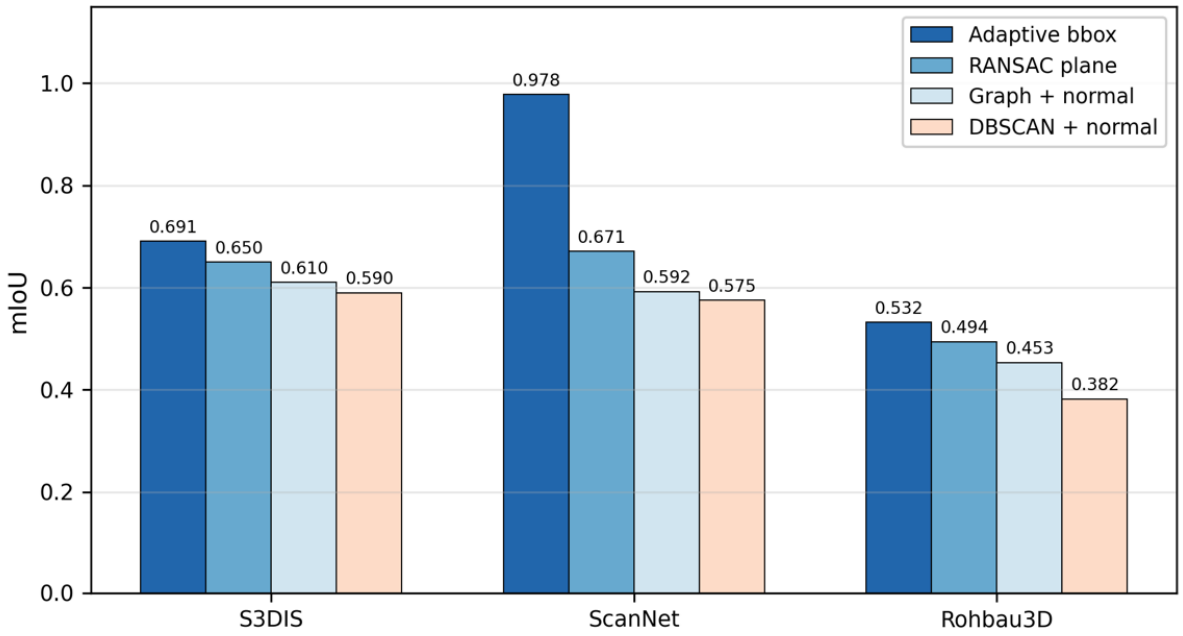


Figure 4.8 Classical operator accuracy (mIoU) on clean point clouds

On clean data, the Adaptive Bounding Box achieved the highest mIoU at 0.978, 0.691 and 0.532 on ScanNet, S3DIS and Rohbau3D respectively. The advantage of the Adaptive Bounding Box over other methods is particularly evident on ScanNet, where the bounding boxes were generated using the BIM model, thus taking advantage of the clearly defined room layout present in RGB-D reconstructions. RANSAC Plane Extraction ranked second among methods and benefited from the fast detection of planar faces on wall and slab classes. Normal Filtered Graph Segmentation performed reasonably well but did not perform better than the two first methods. DBSCAN always had the lowest mIoU values since it depends too heavily on the value of ε which has difficulty adapting to the varying point densities found in typical construction scans.

With the exception of RANSAC Plane Extraction with both S3DIS noisy and Rohbau3D clean (with an F1 CV of 0.009 and 0.001 respectively) and some exceptions noted above, all four methods provided perfect run-to-run determinism (F1 CV=0.000).

The sensitivity to parameter degradation was evaluated by degrading each scene with a controlled method - additive gaussian noise (σ = 0.02 m), 20% random point removal, and surface normal vector

displacement - which simulate the measurement errors of scanners, occlusions, and the rugged, dirty conditions found at active sites. The robustness of an operator can be expressed as the relative decline in mIoU from the clean to the degraded state:

$$\Delta = \frac{mIoU_{\text{clean}} - mIoU_{\text{deg}}}{mIoU_{\text{clean}}} \times 100\% \quad (4.1)$$

Using this standard approach, an operator that experiences a decline greater than 15% is labelled "clutter-sensitive" (i.e., requiring preprocessing under the tested degradation), while those that experience a decline of 15% or less are considered "robust", and meet this study's robustness criterion.

Table 4.6 Robustness Analysis
(Performance drop across datasets. Δ > 15% indicates noise sensitivity)

Operator	S3DIS Clean	S3DIS Noisy	ScanNet Clean	ScanNet Noisy	Rohbau Clean	Rohbau Noisy	Avg Δ%	Verdict
Adaptive BBox	0.691	0.278	0.978	0.722	0.532	0.406	36.6%	SENSITIVE
RANSAC Plane	0.650	0.264	0.671	0.529	0.494	0.366	35.5%	SENSITIVE
Graph + Normal	0.610	0.280	0.592	0.555	0.453	0.343	28.3%	SENSITIVE
DBSCAN + Normal	0.590	0.278	0.575	0.528	0.382	0.300	27.5%	SENSITIVE

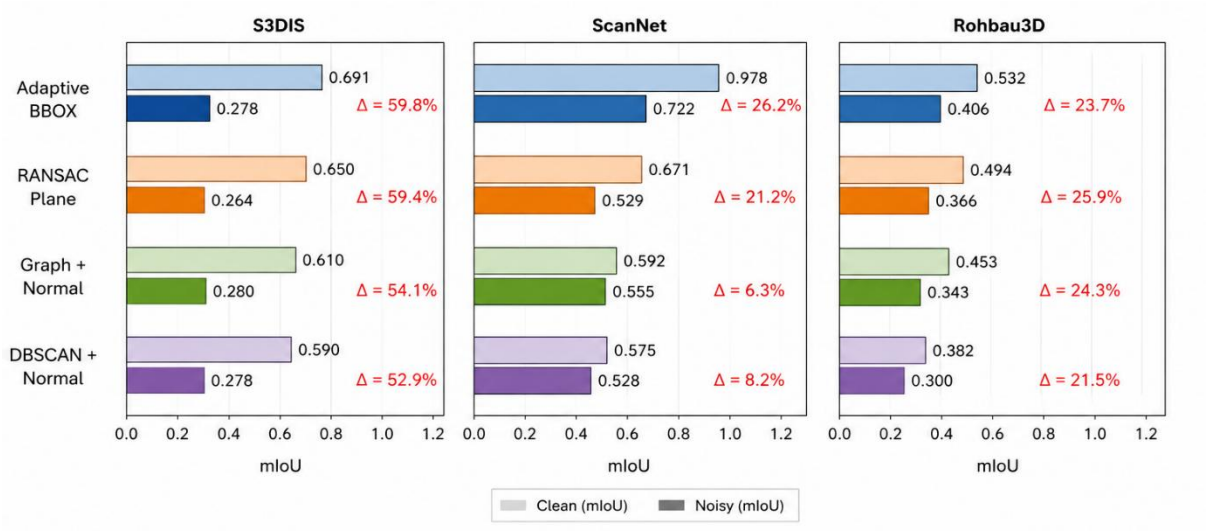


Figure 4.9 Clean vs. noisy mIoU across datasets
(Red annotations show the percentage drop. All operators exceed the 15% sensitivity threshold)

Although all four classical operators have shown to be highly sensitive to noise (>15%), their sensitivity differed. The Adaptive Bounding Box demonstrated the greatest relative loss (59.8% on S3DIS and 26.2% on ScanNet) due to its limited range of parameters designed to fit tightly around clean BIM geometries. When point locations change due to added noise, the bounding box clipped valid structural points. The

graph-based and DBSCAN methods demonstrated smaller relative losses (average 28.3% and 27.5%) since they used connectivity-based logic that required less precise coordinates to produce correct results.

It should be noted that there is a notable exception when evaluating ScanNet. At this database level, both Graph+Normal Filtering (only 6.3% loss) and DBSCAN + Normal Filtering (only 8.2% loss) demonstrate high levels of robustness (<15%). As ScanNet's original base already contained natural RGB-D noise (depth quantization and registration drift), therefore degrading the scan further would result in relatively lower changes than those observed on S3DIS and Rohbau3D's originally cleaner data sets.

In addition to these three previously discussed "noise" conditions, Rohbau3D also provided a unique fourth "noise" condition: A scan taken directly from a construction site prior to cleaning. On this scan, all methods showed improvement over the "nominally" clean condition (Adaptive BBox: 0.605 vs. 0.532). This indicates that Rohbau3D's "clean" dataset likely contained some amount of residual construction clutter (scaffolding, formwork) that can be removed by each of the four operators' normal-based and/or density-based pruning logic.

Table 4.7 Per-element CPU segmentation speed (ms/elem) on clean point clouds

Operator	S3DIS (ms)	ScanNet (ms)	Rohbau3D (ms)
RANSAC Plane	1.562	1.257	1.349
Adaptive BBox	6.480	8.730	12.921
Graph + Normal	30.620	24.668	69.134
DBSCAN + Normal	36.228	31.677	88.957

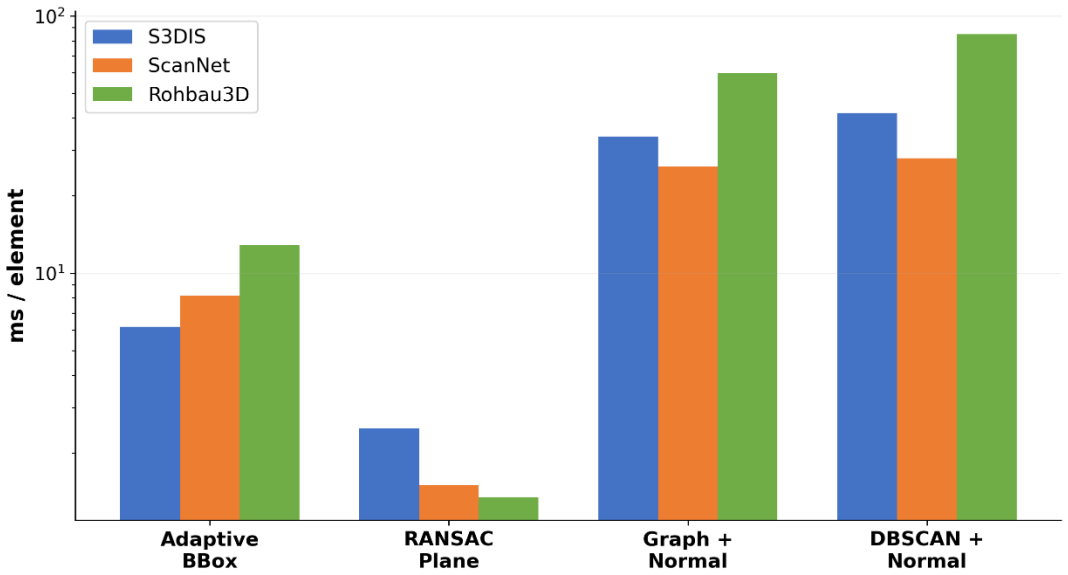


Figure 4.10 Per-element segmentation speed on clean data (log scale)

RANSAC Plane Extraction was consistently the fastest method (processing times of 1.3-1.6 ms/element), approximately four to ten times faster than Adaptive BBox and twenty to sixty-six times faster than the

graph-based and DBSCAN operators. RANSAC's speed was largely due to its early termination consensus logic, which rapidly converged on dominant planar surfaces. The Adaptive Bounding Box fell into a mid-range category (processing times of 6.5-12.9 ms/element) providing a good balance between accuracy and speed. Both Graph-Based and DBSCAN operators were significantly slower (processing times of 25-89 ms/element) as a result of their KNN-graph creation and iterative density reachability queries, whose processing time increased super-linearly with increasing scene sizes such as those found in Rohbau3D.

The Adaptive Bounding Box is recommended as the primary operator for segmenting clean or slightly degraded point clouds using BIM guidance, as it produced the highest accuracy across all three datasets while showing moderate computational costs and full determinism. RANSAC Plane Extraction is preferred if processing speed is considered more important than anything else (e.g., for real-time application in fields using extremely large scans). Due to variations in noise sensitivity between methods, no one method dominated under heavy-noise conditions; therefore, Graph + Normal Filtering exhibited minimal degradation on ScanNet, indicating that a cascaded pipeline which generates several candidate segmentations per element and selects the best by coverage and compactness is preferable to relying on any single operator.

4.5.3 Deep Learning Segmentation Without BIM Priors (RQ 2)

The RQ 2 is influenced by the results that have shown that classical segmentation pipelines depend heavily on the calibration of geometry-derived parametric settings. Since these classical segmentation operators rely on BIM-generated bounding box locations, voxel grid size, and threshold normals which were optimized using clean building geometries, they degrade significantly as the amount of noise increases. In contrast to classical segmentation methods, deep-learning methods do not use BIM-generated geometric priors during inference, but instead create feature representations of the data being segmented based solely on data. This raises the question of whether this lack of dependence on BIM-generated geometric priors will come at some cost with respect to segmentation accuracy, inference time, and generalizability across domains.

Benchmarking was performed on three typical architectures spanning the primary categories of 3D segmentation of point clouds: PointNet++ (SSG) (0.97 million parameters), a hierarchical point-based segmentation architecture; Point Transformer V3 (PTv3) (46.2 million parameters), an attention-based segmentation architecture; and SparseConv U-Net (SpU-Net) (39.2 million parameters), a sparse voxel-based convolutional segmentation architecture. All models were run using previously trained weights from the Pointcept framework on NVIDIA GPUs (using CUDA).

Three additional architectures; KPConv, RandLA-Net, and Point Transformer v2 were also explored but removed from consideration due to issues related to the toolchain used to implement them (for example incompatible CUDA kernel version requirements, obsolete dependency packages).

Table 4.8 Per-class IoU on ScanNet validation (50 scenes, IFC-mapped classes)

IFC Class	PTv3 (IoU / F1)	PointNet++ (IoU / F1)	SpU-Net (IoU / F1)	Best Model
IfcDoor	0.000 / 0.000	0.000 / 0.000	0.000 / 0.000	—
IfcSlab	0.873 / 0.931	0.559 / 0.711	0.524 / 0.672	PTv3
IfcWall	0.742 / 0.835	0.524 / 0.675	0.274 / 0.414	PTv3

IFC Class	PTv3 (IoU / F1)	PointNet++ (IoU / F1)	SpU-Net (IoU / F1)	Best Model
IfcWindow	0.000 / 0.000	0.000 / 0.000	0.001 / 0.002	—
mIoU-IFC	0.4038	0.2708	0.1998	PTv3

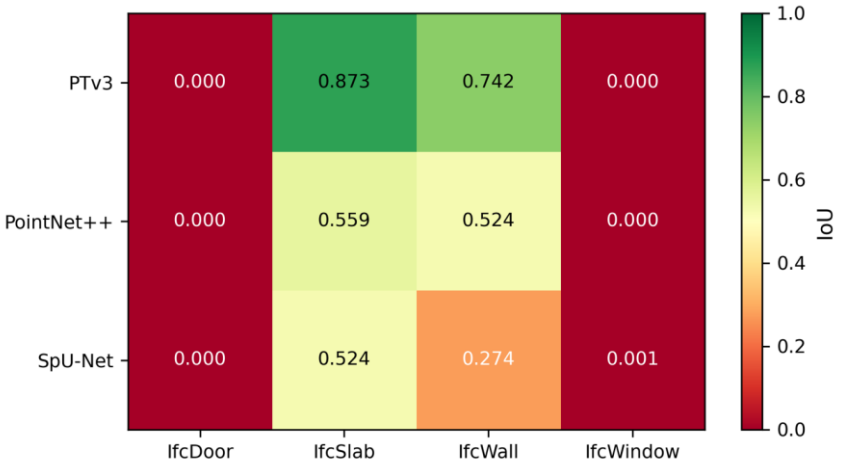


Figure 4.11 Per-class IoU heatmap on ScanNet

Table 4.9 Per-class IoU on S3DIS Area 5 (68 rooms, IFC-mapped classes)

IFC Class	PTv3 (IoU / F1)	PointNet++ (IoU / F1)	SpU-Net (IoU / F1)	Best Model
IfcBeam	0.001 / 0.002	0.000 / 0.000	0.000 / 0.000	—
IfcColumn	0.000 / 0.000	0.000 / 0.000	0.000 / 0.000	—
IfcDoor	0.312 / 0.452	0.052 / 0.092	0.244 / 0.374	PTv3
IfcSlab	0.934 / 0.964	0.738 / 0.845	0.956 / 0.977	SpU-Net
IfcWall	0.728 / 0.834	0.611 / 0.751	0.755 / 0.853	SpU-Net
IfcWindow	0.041 / 0.069	0.086 / 0.129	0.174 / 0.242	SpU-Net
mIoU-IFC	0.3360	0.2478	0.3548	SpU-Net

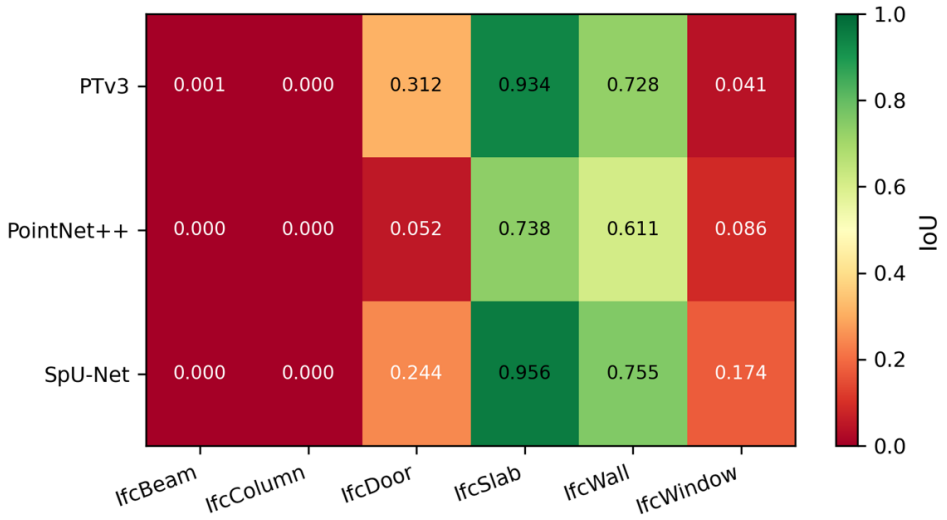


Figure 4.12 Per-class IoU heatmap on S3DIS

Table 4.10 Per-class IoU on Rohbau3D (3 construction-site scenes, cross-domain transfer)

IFC Class	PTv3 (IoU / F1)	PointNet++ (IoU / F1)	SpU-Net (IoU / F1)	Best Model
IfcBeam	0.010 / 0.019	0.006 / 0.013	0.010 / 0.020	SpU-Net
IfcColumn	0.001 / 0.002	0.000 / 0.000	0.000 / 0.000	—
IfcDoor	0.000 / 0.000	0.000 / 0.000	0.000 / 0.000	—
IfcSlab	0.553 / 0.705	0.205 / 0.338	0.840 / 0.907	SpU-Net
IfcWall	0.594 / 0.741	0.216 / 0.354	0.726 / 0.836	SpU-Net
IfcWindow	0.000 / 0.000	0.000 / 0.000	0.000 / 0.000	—
mIoU-IFC	0.1930	0.0712	0.2627	SpU-Net

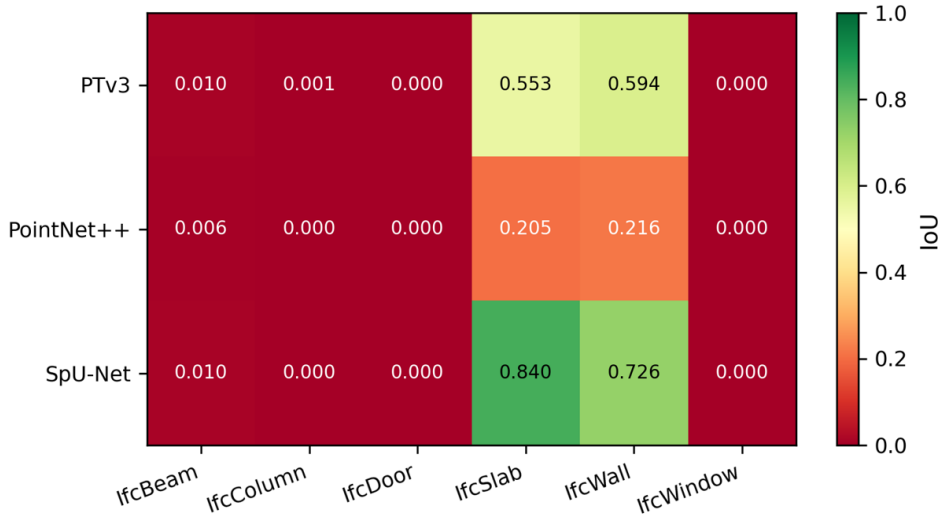


Figure 4.13 Per-class IoU heatmap on Rohbau3D (cross-domain)

All three datasets show structural-class dominance: each model concentrates most of its performance in IfcWall and IfcSlab. PTv3 achieved IoUs of 0.873 for IfcSlab and 0.742 for IfcWall on ScanNet, while SpU-Net achieved 0.840 and 0.726 for the same classes on Rohbau3D. Minority-class performance remained weak. Beam and column IoUs were near zero, while the strongest door and window results were 0.312 for PTv3 and 0.174 for SpU-Net on S3DIS Area 5. Using the macro-average of the displayed IFC-class IoUs, PTv3 ranks first on ScanNet (mIoU-IFC = 0.404), while SpU-Net ranks first on S3DIS (0.355) and Rohbau3D (0.263). PointNet++ ranks second on ScanNet and third on S3DIS and Rohbau3D.

Table 4.11 Cross-dataset summary: mIoU-IFC, consistency, inference time, and multi-criteria ranking

Model	ScanNet mIoU	S3DIS mIoU	Rohbau mIoU	σ(mIoU) Avg	Infer (s) Avg	Params (M)	GPU (MB)	Rank: Accuracy	Avg Rank
PTv3	0.404	0.336	0.193	0.051	157.1	46.2	330 – 1241	1.7	2.2
SpU-Net	0.200	0.355	0.263	0.056	92.7	39.2	356 – 865	1.7	1.8
PointNet++	0.271	0.248	0.071	0.044	615.3	0.97	429	2.7	2.1

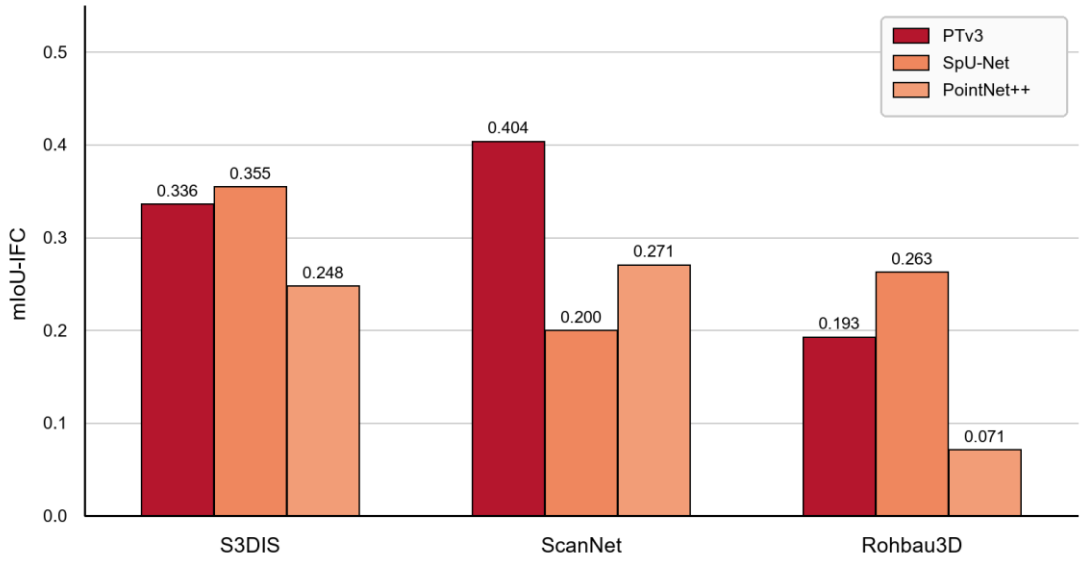


Figure 4.14 Deep learning mIoU-IFC across three datasets
(No model consistently dominates; PTv3 excels on ScanNet while SpU-Net excels on S3DIS and Rohbau3D)

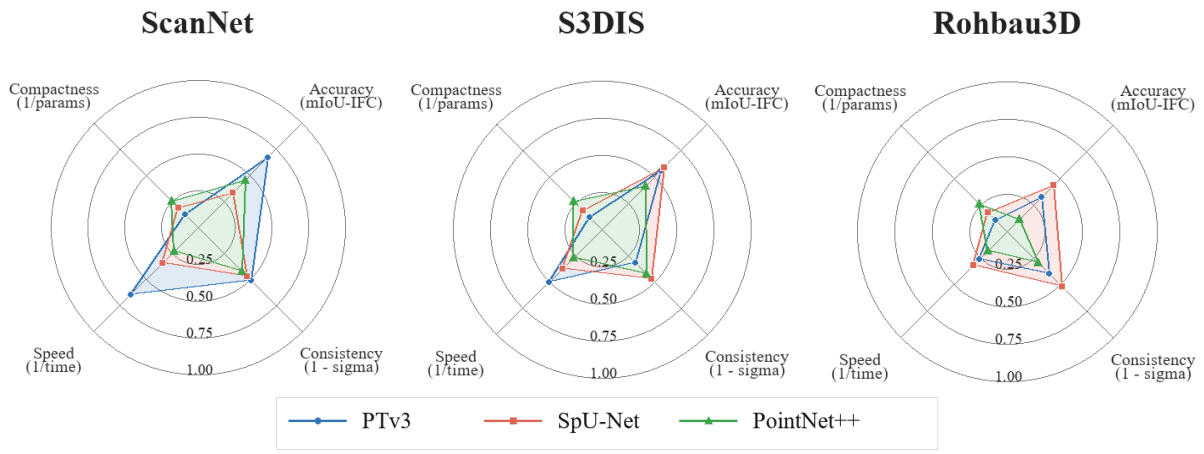


Figure 4.15 Normalised multi-criteria radar plots by dataset (accuracy, consistency, speed and compactness)

4.5.3.1 *Cross-Domain Generalization*

Cross-domain transfer was evaluated on Rohbau3D using pretrained models without updating their weights. Relative to furnished indoor data, construction scans differ in sensor modality, scene content, point density and class distribution. These factors can affect transfer performance, so the checkpoint training source and evaluation protocol must be identified for each result.

The macro-averaged results do not show a uniform decline between ScanNet and Rohbau3D. SpU-Net increases from 0.200 to 0.263 (approximately 31.5%), whereas PTv3 decreases from 0.404 to 0.193 (approximately 52.2%) and PointNet++ decreases from 0.271 to 0.071 (approximately 73.7%). These comparisons require caution because ScanNet contains four mapped IFC classes, whereas Rohbau3D contains six, and the checkpoint and scene conditions must remain consistent. The final hybrid pipeline uses an S3DIS-trained SpU-Net; its S3DIS score of 0.355 is the highest macro-averaged mIoU-IFC among the evaluated S3DIS models.

The variances for each individual scene in Rohbau3D also provide some insight into why certain scenes seem to be easier/harder than others. Specifically, it is apparent that large scenes (e.g., Scene_07000 with approximately 5.8 million points) yield the poorest performance for all models (PTv3 = 0.174, SpU-Net = 0.254, PointNet++ = 0.068) since inference on these types of mega-scenes cause both memory fragmentation and tiling artifacts that negatively impact edge prediction accuracy.

4.5.3.2 *Inference Time and Computational Profile*

Inference times for all of the deep neural network models were determined on an NVIDIA GPU-based platform (CUDA). The average inference times across each dataset are provided in Table 4.11. As expected, SpU-Net is always the fastest among all models for each dataset (with average inference time of 56–167 seconds) followed by PTv3 (average inference time of 110–217 seconds) and finally PointNet++ (the slowest of all, with average inference time of 560–692 seconds). While one might expect PointNet++ to have faster inference times given its relatively smaller number of parameters compared to SpU-Net and PTv3 (approximately 0.97 M vs. 39–46 M respectively), this is clearly not the case. The primary reason for this anomaly is due to PointNet++'s reliance on expensive point-based set abstraction layers such as farthest-point sampling and ball-query neighbourhood search functions, rather than the sparse-convolution or attention kernels used by SpU-Net and PTv3.

Memory usage profiles (GPU VRAM) are fairly modest: PointNet++ uses a constant amount of memory regardless of scene size (i.e., 429 MB); SpU-Net varies depending on scene size (from 356–865 MB); PTv3 also varies depending on scene size (from 330 - 1241 MB). Although the memory usage requirements

for each model will easily fit within the confines of most standard consumer-grade GPUs (typically ranging from 4–8 GB), it should be noted that the primary limitation for practical deployment will be their strict dependence on GPU-based computing: CPU-only inference was not timed, so compliance with the pipeline's time budget on CPU alone has not been demonstrated.

4.5.4 Hybrid Segmentation Strategy (RQ 2)

Sections 4.5.2 and 4.5.3 have shown that the classical BIM-guided operators and deep learning models are not competing alternatives but alternative methods that are suitable for different sections of the segmentation process and different types of failures in the segmentation pipeline. The classical operators perform best when there is accurate BIM geometry available (tight bounding boxes and normal priors) but fail in noisy conditions. The deep learning models do not need BIM priors but perform poorly on minority classes and in cases of cross-domain transfer. Therefore, the complementarity between these two options led to the development of a hybrid process which was tested using two real-world case studies.

4.5.4.1 *Accuracy vs Speed Trade-off*

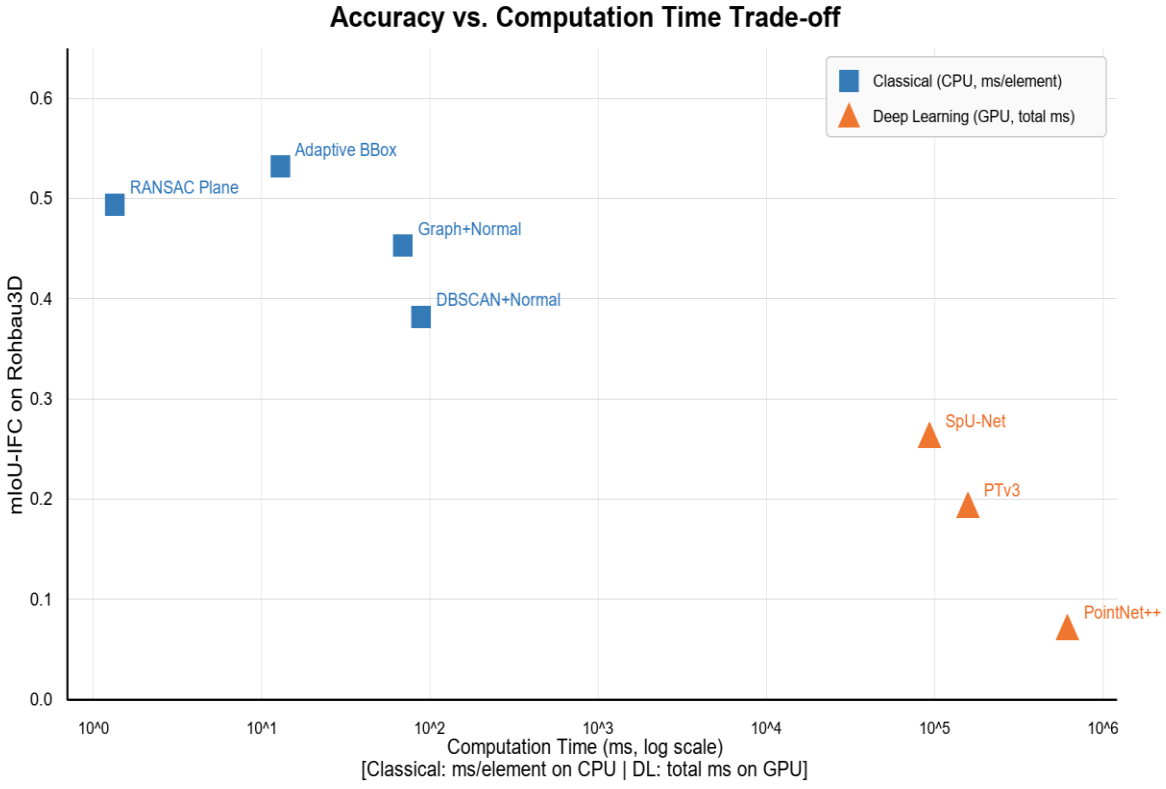


Figure 4.16 Accuracy vs. computation time on Rohbau3D

Figure 4.16 plots classical per-element CPU time alongside total GPU inference time for learned models. These quantities have different denominators and cannot establish a speed-up factor. On Rohbau3D, Adaptive BBox reports mIoU 0.532 at 12.921 ms per element; SpU-Net reports mIoU-IFC 0.263 and an average inference time of 92.7 s across the three datasets (approximately 166.7 s for the reported Rohbau3D workload). A fair runtime comparison requires the same scenes, point counts, output task and timing boundaries. The learned model is therefore assessed as a complementary semantic filter rather than assigned a numerical per-element speed disadvantage.

Lightweight deep-learning architectures can segment construction point clouds without BIM-derived geometric priors, but the reported results show important limitations. PTv3 achieved the highest macro-averaged mIoU-IFC on ScanNet (0.404), mainly through its wall and slab performance. On Rohbau3D, SpU-Net achieved the highest value (0.263). Between ScanNet and Rohbau3D, PTv3 and PointNet++ decreased by approximately 52.2% and 73.7%, respectively, while SpU-Net increased from 0.200 to 0.263. Minority-class IoUs remained low across the models. The learned predictions can therefore support clutter filtering and element recovery, but the presented evidence does not establish them as a stand-alone source of element-level correspondence for quality assessment.

4.5.4.2 *Hybrid Pipeline (S3DIS SpU-Net + BIM-Guided Segmentation)*

The final DL-First Hybrid Cascade uses an S3DIS-trained SpU-Net to provide semantic labels for clutter filtering and geometric tie-breaking. This model achieved a macro-averaged mIoU-IFC of 0.355 on S3DIS and 0.263 on Rohbau3D under the reported evaluations. The cascade combines those predictions with BIM-guided segmentation and proximity recovery. The five phases are described below.

Phase 1: Conducts a single-pass global DL inference (S3DIS SpU-Net) across the entire cropped scan to classify structural features.

Phase 2: Utilizes the DL-inferred values to eliminate non-structural points (furniture/clutter) from the working cloud using a confidence threshold value of 0.30.

Phase 3: Implements BIM-guided segmentation (adaptive bounding-box + surface-normal agreement + DL-class-tiebreaker) on the cleaned cloud.

Phase 4: Conducts proximal recovery on the full scan to recover elements that failed Phase 3.

Phase 5: Produces a coverage limitation report indicating areas where there are scanner-coverage limitations versus algorithm failures.

The primary architectural differences between this DL-first implementation and the BBox-first implementation presented earlier include that DL-inference occurs prior to BIM-guided segmentation allowing for improved Phase 3 accuracy as well as using the DL-inferred class values to break ties when determining whether an area is a wall or slab joint.

Table 4.12 DL-First Hybrid Cascade results on the UC Building and Reservoir case studies (S3DIS SpU-Net)

Metric	UC Building	Reservoir
Total IFC elements	384	106
Elements considered by the hybrid pipeline	384	106
Detected elements	350	63
Detection rate (of total)	91.1%	59.4%
Phase 3 (BBox+Normal+DL)	262	40
Phase 4 (Proximity Recovery)	88	23
Undetected (low coverage / no scan)	34	43

Metric	UC Building	Reservoir
Pre-registration purge (points)	-	68,880 of 209,093 (32.9 %)
Registration cloud (points)	-	140,213
Non-structural points purged	81,627 (18.9%)	56,197 (44.3%)
Scan points (down sampled 0.1 m)	431,657	126,959
Inference time (SpU-Net S3DIS)	30.3 s (295 crops)	6.8 s (78 crops)
Peak GPU VRAM	577 MB	176 MB
Detected storeys	8	5

(a) UC Building *(b) Reservoir*

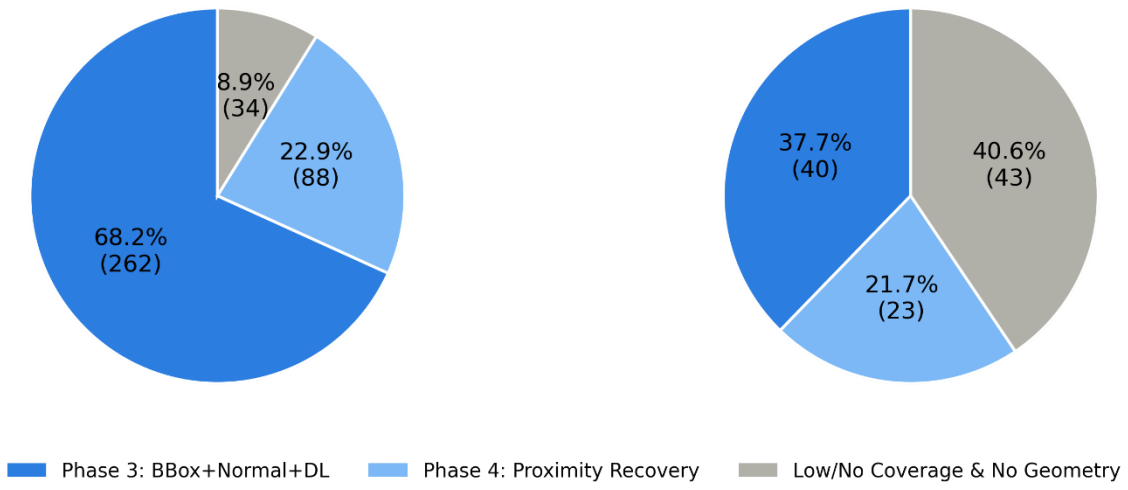


Figure 4.17 Element detection breakdown for the hybrid pipeline on two case studies

Using the university building (educational multi-storey building; 431,657 working-scan points at 0.1 m voxel resolution; 384 IFC elements), the DL-first hybrid cascade identified 350 elements. Of those, 262 elements (68.2%) were successfully segmented via Phase 3 (BBox+Normal+DL on the cleaned cloud) and 88 elements (22.9%) were recovered via Phase 4 (proximity recovery). Prior to BIM-guided segmentation, the DL-first approach eliminated 81,627 non-structural points (18.9% of working-scan points) corresponding to furniture, bookcases and similar classes selected using the reported confidence threshold of 0.30 by the S3DIS SpU-Net. SpU-Net inference took approximately 30.3 seconds and covered 295 overlapping crops spanning 426,803 points at 0.04 m voxel resolution. The remaining thirty-four undetected elements are predominantly low-coverage curtain-wall mullions and panels with small cross-sectional dimensions (65×125 mm) that fall below the minimum point count needed to compute deviations, together with six stairs whose IFC geometry could not be tessellated.

Utilizing the Reservoir (hydraulic reinforced concrete structure; total number of working-scan points: 126,959; total number of IFC elements: 106), the detection rate was found 59.4%. Of those identified, forty elements were segmented via Phase 3 (BBox+Normal+DL); twenty-three elements were recovered via Phase 4 (Proximal-Recovery). Utilizing the S3DIS-trained SpU-Net's DL-clutter-purge, a total of

56,197 non-structural points (44.3% of working-scan-points) were removed from the working-cloud which resulted in a significantly larger percentage than was observed during testing on the University Building due to much greater amounts of construction site clutter being present in the terrestrial-laser-scan. A total of 43 unsegmented elements existed, divided into three groups: sixteen elements had no points within 1.5 metre and consisted of four walls, two slabs, ten windows. All due to either physical occlusions or lack of line-of-sight by the scanner; twenty elements had points close-by but fell short of segmentable-threshold-values, and comprised three walls, five slabs, five windows, seven roofs having less than full coverage; seven elements could not be meshed from their respective IFC representations due to no valid geometry. Failures of this type occurred because of limitations associated with scanning methods employed and/or modelling-gaps in BIM methodologies although algorithmic contributions cannot be excluded without independent segmentation ground truth.

For sufficiency regarding quality assessment in downstream processes, a detection rate of 91.1% on the University Building implies that 350 elements will go forward to perform quality assessments and measurements (MQA). The thirty-four undetected elements (low-coverage curtain-wall mullions and panels, together with six stairs lacking valid geometry) will be marked "Need Validation," thereby preventing them from affecting computed deviation-statistics as opposed to producing spurious zero-deviation-readings. On the reservoir, 63 elements proceeded to the CQA stage thus providing enough coverage regarding major structural categories (wall, slab, beam) to enable meaningful quality assessments to occur. Furthermore, open reporting of unsegmented elements demonstrates that the DL-first cascade architecture, when implemented with an S3DIS-trained SpU-Net provides measurable element coverage for quality assessments occurring downstream of partial scans and/or poor scans.

4.5.5 Fallback Frequency and Validation Status (Robustness)

A cascading fall-back registry was created using Sections 3.6.3 and 3.9 to create a measure of robustness: the pipeline will record where the cascade fails so the frequency of each validation status can be directly and honestly measured as a level of robustness.

Table 4.13 Distribution of validation statuses across both case studies (DL-First Hybrid Cascade)

Deviation Status	UC Count	UC %	Reservoir Count	Reservoir %
Valid (BBox+Normal+DL)	262	68.2%	40	37.7%
Valid (Proximity Recovery)	88	22.9%	23	21.7%
Need Validation (Low Coverage)	0	0.0%	20	18.9%
Need Validation (No Scan Coverage)	28	7.3%	16	15.1%
Need Validation (No Geometry)	6	1.6%	7	6.6%
Total	384	100%	106	100%

The UC Building has 91.1 percent of its total elements detected (350 of 384). 68.2 percent of the elements were validated during Phase 3 BBox + Normal + DL processing and 22.9 percent were recovered through Phase 4 Proximity Recovery. The 8.9 percent (34 elements) that could not be validated are divided into 28 elements (7.3 percent) with no usable scan coverage - predominantly

curtain-wall mullions and panels with insufficient scan-point density, together with one wall - and 6 elements (1.6 percent), all stairs, whose IFC geometry could not be tessellated into valid B-Rep form. No element required manual override. The presence of both a no-coverage group and a no-geometry group shows that the two failure modes the validation-status system is designed to separate both occur on this case study, and reporting them as distinct statuses rather than as spurious zero-deviation readings is what keeps the deviation statistics of Section 4.7 honest.

The reservoir building contains 59.4 percent of its total elements detected. 37.7 percent of those elements were validated through Phase 3 BBox + Normal + DL and 21.7 percent of those elements were validated using Phase 4 Proximity Recovery. Of the 40.6 percent of the total elements that could not be segmented, 15.1 percent contain no scan points within 1.5 metres of them (walls with their farther face out of view by the scanner's angle of sight, and recessed ventilation windows in concrete walls). 18.9 percent of the non-segmented elements contain near-by scan points; however, their usable point support was insufficient for successful segmentation (roof components which were partially visible, or walls with the scan plane at a shallow angle). Finally, 6.6 percent of the non-segmented elements lack valid IFC geometry. These failures were due to insufficient amounts of scan data or missing details in the BIM models, although the counts do not independently rule out algorithmic failure, which is exactly why a validation status system should be used to report such failures.

The validation status system creates a graded level of confidence about the output. Elements that were designated as "No Scan Coverage" or "Low Coverage" do not by themselves establish an algorithmic false-negative rate; they indicate that there is a known issue regarding available data for that specific element. Separating issues related to confidence from issues related to measurements is important for assessing the compliance/quality of construction work. For example, an element incorrectly reported as "Compliant" simply because it lacked scan coverage would represent greater risk than an element correctly reported as "Unassessable". The overall detection rate of 91.1 percent obtained for the UC Building indicates a good operational condition when there was sufficient scanning and BIM detail for that building. The detection rate of 59.4 percent obtained for the Reservoir indicates a less favorable operational condition for the scanning and BIM detail relative to that building.

4.6 Construction Quality Assessment

This section will report on the point-to-plane deviation statistics that have been computed for the reservoir case study; the resulting context aware severity classifications; whether or not the severity classifications agree; and the extent to which the choice of tolerance profile affects the severity classifications. As such, it will provide evidence as to how effective and reliable the proposed framework is at assessing the quality of construction.

RQ 3 asks how efficient and reliable the framework is for construction quality assessment. This section reports the point-to-plane deviation statistics, the resulting context-aware severity classification.

4.6.1 Deviation Statistics by Element Type (RQ 3)

As a basis for the analysis, the framework calculates for each segmented component the point-to-plane distance according to Equation (3.7) between the as-built scan data associated with that component and the related BIM reference surfaces. Each resulting element-specific distribution is evaluated based on its mean, maximum absolute deviation, standard deviation and RMSE. The evaluation of the aggregation of these metrics by IFC element-type allows for the identification of specific trends

regarding the accuracy of constructions depending on the type of structural and non-structural components.

Table 4.14 Point-to-plane deviation statistics aggregated by IFC element type (mm)

IFC Type	n (Elements)	Mean dev.	Max dev.	Std dev.	RMSE
IfcWall	38	46.9	447.9	21.6	61.9
IfcSlab	10	47.6	241.7	34.3	63.5
IfcBeam	1	53.6	194.6	—	57.3
IfcRoof	3	25.7	55.5	10.0	28.2
IfcWindow	9	29.3	67.8	11.6	32.2
IfcDoor	2	82.3	131.9	5.6	90.6
All elements	63	44.7	447.9	24.4	56.9

The table above provides evidence for many trends. Door types have the largest type-Level mean deviation (82.3 mm), indicating a measured mismatch with the BIM reference, whose cause requires independent verification. Beams and slabs follow (53.6 mm and 47.6 mm, respectively), with Slab 5E having the highest element-Level mean deviation (102.2 mm) and the maximum per point deviation of 241.7 mm. Walls which represent the majority of elements (n=38) have a type-Level mean deviation of 46.9 mm, with a standard deviation of 21.6 mm. The maximum per point deviation measured across all walls (447.9 mm) can be attributed to wall 1A located on Level 1, whose scan points contain a localized area of high deviation near the top edge of this wall. The highest per point deviation measured across all elements was also 447.9 mm.

Windows have a type-level mean deviation of 29.3 mm. The element classifications reported in Section 4.6.2 are two compliant, one minor and six moderate windows; the single beam has a mean deviation of 53.6 mm. Roofs average 25.7 mm and doors 82.3 mm. Across the 63 assessed elements, the element-weighted mean deviation is 44.7 mm and the reported aggregate RMSE is 56.9 mm, approximately 8.9 and 11.4 times the 5 mm wall/slab compliance threshold, respectively. These comparisons describe observed mismatch and do not establish installation defects or good alignment independently of measurement uncertainty.

Table 4.15 Summary of measured geometric deviations and average tolerance by storey

Storey	n (Elements)	Average Deviation (m)	Maximum element mean deviation (m)	Average Tolerance (m)	Storey
Level 1	10	0.0464	0.0847	0.0050	Level 1
Level 2	11	0.0422	0.0764	0.0050	Level 2
Level 3	14	0.0498	0.0879	0.0107	Level 3

Storey	n (Elements)	Average Deviation (m)	Maximum element mean deviation (m)	Average Tolerance (m)	Storey
Level 4	23	0.0361	0.0974	0.0067	Level 4
Level 5	5	0.0717	0.1022	0.0060	Level 5

A breakdown of storeys (as shown in Table 4.15) indicates that the average deviation is highest on Level five (71.7 mm), second highest on Level three (49.8 mm), third highest on Level one (46.4 mm), fourth highest on Level two (42.2 mm), and fifth highest on Level four (36.1 mm). While Level four contains the highest deviating wall (wall 4a at 97.4 mm), the lower average for this storey can be explained due to the high number of moderately deviating elements contained within this storey (n = 23). Thus, even though Level four represents the best performing storey, its mean deviation is approximately 5.4 times its reported average tolerance (36.1 mm / 6.7 mm).

4.6.2 Severity Classification (RQ 3)

Based upon comparison of the mean distance between points and planes versus the context dependent thresholds of the Hydraulic Structure Tolerance Profile (see Table 3.7) each element will be given one of four labels - Compliant, Minor, Moderate or Major.

Under the HydraulicStructure profile in Table 3.7, walls and slabs use thresholds of 5 mm, 15 mm and 30 mm for the compliant, minor and moderate bands. Roofs use 8 mm, 20 mm and 40 mm; doors and windows use 15 mm, 30 mm and 50 mm. Values above the moderate threshold are classified as Major. Figure 4.18 summarises the reported distribution by IFC type.

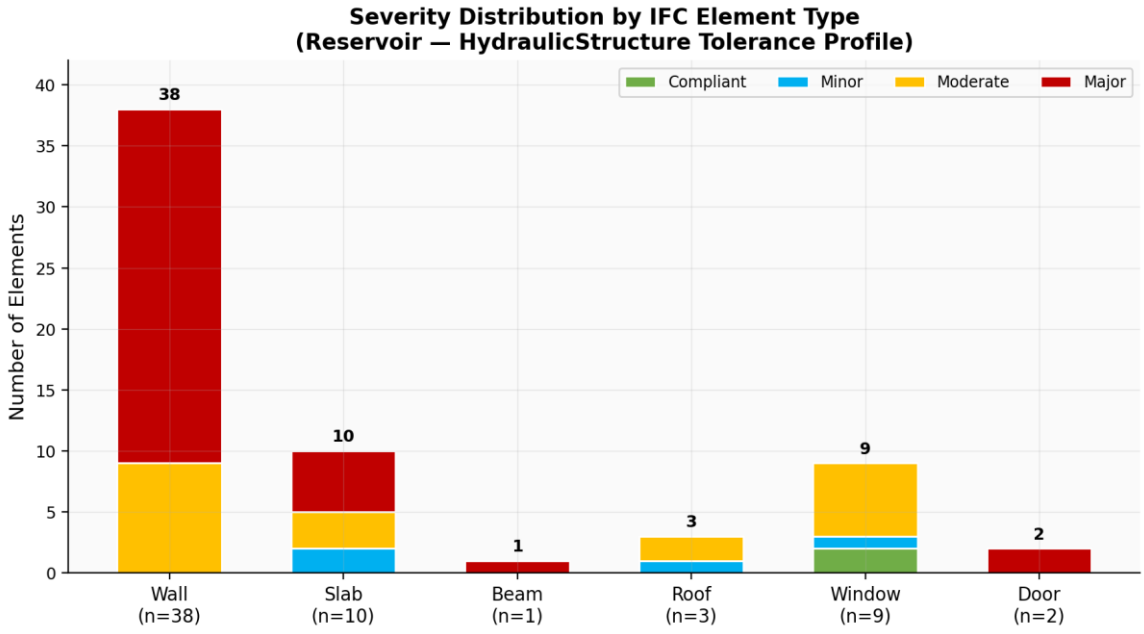


Figure 4.18 Severity distribution by IFC element type (Reservoir case study)

It can be seen that out of those 63 elements, two elements were found to be compliant (3.2%), four elements were minor (6.3%), twenty elements were moderate (31.7%), and thirty-seven elements were major (58.7%). Major-element percentages indicate how much stricter the tolerance profiles for hydraulic structures are compared to the actual geometric variations. Of the thirty-eight total wall

elements, 29 elements were designated major (76.3%), while nine were designated moderate (23.7%). No wall element was given either a compliant or minor designation. All ten slab elements exceeded the five-millimetre compliant designation, and of these five were categorized as major, three as moderate, and two as minor. Both door elements had mean deviation measurements of 76.8 through 87.9 millimetres, so both door elements were also categorized as major. A total of six window elements were categorized as moderate, one as minor and two as compliant, which are the only compliant elements in this data set.

The four minor-categorised elements, Slab 1A (9.6 mm), Slab 2B (10.7 mm), Roof 4B (12.7 mm), and Window 4D (20.7 mm), represented the most approximate examples of being able to meet compliance within this dataset. Along with the two compliant windows, it is clear that there were several building components that were built inside or very close to their tolerance band; however, the bulk of the building's geometry varied significantly enough from the BIM model such that 57 of 63 assessed elements fell into the moderate or major categories.

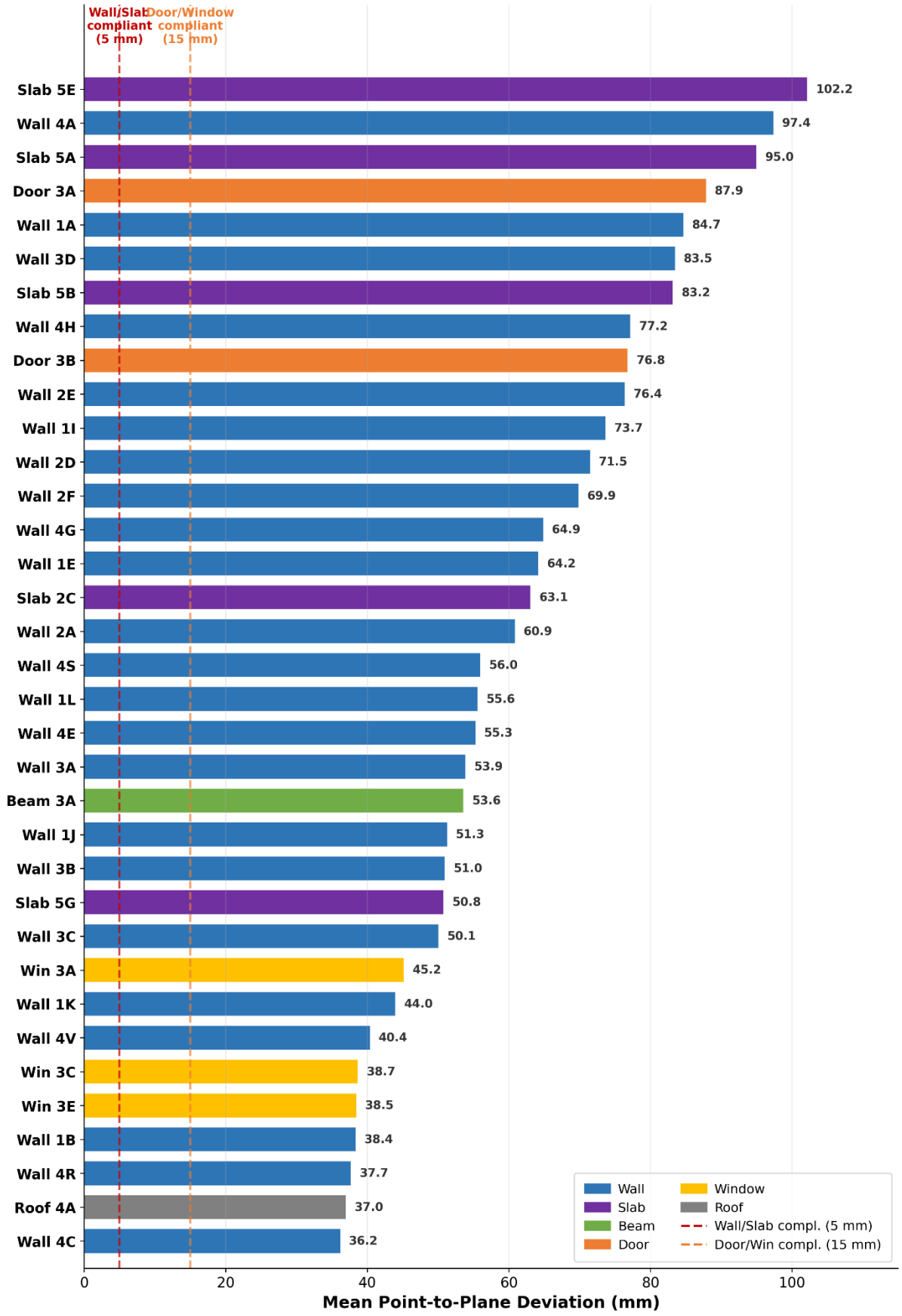


Figure 4.19 Top 35 elements ranked by mean point-to-plane deviation with tolerance thresholds

Figure 4.19 shows the 35 most deviated items from an ordered list. The five-millimetre red dashed line is the wall/slab compliance limit while the fifteen-millimetre orange dashed line is the window/door compliance limit. The colour coding for each item type shows large geometric deviation concentrations in both slab (purple) and wall (blue) items over all levels. Deviations were largest at level 5 (slab 5E), and level 4 (wall 4A); windows are clustered primarily at levels 3 and 4.

Each element is assigned a severity (Compliant, Minor, Moderate, Major/Critical) by comparing its mean deviation against the HydraulicStructure thresholds of Table 3.7 using the logic of Table 3.8. The distribution of severities is the principal construction-quality outcome.

4.6.3 CQA Computational Cost

The overall computational cost of executing the CQA stage (which includes the point-to-plane distance calculation for each of the designated scan points, statistical evaluation for each element, retrieving a tolerance profile by comparison, determining the level of severity based on the retrieved tolerance profile, and determining the validation status of each element) was 3.0 s using the consumer-grade workstation of Table 4.3, which computed the CQA stage for 63 elements. Therefore, the average time taken to execute the CQA stage per element was approximately 48 ms. The majority of this cost was due to performing the point-to-plane projection and searching for the nearest neighbour as opposed to processing the classification logic. In addition, it is noted from Section 4.8 that the CQA stage accounts for 2.4% of the total time required to execute the entire pipeline from start to finish and is therefore considered to be one of the fastest stages; and further supports the conclusion that there are no significant added costs associated with assessing quality when compared to the high computational intensity of both registering and segmenting data.

4.7 Model Quality Assessment

The MQA evaluates the fitness of the BIM model itself, independent of the construction quality of the physical structure. Following the design of Section 3.8, the MQA produces a tolerance-based geometric deviation score for each element together with an independent engineering pass/fail safety net. The MQA was applied to the UC Building case study, which contains 384 IFC elements spanning nine element types across eight storeys. Of the 384 IFC elements, 350 satisfied the assessability conditions defined in Section 3.8.1 and were included in the quality calculation; the remaining 34 were not assessed (28 returned no usable scan coverage and 6 could not be tessellated into valid B-Rep geometry).

4.7.1 Element-specific Quality Scores and Grades (RQ 3)

Each element receives a quality score on a 0–100 scale derived from the ratio of its mean deviation to the element-specific tolerance (Equation 3.9), supplemented by an independent pass/fail safety-net check (Equation 3.10) that flags any element whose RMSE exceeds 1.5 times the tolerance. The scoring formula maps an element with zero mean deviation to 100 and an element whose mean deviation reaches its tolerance to 0, decreasing linearly in between. The letter-grade mapping follows Table 3.10: A (≥ 90), B (75 ≤ score < 90), C (60 ≤ score < 75), and D (< 60).

Table 4.16 Tolerance-specific quality scores, grades, and pass/fail verdicts by element type

IFC Type	Tolerance	Mean Score	Grade Dist. (A/B/C/D)	% Pass
IfcColumn	±10 mm	19.4	1 / 2 / 0 / 14	0.0%

IFC Type	Tolerance	Mean Score	Grade Dist. (A/B/C/D)	% Pass
IfcMember (mullions)	±10 mm	3.1	0 / 2 / 1 / 92	0.0%
IfcPlate (curtain-wall panels)	±10 mm	0.0	0 / 0 / 0 / 44	0.0%
IfcWall	±15 mm	13.7	4 / 4 / 5 / 134	0.0%
IfcSlab	±15 mm	27.2	3 / 0 / 1 / 17	0.0%
IfcDoor	±15 mm	7.8	0 / 0 / 1 / 16	0.0%
IfcWindow	±15 mm	6.8	0 / 0 / 0 / 5	0.0%
IfcCovering	±15 mm	21.5	0 / 0 / 1 / 3	0.0%
All elements	—	9.9	8 / 8 / 9 / 325	0.0%

Table 4.16 shows that no element category passed once the element-specific tolerance and the RMSE safety net are applied. Across all 350 assessed elements the mean quality score is only 9.9 of 100 and, classified by deviation severity, the assessed set comprised 49 Compliant (14.0%), 44 Minor (12.6%), 56 Moderate (16.0%) and 201 Critical (57.4%). Element categories nonetheless differ markedly: slabs (27.2), coverings (21.5) and columns (19.4) achieve the highest mean scores, walls sit in the middle (13.7), and the openings and curtain-wall components score lowest - doors 7.8, windows 6.8, mullions (IfcMember) 3.1 and curtain-wall panels (IfcPlate) 0.0. The mean absolute deviation across the assessed set is 67.1 mm against a 10–15 mm tolerance band, but these results alone cannot separate measurement-chain error from discrepancies in BIM geometry. The lowest scores concentrate on the slender curtain-wall panels and mullions, which carry the tighter ±10 mm tolerance and are simultaneously the hardest elements for the pipeline to segment and measure.

Importantly, the safety-net check (Equation 3.10) failed several elements that would have received a passing score based on mean deviation alone, but whose RMSE exceeded 1.5 times the element-specific tolerance. These typically include curtain-wall panels where a single corner vertex deviates while the panel's average remained within tolerance, and doors whose hinge-side geometry exhibits localised distortion. This illustrates the purpose of the safety net, although the underlying cause of a local mismatch still requires independent verification.

4.8 END-TO-END COMPUTATIONAL FEASIBILITY

The whole as-built-to-report process took 125.7 seconds (as presented in Table 4.17) when run on a commodity workstation assessing the 63 segmentable components of the reservoir. Peak resident memory used was 305 MB (peak CPU host RAM, a resource distinct from the GPU VRAM used by the deep-learning segmentation model, which reaches up to ~1.2 GB per Table 4.12) when computing per-element segmentation. As such both time and memory usage are far below the limits set out in the opening of Chapter 1; approximately 2 min for the 5 minutes limit, and under 1/3 of a gigabyte for the 1 GB limit.

Registration is the largest stage in the integrated reservoir run: 74.9 s, or 59.6% of the 125.7 s total. Semantic inference and filtering add 8.8 s. Relative to the approximately 164 s unfiltered registration

described for the same integrated configuration, the reported gross registration saving is about 89 s and the net saving after inference is about 80 s. This comparison is distinct from the standalone FGR registration benchmark of 272.4 ± 24.1 s in Table 4.4. The remaining integrated stages are segmentation (23.2 s), preprocessing (7.6 s), BIM processing (5.1 s), quality assessment (3.0 s), ingestion (1.9 s) and export (1.2 s).

Removing the points prior to registration also affects how the final alignment results compare to those obtained using the full cloud data. The final fitness value increased from 0.520 to 0.665, but there was little if any effect on the Inlier RMSE, changing from 76.8 mm to 76.1 mm. An important aspect of interpreting this pair of values is to avoid over-interpreting their comparison. Fitness represents the percentage of source points that have a target correspondence within the specified evaluation distance; removing 68,880 points that would have been impossible for any source point to ever have a correspondence with can increase this fraction by reducing its denominator regardless of whether the alignment position had actually changed significantly. This is demonstrated by the fact that even though the final fitness value increased, the Inlier RMSE decreased by only less than one millimetre indicating that the alignment position did not change much at all. Thus, what removing these points achieves is a faster execution time for an alignment of similar geometric accuracy, and also one that produces a fitness score for a cloud that reflects that geometric accuracy better since it is no longer affected by points that the model was not expected to explain.

The reported reservoir run demonstrates completion of the integrated workflow in approximately two minutes on the tested CPU/GPU workstation. It does not establish a CPU-only runtime or a direct segmentation-accuracy advantage over learned models: the classical and learned benchmarks use different priors, and the case studies have no independent segmentation ground truth.

Table 4.17 Stage-wise wall-clock time and peak memory for the complete pipeline

No	Pipeline Stage (Ch. 3)	Section	Time (s)	% of Total	Peak Mem. (MB)
1	Ingestion & coordinate reconciliation	3.3	1.9	1.5 %	7
2	BIM processing / artificial PCD	3.4	5.1	4.1 %	26
3	Pre-processing (voxel, normals, FPFH)	3.5.1	7.6	6.0 %	54
3.5	SpU-Net semantic inference & pre-registration clutter purge	3.5	8.8	7.0 %	176
4	Candidate registration on the purged cloud (FGR / ICP / yaw)	3.5	74.9	59.6 %	175
5	Element segmentation (reuses Step 3.5 predictions)	3.6	23.2	18.5 %	305
6	CQA / MQA	3.7–3.8	3.0	2.4%	230
7	Export and reporting	3.10	1.2	1.0 %	231
Total end-to-end		—	125.7	100 %	305

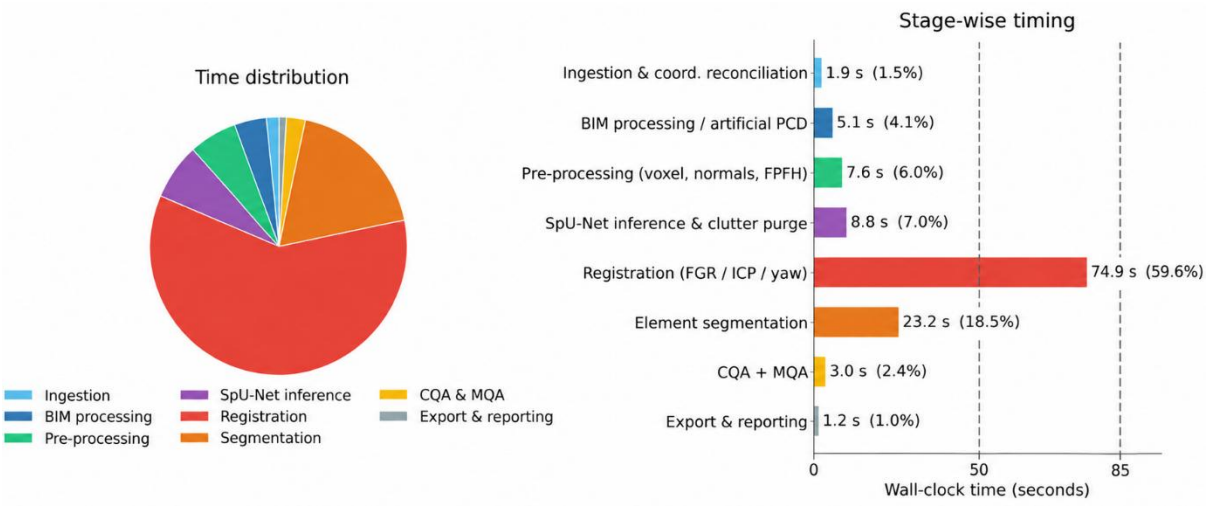


Figure 4.20 Breakdown of end-to-end processing time by pipeline stage

As anticipated from the prior assessments of the application, the majority of the application's execution time was used to perform geometric operations. The longest time was consumed when registering candidates, which took 74.9 seconds (59.6%) and then element segmentation at 23.2 seconds (18.5%). It took 8.8 seconds (7.0%) to perform semantic inference and remove pre-registration clutter. Point cloud preprocessing consisting of voxel down-sampling, surface normal estimation and extraction of FPFH features used up 7.6 seconds (6.0%). The other phases of the process; ingest data, process BIM, assess quality and export, all together consume less than 10% of the total execution time. Therefore, these results suggest that the computational burden imposed by the assessment and report generation modules are relatively minor compared with those due to geometric processing.

The time distribution during execution is illustrated as shown in Figure 4.20. From this illustration it can be observed that roughly 91.1% of the entire processing time was utilized performing geometric operations, namely preprocessing, semantic inference, registration and segmentation, while the quality assessment and reporting component consumed an extremely minute amount of the total runtime. This time distribution clearly demonstrates that the fundamental design goal of developing lightweight assessment modules which may be repeatedly invoked on previously processed geometry without invoking the computationally expensive registration and segmentation phases has been achieved. The entirety of the Scan-to-BIM pipeline therefore ran in 125.7 seconds using a maximum of 305 MB of RAM; thus, demonstrating that the framework can be completely executed on a standard desktop equipped with a consumer-grade CPU and GPU.

The integrated timing results support computational feasibility on the evaluated CPU/GPU workstation. Segmentation coverage supports downstream demonstration of quality assessment, but does not independently establish segmentation accuracy or metrological conformity.

4.9 DISCUSSION

When viewed collectively, the experiments provide support for the central premise of the thesis; a lightweight Scan-to-BIM framework can produce actionable assessments of construction and model quality while remaining computationally feasible on commodity hardware. In this section, the results from Sections 4.5 through 4.8 will address each of the research questions raised and be synthesized into one discussion.

Regarding registration (RQ1), the FGR-based pipeline matches the reservoir result and achieves higher final UC fitness than the RANSAC-based pipeline, with shorter total registration time in both cases. Its UC inlier RMSE and runtime coefficient of variation are higher, so the evidence does not support a general claim of lower error or greater stability. The candidate sweep is important for resolving the reservoir's yaw ambiguity.

The reasoning behind the best candidate approach is clearly reflected as expected. Because each of the classical operators worked well with one or another of the various types of structural elements, and did poorly with some other type of element; using the best-candidate approach to generate multiple candidate-segmentation options for each element, and select from among them, based on coverage and compactness resulted in better segmentation options for each element than would have been possible by selecting a single classical segmentation option.

Regarding segmentation (RQ2), Adaptive BBox has the highest reported clean-data mIoU among the classical operators. The original claim of a 15-20% precision loss relative to learned baselines is not supported by the presented tables. Using the corrected macro-averages, PTv3 ranks first on ScanNet (0.404), while SpU-Net ranks first on S3DIS (0.355) and Rohbau3D (0.263). Direct speed comparisons remain limited because CPU time is reported per element and GPU inference time per scene or workload. These findings motivate a hybrid architecture but do not by themselves prove its superiority over every individual operator.

Regarding Quality Assessment (RQ 3), the MQA scores for the UC Building illustrate that the geometric shortfall is systematic and type-specific rather than random. The slender curtain-wall panels and mullions scored lowest (0.0 and 3.1) because they carry the tighter ±10 mm tolerance and the cross-sections hardest for the lightweight pipeline to segment; this is consistent with the difficulty of measuring slender curtain-wall systems from a lightweight scan.

Finally, to address the overall computational budget required to perform all steps within the end-to-end process, the entire as-built-to-report process was executed on commodity hardware using 125.7 seconds with 305 MB as the highest level of memory utilized. The introduction of semantic filtering added 8.8 seconds for the inference task but removed over 89 seconds from the registration task via eliminating non-structural data points that would have been unable to generate corresponding features. Therefore, the increased computational burden due to the additional inference task is more than outweighed by the decrease in computation time for the registration task.

4.9.1 Threats to validity

The case studies differ in element count, geometry, acquisition method and scan coverage, limiting direct comparison. Their point clouds were not independently segmented manually, so detection rates cannot establish segmentation accuracy or eliminate false positives. The public benchmark annotations provide independent semantic labels, but their uncertainty, class mapping and the source of geometric priors must be considered when interpreting the benchmark results.

The deep learning baselines were initialised from pretrained weights on indoor benchmarks and therefore the accuracy reported includes both in-domain evaluation and cross-domain transfer, rather than their best achievable performance after target-domain adaptation. A more accurate measure would be to evaluate labelled construction datasets of sufficient size to allow fine tuning of those same networks although such datasets do not currently exist at scale.

The artificial reference cloud is only as good as the IFC tessellation generated by the authoring tool. Therefore, if Revit uses a default tessellation that introduces geometrical artifacts (i.e. faceting of curved surface, simplification of junctions) then this will affect the point-to-plane distance measurement independently of any errors present due to construction.

Furthermore, since the 0.10 m voxel resolution utilized throughout the pipeline sets an absolute measurement limit which is higher than many of the tolerances listed in the tolerance table; conclusions regarding elements which are measured as being very close to this measurement floor (approximately 50 mm) should be made knowing that the measurement accuracy is limited by the resolution of the downsampled point cloud.

Another issue is the circular dependence in BIM guided segmentation: the pipeline utilizes BIM geometry to establish the bounds of each element and the normals on the surfaces for segmentation, and then calculates the deviations from the segmented points relative to the same BIM geometry. Therefore, if there is an error in the BIM model (i.e., it is not simply a tessellation artifact), both the segmentation process and the calculation of deviations will be impacted.

The MQA module (Section 3.8) addresses this problem to some extent by grading the model geometrically against its own type-specific tolerance (and also flags elements in the validation status system where BIM geometry is missing or degenerate). However, the framework cannot identify instances where although geometrically correct, the BIM model does not represent what was intended.

4.9.2 Error Budget and Noise Sensitivity

Deviation values listed within Section 4.6 and Section 4.7 provide the cumulative difference between the as-constructed scans and the as-designed BIM models. These cumulative differences represent the accumulation of many different error sources. To understand the correct interpretation of the Quality Assessment Verdicts (QAVs) provided in Section 4.10, it is necessary to identify the primary contributors to the cumulative deviation:

1. Registration Error
   The standalone registration benchmarks report final inlier RMSE values of 76.8 mm for the reservoir and 104.3 mm for the UC FGR pipeline; the semantically filtered integrated reservoir run reports 76.1 mm. Inlier RMSE is a correspondence residual, not a direct estimate of a uniform translation or an independently measured registration uncertainty. Residual rotation can produce spatially varying deviation patterns, and even a translation projects differently onto surfaces with different normals. Local patterns therefore cannot rule out registration error. The alignment provenance of the pre-aligned UC assessment must be documented separately.

2. Segmentation Error
   When the segmentation cascade incorrectly identifies scan points belonging to an element (for example, when points from an adjacent wall leak into a column segment), then the deviation statistics calculated for that element are compromised. The Validation Status System (see Section 3.9) helps mitigate this problem by identifying those elements for which a fallback was required during the segmentation process. On the UC Building, 91.1% of Elements were detected through primary segmentation or proximity recovery; on the Reservoir, 59.4% of elements were validated. Therefore, care must be taken when interpreting deviation statistics for Elements designated as Need Validation since there is no assurance regarding the segmentation quality.

3. Scanner Noisy Positional Data
   Scanner noise depends on the device, range, surface and acquisition conditions. The dissertation does not provide a calibration sufficient to establish a sub-millimetre error for the reservoir or a precise uncertainty for the Matterport data. Voxel downsampling reduces point density and can smooth high-frequency variation, but its effect on deviation uncertainty must be measured or derived under an explicit model. Small-tolerance verdicts therefore require independent validation.

4. Actual Construction Defects
   True differences between construction and design are one possible contribution to measured deviations. Because the study does not independently quantify registration, scanner and segmentation errors at element level, their contributions cannot be separated conclusively. Physically plausible patterns warrant investigation but are not proof of defects. Repeated element-level verdicts have not been formally evaluated; this remains future work in Section 5.5.

The discussion above identifies the principal contributors to cumulative deviation, while Table 4.18 outlines how various stages in the pipeline address specific types of noise.

Table 4.18 Noise types and their treatment in the pipeline

Noise type	Source	Pipeline mitigation	Residual effect
Random positional error	Scanner precision	Voxel downsampling (spatial averaging)	Suppressed below voxel resolution
Trajectory drift	MLS/SLAM accumulation	Not directly addressed; TLS data used for primary case study	Systematic bias in MLS scenarios (limitation)
Occlusion	Line-of-sight obstruction	Validation status: "No Scan Coverage" flag	Elements without data are excluded, not falsely reported
Clutter	Non-structural objects	DL clutter purge (Phase 2 of Hybrid Cascade)	DL clutter purge, executed once before registration and reused before segmentation; 32.9 % of the registration cloud and 18.9–44.3 % of the working scan removed; residual clutter may remain
Registration residual	Coarse-to-fine alignment	Candidates sweep resolves yaw ambiguity; ICP minimizes residual	Residual alignment error may affect deviation estimates. The integrated reservoir run reports 76.1 mm inlier RMSE; the independent benchmarks report 76.8 mm (reservoir) and 104.3 mm (UC FGR). The pre-aligned UC input requires a separately documented alignment-accuracy assessment.
Segmentation leakage	Adjacent-element contamination	Normal-consistency filtering and cascading fallback registry	Flagged via validation status; not eliminated

The results of this study demonstrate how a quality evaluation process can be applied in practice. Deviations substantially larger than the registration RMSE (for example, > ~100 mm) warrant investigation, but the current evidence cannot distinguish construction defects from measurement

uncertainty with conclusive confidence. When the deviations measured by the proposed quality evaluation process are close to or smaller than the registration RMSE, then these evaluations should be viewed as indications rather than absolute. It is an inherent limitation of any Scan-to-BIM quality assessment process using rigid body registration. Formal confidence limits could be assigned to each indication based upon the outcomes of future research related to uncertainty estimation and therefore provide a basis for making decisions based upon a risk factor.

4.10 SUMMARY

This section discusses the results of analysis that involved two different cases. Case Study 1 investigated the inspection of a reinforced concrete hydraulic reservoir based upon its execution class 2 EN 13670 tolerance requirements (i.e., CQA) while Case Study 2 inspected a University Building and associated BIM model that contained 384 IFC objects across multiple object types (MQA). In both cases, the segmentation phase utilized a DL-First Hybrid Cascade Architecture which used the SpU-Net trained for S3DIS-based structural features globally and clutter removal at each level of detail, then segmented each level of detail in the BIM models using a DL tiebreaker. All three stages of registration, segmentation, and quality assessment were examined individually and collectively as a complete workflow, and the computational cost of each stage was summarized within an end-to-end feasibility budget on commodity hardware.

The primary findings can be described as follows:

Firstly, when the CQA module was run for the reservoir using the hydraulic-structure tolerance profile, it classified 58.7% of elements as major and 31.7% as moderate. These classifications indicate large measured deviations relative to the BIM model, but not conclusive construction defects because of registration uncertainty.

Second, when the MQA module graded the 350 of 384 assessable UC Building elements against their element-specific geometric tolerances (±10 mm and ±15 mm by type), no element passed: the deviation-severity distribution was 49 Compliant (14.0%), 44 Minor (12.6%), 56 Moderate (16.0%) and 201 Critical (57.4%), and the mean quality score was 9.9 of 100, distributed across four grade bands as 8 A, 8 B, 9 C and 325 D. Slabs, coverings and columns scored highest and the curtain-wall mullions and panels lowest. The low scores are geometric: a mean absolute deviation of 67.1 mm against a 10–15 mm tolerance band shows a measured mismatch but does not identify how much arises from the measurement chain or the BIM geometry.

Third, the full end-to-end processing chain consumed at its highest level of memory use roughly 305 megabytes, and took about 125.7 seconds to finish using typical desktop computer hardware. The addition of another inference step due to semantic filtering did add some additional computational load; however, the reduction in registration time caused by removing many of the non-structural point objects was enough to make up for it, as well as result in a very computationally efficient process overall.

5
CONCLUSIONS

5.1 SUMMARY OF WORK

This dissertation set out to address a structural disparity in the construction industry: the rapid sophistication of design-stage digital tools has not been matched by an equally efficient means of verifying what is actually built on site. BIM provides a rich digital representation of design intent, and laser scanning provides a dense record of as-built reality, yet the Scan-to-BIM workflow connecting the two—registration, segmentation and quality assessment—has remained fragmented, manual in parts, and dependent on computationally intensive workflows that often require high-end GPU infrastructure for deep learning components. In response, the thesis proposed, implemented, and evaluated a lightweight, end-to-end Scan-to-BIM framework that performs the complete as-built-to-report workflow on consumer-grade CPU and GPU hardware, integrating construction quality assessment and BIM model quality assessment within a single pipeline.

This concluding chapter draws the work together. Section 5.2 revisits each of the four research questions (one primary and three sub-questions) and answers it from the evidence of Chapter 4. Section 5.3 states the methodological, technical, and practical contributions. Section 5.4 sets out the limitations honestly. Section 5.5 recommends directions for future research, and Section 5.6 offers concluding remarks.

5.2 ANSWER TO RESEARCH QUESTIONS

This study was driven by one primary research question and three supporting questions developed in Section 1.2.2. Below are each of these questions recited along with answers referencing experiments within Chapter 4.

5.2.1 Primary Research Question - A lightweight End-to-end Feasibility

The main research question asked whether an end-to-end Scan-to-BIM process running on commodity hardware could generate indicative construction and BIM model quality assessments. Chapter 3 defines how the pipeline works. Section 4.8 demonstrates that the framework can complete the entire as-built-to-report pipeline in 125.7 seconds using 305 MB of RAM and generate BIM-integrated deliverables (a PSet-annotated IFC4 model, an IfcGroup for scheduling, a deviation-coloured E57, and a CSV schedule) that can be directly integrated into the engineers' authoring environment.

In comparison to the three constraints listed in the opening of Chapter 1, the framework meets each constraint. As shown in Section 4.8, execution time is approximately 126 seconds, which is well under

the five-minute limit. The maximum amount of peak memory usage is 305 MB (CPU host RAM) which is also below the 1 GB limit. While the learned component improves segmentation and substantially reduces registration time (from approximately 164 s to 75 s), it does not prevent the output generation. Therefore, the evidence supports an affirmative answer to the primary research question; however, there is one important limitation that should be explicitly stated – all deep learning results reported in this thesis were run on an NVIDIA RTX 3070 Ti and CPU inference performance was not tested for the semantic filter.

As illustrated in Figure 4.20, registration accounts for approximately 59.6% of the total processing time (approximately 75 seconds). Registration is not overhead. Section 4.4 illustrates that on self-similar concrete geometry no global operator will recover the correct pose and the best candidate is yaw. The framework spends seventy-five seconds to obtain an acceptable alignment instead of a quick alignment and the DL first architecture provides a secondary benefit by removing 68,880 non-structural points prior to registration thereby reducing those seventy-five seconds from what would have been 164. These pipeline timings (approximately 164 seconds for the unfiltered cloud and approximately 75 seconds [74.9 seconds; see table 4.17] post-semantic purge) were captured during an end-to-end execution and represent times distinctly different from the independently executed full-resolution FGR benchmark time of 272.4 ± 24.1 seconds presented in table 4.4 (section 4.4) where the registration operator was evaluated independent of its role as part of the complete pipeline.

5.2.2 RQ 1 - Registration

The FGR-based pipeline matches the RANSAC-based pipeline's final reservoir fitness (0.520) and gives higher final fitness on the UC case (0.490 versus 0.344). Total registration is approximately 1.21–1.27 times faster. However, FGR has higher UC inlier RMSE and higher UC runtime variability, so fitness, error and stability must be reported separately. The candidate sweep resolves the reservoir's yaw ambiguity in both pipelines. These case-specific findings support choosing FGR as the default global operator in the evaluated framework.

5.2.3 RQ 2 - Segmentation

Three separate analyses are captured within RQ 2. The first is the analysis comparing four classical segmentation operators (Section 4.5.2) to determine whether there exists an operator that can outperform the others for each type of element and level of noise. This was completed to confirm the decision to use a best-candidate selection architecture based upon these results. The results were that RANSAC plane extraction was the fastest of the four options at 1.3 – 1.6 milliseconds per element; The Adaptive Bounding Box had the best clean-data accuracy (up to mIoU = 0.978 on ScanNet) of the four options. All four options experienced a degradation of approximately 27.5–36.6% on average across the three datasets when operating in noisy environments.

The second analysis benchmarked three deep-learning architectures (Section 4.5.3). Using the macro-average of the displayed IFC-class IoUs, PTv3 achieved the highest ScanNet result (mIoU-IFC = 0.404) and decreased by approximately 52.2% on Rohbau3D (0.193). SpU-Net achieved the highest results on S3DIS (0.355) and Rohbau3D (0.263). Minority-class performance was weak: beam and column IoUs were near zero, while door and window IoUs reached 0.312 and 0.174, respectively, on S3DIS. Under the pre-trained conditions examined here, learned segmentation alone therefore did not provide reliable element-level correspondence for quality assessment.

The third analysis, the DL-First Hybrid Cascade (Section 4.5.4) identified 91.1% of the elements in the UC Building (350 / 384) and 59.4% of the elements in the Reservoir (63 / 106) through this process. Undetected elements were labelled by scan-coverage and geometry-related failure categories; those labels do not independently exclude algorithmic causes.

5.2.4 RQ 3 - Quality Assessment

RQ 3 examined how reliable the quality judgments made by the framework were. The CQA module (Section 4.6) provided tolerance-based classification of 63 Reservoir elements against EN 13670 Class 2 tolerances, demonstrating consistent classification across all tested Reservoir elements. The MQA module (Section 4.7) graded the 350 assessable UC Building elements (of 384) against their element-specific geometric tolerances and recorded a deviation-severity distribution of 49 Compliant (14.0%), 44 Minor (12.6%), 56 Moderate (16.0%) and 201 Critical (57.4%) with a mean quality score of 9.9 of 100. Mean scores ranged from 27.2 (slabs) and 21.5 (coverings) down to 3.1 (mullions) and 0.0 (curtain-wall panels), and the four grade bands were distributed as 8 A, 8 B, 9 C and 325 D. The low ratings are geometric rather than informational: the mean absolute deviation across the assessed set is 67.1 mm against a 10–15 mm tolerance band, and the pass/fail verdict is driven by the RMSE safety net in every case. These are preliminary conclusions, because the deviations exceed the applied tolerances by a wide margin (see Section 5.4); it therefore cannot yet be determined whether errors of this magnitude reflect construction variability or measurement uncertainty. The value of RQ 3 lies in demonstrating the technical feasibility of integrating context-aware assessment into the pipeline, rather than in providing a definitive conformity ruling.

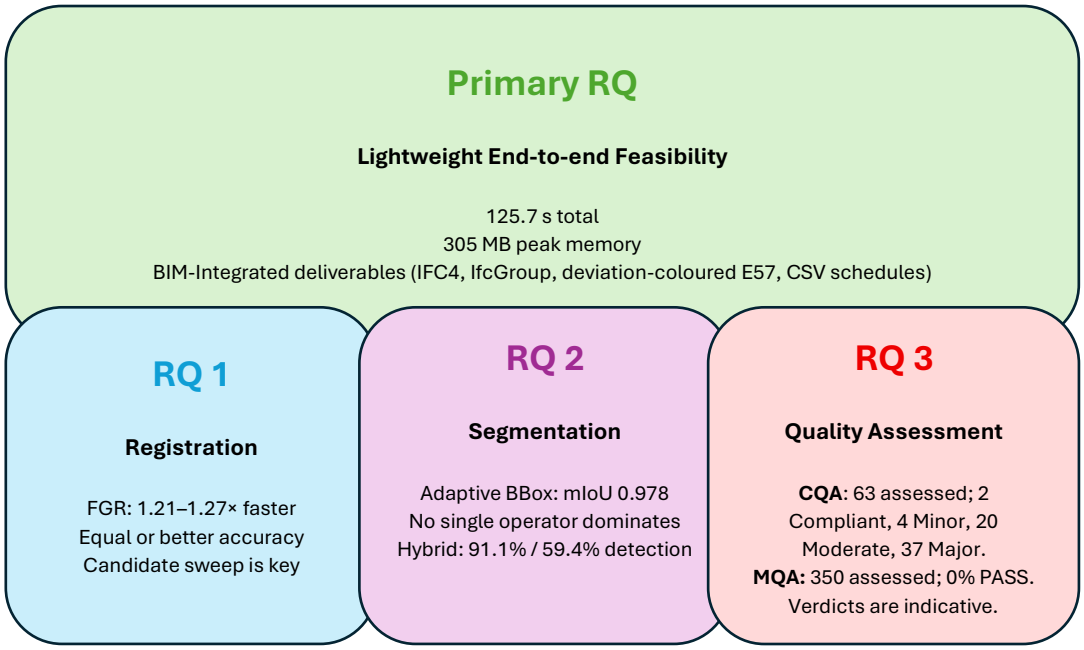


Figure 5.1 Summary of research questions and their evidenced answers

5.3 RESEARCH CONTRIBUTIONS

The dissertation presents the following research contributions. Together they represent advances in the state-of-the-art described in Chapter 2.

5.3.1 Methodological Contributions

- An integrated, single-pipeline formulation that combines the individual processes of registration, segmentation, geometric deviation evaluation, construction quality evaluations and BIM model quality assessments into a unified end-to-end process - a combination of these separate task processes is generally addressed in isolation in the literature.
- A separation of the confidence associated with measurements (i.e., the status system for validating measures) from the values themselves (i.e., deviations), designed to prevent false precision often found when automating quality assessment.
- A tolerance classification that uses context-aware classifications for establishing severity rankings of measured deviations relative to type specific building codes (e.g., EN 13670, EN 1992-3, ISO 4463-1) instead of relative to a single absolute threshold.
- A DL-first pipeline architecture that performs a single semantic inference before both registration and segmentation, enabling semantic predictions to be reused across both stages while reducing redundant computation.

5.3.2 Technical Contributions

- The development of a fully functional, lightweight client–server implementation capable of executing the complete workflow on commodity hardware, incorporating a centroid-seeded coarse-to-fine registration approach with a three-candidate sweep, selecting the transformation with the highest alignment score.
- The development of a Model Quality Assessment that grades each element geometrically against the tolerance assigned to its type (±10 mm or ±15 mm), producing a 0–100 quality score, an A–D letter grade, and an independent PASS/FAIL safety net that fails any element whose maximum local deviation exceeds its tolerance or whose RMSE exceeds 1.5 times that tolerance. Separating the score from the verdict lets an element grade well yet still fail, which flags elements that are acceptable on average but locally defective.
- Development of a collection of BIM-integrated and interoperable products (a P-set annotated IFC4 model, an IfcGroup for Revit scheduling, and a deviation-coloured E57) that provide assessment results back to the engineer's original authoring environment.

5.3.3 Practical Contributions

- Empirical data from two challenging test cases (the hydraulic reservoir assessed against EN 13670 Class 2 tolerances and the multi-storey university building with 384 IFC objects) show that a geometrically driven pipeline extended by selective DL filtering can support practical deployment on the evaluated datasets while reducing computational requirements. Its quality verdicts remain indicative until independently validated at the required tolerance scale.
- A reusable comparative protocol for reproduction: voxel resolution, neighbourhood configuration, and operator, which other researchers can reproduce to compare registration and segmentation algorithms fairly.

5.4 LIMITATIONS

The conclusions are bounded by the following limitations, several of which were anticipated in the scope and delimitations of Section 1.2.4 and the threats to validity of Section 4.9.1.

- **Two case studies:** The framework has been tested with two case studies: a Reinforced Concrete Hydraulic Reservoir (106 IFC elements / terrestrial laser scan) and a Multi-Storey University Building (384 IFC elements / Matterport scan). These two studies are representative of different structural typologies (infrastructure and educational buildings), construction methodologies (reinforced concrete and mixed construction), scanning technologies (TLS and structured-light RGB-D), and numbers of elements checked (CQA 63; MQA 350 of 384 assessable). Generalization to other building typologies (industrial steel-framed, timber, etc.) and to other acquisition systems (MLS SLAM-based, UAV photogrammetry, etc.) is, however, still needed.
- **Idealization based on the quality of the IFC Tessellation**: When the Authoring Tool exports via IFC, it creates an artificial Reference Cloud. If the Exporter tessellated the geometry badly, or omitted portions of it, then those issues will propagate through to the Deviation measurement, and also as "NO_GEOMETRY" flags in the MQA.
- **Learning is used primarily as a segmentation enhancer:** No additional network training was performed in this study. Although the final hybrid pipeline uses the trained S3DIS-SpU-Net for clutter purging and class tie breaking, it only uses the deep learning component as an aid for filtering and semantics, not as a main segmentation mechanism. Therefore, the upper limit on accuracy that could be reached by having a more thoroughly embedded learned component (for example, for semantically differentiating self-similar objects) is not being utilized.
- **Sensitivity to Parameters**: For the sake of fairness, a single voxel size and neighbourhood configuration was used across all points. However, performance with point clouds with drastically differing densities may require re-tuning, and this work did not automate that process.
- **Absence of an assessor study**: no independent multi-assessor validation or inter-rater reliability study is reported. Such a study would provide an external reference for severity labels and help distinguish measurement uncertainty from actual construction defects.
- **Deterministic Band Tolerance**: Elements that have mean deviations close to a severity band's boundary can vary depending on small differences in how the segmentation was done. While a "review" designation will flag these elements for review, a formally defined run-to-run repeatability study has not been conducted (as noted in Section 5.5), therefore it is recommended that this be studied at some point.
- **Propagation of the registration error**: The last registration achieved an inlier RMSE of 76.8 mm (Reservoir), and 104.3 mm (UC Building) respectively. Both of these RMSE values greatly exceed many compliance thresholds in the tolerance table (for example, the 5 mm threshold for walls under EN 13670 Class 2), by greater than one order of magnitude. Although the deviation maps show localised spatial patterns, the system cannot provide an analytical decomposition of the total measured deviation into registration error and true construction defects. Therefore, judgments made regarding deviations within a range of or less than the registration RMSE should be viewed as indicative rather than conclusive. This limitation exists due to the nature of all scan to BIM quality evaluation systems using rigid body registration.
- **Spatial measurement resolution floor**: The most commonly used voxel resolution of 0.10 m acts as a spatial low pass filter which reduces geometric detail smaller than about 50 mm. Many compliance thresholds in the tolerance table fall below the limits of resolution of the system (i.e. they fall below the ability of the system to distinguish between 5 mm construction defects

and 5 mm discretization noise at this resolution). A lower voxel size will result in better resolution; however, the computational processing time and memory requirements will increase nonlinearly.

- **Robustness to noise**: The degradation of the classical segmentation operators when subjected to synthetic noise conditions was found to be in the range of approximately 27.5–36.6% on average across datasets (See Section 4.5.2, Table 4.6). Although the DL-First Hybrid Cascade can remove clutter via its semantic purge phase, no specific noise reduction techniques have been incorporated into the pipeline other than voxel downsampling. In addition, because both case studies utilised TLS or Structured-Light scanning technologies, there was no systematic registration bias introduced by MLS-derived point cloud data collected via trajectory drift; therefore, the coarse to fine approach employed in this study did not need to account for this type of issue. However, deployment onto SLAM-based MLS data sets requires additional verification outside of what has been reported here.
- **Circular dependency between BIM guidance and measured deviations:** the pipeline uses BIM geometry both to guide segmentation and as the reference for deviation measurement. MQA uses those same deviations and does not provide an independent geometric validation that removes this dependency. Independent survey control, manually verified segments or other external reference data are needed to test model errors and selection bias.

5.5 FUTURE DEVELOPMENTS

Current research uses deviation statistics and severity classification for each component; however, it is not explicitly stated whether there is an inherent consistency between the individual results from previous runs. Even though the deployed segmentation operators are deterministic, a component which has a mean deviation value close to a severity band can nonetheless be classified differently between runs owing to small perturbations from registration residuals and voxelization. A future area of research could include conducting a formal repeatability experiment. This would involve running the entire pipeline at least R = 10 times on every case study with exactly the same input parameters and calculating the coefficient of variation of the per-element root mean squared error, the proportion of elements whose severity label remained constant over all experiments and identifying systematic factors contributing to inconsistencies in the classification of elements such as those located within 2 mm of the band boundaries. Additionally, representative physical defect pattern examples at the element level (i.e. formwork displacement gradient, localized spallation, plumbness drift) should be collected and compared to independent field inspection records to ensure that the deviation maps generated by the system reflect real-world construction defects rather than artifacts resulting from registration residuals or segmentation leakages. The limitations above point to concrete and worthwhile directions for further research.

Broader Empirical Validation

Validate the proposed methodology using three other types of buildings (heavy civil engineering structure, steel framed commercial building, and timber residential), but use a variety of ways of collecting the scans (slam based mobile laser scanning, uav photogrammetry). Use an independent group of experts to rate the severity of deviations found in each type of building and to measure how well the method compares across all of the types of buildings (inter rater reliability).

Adaptive and Self-tuning Parameters

Change the fixed voxel size (v= 0.10 m) and radius of the neighbourhood used for each voxel, so that it is density dependent. For example, the local density percentiles for each element's bounding box crop could be computed and then adjust the voxel size accordingly. This would increase the number of voxels in areas of high density, which typically correspond to small tolerances defined by the BIM.

Lightweight Learned Components

Create a lightweight version of SpU-Net by distilling its knowledge or by quantizing it, such that it can run quickly enough on consumer-grade CPU and GPU hardware to meet the proposed 30-second time target for running a full building scan. Research if creating a hybrid network by taking PTv3's attention-based feature extractor and combining it with SpU-Net's sparse voxel convolutional layer will result in higher cross domain mIoU-IFC than either architecture alone, while also meeting the sub-second time constraint for running per element inference on commodity hardware.

Extended Geometric Coverage

Expand the segmentation algorithm to support doubly curved surfaces (e.g., barrel vaults, domes) and parametric MEP assemblies by developing a curvature aware region-growing component. Also develop an occlusion aware completion module that estimates the geometry of partially scanned elements from BIM priors and neighboring scan data.

Uncertainty quantification

Add formal confidence intervals to each deviation measurement and each severity determination by propagating the registration errors, segmentation coverage fractions, and local densities into a per-element uncertainty estimate. In addition to being able to assign a validity value to each element (valid/needs validation), the system will now be able to generate a confidence value for each element.

Real-time and digital-twin operation

Modify the sequential single worker architecture to accommodate incremental processing, where new scan data is continuously registered and evaluated against the model during construction progression, providing real-time progress monitoring and digital twin updates.

Construction-Specific Benchmark Datasets

Conduct community driven annotation campaigns to generate open-source, large-scale, semantically labelled point cloud benchmark datasets representing various aspects of the construction environment (hydraulic structures, bridges, tunnels, etc.) at several stages of construction development. The proposed benchmark dataset(s) will utilize annotation standards similar to those employed by established indoor benchmarks (S3DIS, ScanNet), however address specific construction-related issues such as formwork, scaffolding, temporary works, and adverse environmental conditions.

5.6 FINAL REMARKS

This dissertation has shown that an appropriate sequence of classical geometry-based operators along with selectively applied deep learning filtering can provide a computationally efficient, indicative workflow for assessing construction and BIM model quality in approximately two minutes. Furthermore, the proposed framework provides a documented processing sequence for the evaluated dataset; it

identifies which elements need to be inspected and presents quality assessment output in a format compatible with BIM.

There are several real limitations to this work: only two case studies were provided; the approximately 50 mm resolution floor limits the reliability of individual verdicts; and registration residual errors exceeded typical construction tolerances. It was deliberately chosen to clearly articulate these limitations rather than to obfuscate them. A quality assessment tool that exaggerates its reliability will likely have greater negative consequences than one that articulates the extent of its limitations. The above-described framework represents a comprehensive, operational, and replicable methodological pipeline from as-constructed laser scanning data to actionable quality reports, where registration, semantic segmentation, deviation analysis, and BIM-quality assessments all represent steps within an integrated end-to-end process that may be performed using commodity computing hardware.

6
REFERENCES

Abreu, N., Pinto, A., Matos, A., & Pires, M. (2023). Procedural Point Cloud Modelling in Scan-to-BIM and Scan-vs-BIM Applications: A Review. *ISPRS International Journal of Geo-Information*, *12*(7). https://doi.org/10.3390/ijgi12070260

Aditya, A., Lohani, B., Aryal, J., & Winter, S. (2024). Benchmarking Deep Learning Architectures for Urban Vegetation Point Cloud Semantic Segmentation From MLS. *IEEE Transactions on Geoscience and Remote Sensing*, *62*, 1–14. https://doi.org/10.1109/TGRS.2024.3381976

Anil, E. B., Tang, P., Akinci, B., & Huber, D. (2011). *Assessment of Quality of As-is Building Information Models Generated from Point Clouds Using Deviation Analysis*.

Anil, E. B., Tang, P., Akinci, B., & Huber, D. (2013). Deviation analysis method for the assessment of the quality of the as-is Building Information Models generated from point cloud data. *Automation in Construction*, *35*, 507–516. https://doi.org/10.1016/j.autcon.2013.06.003

Bai, X., Luo, Y., Jiang, L., Gupta, A., Kaveti, P., Singh, H., & Ostadabbas, S. (2023). *Bridging the Domain Gap between Synthetic and Real-World Data for Autonomous Driving*. http://arxiv.org/abs/2306.02631

Bassier, M., Vincke, S., de Winter, H., & Vergauwen, M. (2020). Drift invariant metric quality control of construction sites using BIM and point cloud data. *ISPRS International Journal of Geo-Information*, *9*(9). https://doi.org/10.3390/ijgi9090545

Besl, P., & Mckay, N. (1992). *A method for registration of 3-D shapes - Pattern Analysis and Machine Intelligence, IEEE Transactions on*.

Bonduel, M., Bassier, M., Vergauwen, M., Pauwels, P., & Klein, R. (2017). Scan-to-bim output validation: Towards a standardized geometric quality assessment of building information models based on point clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *42*(2W8), 45–52. https://doi.org/10.5194/isprs-archives-XLII-2-W8-45-2017

Bosché, F. (2010). Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. *Advanced Engineering Informatics*, *24*(1), 107–118. https://doi.org/10.1016/j.aei.2009.08.006

Bosché, F. (2012). Plane-based registration of construction laser scans with 3D/4D building models. *Advanced Engineering Informatics*, *26*(1), 90–102. https://doi.org/10.1016/j.aei.2011.08.009

Bosché, F., Ahmed, M., Turkan, Y., Haas, C. T., & Haas, R. (2015). The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components. *Automation in Construction*, *49*, 201–213. https://doi.org/10.1016/j.autcon.2014.05.014

Chen, J., & Cho, Y. K. (2018). *Point-to-point Comparison Method for Automated Scan-vs-BIM Deviation Detection*. https://api.semanticscholar.org/CorpusID:219525481

Cheng, L., Chen, S., Liu, X., Xu, H., Wu, Y., Li, M., & Chen, Y. (2018). Registration of laser scanning point clouds: A review. In *Sensors (Switzerland)* (Vol. 18, Number 5). MDPI AG. https://doi.org/10.3390/s18051641

Chetverikov, D., Stepanov, D., & Krsek, P. (2005). Robust Euclidean alignment of 3D point sets: The trimmed iterative closest point algorithm. *Image and Vision Computing*, *23*(3), 299–309. https://doi.org/10.1016/j.imavis.2004.05.007

Choy, C., Gwak, J., & Savarese, S. (2019). 4D spatio-temporal convnets: Minkowski convolutional neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*, 3070–3079. https://doi.org/10.1109/CVPR.2019.00319

Do, S. T., Hoang, H., & Che, D. H. Q. (2023). Error Assessment of Point Cloud and BIM Models to Actual Works. *Lecture Notes in Civil Engineering*, *268*, 253–262. https://doi.org/10.1007/978-981-19-3303-5_20

Dong, J., Tian, M., Yu, J., Li, G., Wang, Y., & Su, Y. (2025). DFPS: An Efficient Downsampling Algorithm Designed for the Global Feature Preservation of Large-Scale Point Cloud Data. *Sensors*, *25*(14). https://doi.org/10.3390/s25144279

Erdélyi, J., Honti, R., Funtik, T., Mayer, P., & Madiev, A. (2022). Verification of Building Structures Using Point Clouds and Building Information Models. *Buildings*, *12*, 2218. https://doi.org/10.3390/buildings12122218

Escudero, P. A. (2023). Scan-to-HBIM: automated transformation of point clouds into 3D BIM models for the digitization and preservation of historic buildings. *Vitruvio*, *8*(2), 52–63. https://doi.org/10.4995/vitruvio-ijats.2023.20413

Fassi, F., Fregonese, L., Ackermann, S., & De Troia, V. (2013). *COMPARISON BETWEEN LASER SCANNING AND AUTOMATED 3D MODELLING*.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). *Efficient Graph-Based Image Segmentation*.

Fischler, M. A., & Bolles, R. C. (1981). *Graphics and Image Processing Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*.

Fu, Y., Li, Z., Wang, W., He, H., Xiong, F., & Deng, Y. (2021). Robust coarse-to-fine registration scheme for mobile laser scanner point clouds using multiscale eigenvalue statistic-based descriptor†. *Sensors*, *21*(7). https://doi.org/10.3390/s21072431

Ghule AG, & Deshmukh PR. (2012). *IMAGE SEGMENTATION AVAILABLE TECHNIQUES, OPEN ISSUES AND REGION GROWING ALGORITHM Journal of Signal and Image Processing*. *3*(1). http://www.bioinfo.in/contents.php?id=48

Gomaa, M., Jabi, W., Veliz Reyes, A., & Soebarto, V. (2021). 3D printing system for earth-based construction: Case study of cob. *Automation in Construction*, *124*. https://doi.org/10.1016/j.autcon.2021.103577

Grilli, E., Menna, F., & Remondino, F. (2017). A review of point clouds segmentation and classification algorithms. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *42*(2W3), 339–344. https://doi.org/10.5194/isprs-archives-XLII-2-W3-339-2017

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., & Bennamoun, M. (2021). Deep Learning for 3D Point Clouds: A Survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Vol. 43, Number 12, pp. 4338–4364). IEEE Computer Society. https://doi.org/10.1109/TPAMI.2020.3005434

Hahsler, M., Piekenbrock, M., & Doran, D. (2019). Dbscan: Fast density-based clustering with R. *Journal of Statistical Software*, *91*. https://doi.org/10.18637/jss.v091.i01

Hajian, H., & Becerik-Gerber, B. (2010). *Scan to BIM: Factors Affecting Operational and Computational Errors and Productivity Loss*.

Hu, Y., Li, Y., & Song, R. (2017). *Robust Interpolation of Correspondences for Large Displacement Optical Flow*.

Hua-Ping, W., Wen-Jie, Z., Yi, C., Yaozhi, L., & D, T. M. (2025). An Efficient Three-Dimensional Point Cloud Segmentation Method for the Dimensional Quality Assessment of Precast Concrete Components Utilizing Multiview Information Fusion. *Journal of Computing in Civil Engineering*, *39*(3), 04025028. https://doi.org/10.1061/JCCEE5.CPENG-6303

Huber, D. (2011). *The ASTM e57 File Format for 3D Imaging Data Exchange*.

IEEE Staff, . (2011). *2011 International Conference on High Performance Computing and Simulation*. IEEE.

Jansen, D., Richter, V., Maier, L., Frisch, J., Treeck, C. van, & Müller, D. (2025). Open-source framework for automated generation of building energy performance simulation models and beyond from BIM Data. *Automation in Construction*, *179*. https://doi.org/10.1016/j.autcon.2025.106427

Javadnia, E., Abkar, A. A., & Schubert, P. (2017). Estimation of high-resolution surface shortwave radiative fluxes using SARA AOD over the Southern Great Plains. *Remote Sensing*, *9*(11). https://doi.org/10.3390/rs9111146

Kavaliauskas, P., Fernandez, J. B., McGuinness, K., & Jurelionis, A. (2022). Automation of Construction Progress Monitoring by Integrating 3D Point Cloud Data with an IFC-Based BIM Model. *Buildings*, *12*(10). https://doi.org/10.3390/buildings12101754

Kim, C., Kim, C., & Son, H. (2013). Automated construction progress measurement using a 4D building information model and 3D data. *Automation in Construction*, *31*, 75–82. https://doi.org/10.1016/j.autcon.2012.11.041

Kim, J., Kim, J., Lian, Y., & Kim, H. (2025). Point Cloud-Based Defect Inspection with Multisensor Fusion and Deep Learning for Advancing Building Construction Quality. *Journal of Computing in Civil Engineering*, *39*(3). https://doi.org/10.1061/jccee5.cpeng-6060

Li, P., Wang, R., Wang, Y., & Tao, W. (2020). Evaluation of the ICP Algorithm in 3D Point Cloud Registration. *IEEE Access*, *8*, 68030–68048. https://doi.org/10.1109/ACCESS.2020.2986470

Li, Y., & Sheng, H. (2023). A single-stage point cloud cleaning network for outlier removal and denoising. *Pattern Recognition*, *138*. https://doi.org/10.1016/j.patcog.2023.109366

Lin, J., & Golparvar-Fard, M. (2020). *Construction Progress Monitoring Using Cyber-Physical Systems* (pp. 63–87). https://doi.org/10.1007/978-3-030-41560-0_5

Liu, K., Gu, J., He, X., & Zhang, L. (2024). Optimization algorithms for dynamic environmental sensing and motion planning of quadruped robots in complex environments on unmanned offshore platforms. *Measurement Science and Technology*, *36*. https://doi.org/10.1088/1361-6501/ad8fc2

Liu, Z., Liu, Z., & Sun, Z. (2024). Massive Point Cloud Processing for Efficient Construction Quality Inspection and Control. *Sensors*, *24*(21). https://doi.org/10.3390/s24216806

Lv, Y., Kang, C., Liu, J., & Zhou, H. (2025). A Point Cloud Registration Method for Steel Tubular Arch Rib Segments of CFST Arch Bridges Based on Local Geometric Constraints. *Buildings*, *15*(12). https://doi.org/10.3390/buildings15122130

Lyu, W., Ke, W., Sheng, H., Ma, X., & Zhang, H. (2024). Dynamic Downsampling Algorithm for 3D Point Cloud Map Based on Voxel Filtering. *Applied Sciences (Switzerland)*, *14*(8). https://doi.org/10.3390/app14083160

Ma, Z., Zhu, H., Xiang, X., Turk, Ž., & Klinc, R. (2024). Automatic compliance checking of BIM models against quality standards based on ontology technology. *Automation in Construction*, *166*, 105656. https://doi.org/10.1016/J.AUTCON.2024.105656

Mahmood, B., Han, S. U., & Lee, D. E. (2020). BIM-based registration and localization of 3D point clouds of indoor scenes using geometric features for augmented reality. *Remote Sensing*, *12*(14). https://doi.org/10.3390/rs12142302

Mei, G., Zhang, J., & Wu, Q. (2023). *Robust 3D Point Cloud Registration Based on Deep Learning and Optimization*.

Mengiste, E., Garcia de Soto, B., & Hartmann, T. (2024). Automated integration of as-is point cloud information with as-planned BIM for interior construction. *International Journal of Construction Management*, *24*(2), 137–150. https://doi.org/10.1080/15623599.2023.2211487

Mora, R., Martín-Jiménez, J. A., Lagüela, S., & González-Aguilera, D. (2021). Automatic point-cloud registration for quality control in building works. *Applied Sciences (Switzerland)*, *11*(4), 1–21. https://doi.org/10.3390/app11041465

Ntiyakunze, J., & Inoue, T. (2023). Segmentation of Structural Elements from 3D Point Cloud Using Spatial Dependencies for Sustainability Studies. *Sensors*, *23*(4). https://doi.org/10.3390/s23041924

Park, S., Ju, S., Hong, J., & Heo, J. (2026). Automated and Fast Point Cloud Labeling Method for Training Data Augmentation Using Scan-versus-BIM Approach. *Journal of Computing in Civil Engineering*, *40*(3). https://doi.org/10.1061/JCCEE5.CPENG-7153

Park, S., Ju, S., Nguyen, M. H., Yoon, S., & Heo, J. (2024). Automated Point Cloud Registration Approach Optimized for a Stop-and-Go Scanning System. *Sensors*, *24*(1). https://doi.org/10.3390/s24010138

Pauwels, P., van den Bersselaar, E., & Verhelst, L. (2024). Validation of technical requirements for a BIM model using semantic web technologies. *Advanced Engineering Informatics*, *60*. https://doi.org/10.1016/j.aei.2024.102426

Pedro, L., & Sanhudo, N. (2021). *ARTIFICIAL INTELLIGENCE FOR AN ENHANCED AS-IS BIM ENERGY ANALYSIS Enabling an efficient energy retrofit through the automation of the scan-to-BIM process*. http://www.fe.up.pt

Pepe, M., Klapa, P., Crisan, A., Dewedar, A. K. H., & Palumbo, D. (2026). A Suitable Scan-to-BIM Process Using OS Software and Low-Cost Sensors: Trend, Solutions and Experimental Validation. *Architecture*, *6*(1), 24. https://doi.org/10.3390/architecture6010024

Perez-Perez, Y., Golparvar-Fard, M., & El-Rayes, K. (2021). Scan2BIM-NET: Deep learning method for segmentation of point clouds for scan-to-BIM. *Journal of Construction Engineering and Management*, *147*(9). https://doi.org/10.1061/(asce)co.1943-7862.0002132

Pomerleau, F., Colas, F., Siegwart, R., & Magnenat, S. (2013). Comparing ICP variants on real-world data sets: Open-source library and experimental protocol. *Autonomous Robots*, *34*(3), 133–148. https://doi.org/10.1007/s10514-013-9327-2

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*. http://arxiv.org/abs/1706.02413

Qi, L., Wu, F., Ge, Z., & Sun, Y. (2022). DeepMatch: Toward Lightweight in Point Cloud Registration. *Frontiers in Neurorobotics*, *16*. https://doi.org/10.3389/fnbot.2022.891158

Qin, H., Zhang, Y., Liu, Z., & Chen, B. (2022). Rigid Registration of Point Clouds Based on Partial Optimal Transport. *Computer Graphics Forum*, *41*(6), 365–378. https://doi.org/10.1111/cgf.14614

Quan, S., Yin, K., Ye, K., & Nan, K. (2022). Robust Feature Matching for 3D Point Clouds with Progressive Consistency Voting. *Sensors*, *22*(20). https://doi.org/10.3390/s22207718

Rashdi, R., Martínez-Sánchez, J., Arias, P., & Qiu, Z. (2022). Scanning Technologies to Building Information Modelling: A Review. *Infrastructures*, *7*(4). https://doi.org/10.3390/infrastructures7040049

Rauch, L., & Braml, T. (2025). Rohbau3D: A Shell Construction Site 3D Point Cloud Dataset. *Scientific Data* , *12*(1). https://doi.org/10.1038/s41597-025-05827-7

Rausch, C., Nahangi, M., Haas, C., & West, J. (2017). Kinematics chain based dimensional variation analysis of construction assemblies using building information models and 3D point clouds. *Automation in Construction*, *75*, 33–44. https://doi.org/10.1016/j.autcon.2016.12.001

Remondino, F., Spera, M. G., Nocerino, E., Menna, F., & Nex, F. (2014). State of the art in high density image matching. *Photogrammetric Record*, *29*(146), 144–166. https://doi.org/10.1111/phor.12063

Rusinkiewicz, S., & Levoy, M. (2001). *Efficient Variants of the ICP Algorithm*.

Rusu, R. B., Blodow, N., & Beetz, M. (2009). *Fast Point Feature Histograms (FPFH) for 3D Registration*.

Sacks, R., Eastman, C., Lee, G., & Teicholz, P. (2018). *BIM Handbook*.

Samsami, R. (2024). A Systematic Review of Automated Construction Inspection and Progress Monitoring (ACIPM): Applications, Challenges, and Future Directions. *CivilEng*, *5*(1), 265–287. https://doi.org/10.3390/civileng5010014

Segal, A. V, Haehnel, D., & Thrun, S. (2009). *Generalized-ICP*.

Sheik, N. A., Veelaert, P., & Deruyter, G. (2022). Registration of Building Scan with IFC-Based BIM Using the Corner Points. *Remote Sensing*, *14*(20). https://doi.org/10.3390/rs14205271

Singh, D. P., & Yadav, M. (2024). Deep learning-based semantic segmentation of three-dimensional point cloud: a comprehensive review. In *International Journal of Remote Sensing* (Vol. 45, Number 2, pp. 532–586). Taylor and Francis Ltd. https://doi.org/10.1080/01431161.2023.2297177

Solihin, W., Eastman, C., & Lee, Y. C. (2015). Toward robust and quantifiable automated IFC quality validation. *Advanced Engineering Informatics*, *29*(3), 739–756. https://doi.org/10.1016/j.aei.2015.07.006

Tam, G. K. L., Cheng, Z. Q., Lai, Y. K., Langbein, F. C., Liu, Y., Marshall, D., Martin, R. R., Sun, X. F., & Rosin, P. L. (2013). Registration of 3d point clouds and meshes: A survey from rigid to Nonrigid. In *IEEE Transactions on Visualization and Computer Graphics* (Vol. 19, Number 7, pp. 1199–1217). https://doi.org/10.1109/TVCG.2012.310

Tan, Y., Wen, Z., Chen, Y., Deng, T., Yue, H., Hao, D., & Wang, Q. (2025). Automated approach for indoor geometric quality inspection data collection based on BIM and quadruped robot equipped with LiDAR. *Journal of Building Design and Environment*. https://doi.org/10.70401/jbde.2025.0019

Tang, P., Huber, D., Akinci, B., Lipman, R., & Lytle, A. (2010). Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. In *Automation in Construction* (Vol. 19, Number 7, pp. 829–843). Elsevier B.V. https://doi.org/10.1016/j.autcon.2010.06.007

Tsige, G. Z., Alsadik, B. S. A., Oude Elberink, S., & Bassier, M. (2025). Automated Scan-vs-BIM Registration Using Columns Segmented by Deep Learning for Construction Progress Monitoring. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLVIII-G-2025*, 1455–1462. https://doi.org/10.5194/isprs-archives-XLVIII-G-2025-1455-2025

Valinejadshoubi, M., Moselhi, O., Iordanova, I., Valdivieso, F., Shakibabarough, A., & Bagchi, A. (2024). The Development of an Automated System for a Quality Evaluation of Engineering BIM Models: A Case Study. *Applied Sciences (Switzerland)*, *14*(8). https://doi.org/10.3390/app14083244

Vicente, S., Kolmogorov, V., & Rother, C. (2008). *Graph cut based image segmentation with connectivity priors Technical report*.

Wang, B., Li, M., Lin, F., Liang, Z., & Cheng, J. (2025). Construction Quality Inspection by Local Point Cloud to BIM Registration and Component Association Based on Vision Foundation Model. In *Title of the Journal* (Vol. 2025).

Wang, Y., Feng, H. Y., Delorme, F. É., & Engin, S. (2013). An adaptive normal estimation method for scanned point clouds with sharp features. *CAD Computer Aided Design*, *45*(11), 1333–1348. https://doi.org/10.1016/j.cad.2013.06.003

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. (2019). Dynamic graph Cnn for learning on point clouds. *ACM Transactions on Graphics*, *38*(5). https://doi.org/10.1145/3326362

Wang, Z., Sacks, R., & Yeung, T. (2022). Exploring graph neural networks for semantic enrichment: Room type classification. *Automation in Construction*, *134*. https://doi.org/10.1016/j.autcon.2021.104039

Weinmann, M., Jutzi, B., Hinz, S., & Mallet, C. (2015). Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, *105*, 286–304. https://doi.org/10.1016/j.isprsjprs.2015.01.016

Wu, J., & Zhang, J. (2019). New Automated BIM Object Classification Method to Support BIM Interoperability. *Journal of Computing in Civil Engineering*, *33*(5), 4019033. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000858

Wu, X., Jiang, L., Wang, P.-S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., & Zhao, H. (2024). Point Transformer V3: Simpler Faster Stronger. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4840–4851. https://openaccess.thecvf.com/content/CVPR2024/html/Wu_Point_Transformer_V3_Simpler_Faster_Stronger_CVPR_2024_paper.html

Xiao, A., Zhang, X., Shao, L., & Lu, S. (2024). *A Survey of Label-Efficient Deep Learning for 3D Point Clouds*. http://arxiv.org/abs/2305.19812

Yang, K., Deng, X., Bai, Z., Wan, Y., Xie, L., & Zeng, N. (2024). A Registration Method Based on Ordered Point Clouds for Key Components of Trains. *Sensors*, *24*(24). https://doi.org/10.3390/s24248146

Yang, L., Lin, Y.-C., Cai, H., & Habib, A. (2024). From Scans to Parametric BIM: An Enhanced Framework Using Synthetic Data Augmentation and Parametric Modeling for Highway Bridges. *Journal of Computing in Civil Engineering*, *38*(3), 4024008. https://doi.org/10.1061/JCCEE5.CPENG-5640

Yin, X., Liu, H., Chen, Y., & Al-Hussein, M. (2019). Building information modelling for off-site construction: Review and future directions. In *Automation in Construction* (Vol. 101, pp. 72–91). Elsevier B.V. https://doi.org/10.1016/j.autcon.2019.01.010

Yue, H., Wang, Q., Zhao, H., Zeng, N., & Tan, Y. (2024). Deep learning applications for point clouds in the construction industry. In *Automation in Construction* (Vol. 168). Elsevier B.V. https://doi.org/10.1016/j.autcon.2024.105769

Zhang, J., & El-Gohary, N. M. (2017). Semantic-Based Logic Representation and Reasoning for Automated Regulatory Compliance Checking. *Journal of Computing in Civil Engineering*, *31*(1), 4016037. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000583

Zhang, Y., Chen, L., Lu, Q., Zou, Y., Xiahou, X., Sølvsten, S., & Hancock, C. (2026). Semi-automated localised updating for as-built BIM of piping systems using point cloud data. *Automation in Construction*, *181*. https://doi.org/10.1016/j.autcon.2025.106609

Zhao, H., Jiang, L., Jia, J., Torr, P., & Koltun, V. (2021). *Point Transformer*.

Zhou, Q.-Y., Park, J., & Koltun, V. (2016). *Fast Global Registration*.

Zou, Z., Lang, H., Lou, Y., & Lu, J. (2023). Plane-based global registration for pavement 3D reconstruction using hybrid solid-state LiDAR point cloud. *Automation in Construction*, *152*, 104907. https://doi.org/10.1016/j.autcon.2023.104907

Zoumpekas, T., Molina, G., Salamó, M., & Puig, A. (2021). Benchmarking Deep Learning Models on Point Cloud Segmentation. *Frontiers in Artificial Intelligence and Applications*, *339*, 335–344. https://doi.org/10.3233/FAIA210152