Machine learning techniques for the creation of brim/fem models applied to bridges

D.R.S.A.T.E.

Doctorate in Risk And Environmental, Territorial And Building Development

Coordinatore: Prof. Francesco Fiorito

XXXVIII CYCLE Curriculum: ICAR/06 – Topography and Cartography

DICATECh
Department of Civil, Environmental, Building Engineering and Chemistry

POLITECNICO DI BARI

11

2026

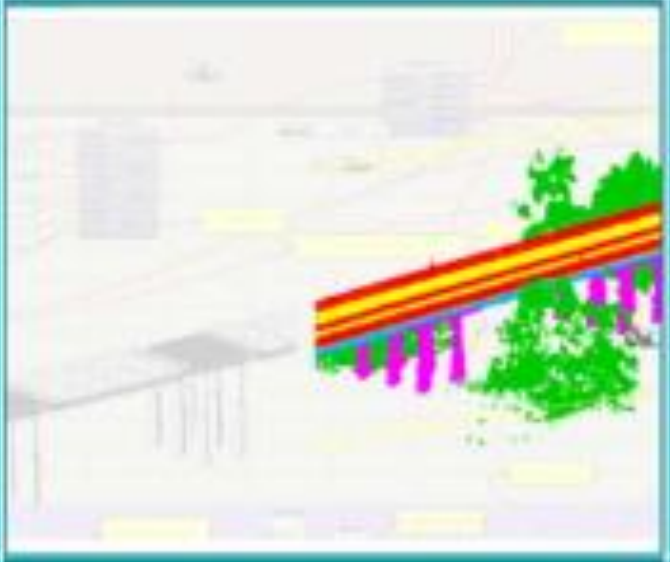
Restuccia Garofalo Alfredo

MACHINE LEARNING TECHNIQUES FOR THE CREATION OF BrIM/FEM MODELS APPLIED TO BRIDGES

Prof. Domenica COSTANTINO
Department of Civil, Environmental, Building Engineering and Chemistry Polytechnic University of Bari

Prof. Vincenzo Saverio ALFIO
Department of Civil, Environmental, Building Engineering and Chemistry Polytechnic University of Bari

Prof. Sorin HERBAN
Department of Overland Communication Ways; Foundation and Cadastral Survey Universitatea Politehnica Timişoara- Romania



D.R.S.A.T.E.

Dctorate in Risk And Environmental, Territorial And Building Development

POLITECNICO DI BARI

11

2026

Coordinator: Prof. Francesco Fiorito

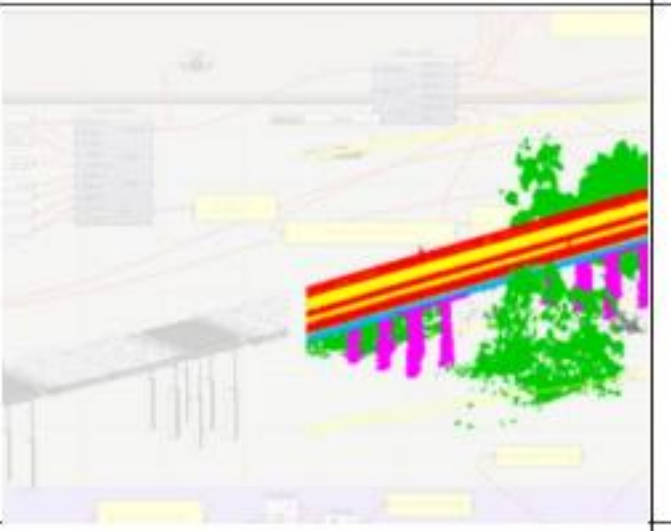
XXXVIII CYCLE Curriculum: ICAR/06 – Topography and Cartography

DICATECh
Department of Civil, Environmental, Building Engineering and Chemistry

Alfredo Restuccia Garofalo

MACHINE LEARNING TECHNIQUES FOR THE CREATION OF BrIM/FEM MODELS APPLIED TO BRIDGES

Prof. Domenica COSTANTINO
Department of Civil, Environmental, Building Engineering and Chemistry
Polytechnic University of Bari
Prof. Vincenzo Saverio ALFIO
Department of Civil, Environmental, Building Engineering and Chemistry
Polytechnic University of Bari
Prof. Sorin HERBAN
Department of Overland Communication Ways, Foundation and Cadastral Survey Universitatea Politehnica Timişoara- Romania



D.R.S.A.T.E.

POLITECNICO DI BARI

11

2026

Dottorato in Rischio e Sviluppo Ambientale, Territoriale ed Edilizio

Coordinatore: Prof. Francesco Fiorito

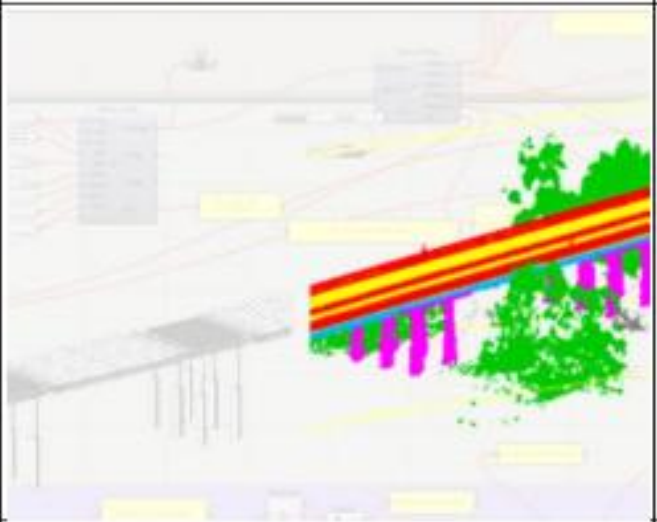
XXXVIII CYCLE Curriculum: ICAR/06 – Topografia e Cartografia

DICATECh
Dipartimento di Ingegneria Civile, Ambientale, del Territorio, Edile e di Chimica

Restuccia Garofalo Alfredo

TECNICHE DI MACHINE LEARNING PER LA CREAZIONE DI MODELLI BrIM/FEM APPLICATE AI PONTI

Prof. Domenica COSTANTINO
Dipartimento di Ingegneria Civile, Ambientale, del Territorio, Edile e di Chimica Politecnico Università di Bari
Prof. Vincenzo Saverio ALFIO
Dipartimento di Ingegneria Civile, Ambientale, del Territorio, Edile e di Chimica Politecnico Università di Bari
Prof. Sorin HERBAN
Dipartimento delle vie di comunicazione terrestri, delle fondazioni e dei rilievi catastali UniversitateaPolitehnicaTimişoara-Romania



EXTENDED ABSTRACT *(eng)*

Bridge and infrastructure management increasingly relies on accurate, up-to-date digital representations to support inspection, maintenance planning and structural assessment. Modern surveys based on terrestrial laser scanning (TLS), Unmanned Aerial Vehicle (UAV) photogrammetry and, when available, mobile mapping systems (MMS) provide dense 3D point clouds that describe the "as-is" condition with metric rigor, but they are inherently unstructured: geometry is available, while semantic information and interoperable model entities are not. This thesis addresses the transformation of raw point clouds into structured, semantically meaningful information that can be consumed by Bridge Information Modelling (BrIM/InfraBIM) and Finite Element Method (FEM) workflows.

A complete, computable pipeline is proposed and formalised, covering: (i) acquisition and pre-processing (registration, georeferencing and quality control); (ii) scalable data structures for neighbourhood queries; (iii) multi-scale extraction of geometric descriptors; (iv) supervised semantic classification of point clouds; and (v) downstream modelling and export towards IFC-based BrIM and FEM environments. The classification stage is built around a Random Forest (RF) model trained on local geometric features computed at multiple scales. The feature set includes spectral descriptors derived from eigenvalues (linearity/planarity/sphericity, omnivariance), curvature-related measures, verticality, and height-based attributes (e.g., z-height / height-above-ground). To reduce manual labelling effort while preserving control on ground truth quality, a semi-automatic iterative workflow is adopted: a manual seed is produced in CloudCompare, an initial RF is trained, predictions are propagated with confidence thresholds, and residual ambiguities are corrected in successive iterations. Class imbalance is handled through balanced class weights and controlled sampling.

The workflow is validated on two bridge case studies: a training project (P3) and an independent validation project (P4). On the training bridge, the RF model reaches overall accuracy OA = 0.892 with macro-F1 = 0.859 and

Cohen's kappa = 0.841, showing stable discrimination for the main structural classes (e.g., piles, deck slab, longitudinal beams). On the independent bridge, the model generalises without re-tuning, achieving Accuracy = 0.86, macro-F1 = 0.84, kappa = 0.83 and mean IoU = 0.78. Error analysis through confusion matrices highlights typical failure modes related to thin elements, local occlusions and scale mismatch in neighbourhood definition. Feature-importance rankings confirm the dominant role of height-based attributes and of multi-scale spectral descriptors in separating context (ground/vegetation) from structural components and in resolving planar/linear patterns.

After semantic segmentation, the thesis discusses how classified point clouds can be converted into modelling primitives suitable for BIM and FEM. Two complementary modelling strategies are compared: Track A, a section-based approach in Rhinoceros/Non Uniform Rational B-Spline (NURBS) oriented to controllable, "clean" solids/surfaces and robust IFC export; and Track B, a parametric approach in Grasshopper oriented to rapid updates, variant generation and mesh-ready geometries for FEM. Practical quality-assurance checks (overlay and cloud-to-model comparisons, class-to-IFC mapping, and go/no-go criteria) are proposed to ensure metric coherence and traceability throughout the conversion chain.

Overall, the thesis contributes an end-to-end, reproducible and interoperable pipeline from survey data to BrIM/FEM-ready models, together with operational guidelines to balance accuracy, effort and export reliability when moving from point-based geometry to information-rich digital representations of bridges.

Point cloud; TLS; UAV photogrammetry; supervised classification; multi-scale geometric features; Random Forest; BrIM/InfraBIM; IFC 4.3; FEM; Rhinoceros/Grasshopper.

Keywords

Restuccia Garofalo Alfredo

Point cloud; TLS; UAV photogrammetry; supervised classification; multi-scale geometric features; Random Forest; BrIM/InfraBIM; IFC 4.3; FEM; Rhinoceros/Grasshopper.

EXTENDED ABSTRACT (Italian)

La gestione e la manutenzione delle infrastrutture richiedono rappresentazioni digitali affidabili dello stato di fatto (as-is), aggiornabili nel tempo e utilizzabili sia per la modellazione informativa (BrIM/InfraBIM) sia per le verifiche strutturali agli elementi finiti (FEM). I rilievi moderni basati su laser scanner terrestre (TLS), fotogrammetria da UAV e, quando disponibile, mobile mapping (MMS) producono nuvole di punti ad alta densità e metricamente rigorose; tuttavia, tali dati sono privi di semantica intrinseca e, se non trasformati, restano difficili da integrare in flussi interoperabili.

La tesi propone e formalizza una pipeline completa e computabile che conduce dalla nuvola di punti grezza a modelli BrIM/FEM utilizzabili, articolata in: (i) acquisizione e pre-processing (registrazione, georeferenziazione e controllo qualità); (ii) strutture dati scalabili per ricerche spaziali; (iii) estrazione multi-scala di descrittori geometrici; (iv) classificazione supervisionata della nuvola; (v) modellazione ed export verso ambienti IFC e FEM. La classificazione è eseguita mediante Random Forest (RF) addestrata su feature geometriche locali calcolate su intorni definiti a scale differenti. Il descrittore include quantità spettrali derivate dagli autovalori (linearità/planarità/sfericità, omnivarianza), misure di curvatura e rugosità, verticalità e attributi di quota (z-height / height-above-ground). Per ridurre la fase di etichettatura manuale mantenendo tracciabilità e controllo, è adottato un ciclo iterativo semi-automatico: seed manuale in CloudCompare, training RF, propagazione con soglie di confidenza, revisione delle aree ambigue e iterazioni successive. Lo sbilanciamento tra classi è gestito tramite pesi bilanciati e campionamento controllato.

La validazione è condotta su due ponti: un progetto di training (P3) e un progetto di validazione indipendente (P4). Sul ponte di training il modello raggiunge OA = 0,892, macro-F1 = 0,859 e kappa di Cohen = 0,841, con buone prestazioni sulle principali classi strutturali (pile, soletta, travi longitudinali).

Sul ponte indipendente, senza ri-tarature, i risultati sono Accuracy = 0,86, macro-F1 = 0,84, kappa = 0,83 e mIoU = 0,78. L'analisi degli errori tramite matrici di confusione evidenzia criticità tipiche legate a elementi sottili, occlusioni locali e scelte della scala degli intorni. Le graduatorie di importanza confermano il ruolo dominante degli attributi di quota e dei descrittori spettrali multi-scala nel separare contesto e componenti strutturali.

Ottenuta la segmentazione semantica, la tesi analizza come tradurre la nuvola classificata in geometrie e oggetti informativi. Sono confrontate due strategie complementari: Track A, approccio section-based in Rhinoceros/NURBS orientato a solidi e superfici "puliti" e a un export IFC più robusto; Track B, approccio parametrico in Grasshopper orientato a revisioni rapide, generazione di varianti e geometrie pronte per il meshing FEM. Sono inoltre proposte verifiche di qualità (overlay e confronti cloud-to-model, mappatura classe→IFC, QA e criteri go/no-go) per garantire coerenza metrica e tracciabilità lungo la pipeline.

Nel complesso, il lavoro fornisce un workflow end-to-end, riproducibile e interoperabile dal rilievo al modello BrIM/FEM, insieme a linee guida operative per bilanciare accuratezza, effort e affidabilità dell'export nel passaggio da geometria puntuale a rappresentazioni digitali informative dei ponti.

Parole chiave

Nuvola di punti; TLS; fotogrammetria UAV; classificazione supervisionata; feature geometriche multi-scala; Random Forest; BrIM/InfraBIM; IFC 4.3; FEM; Rhinoceros/Grasshopper.

INDEX

CURRICULUM 305

INTRODUCTION

This research is developed within a theoretical and methodological framework that addresses the transformation of point clouds acquired through TLS, UAV and, when available, MMS techniques into structured semantic information that can be used for supervised classification, information modelling and finite-element numerical simulation. The point cloud is treated as a high-density metric representation of the as-is condition: geometric data that lack intrinsic semantics and gain meaning through formal descriptor-extraction procedures, supervised learning models, and interoperability rules towards HBIM/InfraBIM and FEM environments.

Infrastructure assets exhibit material and geometric heterogeneity, construction discontinuities and localised defects that make simple parametric models ineffective. Modern acquisitions generate datasets on the order of 10^7–10^8 points: information-rich data that are nevertheless computationally demanding.

To turn this mass of observations into usable knowledge it is necessary to: (i) introduce an explicit mathematical representation of the data and of its neighbourhood; (ii) ensure metric consistency across sensor/world/local reference frames; (iii) adopt scalable data structures and software tools; (iv) maintain continuous quality control through visualisation; and (v) ensure traceability and reproducibility along the whole pipeline.

It is therefore necessary to establish a set of formal procedures that ensure robustness, reproducibility and comparability between the input data and the output data.

In this context, the research focuses on identifying a compact and stable set of local geometric descriptors derived from PCA (Principal Component Analysis) on coherent neighbourhoods, and on assessing their ability to generalise as density, noise and the infrastructure context vary. In parallel, the role of data structures and indexing strategies is analysed in terms of the computational sustainability of the process and of controlling uncertainty

propagation from classification stages towards information modelling and FEM simulation. Technical visualisation is finally assumed as a tool for continuous verification, capable of linking numerical metrics to the actual geometry of the asset and supporting informed methodological choices along the entire pipeline.

The methodology is driven by the following main questions: definition of local geometric features; inter-campaign and inter-structure consistency; influence of data structures on computational convergence; propagation of error in HBIM/FEM stages.

First, the work investigates whether it is possible to identify a minimal and stable set of features derived from PCA analyses on coherent neighbourhoods, robust to noise and density variations.To this end, adaptive windows (k-NN or r-ball) and distance-dependent weightings are considered to stabilise the eigenvalues of the covariance matrix and improve generalisation. The work then analyses how to ensure comparability across surveys carried out on different bridges and/or in distinct campaigns by assuming an ordered chain of transformations sensor→world→local, verified through audits and RMSE/loop-closure analyses, reducing spurious variance in the features. Moreover, the impact of k-d trees, voxelisation and tiling on processing times and accuracy is evaluated, showing how pre-indexing and pre-voxelisation can drastically reduce the cost of spatial queries and stabilise local statistics.

Finally, the work quantifies how classification uncertainty affects the generation of information objects and the computational mesh, maintaining the correspondence points→classes→objects→mesh and measuring how errors propagate on the numerical response.

This research aims to achieve the following objectives:

- an integrated, scalable and reproducible pipeline from point cloud to FEM model;
- a compact formalisation of local features and normal-estimation practices, with attention to invariance and stability;

- a robust registration strategy compatible with multi-sensor scenarios, and operational criteria for managing high-efficiency data structures;
- a Python toolkit validated on real case studies;
- visualisation procedures intended as quality control at every stage;
- interoperability rules between classification, HBIM and FEM, and tools to assess the impact of methodological choices on numerical models.

Specifically, the operational process includes:

Point-cloud pre-processing through voxelisation, filtering and outlier removal.

Neighbourhood construction (k-NN or r-ball) and normal estimation via PCA.

Computation of geometric features (linearity, planarity, sphericity, anisotropy).

Supervised classification of points using Random Forest.

Semantic mapping towards information entities (axes, surfaces, sections) and meshes intended for FEM simulations.

Visual and numerical checks to ensure coherence and traceability of the transfer.

Within this work, the analysed case studies concern existing bridges surveyed through TLS and UAV, with possible MMS integration. The point clouds exhibit heterogeneous density, occlusions and variability typical of infrastructure environments. Training labels are built on representative ROIs, with the aim of obtaining models that generalise across different assets. Each computational stage is driven by configuration files and produces logs containing parameters, transformations and library versions. Reference-frame transformations are stored together with residuals; the resulting datasets adopt uniform conventions for coordinates, colours, normals, features, predicted classes and frame identifiers. The software environment is defined through YAML/TXT files with pinned versions; stochastic algorithms use controlled random seeds.

The thesis is organised as follows:

Chapter 1: Introductory framework and theoretical-methodological foundations, defines the conceptual, mathematical and computational framework that enables the entire development of the thesis. This section introduces the theoretical and methodological fundamentals required to transform a raw 3D point cloud into structured, semantically interpretable information that can be transferred to information models (HBIM/BrIM) and finite-element numerical models.

First, the chapter clarifies the strategic role of the point cloud in the context of infrastructure assets, highlighting its semantic neutrality and its value as a direct representation of the as-is condition. It shows how such neutrality requires formal structuring rules, robust geometric descriptors and scale choices consistent with the anatomy of infrastructure systems, and bridges in particular.

Next, the chapter introduces the mathematical representation of 3D data, formalising the concept of a point enriched with geometric, radiometric and semantic attributes, as well as the notions of distance, neighbourhood and neighbour graph. This formalism provides the basis for extracting local geometric features, for supervised classification, and for the subsequent construction of surfaces and meshes with explicit topology.

Specific attention is devoted to managing reference frames and geometric transformations, clarifying the transition between the sensor frame, the world coordinate system and local component systems. The rigorous definition of rigid transformations and measurement units is presented as a necessary condition to ensure metric coherence, descriptor transferability and comparability across different surveys and successive workflow stages.

The chapter then addresses the data structures and spatial indices required to operate on large datasets, introducing tabular matrices, k-d trees, voxel grids, neighbourhood graphs and topological structures for mesh representation. These tools are discussed not as mere implementation details, but as methodological choices that directly influence the stability of

geometric estimates, the scalability of computation and the quality of the outputs.
The software toolbox adopted in the thesis is also presented, based on the Python ecosystem, clarifying the role of the main libraries for data import, pre-processing, registration, feature extraction, supervised classification and the management of out-of-core workflows. The use of reproducible environments, traceable configurations and systematic logging is framed as an essential requirement for the scientific validation of the results.
Finally, the chapter assigns a central operational role to interactive visualisation, presenting it as a tool for diagnosis, quality control and continuous validation of the entire computational flow. Attribute- and class-based visualisation is described as a key element for linking geometric perception, quantitative metrics and methodological decisions, before moving to information and structural modelling.
Overall, Chapter 1 provides the foundational framework on which the subsequent chapters build: the definition of features and automatic classification (Chapters 2 and 4), the transfer of semantics towards HBIM/BrIM and FEM models (Chapter 5), and the quantitative evaluation of performance and process quality (Chapter 6).
Chapter 2: Methodological foundations and point-cloud classification pipeline introduces and formalises the overall methodological framework of the thesis, describing the entire chain that leads from raw 3D surveying to a semantically classified point cloud, ready for parametric and structural modelling in the subsequent chapters. The chapter aims to define, in a rigorous and reproducible manner, the theoretical, geometric and computational bases underlying the entire automatic-classification workflow applied to bridges.
First, the chapter clarifies the role of the point cloud as an intermediate format between measured reality and information and numerical models, introducing a formal definition of the 3D point enriched with geometric,

radiometric and semantic attributes. The characteristics of structured and unstructured point clouds are discussed, together with the implications of different acquisition techniques (TLS, UAV/SfM photogrammetry) and the importance of metric consistency, reference frame and scale to ensure comparability and transferability of results.

Next, the chapter addresses the data structures and spatial indices needed to manage large point clouds, introducing matrices, k-d trees, octrees and block-processing strategies. These choices are motivated in relation to computational scalability, the stability of local estimates and their direct impact on the performance of geometric descriptors and the classifier.

A central part of the chapter is dedicated to point-cloud pre-processing, including outlier filtering, density control through voxelisation, robust normal estimation and geometric normalisation. These operations transform raw data into reliable input for feature extraction and make it possible to preserve relevant structural discontinuities (edges, borders, transitions between elements).

The chapter then discusses the computation of local geometric features derived from the covariance matrix of k-NN neighbourhoods, and the definition of scalar descriptors that are rotation-invariant, orientation-related and density-related, capable of effectively discriminating structural and non-structural elements of the bridge. Scale and neighbourhood choices are motivated in relation to the asset anatomy and to the need for generalisation across different bridges.

On the basis of these features, supervised classification via Random Forest is introduced, illustrating its principles, the rationale for the selection, class-balancing strategies, and training and validation on independent datasets. The chapter clarifies that classification is not an end in itself, but the key step that enables HBIM/BrIM modelling and the generation of FEM models addressed in the following chapters.

Finally, the chapter describes the ways results are visualised and analysed, combining standard quantitative metrics with spatial checks and visual

inspections in a 3D environment. This analysis makes it possible to assess not only the model's statistical accuracy, but also the geometric consistency of predicted classes, laying the groundwork for overall validation and for the transition to parametric and structural modelling.

Chapter 3: From point cloud to geometric, information and FEM models describes the operational transition from the semantically classified point cloud to the construction of geometric models, information models (BrIM/HBrIM) and finite-element numerical models, defining software environments, workflows and quality criteria that make this transition reproducible and controllable.

The chapter first introduces the working environments adopted and their role in the overall workflow, clarifying the existence of two main tracks: a section-based track oriented to information modelling (cloud → sections/axes → HBrIM/IFC 4.3 → FEM) and a parametric track based on Rhinoceros and Grasshopper with a direct link to structural computation in MIDAS Civil NX. When appropriate, a direct cloud-mesh-FEM procedure is also used for rapidly generating analysis models.

The chapter then examines the use of Rhinoceros as a geometric pre-modelling environment, describing how large point clouds are handled through dedicated plugins, how semantic classes are organised on separate layers and how fundamental geometric references such as axes, sections and control planes are extracted. These references constitute the common basis for both modelling tracks and ensure metric coherence, tolerance control and traceability of transformations.

A substantial part of the chapter is devoted to modelling through NURBS curves, surfaces and solids, and to the generation of triangular and quadrangular meshes from the classified point cloud. Criteria for choosing between TriMesh and QuadMesh depending on structural class, local curvature and downstream use (HBrIM or FEM) are discussed, introducing class-aware strategies that make it possible to control discretisation density,

reduce computational load and preserve meaningful geometric details. Mesh quality is evaluated through geometric indicators and point-to-model deviation metrics, in line with the thresholds defined in the subsequent chapters.

The chapter then addresses parametric modelling in Grasshopper, showing how axes and sections extracted from the point cloud can be transformed into controllable parametric geometries capable of representing cross-section variations, spans and different structural configurations. The role of hierarchical data structures (data trees) in managing element families and in synchronising parametric rules across piers, beams and deck is highlighted.

Specific attention is given to integration with structural analysis software, in particular through the MIDAS GH plugin and the MIDAS Civil NX environment, which allow direct generation of parametric FEM models from the geometric definitions. In parallel, the chapter describes the use of MIDAS CIM as an HBrIM environment for building information models based on libraries, assembly units and semantic mapping compliant with the IFC 4.3 standard, highlighting the consistency between point-cloud classes, IFC entities and FEM typologies.

Finally, the chapter discusses interoperability, automation and quality control, showing how semantic classification of the point cloud helps reduce model complexity, automate the extraction of axes and sections, standardise export workflows and maintain traceability of geometric and information choices. The case studies of the two analysed bridges (a railway bridge used for training and a road bridge used for validation) are used to demonstrate the effectiveness of the proposed procedures and their applicability to different structural typologies.

Overall, Chapter 3 represents the link between automatic point-cloud classification and the subsequent stages of information modelling and structural analysis, providing the operational basis for the quantitative evaluations and comparisons developed in the following chapters.

Chapter 4: Automatic classification of the 3D point cloud is the methodological core of the thesis and is dedicated to automatic classification of the 3D point cloud using supervised machine learning techniques. This chapter formalises and applies the process that transforms the local geometric representation, defined in the previous chapters, into semantic segmentation coherent with the structural breakdown of the analysed bridges.

The chapter systematically describes the extraction of local geometric features from point neighbourhoods, illustrating how descriptors derived from the covariance matrix, orientation indicators, density measures and roughness measures make it possible to characterise different types of structural and non-structural elements. Choices related to analysis scale, number of neighbours and feature stability are motivated in relation to infrastructure morphology and to the need for generalisation across different bridges.

On the basis of these features, the supervised classification problem is set up, defining the semantic classes of interest and the strategies for building the training dataset. The chapter addresses class balancing, sample selection and the control of domain shift between the training bridge and the validation bridge, with particular attention to model transferability in different geometric and environmental contexts.

The reference model adopted is the Random Forest Classifier, selected for its robustness to noise, its ability to handle heterogeneous features and the interpretability of results. The principles of the algorithm are described, together with hyperparameter choices, training and validation procedures, and model serialisation and reuse on independent point clouds.

The chapter devotes ample space to performance assessment, integrating standard statistical metrics (accuracy, precision, recall, F1-score) with more advanced analyses of statistical stability of distributions between training and validation and spatial-coherence indicators. Error analysis is carried out

both globally and locally, highlighting the main confusions between adjacent classes and the criticalities in zones of geometric transition.

Finally, the chapter presents techniques for visualising and analysing the results, showing how the integration of quantitative metrics with visual inspection of the classified point cloud makes it possible to interpret model behaviour, identify residual errors and define corrective actions. The resulting classification is therefore validated not only numerically, but also in terms of geometric-structural coherence, laying the groundwork for reliable transfer of semantics to the information modelling and structural analysis developed in Chapter 5.

Chapter 5: Information and numerical modelling: requirements, operational tracks and quality control defines the functional, geometric and metrological requirements governing the transformation of the 3D point cloud—already classified in Chapter 4—into information models (BrIM/HBrIM in IFC 4.3), parametric models and finite-element numerical models (FEM). The chapter clarifies how the quality of semantic classification and geometric adherence directly affect the reliability of downstream models, and establishes objective acceptance criteria (go/no-go) for entering the modelling and analysis domain.

At the outset, the chapter frames the reference application contexts—design, structural analysis and asset management—and introduces a system of class- and context-dependent thresholds based on reproducible metrics such as F1-score, point-to-model distance (C2C) and inlier ratio at threshold (IR_). These requirements explicitly define which point-cloud elements can be used for information modelling and which should be excluded or treated as mere context.

The chapter then describes the metric and organisational preparation stage in Rhinoceros, where measurement units, tolerances, local reference frames, global shift and naming and layer conventions are set. This stage is presented as indispensable to ensure numerical stability, transformation traceability and consistency of exports towards BIM/HBIM and FEM environments.

A central role is assigned to guiding extracts—axes, sections, edges and control elevations—which act as the hinge between point-cloud semantics and the controllable geometry of models. Extracting these references by structural class makes it possible to constrain modelling to the real data, systematically measure geometric deviations, and consistently feed the different operational tracks.

Starting from the same guiding extracts, the chapter introduces two complementary modelling tracks. The first, section-based (Track A), is oriented to building information models compliant with the IFC 4.3 standard and their subsequent export to structural computation, prioritising traceability, local fidelity to the data and semantic coherence. The second, parametric (Track B), uses visual environments and direct-integration tools with FEM solvers to enable rapid generation of variants while keeping explicit control over the main geometric and structural quantities.

The chapter systematically compares the two tracks, analysing performance, human–computer costs, model reusability and geometric-semantic quality. It shows that the two approaches are not mutually exclusive but can be integrated into a hybrid strategy that, depending on needs, maximises either informational fidelity or speed of design iteration.

In the concluding part, the chapter formalises operational Quality Assurance guidelines, translating classification and geometric-adherence metrics into explicit decision criteria for model acceptance. Quality indicators are thus linked directly to information objects and FEM models, enabling a structured reading of the asset state and providing the basis for the comparative analyses and validation presented in Chapter 6.

Overall, Chapter 5 represents the transition point from automatic point-cloud classification to information and numerical modelling, providing a measurable and reproducible operational framework that coherently links survey, semantics, geometry and structural analysis.

Chapter 6: Results analysis and comparison between training and validation is devoted to a critical and quantitative analysis of the results obtained from the supervised classification pipeline developed in the previous chapters. This section evaluates the effectiveness, robustness and generalisation capability of the machine-learning model, with particular attention to the behaviour of multiscale geometric features and the Random Forest classifier in different contexts.

The chapter first analyses the statistical properties of the extracted geometric features, studying their distribution, stability and evolution across the experimental stages. It discusses the role of spectral features derived from the local covariance matrix, orientation indicators, and roughness and density measures, highlighting how the choice of analysis scale (k-NN and r-ball) directly influences the model's discriminative capability for the different structural and context classes.

Subsequently, the chapter evaluates model performance on the training bridge, providing a detailed reading of global and per-class classification metrics. This analysis makes it possible to verify the internal coherence of the model, identify the main residual confusions between geometrically similar classes, and establish a reliable quantitative baseline for the subsequent comparison.

The central part of the chapter is devoted to validation on an independent bridge surveyed under different conditions than the training dataset. In this context, the model's generalisation capability is analysed, highlighting the effects of domain shift due to variations in density, noise, surface conditions and occlusions. Classification metrics are integrated with geometric-coherence indicators based on point-to-model distances (C2C) and coverage within metric thresholds, directly linking semantic classification quality to the geometric fidelity of the result.

The chapter then proposes a systematic comparison between the performance achieved in training and that observed in validation, interpreting differences in terms of class geometric characteristics, feature

stability and intrinsic limits of the local descriptor. This analysis identifies the most robust classes and those most sensitive to context variations, providing operational indications for improving training, adopting active-learning strategies and, where appropriate, multi-scale refinement.
Finally, Chapter 6 links the numerical and statistical results to parametric modelling and interoperability towards HBIM/BrIM and FEM environments. Through 3D visualisations, overlays between the point cloud and parametric models, and metrological checks on sections and axes, consistency between classification, reconstructed geometry and information models is demonstrated. Quality metrics are thus reinterpreted as operational tools to guide modelling, verification and integration of models and to generate infrastructure-management systems.
Overall, Chapter 6 provides experimental validation of the entire proposed framework, showing how the combination of multiscale geometric features, supervised classification and reproducible quality controls leads to reliable and transferable results, constituting the quantitative basis for the thesis conclusions.

CHAPTER 1 -- INTRODUCTORY FRAMEWORK AND THEORETICAL METHODOLOGICAL BACKGROUND

This chapter defines the conceptual and methodological framework that makes the entire processing chain adopted in the thesis computable, from the raw point cloud to the transfer of geometric and semantic information into BIM/HBIM models and FEM meshes. In particular, it describes:

- the strategic role of the point cloud and its meaning in infrastructure surveying;
- the explicit mathematical representation of 3D data and the concept of neighbourhood;

- coordinate types and geometric transformations between the sensor, world and local reference systems;
- the main data structures for managing large point clouds (matrices, spatial indices, voxel grids);
- the adopted software tools (Python, CloudCompare, Rhinoceros/Grasshopper, MIDAS);
- visualisation as a continuous tool for verification and quality control of the workflow.

The conceptual framework outlined here underpins the entire experimental work and enables a systematic assessment of how methodological choices affect classification, HBIM/FEM interoperability and structural simulation.

1.1 POINT CLOUDS: A STRATEGIC ROLE IN THE CONTEXT OF INFRASTRUCTURE ASSETS

In real infrastructure contexts, the point cloud represents the hinge between inspection, modelling and decision-making. It describes the as-is condition of the asset without imposing a priori parametric schemes, which is fundamental when different materials (reinforced and prestressed concrete, steel, masonry), construction discontinuities and degradation phenomena (cracks, detachments, deformations) coexist. Precisely because it is neutral, the point cloud requires structuring rules (pre-processing, neighbourhood definition, feature extraction) and interoperability procedures capable of transferring the geometric information into semantic representations (classification) and then into information models (HBIM/BrIM) and numerical models (FEM).

From a formal standpoint, a 3D point cloud can be represented as:

$$P = \{ p_i = (x_i, y_i, z_i, a_i) \mid i = 1, \dots, N \} \quad (1)$$

where

$$a_i = (r_i, g_i, b_i, I_i, n_i, c_i) \quad (2)$$

In this formulation r_i, g_i, b_i , the attributes represent colour components (RGB), I_i the return-signal intensity (for LiDAR sensors), $\mathbf{n}_i$ the surface normal associated with the point, and c_i a class label or semantic information derived from classification or segmentation processes.

The strategic value of this representation lies in the ability to integrate metric information with semantic attributes in a traceable manner. In an infrastructure workflow, the point cloud is not merely a collection of coordinates: it is a database that can store information produced at different stages (pre-processing, feature extraction, classification results) and can therefore act as a common interface between acquisition, analysis and modelling. To achieve this, the dataset must be organised, versioned and enriched consistently, so that the same cloud can be imported and processed in different environments (CloudCompare, Python toolchains, Rhinoceros/Grasshopper, MIDAS) without losing the correspondence between points, attributes and the transformations applied.

Within this framework, a distinction is made between organised and unorganised point clouds. Organised point clouds are structured in a 2D matrix form, typical of depth sensors, and preserve the topology of acquisition. By contrast, unorganised point clouds—typical of TLS and photogrammetric surveys—are a set of points without explicit neighbourhood connectivity, for which neighbourhood relations must be reconstructed algorithmically through spatial searches.

In bridge surveys, point density can vary significantly due to distance from the sensor, scan angle, shadowing and occlusions. This variability affects the estimation of local geometric quantities and requires processing strategies capable of adapting to spatial scale. Voxelisation, resampling and careful neighbourhood selection are therefore necessary to ensure stable descriptor computation.

To transform geometric data into semantic information, it is necessary to define descriptors that capture local shape and allow different structural

components to be distinguished. In this thesis, local geometry is described through features derived from the covariance matrix of k neighbourhood points, whose eigenvalues provide a compact representation of dimensionality (linear, planar or volumetric behaviour). This formalisation is central to the supervised classification pipeline described in Chapter 4.

For each point $\mathbf{p}_i$ in the cloud, consider a neighbourhood consisting of its nearest k points. The local covariance matrix is defined as (Hoppe et al., 1992):

$$C_i = \frac{1}{k}\sum_{j=1}^{k}(p_j - \bar{p})(p_j - \bar{p})^T \quad (3)$$

where the local centroid $\bar{\mathbf{p}}$ is computed as

$$\bar{p} = \frac{1}{k}\sum_{j=1}^{k} p_j \quad (4)$$

And $\mathbf{p}_j \in \mathbb{R}^3$.

The analysis of the eigenvalues and eigenvectors of the covariance matrix yields a set of scale-dependent descriptors—such as linearity, planarity and sphericity—used to characterise structural elements. These descriptors are particularly suitable for infrastructure contexts because they capture dominant morphologies (slabs, beams, columns) and can be computed efficiently when supported by appropriate spatial indices (Hoppe et al., 1992; Pauly, Keiser & Gross, 2003)..

From a computational standpoint, point clouds acquired on bridges can reach sizes on the order of 10^7–10^8 points. Processing such datasets therefore requires strategies for data reduction (voxelisation, tiling) and out-of-core management, as well as the use of scalable data structures. These aspects are treated not as marginal optimisations, but as conditions that directly influence the reproducibility and feasibility of the overall pipeline.

In addition, adopting standard file formats (LAS/LAZ, E57, PLY) enables interoperability between different software tools and preserves metadata crucial for traceability (coordinate reference system, acquisition parameters,

units and point attributes). Standardisation also facilitates repeatable workflows and allows surveys performed in different campaigns or on different assets to be compared.

In the context of this thesis, the point cloud is therefore the initial representation from which both semantic segmentation (Chapter 4) and the subsequent information and numerical models (Chapters 5 and 6) are derived. Its strategic role lies in providing a direct link to reality, supporting the decision chain from inspection and asset management to modelling and structural assessment.

The following sections introduce the mathematical and computational tools required to make this transformation systematic: an explicit definition of neighbourhood relations, reference-frame management to guarantee metric consistency, and data structures to scale processing to large infrastructure datasets.

As a final note, a minimal Quality Assurance (QA) checklist is provided: (1) coherence of the EPSG code and measurement units; (2) post-voxel density consistent with the element scale; (3) oriented and stabilised normals (checked through histograms and visual inspections); (4) registration RMSE and loop-closure within thresholds; (5) complete logs of configurations, software versions and random seeds.

1.2 MATHEMATICAL REPRESENTATION OF 3D DATA

Within the proposed framework, a rigorous mathematical representation of the point cloud is necessary to define neighbourhood relations, compute local features and support the construction of information and numerical models. The point is considered as an entity enriched with attributes, and the point cloud as a finite set embedded in Euclidean space.

Formally, a point cloud is defined as a finite set of points in three-dimensional Euclidean space:

$$\boldsymbol{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots\ldots, \mathbf{p}_N\} \subset \mathbb{R}^3 \quad (5)$$

$$p_i = \left(x_i, y_i, z_i, r_i, g_i, b_i, n_{x_i}, n_{y_i}, n_{z_i}, c_i\right)$$

In this definition $p_i = (x_i, y_i, z_i)$, the coordinates represent position, $a_i = (r_i, g_i, b_i)$ the additional attributes include colour, $a_i = \left(n_{x_i} n_{y_i}, n_{z_i}\right)$ surface normals and semantic labels, and c_i the overall set represents the sampling of the asset surface. This abstraction makes it possible to treat the point cloud as a dataset on which operators—filters, neighbourhood queries and feature computations—can be defined in a precise and reproducible manner.

To operate on the point set, a metric must be introduced to measure distances between points. In most practical cases, Euclidean distance is adopted, although alternative metrics can be used when anisotropy or weighted directions must be considered. The choice of metric directly influences neighbourhood construction and therefore the stability of all subsequent geometric estimates. The Euclidean distance between two points $\mathbf{p}_i$ and $\mathbf{p}_j$ is defined as:

$$d\left(p_i, p_j\right) = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2} \quad (6)$$

Given a point $\mathbf{p}_i$, its neighbourhood k is defined as a subset of points located within a prescribed scale. Two main definitions are used: the k-nearest neighbours (k-NN), which select a fixed number of closest points, and the radius-based neighbourhood (r-ball), which includes all points within a given metric radius.

$$N_k(p_i) = \{p_j \in P \; | \| \; p_j - p_i \; \| \quad (7)$$

The scale of analysis is a central concept: it defines the size of the local window used to compute geometric statistics and must be compatible with the characteristic dimensions of structural elements. Choosing too small a neighbourhood makes descriptors sensitive to noise and sampling artefacts; choosing too large a neighbourhood blurs discontinuities and mixes different components.

In point clouds with non-uniform density, k-NN ensures a constant number of samples but may correspond to very different physical extents in space, whereas r-ball guarantees a constant metric extent but yields variable numbers of samples. For infrastructure assets, both definitions are useful and can be compared or combined to improve robustness and generalisation.r

Starting from a neighbourhood, the covariance matrix of the centred coordinates provides a compact representation of local geometry. The eigenvalues quantify the spread along principal directions, while eigenvectors define local axes. This mathematical formalisation is the basis for computing the geometric features used in the classifier.

In addition to feature computation, the mathematical representation supports the notion of invariance. Many descriptors are designed to be rotation-invariant so that they can be transferred across different orientations and reference systems. Orientation-dependent descriptors, when used, must be interpreted with respect to explicitly defined local frames.

Neighbourhood relations can also be expressed through graphs ($\mathbf{P}, \mathbf{E}$)

, in which nodes represent points and edges represent adjacency. Graph representations support the computation of local and global operators, facilitate segmentation processes and make it possible to introduce regularisation terms based on spatial coherence.

To move from point-based representations to surfaces and volumes, it is necessary to make connectivity explicit. This is achieved through mesh representations, which encode topology and enable the construction of continuous geometric entities suitable for modelling and structural analysis.

$$\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F}) \tag{8}$$

In practice, point clouds and meshes coexist in the proposed workflow: the point cloud provides the raw metric sampling and supports feature-based classification, while the mesh supports the construction of modelling

primitives and numerical discretisations. The transition requires explicit rules to ensure that semantic labels and geometric fidelity are preserved.

Different discretisation schemes are available, including triangulations derived from Delaunay/Poisson reconstruction and quadrangular meshes based on surface parameterisation. The choice depends on the structural class, required fidelity and downstream use (information modelling vs FEM).

All these representations must be anchored to well-defined reference systems, since coordinate transformations affect numerical stability, interoperability and the interpretation of orientation-dependent descriptors. These aspects are discussed in §1.3.

The next section introduces the coordinate reference systems used in infrastructure surveys and the geometric transformations needed to move from the sensor frame to global and local modelling frames.

1.3 COORDINATE TYPES AND GEOMETRIC TRANSFORMATIONS

Infrastructure surveys are characterised by the coexistence of different coordinate reference systems. Raw acquisitions are defined in the sensor coordinate system, while modelling and asset management often require world coordinates (georeferenced) and local component-based frames. Managing transformations between these systems is essential to ensure metric coherence and comparability.

In heterogeneous acquisition scenarios, each scan or photogrammetric block is initially expressed in its own sensor frame and must be aligned and, when required, georeferenced. The transformation to world coordinates is usually performed through control points, GNSS constraints or network adjustments, and must be accompanied by residual analysis to quantify uncertainty.

In this thesis, a sensor→world→local chain of transformations is assumed. This ordered chain preserves georeferencing when needed while enabling local representations centred on the asset or on specific structural

components, improving numerical stability and supporting orientation-dependent computations.

Rigid transformations are represented as a combination of a rotation matrix and a translation vector. In homogeneous coordinates, they can be expressed as 4×4 matrices, facilitating concatenation and application to point sets. This formalism is used throughout the pipeline for registration, coordinate shifts and model export.

The choice of local reference systems influences the interpretation of certain descriptors. While many geometric features derived from eigenvalues are invariant to rigid rotations, other quantities—such as normals, orientation indicators and section extraction rules—depend on the adopted frame. For this reason, local frames must be explicitly defined and documented.

From the perspective of numerical stability, working directly in global georeferenced coordinates may lead to large values that degrade floating-point precision in modelling environments and FEM solvers. A local shift (global translation) is therefore commonly applied, ensuring that the geometry remains close to the origin without losing the link to the original world coordinates.

Transformation quality must be evaluated through quantitative indicators and targeted checks, including RMSE values from registration, loop-closure consistency in multi-scan networks and visual inspection of overlays. These checks reduce the risk of propagating misalignments into feature computation, classification and modelling.

Reference systems also play a role in BIM/HBIM modelling, where elements are often defined in local coordinate frames and then positioned in an overall project frame. Ensuring the coherence of transformations between point clouds, IFC entities and FEM models is therefore crucial to achieve interoperable and traceable workflows.

1.4 DATA STRUCTURES FOR POINT CLOUD REPRESENTATION

Point clouds acquired on infrastructure assets are large and heterogeneous; therefore, data structures are not merely implementation choices but methodological components that determine scalability and stability of the entire process. Efficient representations are necessary to support neighbourhood queries, feature computation and class assignment.

A basic representation is a tabular matrix of size N×M, where each row corresponds to a point and columns store coordinates and attributes (colour, normals, features, class labels). This representation is effective for vectorised operations and machine-learning input, but it does not provide spatial connectivity.

To enable efficient spatial queries, indexing structures such as k-d trees are adopted. K-d trees support fast k-NN searches and radius queries, reducing the computational cost of neighbourhood construction and making feature extraction feasible on large datasets.

Neighbourhood graphs provide an explicit representation of adjacency and can be used for segmentation, smoothing and spatial-coherence assessment. Graph-based representations facilitate the propagation of semantic information and the identification of connected components.

Voxel grids discretise space into regular cells and provide controlled resampling of the data. Voxelisation allows point density to be regulated, reduces memory requirements and enables block-based processing strategies, which are essential for out-of-core workflows on large infrastructure point clouds.

Processing large datasets requires balancing geometric precision against computational resources. Choices such as voxel size, tile extent and index reuse directly influence feature stability and processing time. For this reason, parameters are selected based on both geometric considerations (element scale) and computational constraints.

When connectivity must be explicit—for example for surface reconstruction and FEM mesh generation—mesh data structures are required. These

structures store vertices, edges and faces and support checks on topology, quality and geometric deviation from the original point cloud.

Throughout the workflow, interactive visualisation is used to verify the effects of data-structure choices (e.g., voxelisation and tiling) and to ensure that simplification does not remove relevant geometric information needed for classification and modelling.

1.5 PYTHON LIBRARIES FOR POINT CLOUD MANAGEMENT

The processing pipeline proposed in the thesis is implemented in the Python ecosystem, leveraging open-source libraries for point-cloud I/O, geometric processing, spatial indexing, feature extraction, machine learning and visualisation. The selection of libraries is guided by reproducibility, scalability and interoperability requirements.

Table 1.1 summarises the main libraries used throughout the workflow, highlighting their role in each processing stage and their contribution to maintaining traceability and repeatability across experiments.

Tab. 1.1 Main libraries used in the workflow

Library	**Main role in the workflow**	**Primary phase**
Open3D	Geometric processing, filtering, normal estimation	Pre-processing e geometry
PDAL	Import/export, out-of-core management, pipeline on large datasets	I/O and data management
SciPy	Space research facilities (k-d tree) and numerical calculations	Local analysis and features
scikit-learn	Supervised classification (Random Forest)	Machine learning

Library	Main role in the workflow	Primary phase
CloudCompare	Visual inspection, attribute-based analysis, quality control	Visualisation and QA
PyVista / vedo	3D visualisation in the Python environment	Visualisation support

A first group of tools is dedicated to handling point-cloud input and output formats. Data acquired through LiDAR sensors or photogrammetric techniques are stored in standard formats such as LAS/LAZ, E57 and PLY, which differ in compression capabilities, metadata management and flexibility in adding attributes. Dedicated libraries make it possible to import and export these datasets while preserving fundamental information—coordinates, intensity, colour and labels—and maintaining a direct correspondence between the data processed in Python and that visualised or modelled in external software.

Pre-processing operations and geometric management of the point cloud rely on libraries oriented to direct 3D processing. In this phase, controlled density reduction, outlier removal and surface-normal estimation are performed—operations that strongly influence the stability of geometric features and, consequently, the performance of supervised classification. Filtering and estimation parameters are explicitly documented to guarantee reproducibility across different datasets and acquisition campaigns.

Spatial search structures play a central role, enabling efficient computation of neighbourhoods and local statistics on large point clouds. Integrating such structures into the pipeline allows the transition from a purely tabular representation of the data to a local geometric interpretation, indispensable for feature extraction as described in Chapter 4. To contain computational costs, searches are performed on datasets that are pre-reduced and indexed, reusing the same structures across multiple operations whenever possible.

Supervised classification relies on established machine-learning libraries that make it possible to train robust models from heterogeneous features and potentially imbalanced datasets. In this context, adopting a Random Forest classifier provides a good compromise between accuracy, interpretability and computational cost. Trained models are saved and reused in a controlled manner during validation stages, avoiding unnecessary recalculations and ensuring coherence across experiments.

Alongside numerical processing, 3D visualisation plays a fundamental operational role. Dedicated tools make it possible to inspect point clouds and classification results interactively, making phenomena such as localised errors, geometric discontinuities or semantic inconsistencies immediately readable.

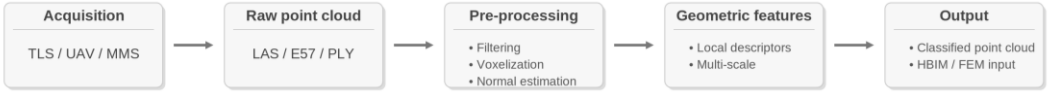


Fig. 1.1 Overall structure of the workflow

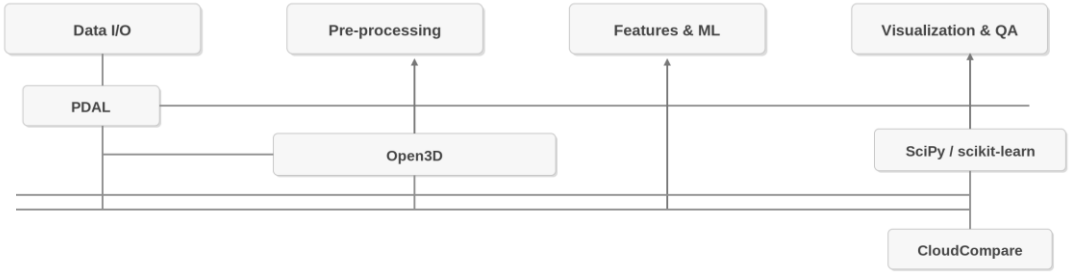


Fig. 1.2 Relationship between software tools and processing stages

The role of visualisation in the overall workflow is illustrated schematically in Figure 1.1 and Figure 1.2, which show, respectively, the general structure of the pipeline and the relationship between software tools and processing stages.

Particular attention is also devoted to process traceability. Each stage of the pipeline is associated with configuration files and logs that record parameters, library versions and data paths. This practice makes it possible to reconstruct the entire workflow a posteriori and represents an essential requirement for scientific validation of the results and for reusing the pipeline in different application contexts.
Overall, the set of adopted Python libraries is not merely a technical support, but constitutes the methodological backbone that enables integration between 3D surveying, machine learning, and information and structural modelling.

1.6 TECHNICAL VISUALISATION AS CONTINUOUS PROCESS VERIFICATION

In the context of this thesis, interactive visualisation is not considered an ancillary or merely illustrative stage (CloudCompare Team, n.d.; Sullivan and Kaszynski, 2019; Zhou, Park and Koltun, 2018), but an operational tool that accompanies the entire processing pipeline, from the raw point cloud to supervised classification and the subsequent transfer towards information and structural models. Visualisation plays an essential role in linking the numerical and statistical quantities introduced in the previous sections with the real geometry of the asset, enabling a critical interpretation of the results. First, it supports preliminary exploration of the data: through 3D rendering it is possible to assess acquisition quality, identify density variations, noise, occlusions and discontinuities. Second, it enables verification of the effects of geometric transformations and filters, by comparing the point cloud before and after voxelisation, outlier removal and alignment operations, and by assessing the stability of normals and local geometric quantities. Third, visualisation helps validate the adopted analysis scales: colour mapping of descriptors such as planarity, linearity and anisotropy allows one to check whether features reflect the expected morphology of structural elements, and guides adjustment of neighbourhood parameters when incoherent patterns

or artefacts appear. Finally, class-based rendering of the classified point cloud makes spatial coherence of predicted labels immediately readable, highlighting systematic errors, fragmentation or confusions between geometrically similar classes. This qualitative control complements quantitative metrics and supports robust decision-making throughout the workflow.

The role of visualisation in the overall workflow is illustrated schematically in Figure 1.3, which shows the integration between numerical analysis, classification and visual control within the processing chain.

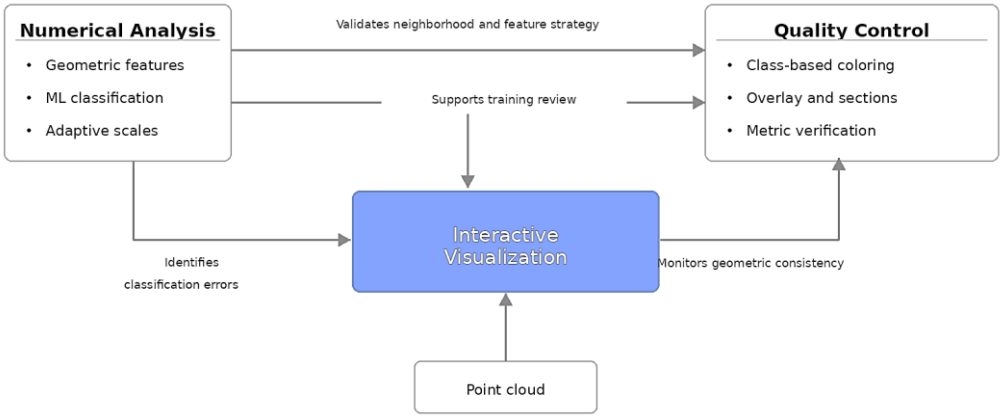


Fig. 1.3 Integration between numerical analysis, classification and visual verification in the processing workflow

A further function of visualisation concerns the transition towards information and structural modelling. Through sections, projections and overlays between the point cloud and geometric models it is possible to verify metric coherence between the surveyed data and the parametric representation. This step makes it possible to identify local deviations, misalignments or uncertainty zones before exporting to HBIM/BrIM and FEM environments, reducing the risk of propagating errors into simulations and information systems.

In the adopted workflow, visualisation is therefore used as a tool for continuous quality control, integrated with the other processing stages and not confined to the mere presentation of results. The same rendering modes, colour palettes and visualisation ranges are kept consistent across different datasets so that analyses are comparable and comparative assessments between case studies are supported. In summary, interactive point-cloud management makes it possible to transform numerical indicators into operational knowledge, bridging the gap between algorithmic processing and the physical reality of the surveyed infrastructure.

CHAPTER 2 — FUNDAMENTALS OF ACQUISITION, PRE-PROCESSING AND FEATURE EXTRACTION

2.0 INTRODUCTION

In this chapter the entire pipeline is described and formalised, leading from raw 3D survey data to the production of information and numerical models that can be used in HBIM/InfraBIM environments and in finite-element codes.

The pipeline follows a coherent design from point cloud to model and has been conceived to be reproducible, traceable and generalisable beyond the original case study. The workflow is trained on a pilot reinforced-concrete bridge and validated cross-bridge on a second bridge with different geometries (pier and deck cross-sections), non-structural components and environmental context. This choice requires dealing with domain shift (changes in the data domain) and maintaining metric and semantic consistency throughout all processing stages.

Starting from the characterisation of organised point clouds (TLS range images) and unorganised point clouds (UAV/SfM photogrammetry, multi-sensor fusion), with attention to reference frames, accuracies and formats (LAS/LAZ, E57, PLY), the availability of XYZ coordinates coherent with the bridge axes is verified. This is a necessary condition to guarantee the transferability of geometric descriptors and to enable a spatial comparison with HBIM/FEM models in the final stages.

Starting from data normalisation and cleaning operations (statistical outlier removal, downsampling, robust normal estimation, registration/translation of the validation cloud into the training reference system (Besl & McKay, 1992; Chen & Medioni, 1992; Rusinkiewicz & Levoy, 2001), and block tiling) for datasets with more than 10 million points, the adopted choices are documented in order to reduce oversmoothing and preserve structural discontinuities (deck edges, pier outlines, walkways). (Ester et al., 1996)

For each point, a k-NN neighbourhood is defined and the covariance matrix of positions is computed; its diagonalisation yields eigenvalues and eigenvectors that feed rotation-invariant descriptors (Hoppe et al., 1992; Pauly, Keiser & Gross, 2003) such as linearity, planarity, sphericity, anisotropy and omnivariance. The set is enriched with oriented features (verticality/horizontality, angle to the vertical axis), local-density measures and, where useful, absolute height and height normalised with respect to the point-cloud centroid. The aim is to obtain compact but informative feature vectors capable of discriminating structural classes (piers, decks, abutments, slabs) and non-structural classes (barriers, walkways, ground, vegetation).
Features are organised in a table (CSV/Parquet) and combined with the coordinates for spatial traceability. The training dataset is balanced (controlled downsampling of majority classes and, when necessary, class_weight=balanced weighting) and then used to train a Random Forest (RF) classifier with reproducible settings (fixed random_state). The model is serialised (joblib) and re-applied to the validation cloud.
The use of RF provides robustness to noise, interpretability (mean feature importance) and computation times compatible with large-scale operational pipelines. The evaluation combines standard macro-averaged metrics (Accuracy, Precision/Recall/F1) with measures of statistical coherence between train/validation distributions (Jensen–Shannon divergence) and spatial checks. Results are presented through normalised confusion matrices, frequency histograms and 2D/3D thematic maps (CloudCompare), preserving alignment with the original coordinates for comparison with mesh/FEM models.

2.1 DEFINITION AND CHARACTERISTICS OF POINT CLOUDS

The point cloud is the format that links the measured reality to computational models. In the infrastructure domain, its semantic neutrality is an advantage, as it avoids premature parametric assumptions on complex geometries characterised by material heterogeneity (RC, prestressed RC, steel, masonry)

and criticalities (discontinuities, edges, deformations). To become information, however, the cloud must be formalised explicitly and reproducibly, because pre-processing, feature extraction and classification rely on this formalisation. The cloud is treated as a discrete set (Equation 5) as a set-theoretic representation, introducing the mathematical framework on which local operations and neighbourhoods are defined. In the proposed methodology, this framework supports neighbour-graph construction, feature extraction and, downstream, classification. Equation 2 defines the enriched pointas an ordered tuple which, in addition to the metric coordinates (x_i, y_i, z_i) , stores the instrumental/radiometric attributes (I_i (o $\tau_i)$; $r_i, g_i, b_i)$, the local geometric attributes ($\mathbf{n}_i$) , and the informational attributes—respectively normal, semantic class and timestamp/ID—making the point traceable along the entire pipeline (features, classification, HBIM/FEM).

$$p_i = (x_i, y_i, z_i, r_i, g_i, b_i, I_i, n_i, c_i, t_i) \quad (9)$$

This formulation then justifies the use of the same columns in CSV/PLY/LAS files and in DataFrames for pre-processing and validation.

Extending the 3D point with attributes makes the information state traceable along the pipeline: the same attributes become columns in DataFrames (exchange via CSV/PLY/LAS) and channels for visualisation (CloudCompare/Open3D). It is good practice to maintain clear column conventions (XYZ, RGB, normal, feature_, predicted_class, frame_id) to align datasets, Python code and HBIM/FEM software. Metric information defines proximity, while the neighbourhood relation (k-NN or r-ball) determines which portion of the structure each point “sees”. For bridges, the choice of k or r must be tied to the anatomy of the asset (for example, too-small windows confuse roughness/noise with morphology, while too-large windows flatten functional edges such as the deck border, ribs and parapets), in order to avoid degradation in index quality. In real scenarios the density varies (occlusions, sensor distance/angle). We therefore refer to an effective density,

commensurate with the minimum meaningful scale of elements (deck-slab thickness, pier cross-section, parapet width). Voxel downsampling and SOR filters make density more uniform and stabilise local statistics (covariance), improving the robustness of planarity/linearity/anisotropy indices that will feed the Random Forest. To describe point density we can use the formulation proposed by Breiman, Hoppe et al., Pauly, Keiser & Gross, and Zhou, Park & Koltun:

$$\rho(p_i) = \frac{3|\mathcal{N}_r(p_i)|}{4\pi r^3} \tag{10}$$

dove:

- $\rho(p_i)$(Web) point density within a spherical neighbourhood of radius r centred at p_i (units typically "points per m³" for coordinates in metres);
- $\mathcal{N}_r(p_i)$(Web) set of neighbours, i.e. all the points $\boldsymbol{p_i}$ in the cloud such that $\|p_j - p_i\| \leq$ r which expresses the cardinality of the set (how many neighbours there are);
- $4\pi r^3$ volume of a sphere of radius r.

The density-estimation equation is translation-invariant and, at a consistent coordinate scale, comparable across scenarios; it is useful to detect outliers/sparse areas and to weight classes during training (e.g., reweighting for low-density classes). It fits well into local-feature pipelines (together with planarity, anisotropy, etc.) to improve separability between ground/vegetation/structures. In practice, neighbours within r are counted and divided by the sphere volume, providing a simple and interpretable estimate of local density around point p_i. Organised range images (TLS) preserve sensor adjacency and facilitate interpolation and hole filling; unorganised clouds (SfM/UAV) offer flexible coverage but require indexing (k-d tree) and more cautious scale policies. The nature of the data directly affects normal estimation (stability), neighbourhood construction and computational costs.

File-format choices are not neutral: LAS/LAZ preserves classes and intensity for LiDAR; PLY is flexible (additional columns, features, predicted class); E57

is suitable for multi-sensor scenarios. It is good practice to fix column conventions so that datasets, Python code and HBIM/FEM environments speak the same language.

2.2 DATA STRUCTURES FOR POINT-CLOUD REPRESENTATION

The operational structuring of point clouds is presented below, in order to represent, index and organise them so as to make searches and computations scalable (features, graphs, classification) on datasets on the order of 10^7–10^8 points. The choice of data structures directly influences runtime, memory, the quality of local estimates (covariance) and, downstream, the performance of the Random Forest and the consistency of HBIM/FEM mapping.

The matrix representation is the simplest and most portable (CSV/TXT/LAS/PLY): each row is a point and each column an attribute (XYZ, RGB, intensity, normal, feature_, predicted_class, frame_id, T_id). This form (9) is ideal for logging and interoperability, but it is insufficient for fast geometric operations based on local neighbourhoods and covariance analysis (Hoppe et al., 1992; Pauly, Keiser & Gross, 2003).

$$P = \begin{bmatrix} x_1 & y_1 & z_1 & r_1 & g_1 & b_1 & \dots \\ x_2 & y_2 & z_2 & r_2 & g_2 & b_2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ x_N & y_N & z_N & r_N & g_N & b_N & \dots \end{bmatrix} \tag{11}$$

For k-NN and r-ball neighbourhood queries, spatial indices are required to reduce the complexity from $O(N)$ to approximately $O(\log N)$ per query. Among the most commonly used structures are k-d trees, based on a recursive partition of space along the axes, and octrees, which subdivide space into eight hierarchical sub-cubes. k-d trees are particularly suited to local searches, such as those used for normal estimation and covariance matrices, whereas octrees effectively support Level of Detail (LOD), decimation and multi-resolution management of point clouds.

From an implementation perspective, established libraries such as Open3D, PCL, CGAL and SciPy provide optimised data structures and dedicated APIs for the efficient execution of single and batched queries. (Zhou, Park & Koltun, 2018; Virtanen et al., 2020)

The neighbourhood is formalised either through k-NN or r-ball criteria; this choice directly affects the robustness of the estimates and the scale of the extracted features (Hoppe et al., 1992; Pauly, Keiser & Gross, 2003). A hybrid criterion, enforcing a minimum radius and a maximum number of neighbours, helps mitigate the effects of non-uniform data density, due for instance to occlusions or to variations in distance and angle with respect to the sensor. (Zhou, Park & Koltun, 2018)

For robust local-feature computation, each point p_i is assigned a neighbourhood of type ε-ball

$$\mathcal{N}_\varepsilon(p_i) = \{\, p_j \in P \mid d(p_i, p_j) \leq \varepsilon \,\} \tag{12}$$

that is the set of samples whose distance from p_i does not exceed a threshold ε according to the metric $d(\cdot,\cdot)$, typically Euclidean or appropriately weighted. This neighbourhood provides the basis for constructing k-NN or -ε -ball graphs and is used in all local estimates, such as the computation of normals, curvature, and planarity and linearity indices.

$$\mathcal{N}_k(p_i) = \{\, p_j \in P \mid d\big(p_i, p_j\big) \leq \varepsilon \,\} \tag{13}$$

From the neighbourhood we build the k-NN graph (weights from distance or angular similarity between normals). On this graph we compute the local covariance and the eigenvalues λ1≥λ2≥λ3; from them we derive planarity, linearity, sphericity, anisotropy and other indices used in §2.5 and in RF training (§2.6). On this neighbourhood we estimate the covariance matrix, centred with respect to the mean; its eigenvalues and eigenvectors (normal) generate the geometric features used in the pipeline (planarity, linearity, sphericity, anisotropy) (Hoppe et al., 1992; Pauly, Keiser & Gross, 2003) and feed classification and QA.

$$C_i = \frac{1}{k}\sum_{j=1}^{k}(p_j - \bar{p})(p_j - \bar{p})^T \quad (14)$$

Voxelisation controls the resolution (ℓ), makes density more uniform and reduces search times. For very large datasets, processing is performed in blocks (tiling) with ghost borders near the boundaries; global indices ensure correct recomposition downstream. For tabular filtering and selection operations it is natural to adopt an Array of Structures (AoS) layout, whereas for numerical routines that stream columns (normals, a specific feature) a Structure of Arrays (SoA) is preferable, reducing cache misses and improving access locality. Building the index only once and caching reusable neighbourhoods significantly reduces runtime in the feature stages. When explicit connectivity is required, one can transition to meshes with half-edge data structures, which provide consistent access to vertices, edges and faces, definition of domains and constraints, and preparation for FEM. The transition should be planned so as not to degrade semantics (class/object) and to control point-to-surface deviation.

Explicitly defining data types and numerical precision (float32 vs float64), the spatial reference system (EPSG and units), as well as the parameters k, r, ℓ, and storing them in configuration files (e.g., YAML), makes the pipeline repeatable and traceable. The same choices are reflected consistently both in result visualisation, in terms of colour palettes and value ranges, and in the computation of the evaluation metrics described in §2.7.

2.3 STORAGE FORMAT MANAGEMENT TECHNIQUES

An integrated view of the main management techniques and storage formats for point clouds is provided below, highlighting the aspects that directly influence feature extraction, semantic segmentation and subsequent HBIM/FEM modelling.

The most widespread formats are LAS/LAZ (ASPRS standard with compression), PLY (geometry plus flexible attributes), E57 (rich metadata and multi-sensor support) and CSV/XYZ (minimal, useful for custom pipelines). Format choices and attributes (RGB, intensity, normal, class label, scan ID, time) affect portability towards HBIM/FEM software and the reproducibility of experiments. The acquired points are assembled into the cloud $P = \{p_i\}$, to which the workflow associates attributes (intensity, colour, time, provisional labels) and metadata (reference system, sensor poses, quality). (PDAL Contributors, n.d.)

A point cloud is said to be structured when it preserves the organisation by scans or by grid (organised clouds), with implicit local relationships (rows/columns, azimuth/elevation) useful for the efficient computation of normals and neighbourhoods. It is defined as unstructured when points are stored as an unordered set.

Classification pipelines benefit from structure in k-NN steps, which help ensure representative geometric content and manageable weights, preserving the quality of boundary regions, as well as from the availability of metadata (scan ID, intensity) and from the definition of robust features.

In general, before feature extraction, it is preferable to apply:

(i) voxel-grid filtering to control density,

(ii) outlier removal (statistical or RANSAC),

(iii) normal estimation and eigenvalues of the covariance matrix over k-NN neighbourhoods.

These steps balance computational cost and metric fidelity, preparing the data for validation metrics (accuracy/macro-F1, train/val JSD, mean spatial error along boundaries between classes).

Management and format choices directly affect feature quality and dataset balancing. The presence of coherent attributes facilitates training the classifier and validating it on independent point clouds. As output, a classified and regularised cloud enables parametric modelling and HBIM/FEM

integration. (Fischler & Bolles, 1981) (Hoppe et al., 1992; Pauly, Keiser & Gross, 2003).

2.4 — POINT CLOUD PRE-PROCESSING AND QUALITY CONTROL

Point-cloud pre-processing is a set of operations that transform raw data into reliable input for subsequent stages (feature extraction, supervised classification, parametric modelling and meshing). The workflow was applied both to the pilot bridge and to the independent validation cloud, ensuring coherence with the HBIM/FEM workflow.

The first step consists in estimating the local scale, measured through the average distance to the k nearest neighbours. For each point p_i of the cloud P (Equation 5), its k-NN are considered and the local density metric d is defined as the average of the norms of the difference vectors. Using Equation (6), the mean distance to the k neighbours is computed for local-scale estimation and, based on the obtained values, a statistical outlier filter is applied. Points with a mean distance above an adaptive threshold are then marked as anomalous, equal to:

$$d_i > \mu_d + \alpha\,\sigma_d \tag{15}$$

where μd is the mean, σd the standard deviation, and the coefficient α controls the trade-off between noise removal and the preservation of thin details. Equation 16 describes the statistical threshold criterion for outlier filtering.

After removing outliers, downsampling is performed to limit the number of points and to make their distribution more uniform. Two strategies were used: isotropic voxel-grid downsampling (2–5 cm step for classification) and minimum-distance subsampling. Both variants produce a controlled reduction N→N′, preserving the normals and curvature trends needed for feature extraction. The following equation

$$N \rightarrow N' \ll N \tag{16}$$

determinations the controlled reduction in the number of points (N→N′).
For classification purposes, robust estimation of normals and PCA-based descriptors (Principal Component Analysis) plays a central role. For each point, the covariance matrix of the associated k-NN neighbourhood is computed, from which rotation-invariant eigenvalues and eigenvectors are derived. From these quantities, the main local geometric descriptors are obtained, such as linearity, planarity, sphericity, anisotropy and entropy. The following formula (Hoppe et al., 1992; Pauly, Keiser & Gross, 2003)

$$C_i = \frac{1}{k}\sum_{j=1}^{k}(p_j - \bar{p}_i)\,(p_j - \bar{p}_i)^T \tag{17}$$

con

$$\bar{p}_i = \frac{1}{k}\sum_{j=1}^{k} p_j\,, \qquad p_j \in \mathbb{R}^3$$

represents the local covariance matrix for normal estimation and PCA-based features. To improve numerical stability in the PCA estimation and to facilitate comparisons between blocks, the translation equation is applied. (Hoppe et al., 1992; Pauly, Keiser & Gross, 2003)

$$\widehat{p_i}(t) = p_i - \bar{p} \tag{18}$$

which translates each neighbourhood with respect to its centroid. This centring reduces scale/offset effects and simplifies the computation of quality metrics.
Pre-processing concludes with a quality-control step that records:

- percentage of points removed as outliers,
- reduction factor N/N′,

- distribution of PCA radii and continuity of normals along the edges of structural components (piers, deck slabs, abutments).

These indicators are later recalled and used in subsequent analyses to interpret the classifier performance (accuracy/macro-F1) and the geometric coherence between training and validation. The normalised cloud thus becomes the direct input for feature extraction and, downstream, for the construction of parametric and FEM models. (Hoppe et al., 1992; Pauly, Keiser & Gross, 2003).

2.5 COMPUTATION OF LOCAL GEOMETRIC FEATURES

After format management and pre-processing (outlier filtering, downsampling and alignment), it is necessary to transform the point cloud into a set of local numerical descriptors suitable for classification. The basic idea is to associate with each point a feature vector capable of synthesising the morphology of its neighbourhood, capturing geometric properties such as linearity, planarity, sphericity and anisotropy.

These descriptors are translation-invariant and almost rotation-invariant, which allows their transfer between different structures (pilot → validation), in line with the objectives of cross-bridge generalisation.

For each point p_i the neighbouring points are selected k, either according to a k-nearest-neighbours criterion or by means of a fixed or adaptive radius. Starting from this local neighbourhood, the centroid $\bar{p}_i$ is then computed and, subsequently, the local covariance matrix C_i , which describes the three-dimensional dispersion of neighbouring points with respect to $\bar{p}_i$.

The matrix $C_i \in \mathbb{R}^{3\times3}$ is symmetric and positive semidefinite and constitutes the basis for defining nearly all the geometric features used.

The choice of the parameter k governs the trade-off between sensitivity to noise and the ability to describe geometric detail. Low values k of allow micro-geometries to be captured, but make estimates more sensitive to noise;

conversely, higher values produce more stable descriptors, at the cost of losing edges and geometric discontinuities, such as transitions between piers and abutments or between the intrados and the deck.

In the analysed bridge datasets, values between $k \in [15,30]$ were found to provide a good compromise, while still ensuring sufficient local adaptivity in areas characterised by variable point density.

The computation of geometric features starts from the construction of the local covariance matrix C_i for each point) $p_i \in \mathbb{R}^3$, defined with respect to its neighbours (Bentley, 1975). Denoting by $\bar{p}_i$ the centroid of the local neighbourhood k, the matrix C_i is diagonalised to obtain the three ordered eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and the corresponding orthonormal eigenvectors (Hoppe et al., 1992; Pauly, Keiser & Gross, 2003)

From this triad, compact and interpretable scalar descriptors are derived, which constitute the input of the classifier. In accordance with the literature and with what is adopted in this research, the following features are used:

- Linearity

$$L = \frac{\lambda_1 - \lambda_2}{\lambda_1} \tag{19}$$

highlights edges/linear elements (guardrails, pipes/ducts).

- Planarity

$$P = \frac{\lambda_2 - \lambda_3}{\lambda_1} \tag{20}$$

- Sphericity

$$S = \frac{\lambda_3}{\lambda_1} \tag{21}$$

- Anisotropy

$$A = \frac{\lambda_1 - \lambda_3}{\lambda_1} \tag{22}$$

- Omnivariance

$$O = (\lambda_1 \lambda_2 \lambda_3)^{1/3} \tag{23}$$

- Normalised curvature

$$C = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \tag{24}$$

- Eigenvalue entropy, which measures the degree of disorder of the spectral distribution.
- *Verticality, defined as*

$$V = \mid n_z \mid, H_0 = \sqrt{n_x^2 + n_y^2} \tag{25}$$

where is the eigenvector associated with , interpreted as the local normal.$n\ \lambda_3$

- Roughness (deviation from the local plane):

$$\rho = \sqrt{\frac{1}{k}\sum_{j=1}^{k}((p_j - \bar{p}_i) \cdot n_i)^2} \tag{26}$$

- Local density (optional):

$$\delta = \frac{k}{\frac{4}{3}\pi r_k^3} \tag{27}$$

with r_k distance of the k -th neighbour.

These quantities are redundant but complementary: combining orientational measures (verticality/anisotropy) with volumetric measures (omnivariance/entropy) increases separability between adjacent classes (e.g., ground–abutment, intrados–pier).

The formula:

$$\mathrm{Var}(z) = \frac{1}{k}\sum_{j=1}^{k}\left(z_j - \bar{z}\right)^2 \tag{28}$$

measures the height variation of points in the neighbourhood, useful for discriminating irregular ground from regular surfaces.

All these features are computed point-by-point and saved to a CSV file or to a compatible format for classification (e.g., a pandas DataFrame).

For the selection of the number of neighbours k, in view of the issues described above, this work adopted k = 20 as an optimal compromise between local detail and statistical robustness.

A further consideration concerns computational scalability. Processing the entire point cloud in a single operation is prohibitive for datasets larger than 10 million points. For this reason, a block-based computation system was implemented: the cloud is divided into manageable subsets, features are computed locally, and the results are subsequently recomposed into a single aggregated dataset. This strategy ensures compatibility with limited-memory machines and improves the overall stability of the process.

Local geometric features constitute the mathematical interface between the point cloud and machine-learning algorithms, representing a conceptual bridge between the unstructured geometric information of the survey and advanced semantic segmentation techniques.

Before the training phase, the features are standardised column-wise, according to:

$$f' = \frac{f - \mu_f}{\sigma_f} \tag{29}$$

This operation improves the convergence of learning algorithms and enables a coherent interpretation of feature importances and SHapley Additive exPlanations (SHAP) analyses, thanks to comparable scales.

To increase the robustness of the process, the following strategies were adopted:

- *Clipping outliers in the geometric features;*
- use of an adaptive value of in regions with low point density;k
- Coherent orientation of normals is enforced using minimum spanning tree (MST) structures and local propagation, in order to avoid sign inversions that can degrade sensitive features such as verticality and roughness.

From an implementation standpoint, neighbour search is performed via k-d tree or ball tree (scikit-learn), local PCA is obtained through eigenvalue

decomposition of the covariance matrix (NumPy), and feature computation is performed in vectorised form (pandas/NumPy).

The final result is a structured DataFrame of the form

$$[X \mid Y \mid Z \mid f_1 \dots f_m], \tag{30}$$

aligned point-by-point with the original cloud and serialised in CSV or Parquet format. This representation constitutes the input to the subsequent classification pipeline and guarantees interoperability with HBIM and FEM applications. (Bentley, 1975) (Hoppe et al., 1992; Pauly, Keiser & Gross, 2003) (Pedregosa et al., 2011).

2.6 CLASSIFICAZIONE SUPERVISIONATA E RANDOM FOREST. (BREIMAN, 2001)

Supervised classification is the step that transforms a set of local geometric features, extracted from k-neighbourhoods and enriched with orientational and height indicators, into semantic classes coherent with the structural breakdown of the bridge (piers, abutments, longitudinal/transverse beams, slabs, pavements) and with non-structural elements (guardrails, power-line poles, ground, vegetation). In the work presented, classification is not an end in itself but the trigger of the whole HBIM/FEM workflow, because from a lightweight thematic point cloud both parametric modelling and simulation meshes are derived.

The model was also conceived with the aim of ensuring cross-bridge generalisation: a classifier trained on a pilot bridge (A) should maintain stable performance when applied to a different bridge (B), characterised by different geometries, structural sections and environmental contexts.

Achieving this objective requires a robust pipeline, including class balancing, the use of interpretable and transferable geometric features, control of domain-shift effects, and a performance evaluation that is both quantitative and spatial.

The geometric information extracted from a 3D point cloud becomes useful when it is interpreted semantically, i.e., associated with categories that are meaningful for structural analysis and modelling. This step is performed through supervised classification, in which each point of the cloud is assigned to a predefined class (for example: ground, deck, piers, vegetation, barriers) based on the local features computed previously.

In the supervised-classification paradigm, the learning process starts from a training set, consisting of a subset of labelled points. The classifier learns the mapping between feature vectors and target classes, and can then predict the class of new, unseen points (validation set).

Given a tabular dataset composed of samples, each described by features, and belonging to one of the possible classes, the problem consists in finding a decision function that maps the feature space to the space of class labels. Classification can therefore be formalised as the following function:NnK

$$f: \mathbb{R}^n \rightarrow \{1, \dots, K\}. \tag{31}$$

The equation represents the mapping from the features in $\mathbb{R}^n$ to the class space $\boldsymbol{\mathcal{C}}$.

Among the various algorithms available, the Random Forest (RF) method introduced by Breiman (2001), has proved particularly suitable for point-cloud classification because of its robustness, ability to handle noisy data, interpretability and high performance even on imbalanced datasets. The algorithm, introduced by Breiman in 2001, combines multiple decision trees to reduce overfitting and improve generalisation. Each individual decision tree partitions the feature space into regions corresponding to each class through a sequence of hierarchical "if–else" rules built on feature values. This represents an example of a learning criterion that minimises empirical risk in the case of a multi-class loss function. Given a training set composed of N samples, the empirical risk associated with a decision function f is defined as:

$$L(f) = \frac{1}{N} \sum_{i=1}^{N} \ell(f(x_i), y_i) \tag{32}$$

where $x_i \in \mathbb{R}^n$ represents the feature vector of the i -th point (made up of local geometric descriptors and, where applicable, spatial coordinates), $y_i \in$

$\{1, \dots, K\}$ is the associated class label and $\ell(\cdot,\cdot)$ indicates the loss function, typically the log-loss or classification error.

In this context, a decision rule can be interpreted as a combination of relevant geometric features, such as planarity and verticality, capable of discriminating between different structural classes.

Random Forest (RF) is a non-parametric supervised-learning ensemble algorithm, composed of M decision trees. Each tree is trained on a bootstrap sample of the training set and, at each split node, evaluates a random subset of the available features. This randomisation mechanism introduces decorrelation among trees, contributing to variance reduction and limiting overfitting.

In the Random Forest model, individual trees produce independent predictions and the final classification is obtained by majority vote among all trees in the ensemble. Formally, the decision can be expressed as:

$$\hat{y}(x) = \arg \max_{k \in \{1,\dots,K\}} \sum_{m=1}^{M} \mathbf{1}(h_m(x) = k) \tag{33}$$

Where $h_m(x)$ is the class predicted by m the-th tree and $\mathbf{1}(\cdot)$ is the indicator function.

The algorithm also provides a measure of feature importance, based on the mean decrease in impurity (Gini impurity) across decision nodes. This property is useful for analysing which geometric characteristics are most discriminative between classes.

In the case study, training was carried out on a balanced dataset composed of 8 main classes:

10 -> ground
12 -> vegetation
20 -> abutments and barriers
30 -> piers
40 -> main beams
50 -> secondary cross-beams
60 -> slab

70 -> ancillary elements

For each point in the training set, the 18 geometric features were computed and the class label was added. Training was performed with a number of trees N = 100 and unconstrained maximum depth, with automatic class balancing through weights inversely proportional to class frequency.

For training, each point is assigned its class label and this label is appended to the feature vector, building a tabular dataset. The Random Forest is trained with 100 trees (n_estimators=100), with max_features set to sqrt(num_features) and with unconstrained max_depth, in order to allow the model to adapt to the variability of the descriptors. The random_state parameter is fixed to ensure reproducibility (Pedregosa et al., 2011)

The evaluation of performance is carried out through:

- accuracy (overall percentage of correct classifications);
- per-class precision/recall/F1-score;
- confusion-matrix analysis;

analysis of feature importance.

$$G(t) = 1 - \sum_{k=1}^{K} p_{k|t}^2 \tag{34}$$

where $p_{k|t}$ represents the proportion of samples belonging to class k in node t . Choosing splits that maximise the reduction in impurity makes it possible to obtain trees that are effective at separating classes.

In addition to the classic metrics, to evaluate the robustness of generalisation we also considered:

1. out-of-bag (OOB) error;
2. k-fold cross-validation on the training set;
3. cross-bridge validation on the independent bridge.
4. Out-of-bag error is computed on the samples left out of the bootstrap procedure and provides an internal estimate of performance without requiring a separate validation set. It is useful for checking whether the number of trees is sufficient and whether overfitting effects are present.

Cross-validation, on the other hand, repeats training across different splits, providing a more stable estimate but at a higher computational cost. In this work, k-fold cross-validation was used on reduced datasets to confirm the hyperparameter choices.

Finally, cross-bridge validation tests the actual generalisation capability of the model: performance is measured on a different bridge (Bridge B), acquired under different conditions and with different geometries, after aligning reference systems and ensuring feature compatibility.

- The training dataset is built starting from a portion of the pilot bridge, selecting representative areas for each class and balancing the number of points per class:
- training cloud: manually labelled subset;
- validation cloud: second structure acquired independently, translated (and, where required, scaled) into the training reference system;
- This split makes it possible to evaluate not only interpolation capability (same domain), but above all cross-bridge generalisation (different domain).

Given the nature of point clouds, the distribution of points across classes is typically highly unbalanced. Certain thin or less extensive components (parapets, walkways, some structural details) are represented by far fewer samples than dominant classes (ground, deck). This makes a balancing strategy necessary.

$$N_k = min(n_k, T) \tag{35}$$

uniform sampling per class c.

in order to obtain a balanced dataset, so that the classifier learns equally from all classes and does not become biased towards the most represented surfaces (ground, deck).

To mitigate the effects of class imbalance, during the training phase a class-weighting strategy was adopted, defined as:

$$w_k \propto \frac{1}{n_k} \tag{36}$$

Where n_k indicates the number of samples belonging to class k . This approach, implemented through the class_weight='balanced' option of the scikit-learn library, assigns a larger weight to under-represented classes, promoting fairer learning and increasing the sensitivity of the model towards minority yet relevant structural elements, such as piers, abutments, parapets and walkways.

This balancing is essential in infrastructure datasets, where ground and deck dominate by area, whereas thin elements such as parapets or piers are represented by fewer points.

Supervised classification based on Random Forest therefore constitutes the core of the semantic-segmentation process developed in this thesis, enabling the transformation of the raw point cloud into a structured, information-rich base organised into semantic classes and ready for the subsequent stages of parametric and structural modelling described in the following chapters (Breiman, 2001; Pedregosa et al., 2011).

After training, the model is applied to the validation cloud. For each point, the predicted class is computed and stored as an attribute in the output file (PLY/CSV), enabling visualisation and subsequent filtering for modelling.

Parameter	**Value / Notes**
Number of trees (N)	100
Criteria	Gini (impurity)
Maximum depth	Unencumbered (None)
Class balancing	class_weight = 'balanced'
Bootstrap (sampling with replacement)	Yes
Random feature selection	Enabled (default RF setting)

Dataset and split

- Training set: “RFI Bridge” point cloud with 8 balanced classes (10, 12, 16, 20, 22,26,28,30).
- Independent validation set: “Evaluation Bridge” point cloud acquired in a different context.
- Feature: 18 Local characteristics per point (calculated on k = 20 neighbors).
- Split strategy: separation by structure (no point-level mixing) to test spatial generalisation.
- Metrics used: overall Accuracy, Confusion Matrix, per-class Precision/Recall/F1.
- Implementation: scikit-learn (RandomForestClassifier), pandas, matplotlib, joblib.

The training set consists of approximately 1.6 million points (after balancing), whereas the validation point cloud contains approximately 2.1 million points. The dataset is split into 70% for training and 30% for internal validation. Out-of-bag error and k-fold cross-validation are used as additional checks. The final test is performed on the independent bridge, assessing cross-bridge generalisation.

Feature importance

- MDI (Mean Decrease in Impurity), defined as the sum, over all splits that involve the feature, of the reduction in the Gini index, weighted by the number of samples in the node.*j*
- Permutation Importance (MDA, Mean Decrease in Accuracy), which measures the variation in performance (Accuracy or F1-score) following the random permutation of the feature on the validation dataset.*j*

The feature-importance ranking makes it possible to identify the most discriminative descriptors for each class and to verify the consistency of the adopted feature set. In this work, orientational features (verticality) and height contribute strongly to the discrimination of piers and parapets, whereas planarity dominates in the separation of deck-slab and ground points; roughness and entropy are useful in identifying vegetation.

Statistical stability

The statistical stability between the training and validation datasets was also assessed using the Jensen–Shannon divergence between the class P and Q distributions and , defined as:

$$D_{\mathrm{JS}}(P \parallel Q) = \frac{1}{2} D_{\mathrm{KL}}(P \parallel M) + \frac{1}{2} D_{\mathrm{KL}}(Q \parallel M), \tag{37}$$

$$M = \frac{P + Q}{2} \tag{38}$$

Where D_{KL} indicates the Kullback–Leibler divergence.

JSD is symmetric and bounded; values close to 0 indicate similar distributions. In this thesis it is used as an indicator of domain shift between datasets, helping interpret potential performance drops when moving from the pilot bridge to the validation bridge.

$$E_{\mathrm{space}} = \frac{1}{N_{edge}} \sum_{i \in edge} d_i \tag{39}$$

where represents the distance of the classified point from the geometric reference of the correct class (mesh or annotations), computed on transition regions (boundary).d_i

Finally, a mean spatial error was computed, defined as the average distance between misclassified points and the boundary of the correct class.

- In the case study, the spatial error was computed by measuring distances to reference meshes extracted from the HBIM/FEM model or to manually delineated regions.

- These metrics, combined with confusion matrices and visual checks in CloudCompare, allow a robust interpretation of the results and support subsequent modelling decisions.
- The classified point cloud is visualised through thematic maps (colour by class) and through histograms of predicted classes. Comparing training and validation helps identify systematic drift and classes that are more sensitive to domain shift.
- To support analysis, confusion matrices are normalised (by rows and/or columns), highlighting where errors concentrate (e.g., abutment vs barrier, slab vs beam).

Spatial filtering (connected components, morphological operations on labels) can be applied as post-processing. In this thesis, however, the focus is on the raw classifier output, in order to preserve traceability between prediction and original data.

In conclusion, the classification results—together with feature importance and statistical stability indicators—constitute the basis for the subsequent chapters, where the thematic point cloud is used to drive parametric modelling and FEM discretisation.

2.7 VISUALISATION AND ANALYSIS OF CLASSIFICATION RESULTS

The results of the supervised classification of the point cloud were analysed and made usable, linking operationally the previous stages (feature extraction, balancing and training) with quantitative checks and qualitative inspection on real datasets. The aim was to rigorously measure the model performance (accuracy, macro-F1, confusion matrix) and to visually validate the geometric-structural coherence of the predicted classes, in order to guarantee the use of the output as a reliable input for parametric modelling and FEM.

The post-processing pipeline followed these steps:

(i) block-wise prediction with XYZ coordinate traceability;
(ii) merging the blocks and generating an intermediate (.CSV) file containing, for each point, the 18 features, the coordinates and the predicted class;
(iii) conversion to .PLY format for 3D inspection (CloudCompare) with a discrete colour palette per class (Figures 2.1 and 2.2 in CloudCompare);
(iv) production of summary plots such as the class-frequency histogram (Figure 2.3);
(v) normalised confusion matrix (Figure 2.4);
(vi) feature importance (Figure 2.5).

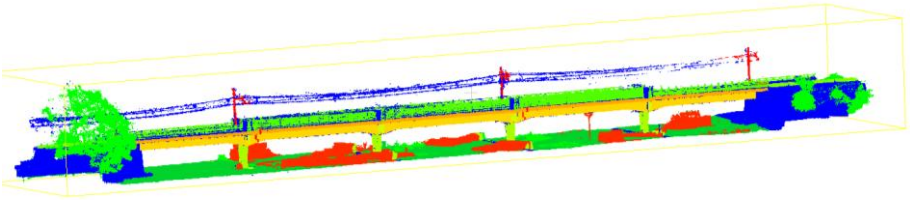


Fig. 2.1 — Classification displayed in CloudCompare for the training-bridge point cloud.

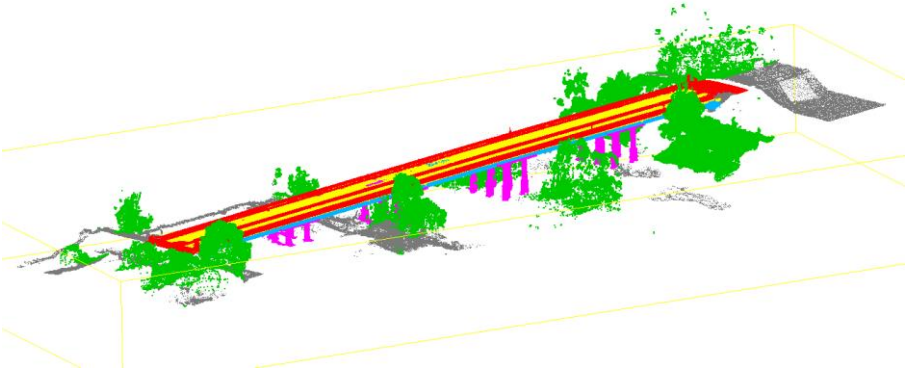


Fig. 2.2 — Classification displayed in CloudCompare for the validation-bridge point cloud.

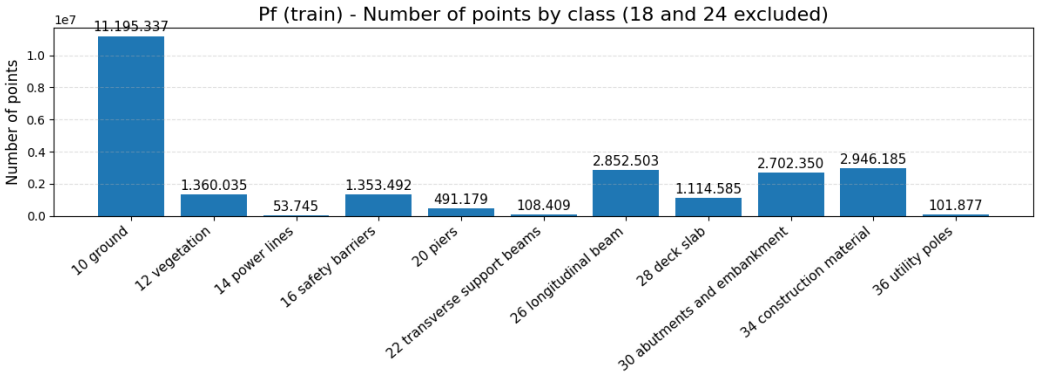


Fig. 2.3 — Class-frequency histogram (validation).

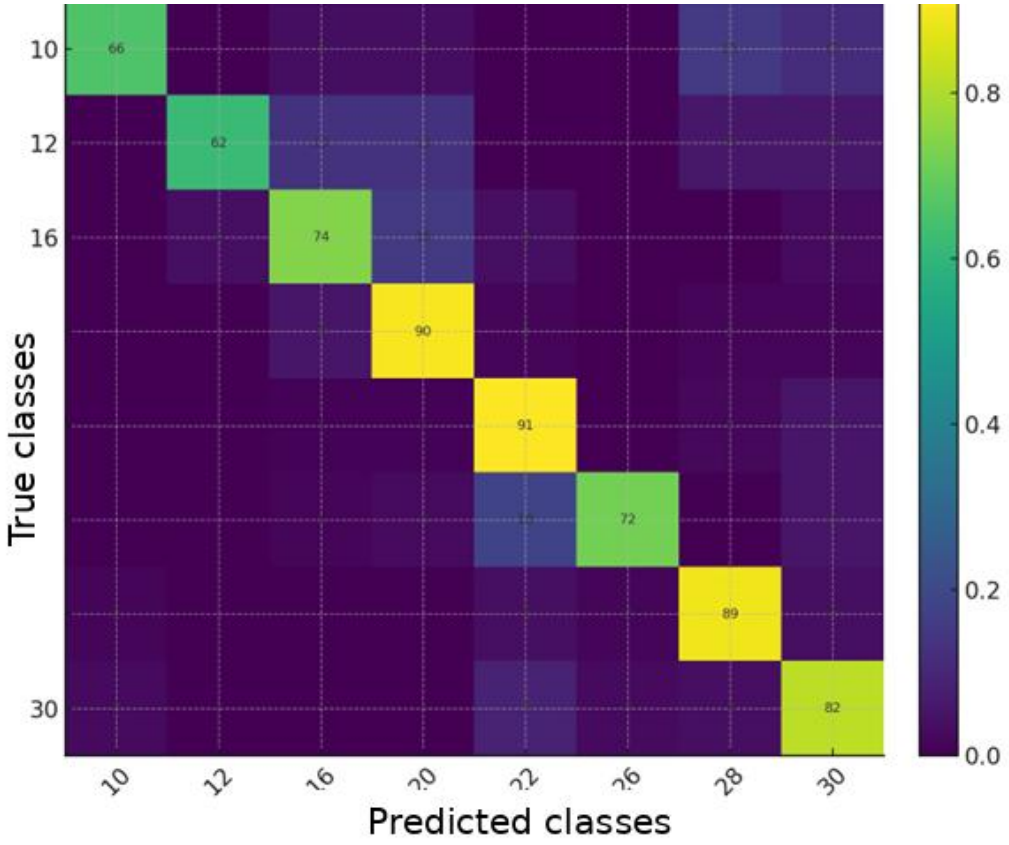


Fig. 2.4 — Normalised confusion matrix (%).

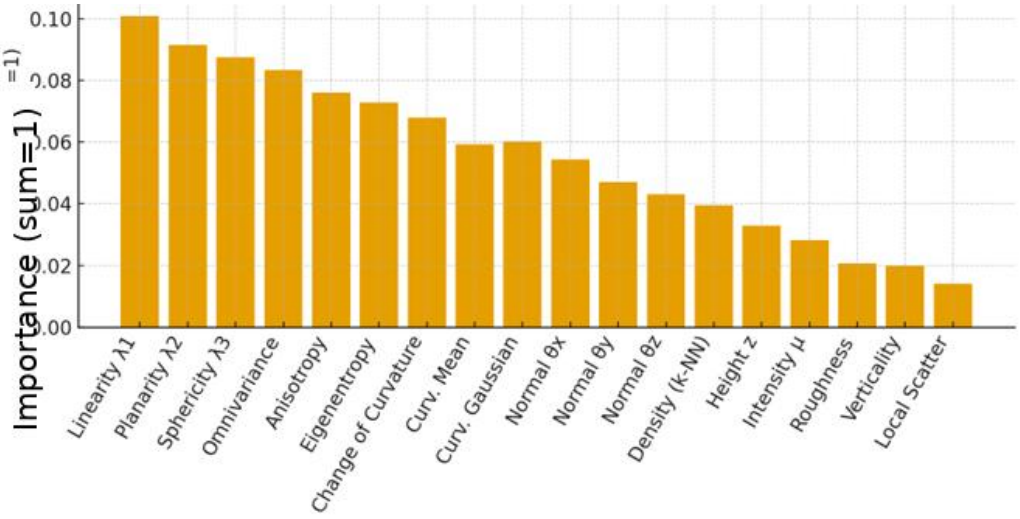


Fig. 2.5 — Mean feature importance (Random Forest).

Preserving the spatial coordinates of points allows a direct comparison between model predictions and reference geometries (HBIM/FEM), enabling a local verification of performance, particularly in transition regions between adjacent classes. Classifier performance is summarised through standard metrics, computed both on the entire dataset and on annotated subsets of the validation cloud.

The adopted metrics are defined below:

- Global accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{40}$$

- Per-class precision k

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k} \tag{41}$$

- Per-class recall k

$$\text{Recall}_k = \frac{TP_k}{TP_k + FN_k} \tag{42}$$

- Per-class F1-score k

$$\text{F1}_k = 2 \cdot \frac{\text{Precision}_k \cdot \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k} \tag{43}$$

- Macro-average of the F1-score

$$\mathrm{F1}_{\mathrm{macro}} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{F1}_k \tag{44}$$

- Jensen–Shannon divergence between the class distributions of the training () and validation () sets (Equation 37):PQ
- Mean spatial error along boundaries between classes (Equation 39)

Equations (40) measure overall classifier correctness; (41) and (42) quantify, respectively, the purity and coverage of predictions for each class; (43) provides a balanced synthesis; (44) assigns equal weight to all classes, regardless of frequency; (37) checks statistical coherence between training and validation; finally, (39) assesses the geometric quality of the segmentation along class transitions.

The normalised confusion matrix (rows: true classes; columns: predicted classes) is the main tool for analysing systematic confusions. Row normalisation allows the recall of each true class to be interpreted immediately, while column normalisation provides a direct measure of prediction precision. Recurring errors emerge in off-diagonal elements, such as the confusion between class 16 (abutments/barriers) and class 20 (piers). For qualitative validation, the classified point cloud was visualised in CloudCompare using a discrete colour scale, assigning a different colour to each class. This inspection highlighted the geometric continuity of the main structural components (piers, slab, deck) and made it possible to localise errors mainly in transition zones or partially occluded areas. In support of the quantitative evaluation, 2D intensity maps (orthogonal projections XY, XZ and YZ) and histograms of relative class frequency were also produced, useful for summarising the most frequent confusions (CloudCompare Team, n.d.).

These indicators are fundamental for identifying weak classes. In particular, classes such as vegetation (12) and ground (10) show lower performance, due to high morphological variability, spatial overlap with other classes and reduced representation in the training set. Conversely, classes characterised by regular and well-defined geometries, such as piers (20) and main beams (22), exhibit high accuracy levels.

Validation was carried out on a dataset independent from the training one. The validation cloud, suitably translated and rotated, was subjected to the same feature-extraction procedure (18 descriptors per point) and classified through block-wise prediction, in order to limit RAM usage. The final recomposition yielded a cloud of more than 10 million labelled points.

To support the analysis, bar charts of class distribution, summary tables and confusion-matrix heat maps were generated via matplotlib. These tools highlighted specific criticalities, such as the partial confusion between class 16 (abutments/barriers) and class 20 (piers), or between secondary beams (26) and main beams (22).

Overall, the adopted validation strategy achieved a high global accuracy (around 85%) and particularly robust performance on the most structurally significant classes. The residual areas of uncertainty were systematically documented and will be discussed in the following chapters, in relation to the transition towards HBIM parametric modelling and FEM analysis. Table 2.1 reports the details of the metrics computed for each class.

Tab. 2.1 — Per-class metrics in the classification.

Class	Precision	Recall	F1-score	Support (n points)
10	10	72,0%	66,0%	68,9%
12	12	68,0%	62,0%	64,9%
16	16	78,0%	74,0%	75,9%

20	20	94,0%	90,0%	92,0%
22	22	92,0%	91,0%	91,5%
26	26	92,0%	91,0%	91,5%
28	28	91,0%	89,0%	90,0%
30	30	85,0%	82,0%	83,5%

Below, the values in the table are commented with reference to the data of the respective classes.

- 10 (ground) is among the weakest classes: a lower recall is observed; part of the true ground is confused with adjacent classes (e.g., pavements/slabs 30 or deck 28) in occluded or irregular zones. It was necessary to increase training support or enrich the local features.
- 12 (vegetation) exhibits high morphological variability and fragmentation, leading to false negatives (recall) and false positives towards nearby vertical surfaces. It was necessary to increase training support or enrich the local features.
- 16 (abutments/barriers) performs well but is penalised by confusion with class 20 at joints/extremities and along edges where normals are unstable. Lower recall indicates portions of true 16 predicted as 20. It was necessary to increase training support or enrich the local features.
- 20 (piers) is a robust class, with many correct predictions on the diagonal and only limited errors with 16 at transitions. Precision is high, with few non-pier surfaces labelled as piers.-
- 22 (main structural class) shows high performance, with slight confusions with 30 (pavements/slabs) or 26 on contiguous elements.

- 26 (main structural class) also shows high performance, with slight confusions with 30 (pavements/slabs) or 22.
- 28 (deck) is compact and continuous, with few residual errors towards 30 in flat areas and towards 10 along the abutments.
- 30 (pavements/slabs) shows good purity but false positives from 22 when normals are unstable or the thickness is limited; recall shows some losses on noisy patches.

The number of true points per class in the validation set corresponds to the sum of each row in the confusion matrix.
Therefore, the main structural classes (piers 20, main beams 22, deck 28) are compact and exhibit high precision/recall, whereas weak or morphologically variable classes (vegetation 12, ground 10) show greater local variability and some confusion near boundaries. The partial confusion between 16 and 20 is localised at extremities and at data discontinuities. Overall, the integrated quantitative and visual analysis confirms the model stability and its transferability; the identified uncertainty areas are revisited in the following chapters to tune parametric modelling and normal-guided meshing.

CHAPTER 3 — WORKING ENVIRONMENTS FOR THREE-DIMENSIONAL MODELLING

3.1 OBJECTIVE AND CONTEXT

In this section we operationally describe the software environments that enable our Scan-to-HBriM/FEM workflow for reinforced-concrete bridges,

clarifying roles, geometric principles and interoperability. The workflow follows two main routes towards the FEM numerical model:

Track A (section-based) classified cloud → references/extracts → HBriM model (CIM/IFC 4.3) → FEM (buildingSMART International, 2022).

Track B (parametric) classified cloud → parametric curves/sections in Grasshopper → direct export to MIDAS Civil NX via MIDAS-GH → FEM.

In parallel, when appropriate, we adopt a "brevi manu" route PC→mesh→FEM, which exploits the direct transformation of the point cloud into a quad/tri mesh to accelerate the generation of the analysis model. In both cases, the competitive advantage comes from semantic classification: work is performed by classes (piers, beams, slab, etc.), reducing data volume and concentrating detail where needed (railway bridge for training and road bridge for validation). Table 3.1 shows a scheme with the roles of the different programs and their respective inputs and outputs.

Tab. 3.1 — Roles and I/O by environment.

Environment	**Main input**	**Output/Exchanges**	**Use in the workflow**
Rhino+ Veesus	PLY/VPC per class; mm units, shift	Curve/Section BREP QuadMesh	Pre-modellazion Track A/B; PC→mesh
Grasshopper	Curves/Sections, parameters	BREP parametric; topology for MIDAS-GH	Track B towards Civil NX
MIDAS-GH	GH definition	Civil NX model (nodes, elements, BC)	Parametric FEM analyses

MIDAS CIM	Curve/Point Library	HBriM (IFC 4.3); STEP	Track A → FEM
Civil NX	IFC/STEP o GH-API	FEM (analysis, post-proc)	Structural outcomes

3.2 RHINOCEROS + VEESUS ARENA4D: FROM POINT CLOUD TO REFERENCE GEOMETRY

For Track A we proceed with the integration between Rhinoceros and Veesus Arena4D, which provides point-cloud management tools inside Rhino. Veesus supports E57/LAS/LAZ formats, automatically handles unit conversions and allows you to set a global shift for high-precision coordinates and to improve computational performance (Veesus Ltd, 2025).
Once the classified cloud has been imported, the semantic classes are represented as Rhino layers and used as a guide for extracting geometric primitives: axes (alignment), cross-sections (deck and piers) and reference planes. The design and modelling phases therefore become interpretable within the CAD environment, aligning survey data and parametric representation.

$$x_{mm} = s\, x_m, \tag{45}$$

with s = 1000

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_0 \tag{46}$$

$$\mathbf{x}' = R\,\mathbf{x} + \mathbf{t} \tag{47}$$

with $.T = \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix}$

The chosen reference system (local) is the one used to discretise bridge sections, manage the assembly and standardise export to the HBriM

environment; an image of the point cloud in Rhino/Veesus is shown in Figure 3.1.

$$\mathbf{C}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u)\, w_i\, \mathbf{P}_i}{\sum_{i=0}^{n} N_{i,p}(u)\, w_i}, u \in [u_0, u_m], \quad (48)$$

$N_{i,p}(u)P_i w_i$ At the geometric modelling stage, NURBS and their B-spline functions are used. B-splines are defined as a linear combination of basis functions , each of which is multiplied by a control point . The parametric curve is given by:

where the control points affect the curve shape and the functions are defined on the knot vector, i.e. a non-decreasing sequence of parameters. The curve is defined in the parametric domain u and its degree p is equal to n (the number of control points) minus 1. The basis functions are defined recursively, with the simplest case being:

A) G^0 continuity ensures that adjacent curves share the endpoint. There are no constraints on tangency, so the curve may present corners and be non-smooth.

B) Higher-order continuity conditions also impose constraints on derivatives between adjacent curves. For G^2 continuity, the first and second derivatives at the endpoint are the same, ensuring a smooth curve and preventing discontinuities that may cause problems in subsequent phases (e.g., during meshing).

C) The modelling adopted uses continuity between curve segments, which ensures that the curve is smooth and that the transitions between sections are consistent.

Given a curve and a point, the objective is to evaluate the closest point to the curve. If the curve is given in parametric form C(u), the objective is to find the parameter u that minimises the distance between C(u) and the point. The problem can be expressed as a minimisation of the function:

- where:

- denotes the projection onto the profile.

The concept is then extended to multiple profiles, possibly combined by Boolean operations. The Loft command uses a set of profiles and generates a surface that connects them. The Loft parameters allow you to control the continuity between adjacent profiles and thus the smoothness of the resulting surface.

$$E(\mathbf{P}_i) = \sum_j w_j \parallel \mathbf{C}(u_j) - \mathbf{p}_j \parallel^2 + \lambda \sum_i \parallel \Delta^2 \mathbf{P}_i \parallel^2 \quad (49)$$

The objective function used for the optimisation is:

where:

$\mathbf{p}_j$ are the section points;w_j

Δ^2 weights (robustness);

discrete second difference (fairness);

λ regularisation weight.

$$d_j = \parallel \mathbf{p}_j - \pi_{\mathbf{C}}(\mathbf{p}_j) \parallel \quad (50)$$

The optimisation task aims to find, for each profile, the set of control points that minimises the objective while maintaining the desired smoothness. In practice, the method allows reliable cross-sections to be extracted from noisy data, controlling the trade-off between fidelity and regularity.$\pi_{\mathbf{C}}$

Validation of the extracted profiles can be performed by measuring their distance from the point cloud and by checking continuity constraints. This verification is essential to avoid geometric artefacts that may propagate to the final model.

$$\mathbf{S}(u, v) = \mathbf{C}(u) + v\,\mathbf{d} \qquad v \in [0, h] \quad (51)$$

Once the profiles are defined, the surface is generated and the solids are built by closing the B-Rep edges. The solid model can then be exported in CAD formats or used as a basis for HBriM modelling.

In the modelling workflow, the Extrude command is used to generate a surface by translating a profile along a direction vector, while the Loft command is used to create a surface that interpolates a sequence of profiles.

These operations are used to produce the deck surface and the main structural elements.

$$\mathbf{S}(u,v) = \sum_k B_k(v)\,\mathbf{C}_k(u) \tag{52}$$

where:

$\mathbf{C}_k$ direction vector;

B_k profile.

In the following, a set of constraints is applied to keep the profiles consistent and to avoid self-intersections, especially in areas with abrupt changes (e.g., near piers and abutments).

The Sweep command, expressed by the following relation, generates surfaces by sweeping a profile along one or more rails and is used for barriers and kerbs (single rail) or box-sections (double rail):

$$\mathbf{S}(s,t) = \mathbf{r}(s) + \mathbf{F}(s)\,\boldsymbol{\gamma}(t) \tag{53}$$

where:

$\mathbf{r}(s)$ rail (axis);

$\boldsymbol{\gamma}(t)$ profile;

$\mathbf{F}(s)$ orthonormal basis (frame) that maintains G^1 by aligning the profile to the rail tangent.

Once the surfaces are obtained, the edges are closed to generate B-Rep solids. Boolean operations correspond to the classic set operations:

; Intersection: ; Difference:$A \cup BA \cap BA \setminus B$ (54)

Continuity between surfaces is evaluated at the G^0 (position), G^1 (tangency) and (curvature) levels: in practice, G^2 avoids visual discontinuities (kinks) in reflections and is preferable for deck/pier surfaces when it affects the quality of the analysis mesh.

Alongside NURBS modelling, the workflow also makes use of Tri/Quad meshes when this simplifies the preparation of the model for HBriM/IFC and FEM, using remeshing and wrapping tools already available in Rhino (Botsch et al., 2010; Jakob et al., 2015).

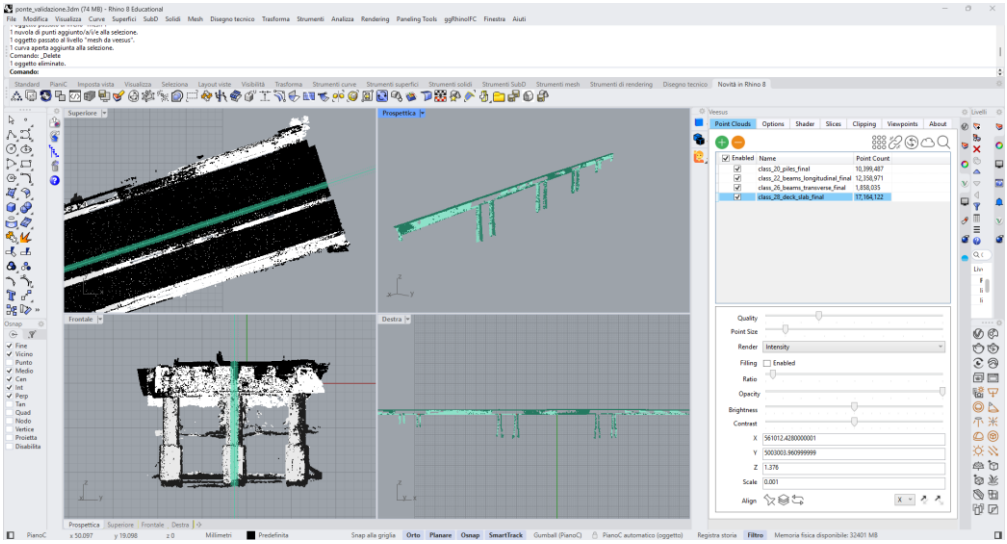


Fig. 3.1 — Rhino + Veesus view of the validation bridge (visible class layers: piles, beams, deck slab).

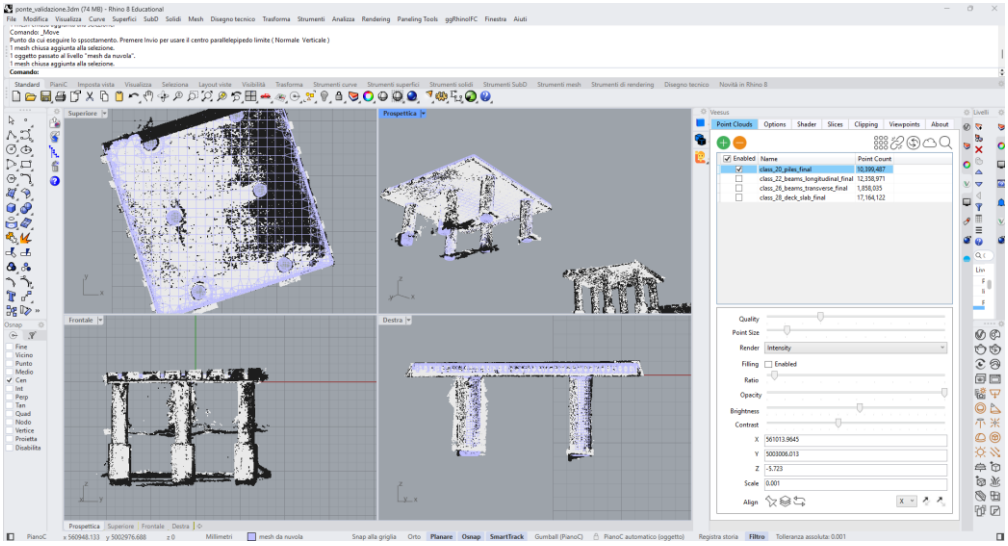


Fig. 3.2 — QuadRemesh/ShrinkWrap on a deck detail.

3.3 GRASSHOPPER: PARAMETRIC MODELLING AND DATA MANAGEMENT

Grasshopper (GH) is the visual programming environment integrated in Rhinoceros, in which geometries are constructed and modified through a

graph of nodes (components) and connections. The key advantage is the direct association between input parameters, intermediate functions and outputs, enabling the rapid generation of many variants and preserving the traceability of the modelling process.

In this project, GH feeds two distinct channels:

- Track B (parametric) towards MIDAS Civil NX, via MIDAS-GH, ensuring continuity between geometry and the structural model.
- Auxiliary channel towards HBriM/IFC or FEM environments, in which the parametric object supports data checking, discretisation and export.

The workflow is organised as a set of clustered components: import and filtering of point-cloud-derived primitives; construction of the parametric axis and sections; generation of surfaces/solids; data-tree management for multiple spans; export blocks.

The management of lists and data trees is central. Unlike traditional scripting, GH allows you to keep a hierarchical structure (branches) to represent spans, classes or groups of sections, and to propagate transformations consistently. This representation is compatible with the semantic approach adopted in the classification stage: each class (deck, pier, barrier, etc.) corresponds to a group of parameters and a set of geometric rules.

- In particular, the definition provides:
- a parametric centreline (axis) extracted from the “deck” class;
- cross-sections defined by control points and updated along the axis;
- local reference frames at discretised stations, used to orient and place the sections.

The GH environment therefore makes it possible to construct the 3D bridge model flexibly and, at the same time, to produce consistent data structures for FEM export.

To support reproducibility, the GH definition is accompanied by a set of input files (cloud, reference polylines, parameter tables) and by the definition of constants (units, tolerances, discretisation step).

$$\mathbf{x}(s,u) = \mathbf{r}(s) + R(s)\,\gamma(u). \tag{55}$$

For the axes, in addition to the geometric centreline, a chainage system is defined (abscissa s along the alignment), which is used to parametrise the variation of sections and to locate accessories and discontinuities (expansion joints, piers, bearings).

Within the definitions, numerical sliders and panels provide interactive control, while more complex configurations (parameter tables) are read from external sources and distributed along the data trees.

The use of custom clusters makes it possible to encapsulate recurring operations (e.g., computing section dimensions, placing stiffeners) and to keep the definition readable and maintainable.

The approach ensures that the modelling is robust with respect to changes: modifying the input alignment or the sampling step automatically regenerates the geometry and the associated export entities, limiting manual rework.

- Piers are modelled through extrusion or lofting of variable sections along the vertical axis;
- The deck is generated through sweep along the road alignment, with variable sections (double T, box, etc.);
- Transverse beams are constructed through parametric patterns constrained to the discretisation of the spans;
- Longitudinal beams with variable cross-section are realised through adaptive lofts specific to each element.

Sections are defined by families of curves $\mathcal{C}_k(u)$ (standard or custom profiles) and by variation functions along the axis $s \in [0, L]$ that control parameters such as web and flange thicknesses, deck depth, fillet radii, and

soffit elevation. In Grasshopper, a variable section is implemented through a weighted combination of reference profiles:

$$\gamma_s(u) = \sum_k \omega_k(s)\,\gamma_k(u), \qquad \sum_k \omega_k(s) = 1. \tag{56}$$

This model allows, within a single span, a continuous transition from a double-T section to a box section (or vice versa), and to manage local widenings or reinforcements (strengthening, stiffeners). When the standard GH components (Extrude/Loft/Sweep) are insufficient, custom clusters were developed to control the weights $\omega_k(s)$ and the variation envelopes, following the approach adopted, for example, for the highly variable longitudinal beam of the training bridge.

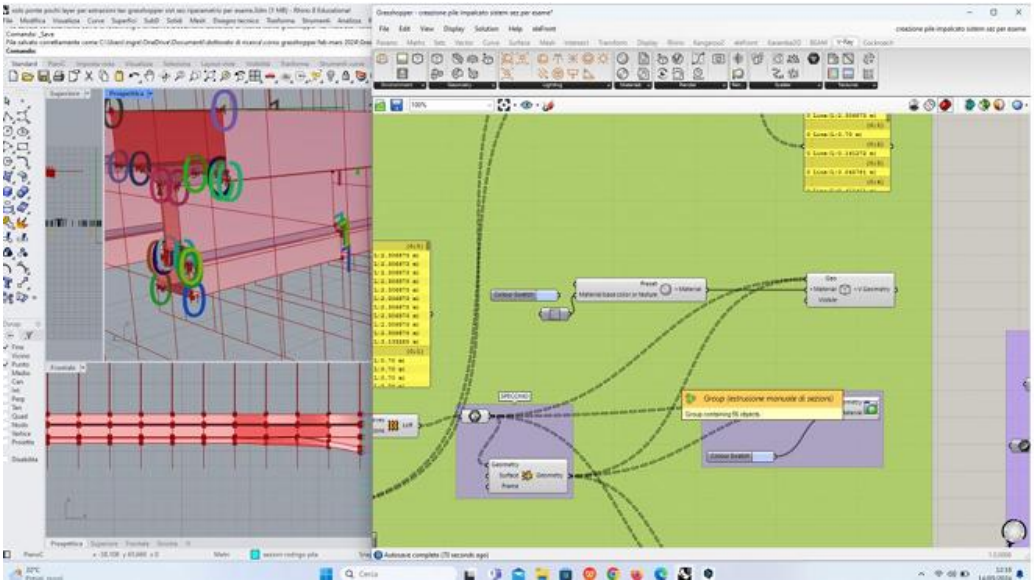


Fig. 3.3 — Grasshopper canvas for the training bridge.

GH definitions are collections of hierarchical lists (data trees). Each branch hosts families of sections, axes and materials; vector operations maintain parallelism between branches (piers, beams, slab), synchronising geometric variations and metadata (materials, class IDs).

3.4 MIDAS-GH (GRASSHOPPER PLUG-IN → MIDAS CIVIL NX)

For Track B, the MIDAS-GH plug-in enables the direct conversion of the Grasshopper parametric model into a FEM model in MIDAS Civil NX. The modelling is based on the definition of structural elements (beam, plate, solid) and on the assignment of properties and boundary conditions. The integration between GH and MIDAS supports an iterative workflow: geometric changes in GH can be propagated to the FEM model without manual re-modelling.θ

$$M(\theta) = T(G(\theta), P(\theta)) \quad (57)$$

MIDAS-GH is structured as a set of components that map GH geometric primitives to MIDAS entities, following an explicit definition of nodes, element connectivity, section properties and materials. The export can also include the creation of load cases and combinations, as well as the definition of analysis settings.

$G(\theta)$The approach is particularly useful for bridges, where the variation of sections and the repetition of spans can be encoded parametrically. The discretisation step defines the density of the FEM model and can be tuned to match computational constraints and the level of detail required.

$P(\theta)$ The conversion process requires special care in managing reference frames, section orientation and the alignment between geometric axis and structural line. For this reason, the workflow includes the generation of local frames along the axis, which are used both to orient the sections and to ensure consistent element connectivity.

$\mathcal{T}(\cdot)$In addition to MIDAS-GH, the project also adopts MIDAS CIM (Computer Integrated Model), an HBriM environment capable of producing an IFC 4.3 compliant model from parametric libraries. CIM is used to build the information model of the bridge, enriched with attributes, class IDs and semantic relationships.

For each configuration , the structural problem solved by MIDAS Civil NX is expressed in the formθ

$$\mathbf{K}(\boldsymbol{\theta})\,\mathbf{u} = \mathbf{f}(\boldsymbol{\theta}), \tag{58}$$

where:

$K(\theta)$ nodal displacements;

u forces;

$f(\theta)$ stiffness matrix;

load vector.

On this basis, the MIDAS model provides the internal forces, displacements and stresses required for subsequent checks and for comparison between Track A and Track B.

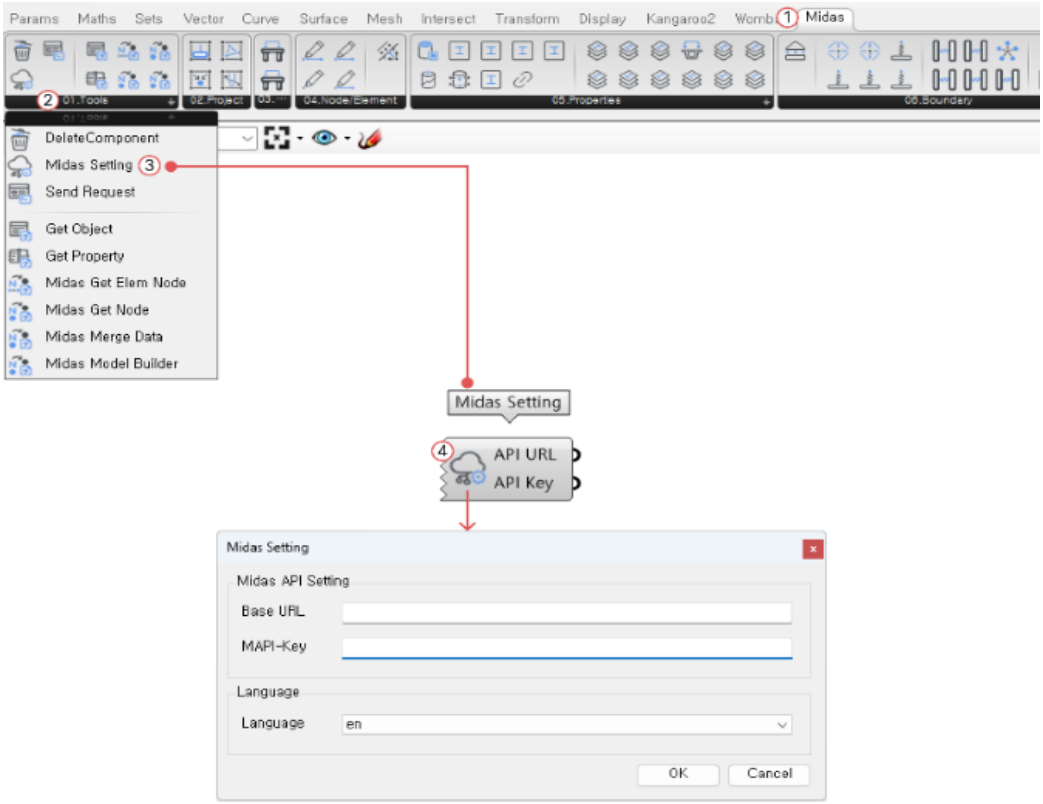


Fig. 3.4 — Setting up the API connection in Grasshopper (Midas Setting).

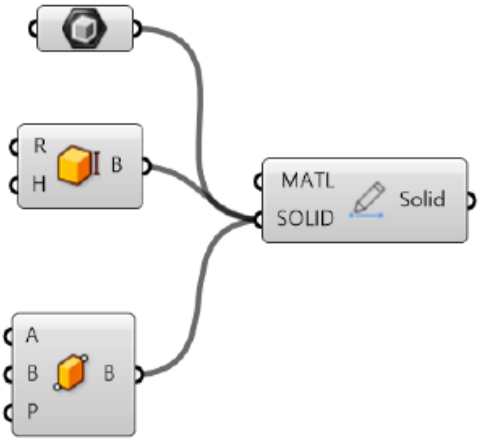


Fig. 3.5 — Examples of “Material/Section/Shape” components and the JSON “previous/next” chain.

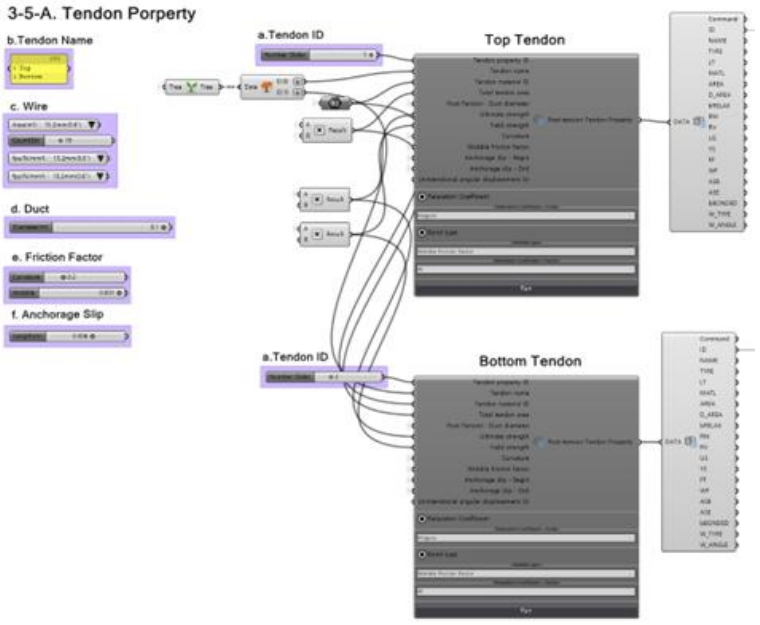


Fig. 3.6 — Grasshopper template for bridges (tendons, profiles, properties).

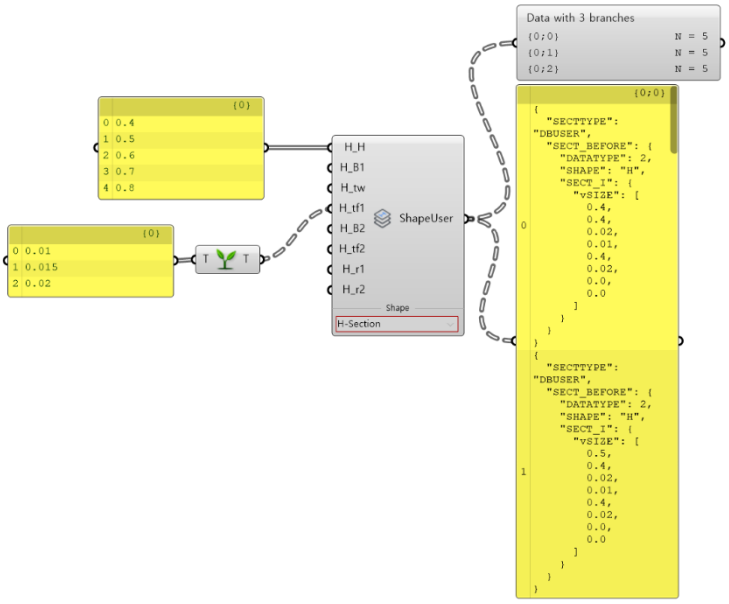


Fig. 3.7 — "Get Property" example with branch-based data handling.

For the generation of the FEM model, MIDAS-GH requires a preliminary discretisation of the geometry into nodes and elements. The definition provides a mapping between GH curves and MIDAS beam elements, and between GH surfaces and MIDAS plate elements, with the possibility of controlling mesh density and element type.

The semantic chain "Shape → Section → Element" is implemented as a composition of functions that generate JSON structures consistent with the MIDAS Civil NX API. The user connects the components according to a "previous/next" logic to define the section (standard or custom) and assign it to the structural elements.

Section properties are defined through parametric templates (rectangular, T, box) or imported from libraries; material properties are assigned consistently with the HBriM model, to maintain correspondence between information and numerical representations.$s \in [0, L]$

$$A(s), I_y(s), I_z(s), J(s) \tag{59}$$

The export process produces a MIDAS file (MCT) or directly opens the model within MIDAS Civil NX, allowing the user to run the analysis and to retrieve results. The iterative loop between GH and MIDAS facilitates sensitivity analyses and optimisation studies.$\theta h(s) t_i(s) b_i(s) \{\text{Node}(s_k), \text{Beam}(s_k)\}$

The Midas Model Builder component enables the creation of geometric and structural entities (nodes, elements, sections, materials, loads and constraints), reducing the typical errors of manual "translation" and guaranteeing atomic and repeatable updates of the model.

With Midas GH (plug-in), the dedicated components (section, material, loads, constraints) map the definitions directly into the Civil NX model, keeping parametrisation active after export; conversely, exporting "baked" geometries towards CIM/IFC or FEA preserves the shape but loses the native parametric links.

MIDAS-GH links Grasshopper and Civil NX, enabling you to create and modify structural analysis models (nodes, elements, constraints, loads) directly from GH and run analyses, connected via API without intermediate steps. This enables Track B (parametric) all the way to FEM and constitutes a key integration point with Track A, which can nevertheless converge into Civil NX via IFC/STEP.

The logic of the two branches (Track A and Track B) was introduced in §3.1 and summarised in Table 3.1; in the following we focus on the integration points with Civil NX and on the implications for FEM discretisation.

The GH definition generates FEM topology (lines/areas/solids) and structural attributes (sections, materials) that Civil NX interprets natively; synchronisation takes place through dedicated API components (connection, geometry push, mesh generation and load cases). With the Midas GH → Civil NX flow, parametrisation remains active and editable in real time (sections/materials/dimensions, real-time updates on the model), whereas exporting to IFC/Parasolid/ACIS yields faithful geometries that are no longer parametrically editable.

MIDAS Civil NX is the infrastructure-oriented FEM environment, capable of handling static analyses, moving loads, construction stages, seismic analysis and mixed beam/plate/shell/solid models. In this work it represents the natural endpoint of Track B and one of the possible endpoints of Track A (via IFC/STEP). The linear static problem is expressed in the form

$$\mathbf{K}\,\mathbf{u} = \mathbf{f} \tag{60}$$

where the stiffness matrix is assembled from beam, plate and shell elements, with boundary conditions that include fixed constraints at the piers and compatibility constraints at the bearings. For modal dynamic analysis one solves:K

$$\mathbf{K}\,\boldsymbol{\phi} = \omega^2\,\mathbf{M}\,\boldsymbol{\phi}, \qquad \mathbf{C} = \alpha\,\mathbf{M} + \beta\,\mathbf{K} \tag{61}$$

where is the mass matrix and the Rayleigh damping matrix.MC

These fundamentals guide the mesh choice: denser in regions with high curvature/connections (pier–deck), coarser on prismatic stretches.

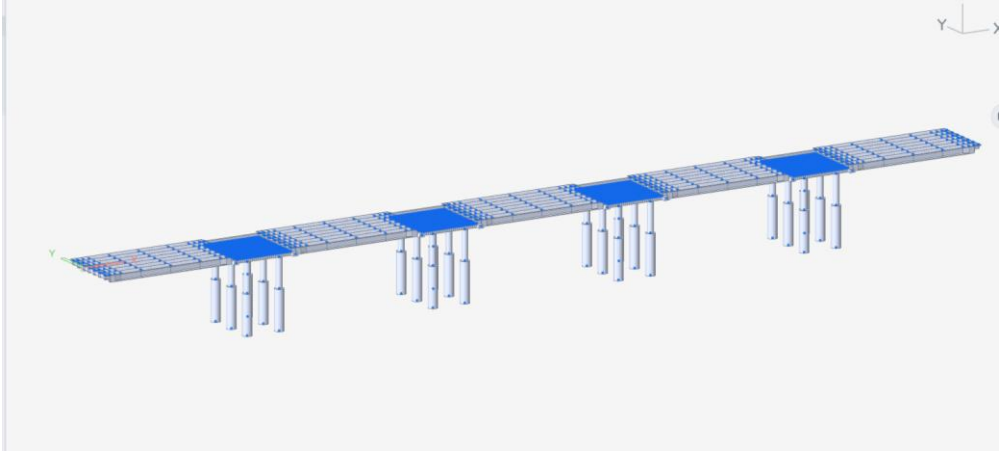


Fig. 3.8 — Civil NX view of the railway bridge with beam/shell elements.

3.5 MIDAS CIM (HBriM): parametric libraries and IFC 4.3

MIDAS Civil NX: FEM solver, linear and non-linear analysis, bridge-specific functions.

- In Table 3.2, the output of the working environment is summarised in terms of the exported formats, destination and QA/interoperability notes.
- The MIDAS CIM environment provides:
- • parametric libraries for common bridge components (piers, decks, barriers, bearings), defined by rules and parameters;

• assignment of attributes and properties (materials, phases, class IDs) to support information modelling;

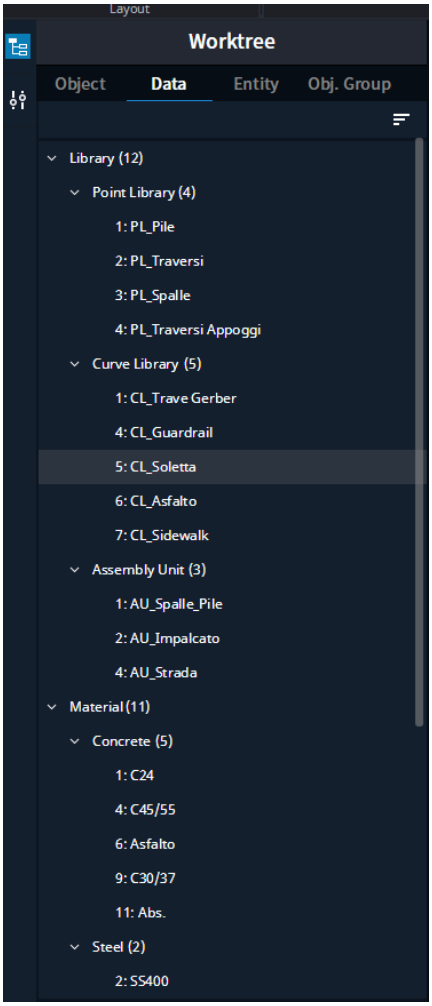


Fig. 3.9 — CIM screen with deck from Curve Library, Point Library and Assembly Unit.

MIDAS Civil NX, instead, is used to:

• import the structural model created via MIDAS-GH or created manually;

• perform analysis and design checks according to the relevant codes;

• manage load cases, combinations, staged construction, and output extraction.$\mathbf{c}(s)\mathbf{T}(s)\mathbf{tnb}\boldsymbol{\gamma}(u) = [\,\xi(u), \eta(u)\,]^{\mathsf{T}}$

$$\mathbf{x}(s,u) = \mathbf{c}(s) + R(s)\,[\,\xi(u)\,\mathbf{n}(s) + \eta(u)\,\mathbf{b}(s)\,] \tag{62}$$

With $R(s)$ rotation matrix from the global coordinate system to $\{\mathbf{t}, \mathbf{n}, \mathbf{b}\}$

A key aspect is ensuring consistency between the information model (HBriM) and the numerical model. The link is established through common identifiers and through the consistent definition of geometry and materials. In this way, results (internal forces, displacements, utilisation ratios) can be traced back to the corresponding HBriM elements.

For interoperability, the IFC 4.3 schema supports infrastructural entities and enables the exchange of alignments, cross-sections and bridge components. The CIM export maps, for example, the deck to IfcBridgePart and the alignment to IfcAlignment, storing geometric information as swept solids or B-Rep representations, and attributing semantics through property sets and classifications.

This approach allows the model to be shared with other BIM platforms and used as a basis for coordination, asset management and future updates.C^1C^2

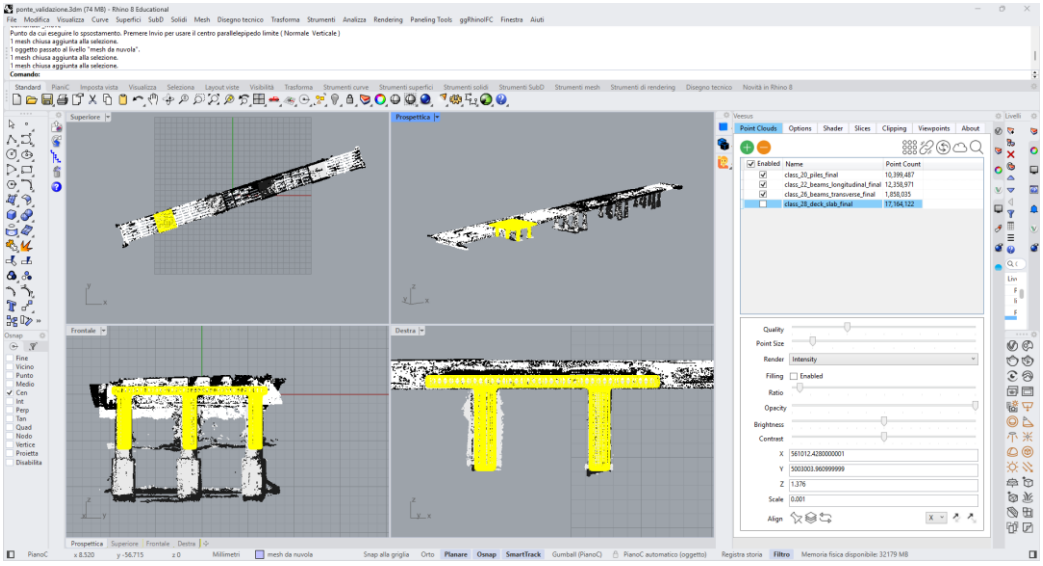
The integration between the environments is therefore essential to guarantee a reliable and reproducible pipeline from survey data to analysis.$s\boldsymbol{\gamma}(u, s)$

The software stack is completed by auxiliary tools (Python, libraries for point cloud processing, mesh processing and visualisation), which support automation and quality control.

The following diagram summarises the conversion pipeline from point cloud to HBriM/FEM, highlighting the role of Rhino/Veesus, Grasshopper and MIDAS environments.

In conclusion, Track A is based on the creation of an information model (CIM/IFC) driven by reference geometry extracted from the cloud, whereas Track B emphasises parametric modelling and direct export to FEM; the availability of both tracks allows you to choose the most appropriate solution depending on the required level of detail, time constraints and interoperability needs.

Table 3.3 summarises the minimal export package for the two tracks, including the geometric model, the classified point cloud with quality fields, and a manifest file documenting parameters, software versions and applied transformations.



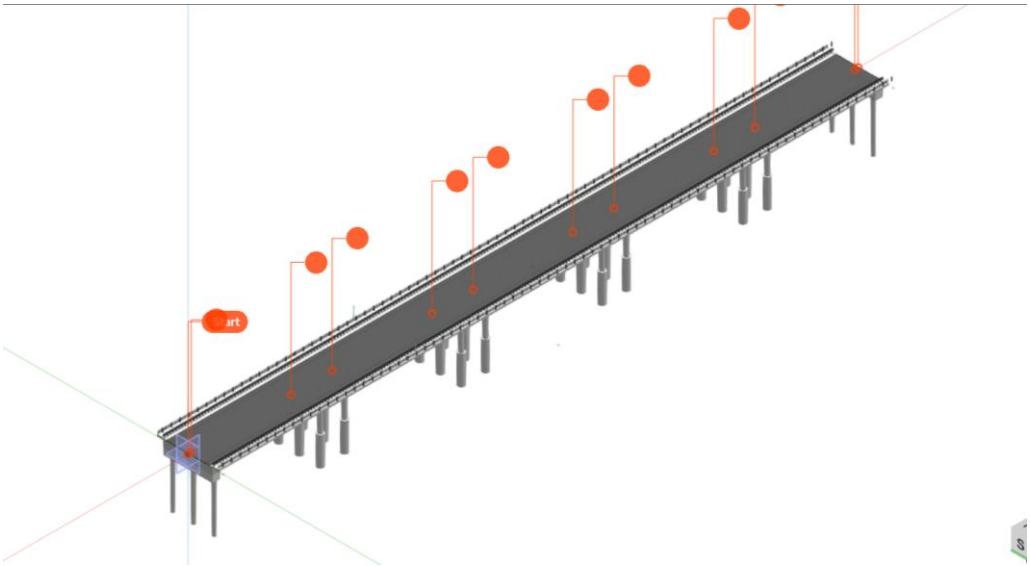


Fig. 3.10 — Semantic mapping and exchanges: classes → Rhino layers → IFC 4.3.

3.6 USING TRIANGULAR MESHES AS A SHORTCUT FOR FEM

When objects have very complex shapes (e.g., arches), Rhino meshing commands can be used to convert surfaces and solids into triangular or quadrangular meshes, which are more suitable for import into an FEM solver. In particular, the TriMesh and QuadMesh commands generate a mesh from a NURBS surface, controlling density through parameters such as chord height and maximum edge length. Triangular meshes are more robust for complex geometries, whereas quad meshes are preferable when a more uniform discretisation is desired. In this way, modelling time is reduced and it is possible to move more quickly to the analysis stage, without necessarily reconstructing the entire geometry as NURBS. Although this approach is less precise in terms of parametric control, it is useful for preliminary analyses and rapid checks.

In CloudCompare, a mesh can also be generated directly from the cloud (Poisson reconstruction, Delaunay triangulation). In Rhino, import and meshing provide additional control and allow manual correction.
The mesh representation is suitable for export to MIDAS Civil NX and other FEM solvers, but it does not preserve the parametric structure of the model; for this reason it is used as a shortcut in Track A only when reference modelling would be too time-consuming.
Mesh simplification algorithms (decimation) can reduce the element count while maintaining the shape; the key is to preserve geometric features that are relevant to structural behaviour.
Mesh quality can be assessed through metrics such as aspect ratio and skewness, and through distance comparisons with the original cloud.

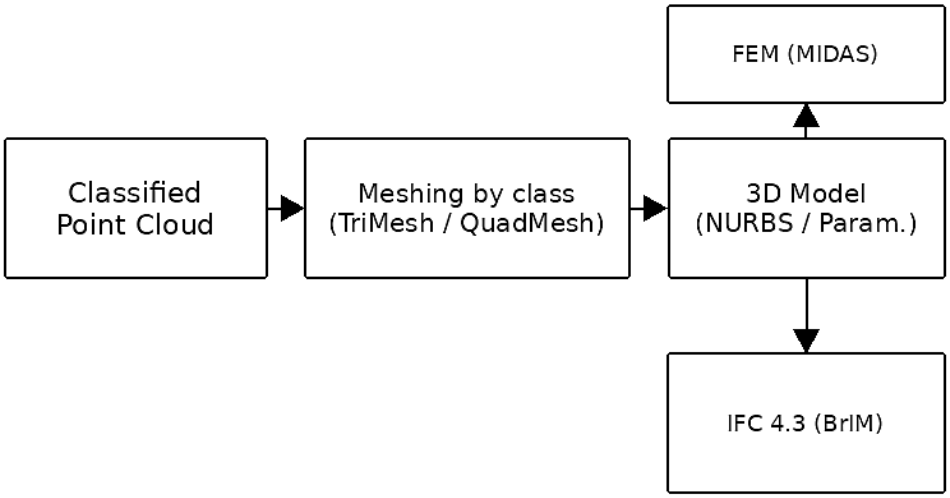


Fig. 3.11 — PC→mesh→IFC/FEM pipeline.

The use of meshes in MIDAS Civil NX typically occurs by importing formats such as STL or OBJ, and the mesh is interpreted as a set of shell elements. It is necessary to check mesh quality (aspect ratio, inverted normals, holes) to avoid numerical issues; if needed, the mesh is cleaned and remeshed in Rhino

or dedicated tools (e.g., MeshLab). In table 3.2 you can see the summary of the workflow.

Tab. 3.2 — PC→mesh→IFC/FEM pipeline summary.

Step	Input	Main operations	Step output	Downstream use
1. Point Cloud (classified)	E57/LAS/LAZ/PLY per class (piers, beams, deck slab, barriers, …)	Import, mm units, global shift, per-class layers	Point cloud ready for meshing, ordered by class	Feed Step 2
2. Meshing by class (Tri/Quad)	PC per class	TriMesh or QuadMesh with per-class parameters (density, smoothing, closures)	Consistent meshes for each class	Feed Step 3
3. 3D model (NURBS/Param.)	Mesh per class	NURBS/Solids (Loft/Sweep) o Parametric definitions (GH)	Consistent 3D model (as-built/parametric)	Final Branch
4A. IFC 4.3 (BrIM)	3D model	Class→IfcEntity mapping (IfcColumn/Beam/Slab/Barrier/…) + Psets	File **IFC 4.3**	HBriM coordination/archive
4B. FEM (MIDAS)	3D model	Export to MIDAS (beam/plate/shell/brick, materials/sections)	FEM model	Structural analysis

In conclusion, the mesh-based shortcut represents a pragmatic compromise between modelling accuracy and time, and it can be used to complement the two main tracks described above.

3.7 CLOUDCOMPARE: PROOF OF CONCEPT FOR HIGH-PRECISION SECTION EXTRACTION

CloudCompare is an open-source software for point-cloud processing. In this work it is used as a proof of concept to test high-precision section extraction and to compute quantitative quality indices. The operations include filtering, slicing, mesh generation and distance computation.

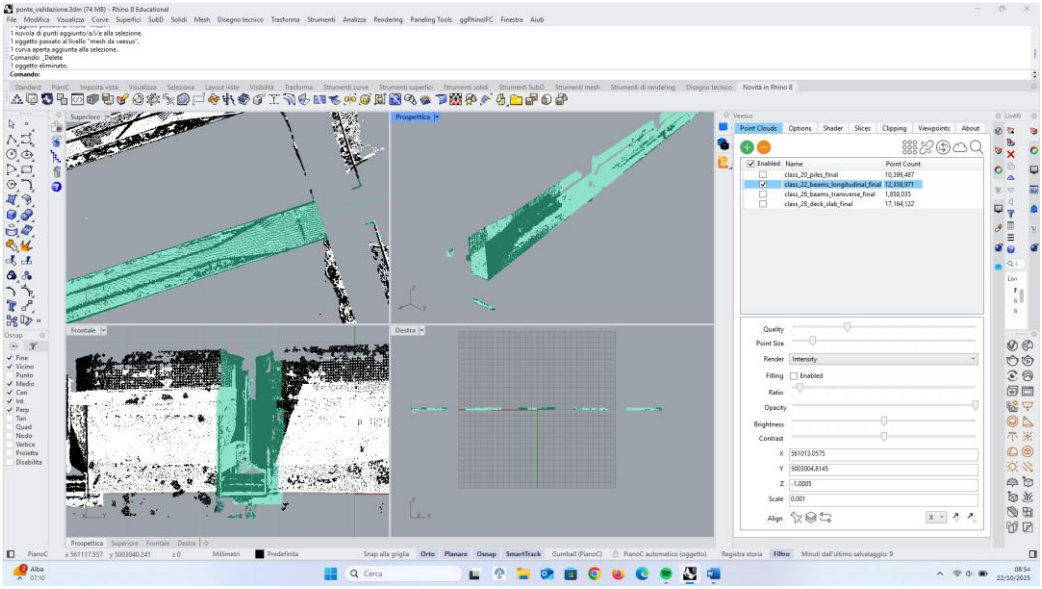


Fig. 3.12 — Example point-cloud portion of a secondary beam.

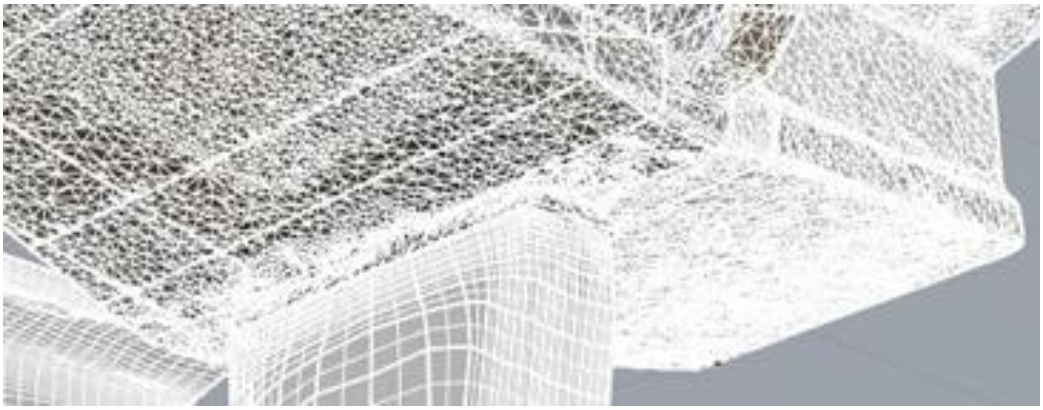


Fig. 3.13 — Mesh with different parameters.

For section extraction, the cloud is sliced with a plane and a thickness τ, obtaining the points within the slab. The resulting set is projected onto the plane and used to fit a polyline/profile.

In CloudCompare, the process is implemented through the “Cross Section” tool, which allows you to define the plane, thickness and step along a direction, producing a series of slices along the alignment.

The tool provides, for each slice, the resulting points and an exported polyline (2D) that can be used as a reference section. The advantage is the ability to process large clouds and to quickly obtain sections at regular spacing.

$$E_{\text{rms}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d(p_i, \mathcal{M})^2}\,, d(p, \mathcal{M}) = \min_{q \in \mathcal{M}} \| p - q \|_2. \qquad (63)$$

However, the method is sensitive to local noise and to point density; for this reason, it is integrated with robust fitting and smoothing techniques, and with semantic filtering by class.

The key indicator introduced is the Inlier Ratio IR_τ, defined as the fraction of points within a threshold distance τ from the reference geometry. The indicator is evaluated for each class and each cross section, allowing problematic areas to be identified (outliers, misclassifications, insufficient density).

Tab. 3.3 — Meshing requirements for structural classes.

Class (ID)	Downstream use (IFC / FEM)	Recommended mesh type	ℓ_base [m]	ℓ_min [m] (critical areas)	Smoothing (iter / λ)	Local closures	QA (target minimum)
Piers (20)	IfcColumn FEM shell/brick	Tri (curvature-aware)	0.05–0.08	0.03–0.05	1–2 / 0.4–0.6	Yes (holes < 2·ℓ_min)	p95 ≤ 0.05 IRτ ≥ 0.90 F1 ≥ 0.90
Longitudinal girders (28)	IfcBeam FEM beam/shell	Quad (oriented along axis)	0.06–0.10	0.04–0.06	1 / 0.4	Only minute defects	p95 ≤ 0.05 IRτ ≥ 0.90 F1 ≥ 0.86
Transverse beams (26)	IfcBeam FEM beam/shell	Quad or Tri at nodes	0.06–0.10	0.04–0.06	1–2 / 0.4–0.5	Same as above	p95 ≤ 0.05 IRτ ≥ 0.90 F1 ≥ 0.85
Deck slab / Deck (22)	IfcSlab FEM plate/shell	Regular Quad	0.08–0.12	0.05–0.08	1 / 0.3–0.4	No (prefer explicit gaps)	p95 ≤ 0.05 IRτ ≥ 0.90 F1 ≥ 0.86

In the workflow presented in Chapter 5, IR_τ is used both in the training phase (railway bridge) and in the validation phase (road bridge). We fixed τ = 20 mm; unit conversion to metres is performed by applying a factor for

1000. The index is paired with the C2C distance and other percentiles for a comprehensive evaluation

To determine the slicing step , we choose it proportional to the local radius of curvature $\rho = 1/\kappa$ of the alignment, with the step defined as:

$$l_{target} \leq \min\left(l_{base}, \rho/k\right), \tag{64}$$

where:

l_{target} is the radius of curvature;

l_{base} is a user coefficient that controls resolution.

ρThe step defines the number of sections and thus the computational cost: smaller steps improve fidelity but increase processing time.

kThe parameter is tuned to guarantee sufficient coverage in areas with high curvature, while avoiding excessive oversampling on straight segments.

In any case, the value of cannot fall below the minimum required by IFC/structural discretisation constraints, so as not to lose relevant geometric features.$l_{target} l_{min}$

Tab. 3.4 — Meshing requirements for non-structural classes / context.

Class (ID)	Downstream use (IFC / FEM)	Recommended mesh type	ℓ_base [m]	ℓ_min [m]	Smoothing	Closures	QA (target minimum)
Barriers / Abutments (16)	IfcBarrier / IfcKerb equivalent loads in FEM	Tri	0.08–0.12	0.06–0.08	1–2 / 0.5	Yes (micro-gap)	p95 ≤ 0.06 IRτ ≥ 0.85 F1 ≥ 0.80
Flooring (30)	IfcCovering Contact surfaces	Tri (reduced)	0.12–0.20	0.10–0.12	1 / 0.5	No	p95 ≤ 0.07 IRτ ≥ 0.85 F1 ≥ 0.80
Ground (10)	Context / masks	Coarse Tri	0.20–0.40	0.15–0.20	1 / 0.6	Yes	p95 ≤ 0.10 IRτ ≥ 0.80 F1 n.d.
Vegetation (12)	Excluded or masks	— (avoid)	—	—	—	—	F1 ≥ 0.75 (if any) No p95 constraint

The following chapter discusses the results of this approach on the two case-study bridges, highlighting benefits and limitations.

For barriers and abutments, however, it is essential to preserve a clean and topologically coherent geometry, as these elements are relevant for clearance and interference checks. For these classes, less stringent tolerances than for primary structural elements can be adopted, while still maintaining an accuracy level adequate for coordination and verification purposes.

3.8 TRIANGULAR MESHES (TRIMESH) AND QUAD MESHES (QUADMESH): PRINCIPLES, ADVANTAGES, LIMITATIONS

TriMesh can be obtained through Delaunay/advancing-front triangulations or reconstructions from point clouds (e.g., ball-pivoting). It is robust on complex topologies, tolerates local holes and adapts well to rapid curvature changes and detailed geometries (Figure 3.13) (Bernardini et al., 1999; Botsch et al., 2010).

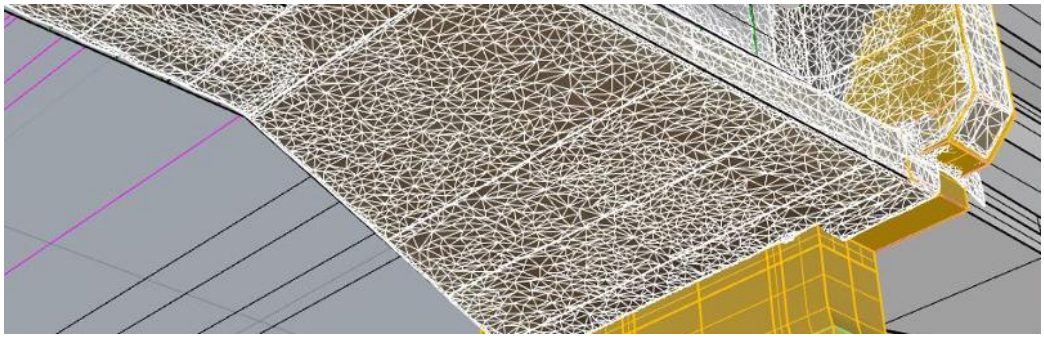


Fig. 3.13 — Triangular mesh of a variable-section beam.

Among the main advantages of a triangular mesh are its natural adaptivity to the curvature gradient and the possibility of obtaining an anisotropic tessellation through advancing-front algorithms. These characteristics make it particularly effective for discretising geometrically complex surfaces.
TriMesh is ideal for representing connections and discontinuity zones (corbels, bearing seats, stiffeners); it is also the most natural choice when the

surface is reconstructed directly from a point cloud or through shrink-wrapping operations around geometries or point sets.

From a qualitative standpoint, a well-conditioned TriMesh is characterised by a high minimum angle $\theta_{\min}$ and a shape ratio $\alpha = \frac{\ell_{\max}}{\ell_{\min}}$ close to one. In FEM this helps to reduce numerical artefacts associated with quadrilateral skewness, which are particularly critical near sharp corners.

Triangular meshing nonetheless has some limitations: among these, the presence of topological irregularities (non-uniform nodal valence), a lower ease of parametrisation and a less direct conversion into NURBS surfaces. Moreover, for nearly planar or laminar surfaces it may require more elements than an equivalent QuadMesh to reach the same geometric accuracy.

For resolution and quality assessment of triangulated surfaces, the following indicators are relevant. The area of a triangle is defined as:

$$A_{\Delta} = 0.5 \cdot \| (p_1 - p_0) \times (p_2 - p_0) \| \tag{65}$$

The aspect ratio of a triangle in the TriMesh case can be expressed as:

$$AR_{\Delta} = \frac{\ell_{max}}{2 \cdot r_{in}} \to 1 \tag{66}$$

where denotes the radius of the inscribed circle.r_{in}

The nodal valence deviation, for triangular meshes, is finally evaluated as:

$$\Delta v = |v - 6| \tag{67}$$

QuadMesh, instead, derives from remeshing processes aligned with the principal curvature directions and tends towards a nodal valence equal to 4. In Rhino, the typical workflow involves quad-remesh tools with the ability to generate regular SubD and subsequently convert to NURBS surfaces (Fig. 3.14).

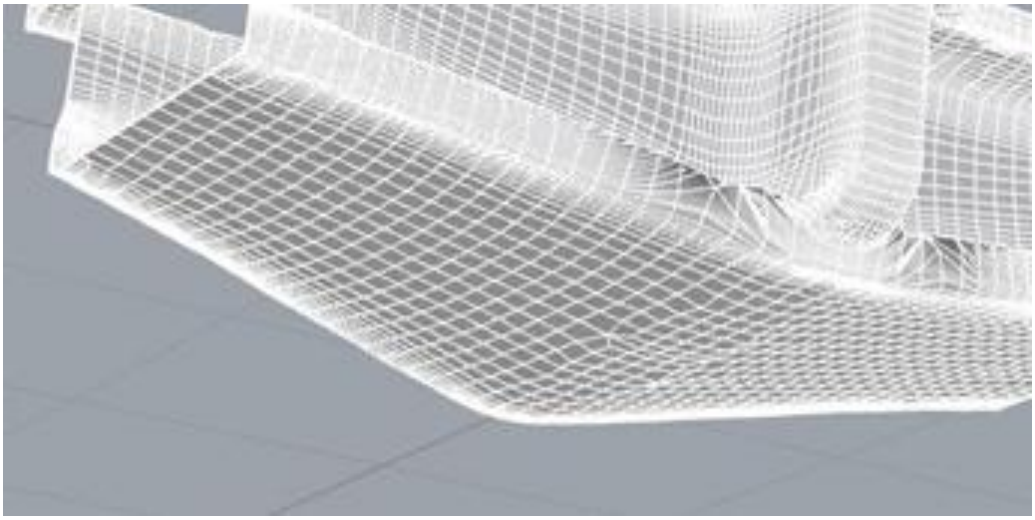


Fig. 3.14 — Quad mesh of a variable-section beam.

The main advantages of QuadMesh lie in its topological regularity, characterised by an almost orthogonal grid, particularly suited to parametric modelling and conversion to NURBS surfaces. It also provides a natural UV mapping, useful for offset operations, thickness definition and technical texturing. On planar or gently curved surfaces, QuadMesh can reach high geometric accuracy with fewer elements than an equivalent TriMesh.

QuadMesh tends to align edge loops with the principal curvature directions and is therefore preferable when a transition to SubD/NURBS is expected, or for FEM use with quadrangular shell elements. In Rhino, the typical workflow is based on the QuadRemesh command, which allows the generation of quads with optimised topology, support for guide curves, symmetry constraints, hard-edge management and direct SubD output, subsequently convertible to NURBS.

The limitations of QuadMesh emerge in the presence of high curvature, sharp details or complex geometric transitions, which require the introduction of extraordinary vertices (valence different from 4) or the management of areas with very small elements. Moreover, compared with TriMesh, QuadMesh is more sensitive to skewness and loss of orthogonality: highly distorted

quadrilaterals cause a rapid degradation of geometric accuracy and numerical performance.

Quadrilateral quality can be expressed through the skewness index:

$$Q_{sq} = 1 - \frac{max_k\ |\theta_k - 90°|}{90°} \to 1 \tag{68}$$

where represents the internal angles of the quadrilateral.θ_k

The average discretisation step for a QuadMesh is defined as:

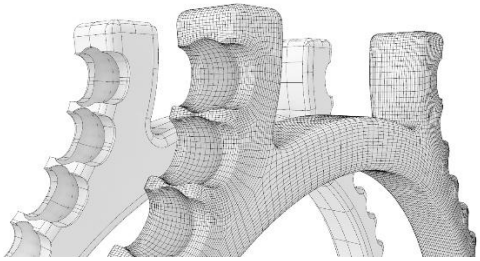
$$\hbar = \sqrt{\frac{A_{tot}}{N_e}}\ (QuadMesh) \tag{69}$$

where is the total area of the surface and the number of elements.$A_{\text{tot}} N_e$

The choice between TriMesh and QuadMesh depends on downstream use (HBriM/IFC and FEM) and is therefore defined by semantic class; the principles and limitations of the two tessellations are discussed in §3.8, whereas here we only report the operational settings in Rhino.

For the railway and motorway case studies analysed, a per-class strategy was adopted: main structural elements (piers, straight beams with predominantly linear geometry, slab) favour the use of QuadMesh, with explicit control of skewness and aspect ratio; for pier caps and variable-section beams, TriMesh is instead used; minor or contextual elements (barriers, abutments, ground and vegetation) are represented through light TriMesh, sufficient to describe their outline without overloading the overall discretisation.

Figure 3.15 provides an illustrative scheme of the two mesh types and their respective application domains.



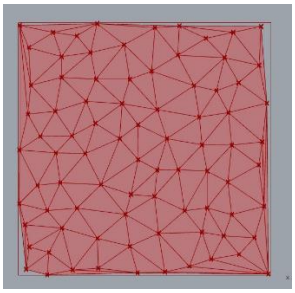


Fig. 3.15 — Irregular TriMesh vs regular QuadMesh.

Pier shafts show variations in section and portions with cylindrical and/or conical geometry. For these elements a mixed discretisation strategy was adopted, with a prevalent QuadMesh along the shaft, to guarantee topological regularity and facilitate subsequent conversion to SubD/NURBS, and local TriMesh at the head and base, where high curvature and connections introduce geometric irregularities.

The target mesh step lies in the 20–40 mm range, with control of aspect ratio and quadrilateral squareness in the regular portions. This combination meets both the geometric-adherence requirements imposed by the quality metrics (C2C, IR) and the quality needs of the future FEM discretisation, whether with shell or brick elements.

All Quad–TriMesh modelling was carried out in Rhinoceros with the goal of obtaining a solid and topologically coherent geometry. Export towards the FEM environment takes place in Parasolid and STEP formats; the imported geometry is subsequently meshed in a structured manner inside the analysis software. The first meshing phase in the modelling environment therefore serves to prepare and qualify the model for subsequent FEM discretisation, and is indispensable to obtain a geometrically correct and sufficiently accurate solid.

Structural elements are ultimately meshed in the FEM software as a function of their structural hierarchy, static/seismic role and the level of detail required by the analysis.

Beams have webs and flanges characterised by gentle curvature and a preferential development along the longitudinal axis. For these elements, a dominant QuadMesh was adopted, aligned to ribs and principal directions through guide curves, with local refinements at connections (beam–slab nodes, diaphragms).

The target discretisation step allows skewness values to be kept low and the aspect ratio to be controlled within ranges compatible with FEM modelling

(typically AR ≤ 5–7), while preserving the transcribability of the elements into beam and shell entities in the analysis environment, as shown in Figure 3.16.

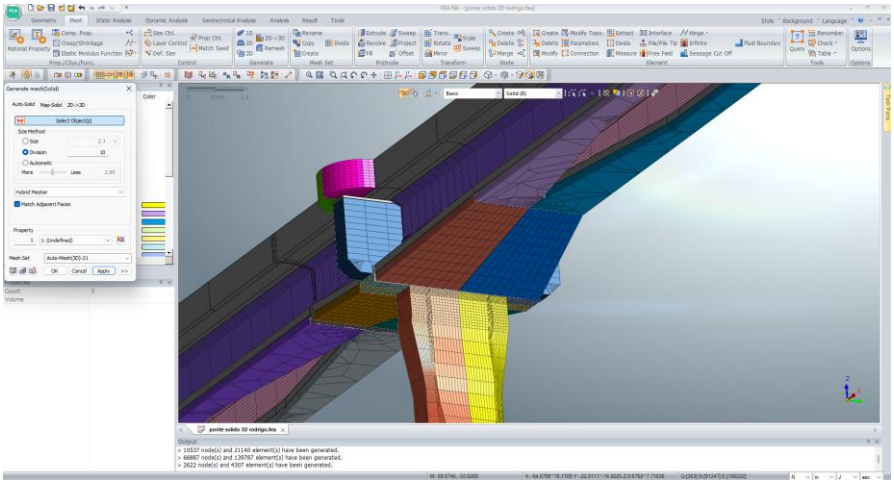


Fig. 3.16 — Structured meshes in Midas FEA.

For the slab/deck (IfcSlab), being a laminar surface with modest curvature, the natural choice is a regular QuadMesh (Figure 3.17) with low skewness and an almost orthogonal grid, to facilitate both NURBS generation (via SubD→ToNURBS) and the extraction of plate/shell elements with uniform thickness. The target step is h = 25–50 mm, with possible densification near joints or supports.

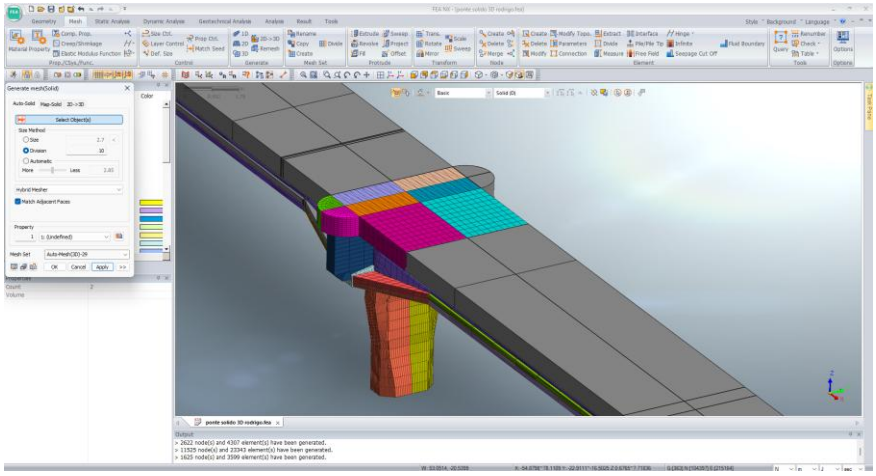


Fig. 3.17 — QuadMesh applied to the structural slab.

For small geometries, sometimes noisy or with non-structural details, a regular QuadMesh with moderate low-pass smoothing is preferred, as in Figure 3.18, to remove micro-asperities without losing the global shape. This representation is sufficient for interference checks and for potential mapping into IfcBarrier/IfcKerb (HBriM).

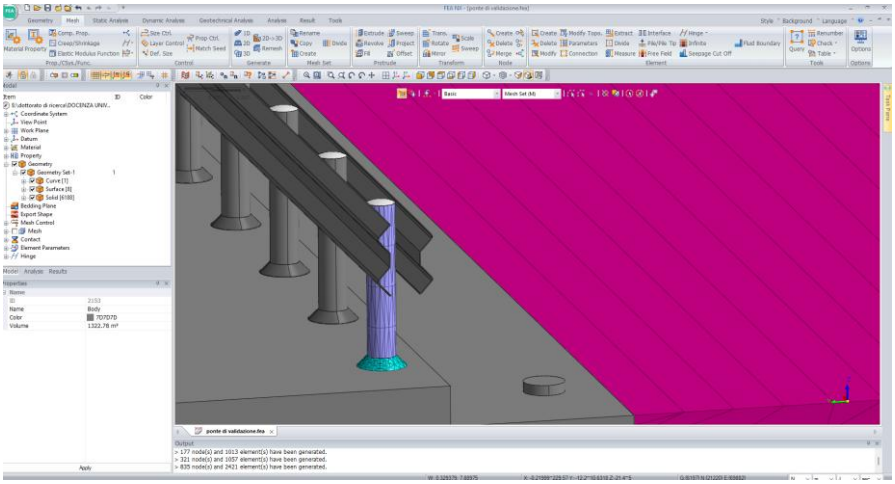


Fig. 3.18 — Regular QuadMesh of a vertical guardrail element.

Ground and vegetation classes do not contribute to structural response and, unless clipping or geometric contextualisation is required, they can be excluded from the calculation model. If they must be kept, a coarse TriMesh is in any case sufficient, adequate for outlining only, without significant impact on computational cost.

Operationally, the recommended presets for structural classes favour QuadRemesh, with edge length differentiated by zone: finer near supports, joints and stiffeners, and coarser in span portions. The use of guide curves on functional edges and the detection of hard edges allow relevant borders and geometric discontinuities to be preserved correctly.

For abutments and barriers, controlled density-reduction strategies can be adopted, accompanied by local repairs where necessary. Surface-closing operations (ShrinkWrap) should be limited to small holes and performed with zero offset, to avoid undesired alterations of the geometry.
In the final phase, the numerical ranges of discretisation parameters must be calibrated according to bridge scale and the quality thresholds defined in the following chapters, verifying indicators such as squareness, aspect ratio and mesh skewness a posteriori. A concise comparison between TriMesh and QuadMesh is reported in Table 3.5.

Tab. 3.5 — Tri vs Quad (pros/cons and applications).

Aspect	TriMesh (Triangles)	QuadMesh (Quadrilaterals)
Topology	Irregular (variable valence)	Almost regular (valence ≈ 4)
Adherence to details	Excellent on curvature and edges	Good on regular surfaces
Parameterisation/NURBS	Harder	Natural (UV/regular loops)
FEM local (connections)	Robust	Requires management skewness
Weight at equal error	Maggiore su superf. piane	Minor on surface. Desserts
Recommended use	Variable piers, connections, details	Deck slab, regular webs, panels

3.9 IMPLEMENTATION IN RHINOCEROS: COMMANDS AND TOOLS

The operational logic can be reduced to five steps (metric preparation, remeshing, local refinements, QA and export), consistent with the general scheme already discussed; below we recall only the most relevant aspects for implementation in Rhino.

1. metric preparation (units in mm, global shift, tolerances);
2. remeshing with geometric constraints;
3. local refinements;
4. quality control (C2C p95, , element metrics);d_H

5. export.

The first step consists in generating a QuadMesh aligned with the principal curvature directions by means of QuadRemesh / SubD. For each class, a target edge length is set, consistent with the adopted density law, integrating—when necessary—guide curves (axes, functional edges, extrados/intrados borders) and symmetry constraints. This approach supports both SubD→NURBS conversion for IFC export and the generation of FEM shells with limited skew and aspect ratio. If remeshing proves excessively rigid in noisy zones or in areas with strong blends, local TriMesh (hybrid strategy) or light preliminary smoothing is used.h

The main control parameters are:

- Target edge length , finer near supports, joints and ends, coarser in span;h
- Guide curves, to orient the tessellation and reduce manual refinements;
- Detect hard edges / preserve creases, to retain sharp edges and functional discontinuities;
- Adaptive quad count, checking the average step $\hat{h} \simeq \sqrt{A_{\text{tot}}/N_e}$ to avoid under-sampling.

A successful procedure produces regular quads (squareness $Q_{sq} > 0.7h$), aspect ratio typically ≤ 5–7 and moderate skew, with RMSE and p95 within the class targets. Locally distorted quads indicate the need to reduce the step or strengthen the guides.

Once a coherent tessellation is obtained, density can be reduced in non-critical areas (barriers, span portions far from joints), preserving edges and average curvature. Decimation, via ReduceMesh / Remesh, is driven by error thresholds (RMSE or C2C p95), ensuring that the d_H local remains compatible with IFC and FEM uses.

In the presence of small gaps or local holes, ShrinkWrap with zero offset can be used, limiting its use to local closures and setting densities consistent with the per-class steps. Options such as Fill Holes and Smoothing must be used

with caution so as not to remove informative discontinuities (joints, gaps). “Repaired” surfaces should be tracked through dedicated layers or attributes. Functional edges (beam webs, deck borders, thickness changes) must always be protected with PreserveEdges / Crease during remeshing, decimation and smoothing, to avoid geometric and numerical artefacts in the FEM.

Before export, it is essential to guarantee edge continuity and normal consistency (Weld, UnifyNormals), checking orientation through visual inspection and tests in the FEM pre-processor. Overlaying the mesh and the reference geometries enables a final verification of the correctness of the discretisation (Fig. 3.19).

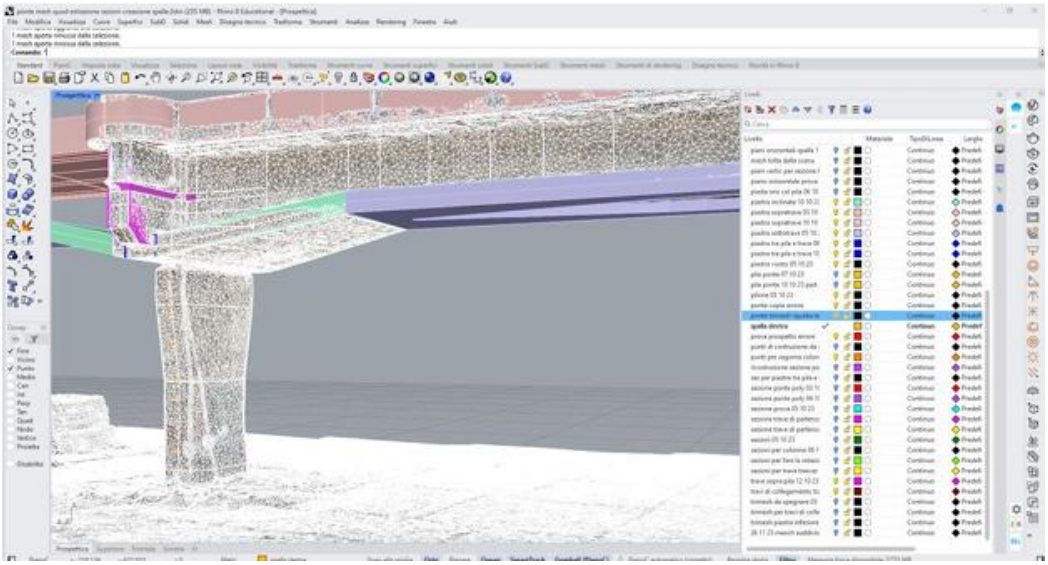


Fig. 3.19 — Correspondence between mesh patterning and geometry.

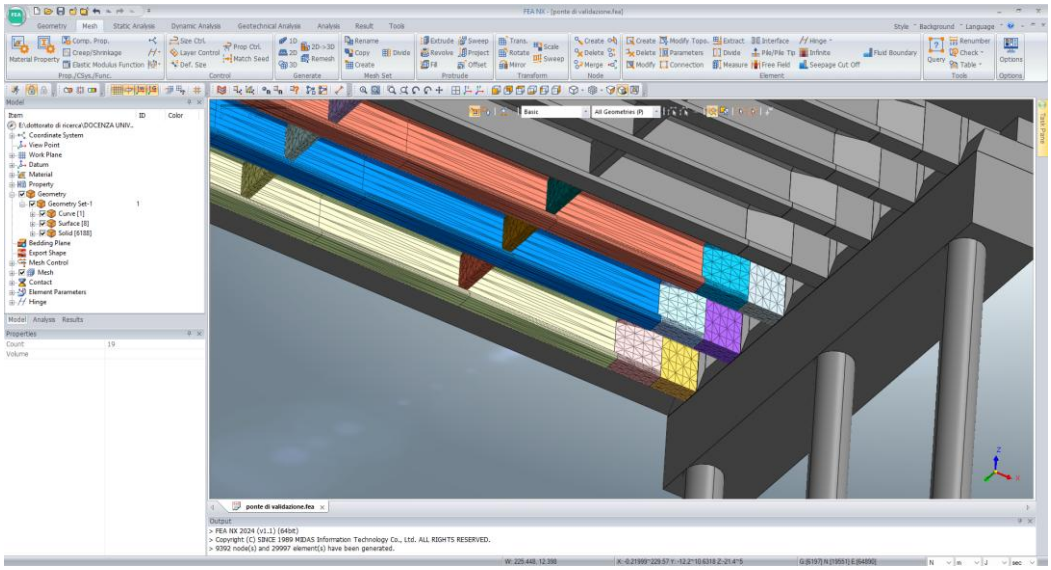


Fig. 3.20 — Transition between TriMesh and QuadMesh in the joint area where the secondary beam connects to the Trave-Pulvino.

For Quality Assurance (QA), we report: the 95th percentile (p95) of point-to-model distances, the computed on critical windows, the total number of elements, the distributions of aspect ratio and skewness, as well as any rework notes.d_H

Systematic application of this practice to structural classes allows and p95 to be kept within the targets defined in Chapter 5, reducing weight and complexity where unnecessary and guaranteeing process interoperability: a single workflow capable of producing both geometrically clean IFC objects and numerically stable FEM models.E_{rms}

Tab. 3.6 — Key Rhino commands and settings: operational summary.

Command	Purpose	Relevant Parameters/Options	Usage notes in our workflow
Mesh	Generates a mesh from surfaces/polysurfaces/SubD (for	Detailed controls globali: Maximum angle, Maximum/Minimum edge length, Maximum distance	Useful when we reconstruct “clean” surfaces (e.g., the

	rendering/analysis or editable).	edge-to-surface, Aspect ratio, Refine mesh, Jagged seams.	deck slab) and want their polygonal equivalent with controlled density.
QuadRemesh	Retopologia in quad da NURBS/mesh/SubD; opz. Exit SubD.	Target Quad Count o Target Edge Length; Guide curves con Curve influence (None/Approx/Edge-Ring/Edge-Loop); Symmetry; Detect hard edges; Preserve mesh boundary.	For beams/deck slab/piers: use guide curves along axes and edges, bridge symmetry, and hard edges on sharp corners.
ShrinkWrap (Mesh Wrap)	Creates a closed "wrap-around" mesh around NURBS/SubD/mesh/point cloud.	Target edge length/Polygon count, Offset, Detect and fill holes, Smoothing.	Excellent for "closing" point clouds or imperfect geometries and obtaining shells that can be printed/analysed; native in Rhino 8.
ReduceMesh	Reduces polygons while minimizing distortion.	Target face percentage or count (triangulate → reduce → re-quadrangulate).	Selectively reduce non-critical meshes (barriers/pavement), preserving density at structural nodes.
TriangulateMesh / QuadrangulateMesh	Strength tri o Try merge in quad.	Rectangularity test and Diagonal constraints for QuadrangulateMesh.	Useful for smoothing out the mesh type on specific components.
MeshRepair / UnifyMeshNormals / FillMeshHoles	Repairs normals, welds edges, closes holes.	Interactive panel; Edge analysis, Advanced repair tools.	Obligatory step before export STL/STEP/IFC o prima di FEM.
MeshFromPoints	Mesh directly from point cloud/points.	Grid size, sampling density, tolerances (in GH: Delaunay).	For PC-derived classes, useful on portions (piers/beams) with dense sampling.

3.10 FROM MESH TO SURFACE (NURBS/SUBD) AND TO THE MODEL (IFC / FEM)

Starting from the per-class meshes (piers, beams, slab, barriers), we proceed with the reconstruction of engineered surfaces and solids suitable for export towards the two main application outputs:

- HBriM / IFC 4.3, for generating information objects (IfcSlab, IfcBeam, IfcColumn) consistent with libraries and naming conventions;
- FEM, to obtain clean geometries from which to derive plate/shell/beam/brick elements with materials, thicknesses and constraints.

The operational strategies described below are complementary; their adoption depends on geometry, the required level of detail and the model destination.

(A) QuadRemesh → SubD → ToNURBS (Figure 3.21)

This is the preferred route for regular surfaces and for the IFC workflow. The procedure involves generating a regular QuadMesh via QuadRemesh, aligned to the principal curvature directions; the mesh is then converted to SubD, preserving its shape, and finally transformed into NURBS surfaces through ToNURBS, obtaining a decomposition into a manageable number of patches. The main advantage lies in the topological cleanliness and coherence of the NURBS surfaces, which are easily editable in a parametric way and particularly suited to IFC export. The effectiveness of the method, however, depends on careful tuning of the parameters (guide directions, symmetries, density) and tends to decrease in the presence of very noisy or irregular geometries.

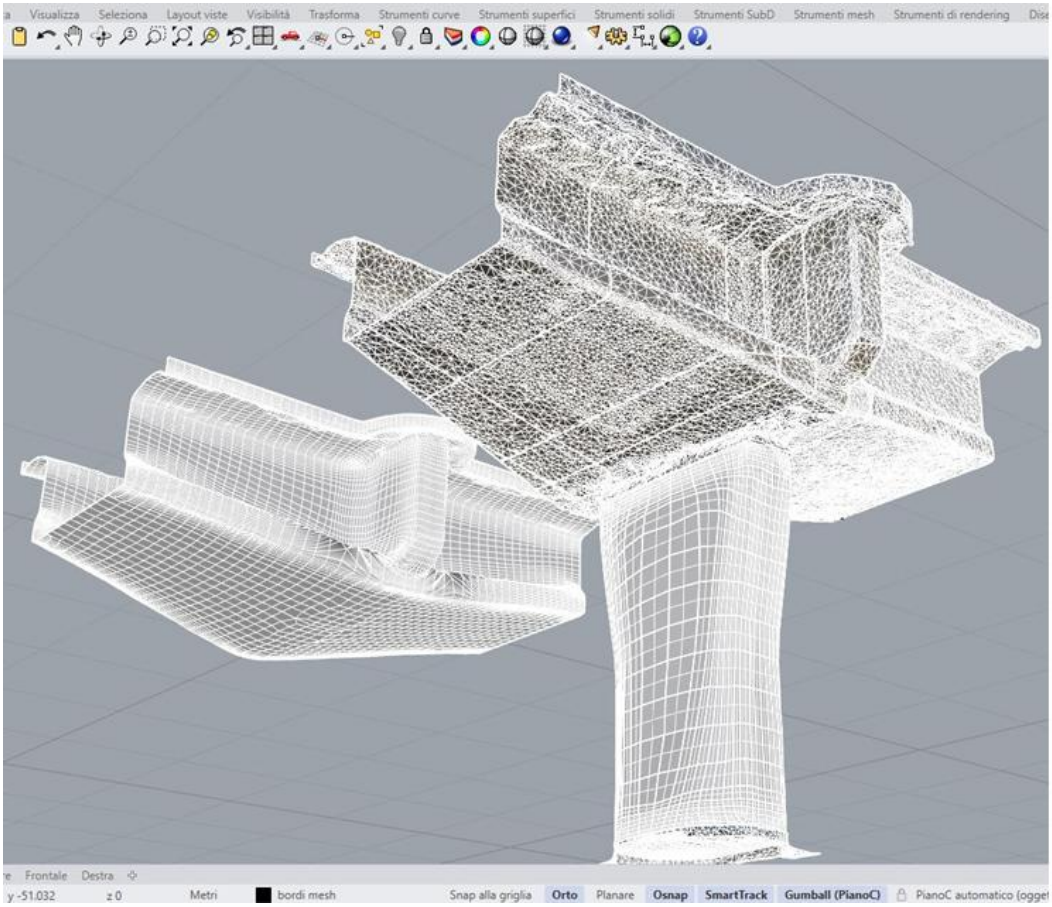


Fig. 3.21 — Transforming a variable-section beam into NURBS.

(B) MeshToNURB / ToNURBS (Figure 3.21)

In this case, conversion takes place directly on the mesh, with a 1:1 correspondence between mesh faces and NURBS patches. The workflow is extremely fast and effective for localised interventions, such as perforations, extractions or point details, where it allows a high geometric adherence to be maintained without intermediate steps. However, when applied to large models, it generates a very high number of NURBS surfaces, with a consequent increase in file weight and management difficulty; for this reason, the approach is recommended only for “surgical” operations on limited areas.

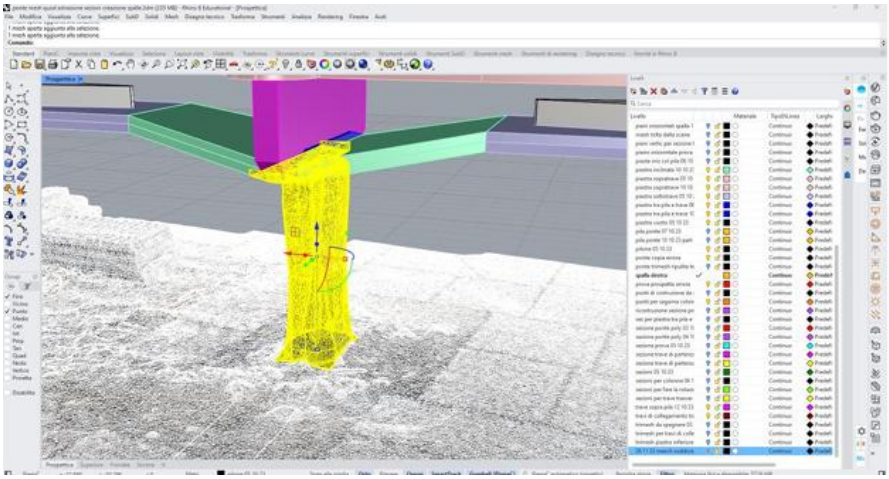


Fig. 3.22 — Direct mesh-to-NURBS transformation of a pier.

(C) Sections from mesh → Loft/Sweep/Patch (Figure 3.23)

This strategy is based on extracting section or contour curves from the mesh, subsequently simplified through controlled tolerances, and on reconstructing surfaces or solids via Loft, Sweep2 or Patch. The method guarantees a high level of engineering control of the geometry, allowing design dimensions, thicknesses and alignments to be imposed, and is therefore particularly effective for structural modelling, both for regular elements and for more complex geometries. The main limitation is the time required to build a sufficiently dense grid of sections to follow local variations (cf. §5.3); this issue is largely overcome by introducing automation procedures in section generation, making this strategy one of the most robust and reliable overall.

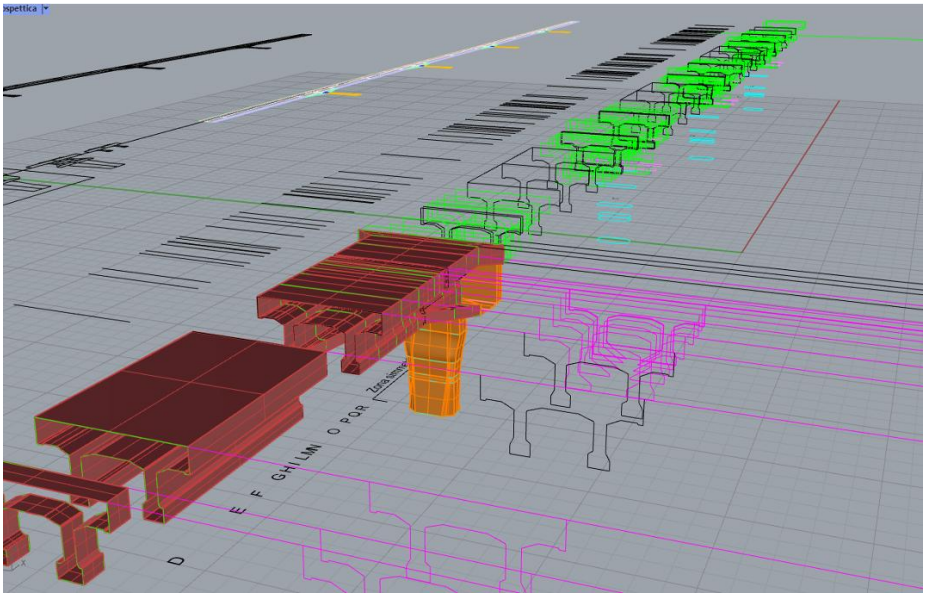


Fig. 3.23 — Geometry construction from sections using Boolean commands.

Use case / Geometry	**A: QuadRemesh→SubD→NURBS**	**B:MeshToNURB direct**	**C: Sections→Loft/Sweep**
IFC_4.3 "philological" (BrIM)	**Excellent (regular topology, Pset on NURBS)**		Good (known sections, regular geometry)
FEM plate/shell	Good (shell from NURBS)	Not recommended	**Excellent (thickness/axes under control)**
Zone_very noisy	Moderate (requires smoothing)	Poor	Good (sections filter out noise)
Dettagli_locali / fori	Good	**Useful**(local only)	Good
Modelling time	Sep	Lows (local) / Highs (global)	Variable (fast if sections are already available)

Quad optimisation aims to align edges with the principal directions (principal curvature); conceptually, it can be thought of as minimising$(\mathbf{e}_1, \mathbf{e}_2)$

$$\mathcal{E}_{\text{align}} = \sum_{f \in F} (1 - \cos^2 \angle(\mathbf{t}_f, \mathbf{e}_1)) + (1 - \cos^2 \angle(\mathbf{s}_f, \mathbf{e}_2)), \tag{70}$$

where $\mathbf{t}_f, \mathbf{s}_f$ are the directions of the sides of the quad f. QuadRemesh implements a proprietary strategy with guide curves and symmetries to control such alignment.

The surface-to-data mean squared error (on N points p_i, model $\mathcal{M}$) is defined as in (67).

The operational procedures described can be read as a compact sequence of five steps: (1) import and preparation of the per-class sub-cloud with units/tolerances and, if necessary, global shift; (2) curvature-aware remeshing (Tri/Quad) preserving functional edges; (3) reconstruction of surfaces/solids with strategies consistent with downstream use; (4) QA control through point–surface distances and element metrics; (5) export towards IFC/HBriM or FEM. A synthesis of the flow is already reported in Figure 3.11 and Table 3.2.

Tab. 3.7 — Tri vs Quad (pros/cons and applications).

Appearance	TriMesh (Triangles)	QuadMesh (quadrilateri)
Topology	Irregular (variable valence)	Almost regular (valenza ≈ 4)
Adherence to details	Excellent on curvature and edges	Good on regular surfaces
Parameterisation/NURBS	Harder	Natural (UV/loop regular)
FEM local (connections)	Robust	Requires skewness management
Weight at equal error	Major on surface. flat	Minor on surface. Desserts
Recommended use	Variable piers, connections, details	Deck slab, regular webs, panels

Class (use)	Tipo mesh	Target:h [mm]	Operational notes
Piers (IfcColumn / FEM shell)	Quad +Tri locale	20–40	Quad regular on drums; Tri in testa/base; spigoli preservati
Long./transv. beams (IfcBeam)	Quad dominant	15–30	Alignment with ribs; loops consistent with NURBS

Deck slab (IfcSlab)	Quad	25–50	Regular pattern; skewness < 0.3; local planarity
Barriers/Abutments (IfcBarrier)	Tri	40–80	Priority to lightness; low-pass smoothing

3.11 PARAMETRIC MODELLING TOWARDS FEM

The aim of this subsection is to describe the transition from the classified point cloud to a parametric model in Rhinoceros/Grasshopper (and in the MIDAS Civil GH plug-in), up to the construction of an FEM model consistent with the accuracy and interoperability requirements defined in Chapter 5. The key principle is the separation between reference geometry (axes and sections) and parametric rules (dimensions, constraints, materials), so as to allow the coherent regeneration of the model as parameters, construction stages or design scenarios change.

In the railway (training) and road (validation) case studies, semantic classification and the use of section planes made it possible to extract sections by structural class, from which to derive the axes, reducing computational burden and controlling the level of detail as a function of downstream use (HBriM/FEM). The minimal formalisation of the model is therefore made up of axes and families of sections.

The axis is modelled as a guide curve , with :$\gamma(u) u \in [0,1]$

$$\gamma = \gamma(u), \qquad u \in [0,1] \tag{71}$$

The sections are defined as a family of profiles , obtained by weighted interpolation of prototypes through blending functions :$\Sigma(u) \Sigma_i b_i(u)$

$$\Sigma(u) = \sum_i b_i(u)\, \Sigma_i\,, \quad \sum_i b_i(u) = 1 \tag{72}$$

The placement of the sections in space is performed through a Rotation Minimising Frame, which avoids spurious twists in Sweep operations (Bishop, 1975):

$$\mathbf{X}(u, v) = \gamma(u) + R(u)\, \mathbf{s}(u, v), \text{ con } R(u) \in SO(3). \tag{73}$$

This formalisation is directly implementable in Grasshopper (Evaluate Curve, Perp Frames, Loft, Sweep1/2 components) and in the GH Civil plug-in (Bridge, Section, Material, Assembly).

ΔsAxes are derived from mean trajectories of the slab/deck or of the main webs (methods: centroids of regular sections, local PCA, outlier filtering); sections are created on planes orthogonal to the axis with a variable step that is larger in quasi-prismatic regions and denser near constraints, bearings, curvature changes or joints. The quality requirement is twofold (Jolliffe, 2002):

geometric adherence, with point–profile distance (median and p95) within the per-class threshold;

Parametric stability is guaranteed by normalising the domain and adopting key sections consistently along spans (0, 0.25, 0.5, 0.75, 1), so as to ensure robust and comparable interpolations even between different spans. $\boldsymbol{u} \in [\mathbf{0}, \mathbf{1}]$

For the railway deck, a step of m was used in the central span due to the strongly variable profile and m in quasi-prismatic stretches. In the road bridge case, with six double-T (I-girder) beams, detailed sections were instead introduced at diaphragms and cross-beams, with a local step between and .$\Delta s = 0.5m - 1\Delta s = 3$ m-40.51.0 m

In Grasshopper, the minimal parametric chain includes: inputs (axes, sections, elevations), rules (blending, offsets, constraints) and outputs (surfaces/solids and associated metadata). Information management is performed through Data Trees, which allow the same rules to be applied to families of elements (piers, beams, rails) in a coherent and scalable manner.

The basic geometric rules are governed by NURBS, with curves and surfaces described through B-spline bases and weights w_i . The NURBS curve is defined as in (54). The intrinsic continuity of NURBS C^{p-1} applies within single patches; at joints between geometric lots, instead, the levels of geometric continuity $G0/G1/G2$ are controlled, corresponding respectively

to position, tangency and curvature continuity. Respecting these conditions is crucial to obtain Loft and Sweep operations free of folds or spurious discontinuities.

The main operators used in GH/Rhino are:

– Loft, for section interpolation and continuity along the parameter ;$\Sigma(u_k)u$

– Sweep1/Sweep2, with one or two guide curves γ , for robust placement based on perpendicular frames;

– Extrude, Offset and Shell, for generating webs and plates with variable thickness $t(u)$;

– Boolean operations (Union, Difference, Split), for defining functional connections and recesses (e.g., diaphragms and service openings).

To overcome the limitations of standard operators (Loft/Sweep), a parametric cluster was developed in Grasshopper that generates the section starting from the upper and lower envelopes :$z_{\text{sup}}(u)z_{\text{inf}}(u)$, and from the lateral widths $b_L(u), b_R(u)$, with height

$$h(u) = z_{\text{sup}}(u) - z_{\text{inf}}(u) \tag{74}$$

Control nodes are anchored to elevations and offsets measured on the sections extracted from the cloud. The cluster integrates automatic checks for geometric consistency, in particular: loft fairness (limiting curvature variations $\kappa(u)$), respect of minimum thicknesses $t(u) \geq t_{\min}$ for FEM checks, and domain normalization $u \in [0,1]$ for reusing the rules across different spans.

The resulting parametric architecture guarantees metric consistency and common references for both case studies (railway and road bridges), keeping units, orientations, global shift and rigid transformations consistent with what is defined in Chapter 5, to the benefit of HBriM/FEM interoperability. Axes and sections extracted from already classified clouds can be reused directly to feed GH definitions of piers, longitudinal and transverse beams and slab, with section families .$C_k(u)$ and loft/sweep operations governed by a small set of continuous parameters.

Alignment with the FEM model is also ensured: the Section/Material → Beam/Plate/Solid passage allows the direct generation of 1D/2D/3D elements in MIDAS Civil NX, using standard libraries or custom sections, including bridge templates and prestressing components. Once parametric dependencies are defined, the Model Builder orchestrates in a single run materials, sections, geometries, constraints and loads, enabling sensitivity analyses and variant studies (strengthening, widening, pier relocation) without the need for manual rework.

3.12 GH FOR MIDAS CIVIL (GH-CIVIL): END-TO-END PARAMETRISATION

The GH-Civil plug-in introduces dedicated components for Materials, Sections (libraries of standard and composite profiles), Assemblies (girders, deck, diaphragms), Load Cases and Direct export to MIDAS Civil NX. The substantial difference with respect to "pure" GH is the persistence of parametrisation within the structural software: changes to sliders/nodes in GH update the Civil model in real time (geometry, properties, constraints). Typical parameters are: Section, Material, Deck geometry, Diaphragms/cross-beams, Constraints and bearings. Each parameter is characterised by a typical mapping:

- Section: type (I-girder, box, T), dimensions (b_f, t_f, t_w, h) , variation law $h(u)$
- Material: E, ν, ρ, fck/fyk; standard libraries (RC/Steel).
- Deck geometry: road axis $\gamma(u)$, crossfall, superelevation, plan alignment
- Diaphragms/cross-beams: spacing p_d , thicknesses, connections.

- Constraints and bearings: groups of nodes/lines; construction staging (optional).

The plug-in directly generated beam/plate/shell elements with coherent connectivity (shared nodes) and allowed structured meshing (regular divisions) along preferential directions, along axes and cross-beams, which is useful for dynamic and fatigue checks. Grasshopper produces B-Rep/mesh/curve geometry for geometric modelling and, at most, exports files (IGES/STEP/SAT) without structural semantics: mechanical and loading properties are lost or must be reassigned downstream (with a consequent loss of parametrisation). MIDAS GH, instead, exposes directly in GH the FEM-domain constructs (materials, sections, constraints, loads, groups, nodes/elements) and synchronises them with Civil NX through the API. This enables: persistent parametrisation from the graph to the solver, selective import from the analysis model (Get Node/Get Property/Import Civil NX) to "close the loop", and atomic execution and rebuild of the model via Builder/Run.

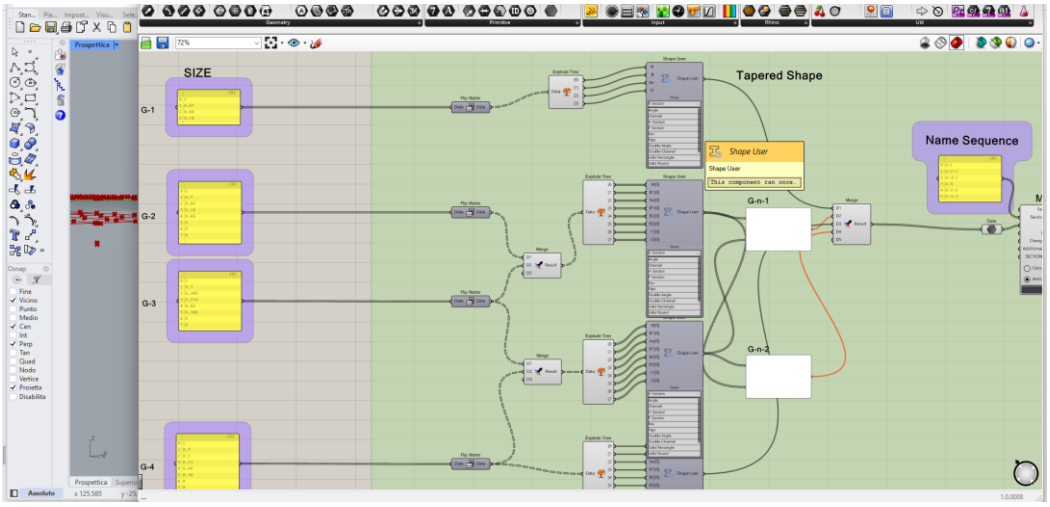


Fig. 3.24 — Grasshopper algorithm for Midas Civil.

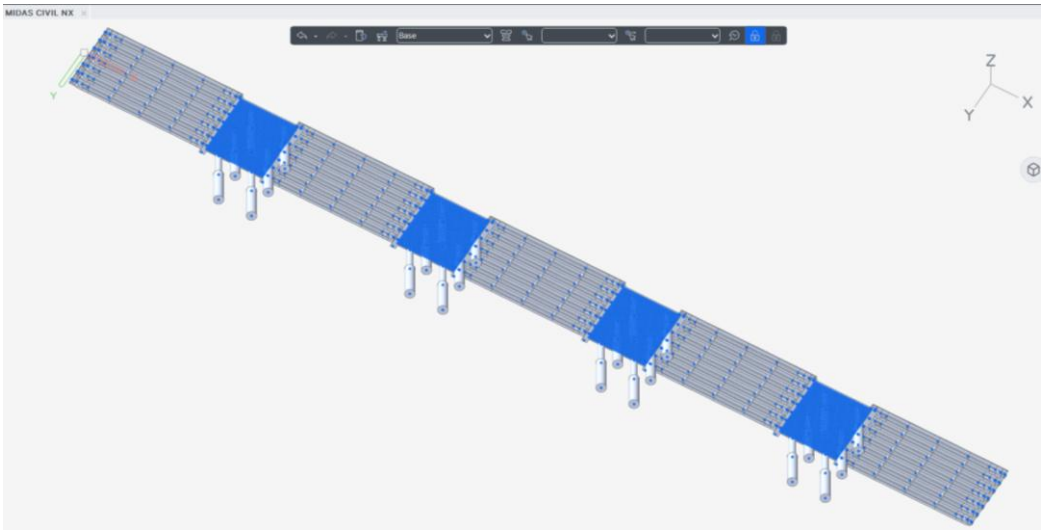


Fig. 3.25 — Midas Civil model: perspective view of the entire validation bridge.

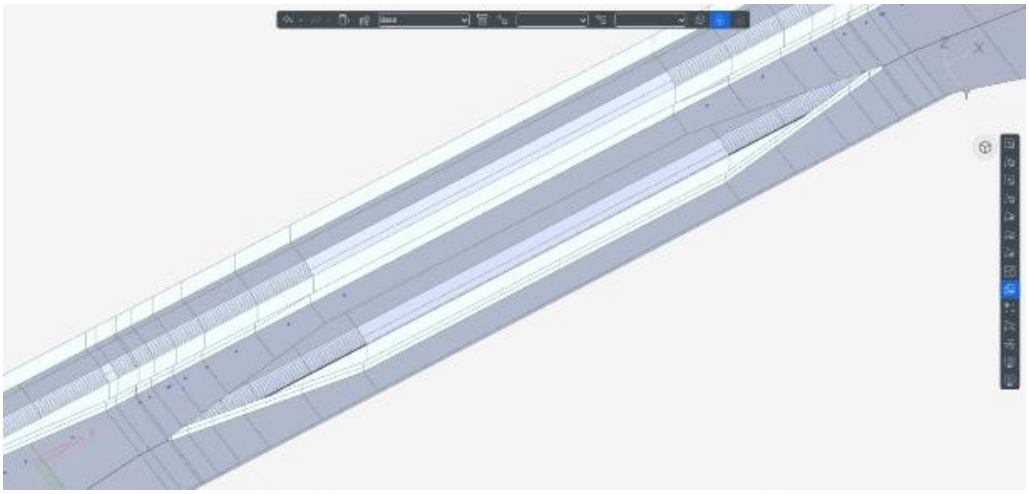


Fig. 3.26 — Midas Civil model: detail from below of the variable-section beam.

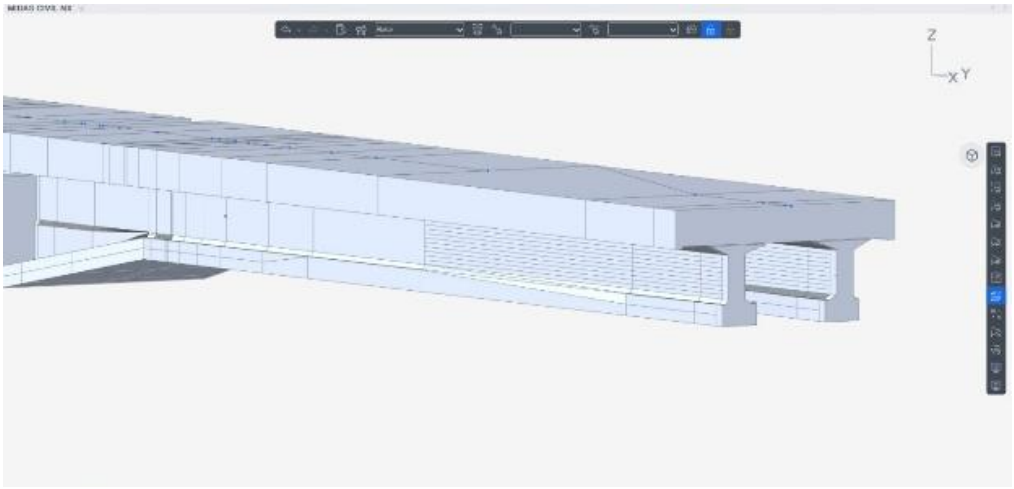
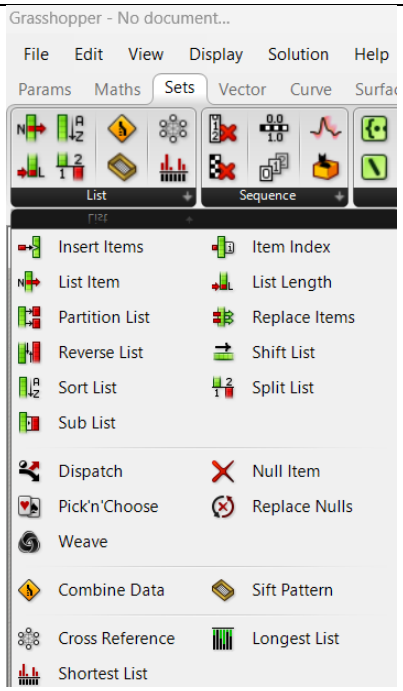


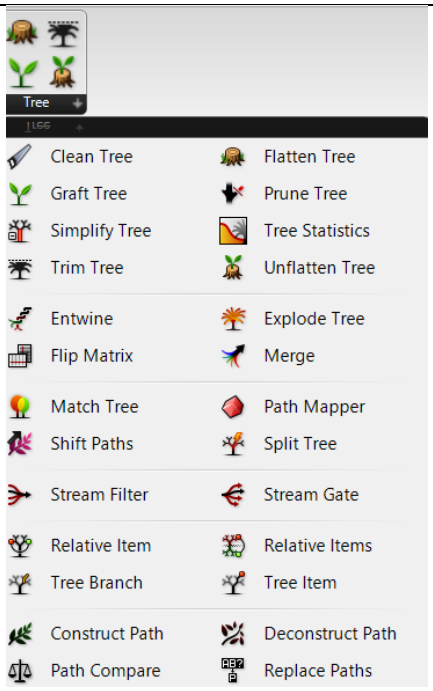
Fig. 3.27 — Midas Civil model: perspective detail of the variable-section beam.

In Grasshopper, each component processes data streams organised in lists and data trees; model robustness depends on correct stream alignment (matching), path management and appropriate use of structure operators (Graft, Flatten, Simplify, Path Mapper). In the adopted workflow, the semantic classes of the cloud (piers, longitudinal/transverse beams, slab) feed distinct branches of the tree, enabling differentiated rules for section, step and density, and keeping the model light by excluding non-structural elements, in line with Chapter 5.

Correct management of lists and data trees is ultimately decisive for the parametric modelling of variable-section beams and piers and for controlling their position and continuity along the axis.



(a) Comandi delle liste



(b) Comandi degli alberi

Fig. 3.28 — Grasshopper list (a) and data tree (b) commands.

A list is an ordered sequence of elements $[x_0, x_1, \dots, x_{n-1}]$. A data tree is an indexed collection of lists in branches (paths):

$$\mathcal{T} = \{\ \mathcal{L}^{\{p\}} \mid p \in \mathcal{P}\ \}, \quad \mathcal{L}^{\{p\}} = [x_0^{\{p\}}, \dots, x_{n_p-1}^{\{p\}}] \tag{75}$$

The path $\{i; j; k\}$ identifies a branch of the tree. The fundamental operators are: Graft, which promotes each element to a list in a dedicated branch; Flatten, which collapses the hierarchy; Simplify, which reduces redundant paths; Path Mapper, which enables explicit re-mapping (for example $\{a; b\} \rightarrow \{b; a\}$).

In the workflow adopted for the bridges under study, one branch per semantic class was maintained (e.g., {20} piers, {28} longitudinal beams,

{22} cross-beams, further articulated by span), in order to control sampling density and local rules independently.

Given the axis of an element , the unit tangent vector is defined as:$\mathbf{C}(u) \in \mathbb{R}^3, u \in [0,1]$

$$\mathbf{t}(u) = \frac{\mathbf{C}'(u)}{\| \mathbf{C}'(u) \|}, \text{con frame } \{\mathbf{t}, \mathbf{n}, \mathbf{b}\} \tag{76}$$

obtained through Perp Frames or Rotation Minimising Frames (parallel transport), to avoid spurious rotations (flip) in the presence of weak curvature:

$$\mathbf{t}(u) = \mathbf{C}'(u)/\| \mathbf{C}'(u) \|. \tag{77}$$

Stations along the axis are generated through Range or Series (parameters u_i); section planes derive from the local frames, while the section curves Γ_i are subsequently used in Loft or Sweep operations along C.

The final solid geometry is obtained from B-Reps generated with Extrude, Loft and Sweep. Boolean operations (Solid Union, Difference, Intersection) formally implement:

$$\Omega = \bigcup_i \Omega_i \, , \Omega' = \Omega \setminus \Theta \tag{78}$$

The section step Δs and the density of profiles are modulated according to geometric and structural criticality: connection zones (pier–deck), high curvature or section variations require a greater number of stations, whereas uniform stretches can be sampled more coarsely. This strategy is consistent with the approach in Chapter 5, based on accuracies and discretisations differentiated by class and by intended use (HBriM/FEM).

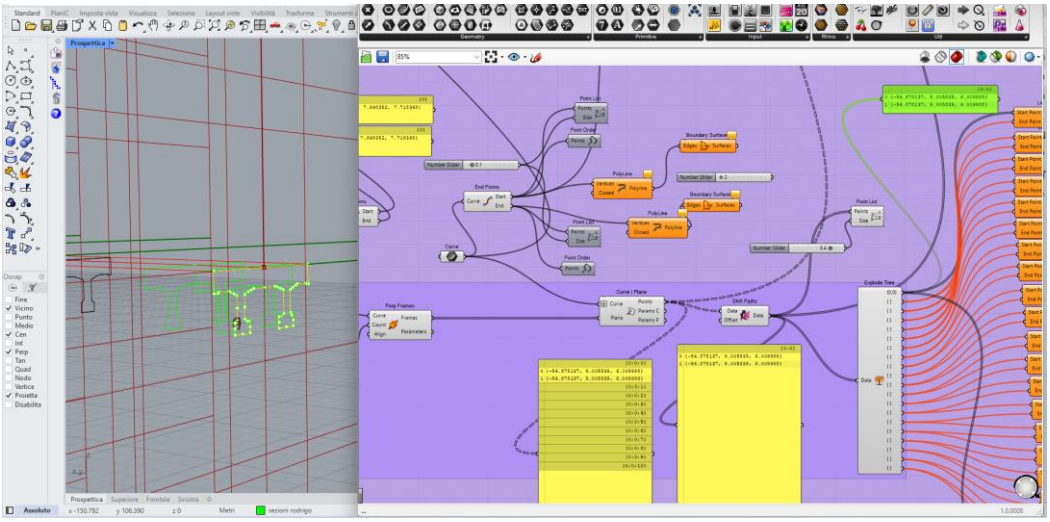


Fig. 3.30 (schematic) GH data tree with branches by class (diagram: path, Graft/Flatten/Path Mapper nodes).

3.13 CASE STUDY: VARIABLE-SECTION LONGITUDINAL BEAM AND PARAMETRIC PIER GENERATOR (TRAINING RAILWAY BRIDGE)

For the railway bridge (training dataset), a main double-T beam with variable cross-section was modelled and a parametrically controlled pier generator was built along the axis. The inputs are: (i) deck axis $\mathbf{C}(u)$, (ii) stations u_i for the sections, (iii) rules for the variation of section dimensions, (iv) spans and spacings (lists). The curves/sections come from the classified cloud (beam/slab/pier classes) segmented for Track B (GH) and Track A (HBriM sections).

a) Variable-section beam (double-T)

The custom variable section of the railway deck proved difficult to model using standard commands. Variable-section structures are based on guide lines that, in this case study, started from a unique point but ended on multiple different points due to the strong variation of the section. For this reason, the traditional Loft and Sweep commands were not able to correctly

define the end points of the generators. A modelling approach based on the parametrisation of dimensions was therefore adopted.

Let be $s \in [0, L]$ the real abscissa along **C** (with $u = s/L$). We define smooth functions (e.g., Graph Mapper → Bezier/Sine): flanges (width) $b_f(s)$, web (thickness) $t_w(s)$, flange thickness $t_f(s)$, fillets $r_c(s)$. A regular interpolator is given by the following relation:

$$b_f(s) = b_{f,0} + (b_{f,1} - b_{f,0})\,\phi(s), \phi(s) = \text{smoothstep}\,(\frac{s}{L}) \tag{79}$$

At stations s_i the section Γ_i is built in the frame $\{\mathrm{t}, \mathrm{n}, \mathrm{b}\}$ as a union of primitives (rectangles/lines/arcs) with constraints G^1 on the chamfers; finally, Boundary Surfaces closes the section. The beam surface is obtained with

$$\mathbf{S}(v, w) = \sum_i N_{i,p}(v)\,\Gamma_i(w) \tag{80}$$

The station $[s_i]$ lists and the curve lists $[\Gamma_i]$ live on branches $\{28; span\}$, i.e., stations and curves are organised in data structures (trees) on branches corresponding to the different spans (span), so that each span has its own set of sections and curves.

Graft on Γ_i and Flatten on s_i guarantee a coherent Longest List behaviour in Loft, avoiding incorrect pairings between curves and sections and ensuring a consistent generation of the surface.

Finally, Weave/Entwine are used to combine multiple sets of curves or sections, allowing the different structural parts of the deck—such as flanges, web and reinforcements—to be modelled simultaneously.

Subsequently, minimum thickness and fillet-radius thresholds are imposed (GH slider with Expression), and curvature $\kappa(u)$ is checked to prevent undesired frame twists (Parallel Transport option). Grasshopper modelling continues by building the solid from the surface and finally reaching the FEM environment.

For exporting the model to the FEM environment (e.g., MIDAS FEA NX), it is possible to:

- leave the analysis mesh generated in the FEM environment;
- convert to NURBS (ACIS/Parasolid) with MeshToNurb/Boundary, respecting thresholds and naming consistent with the classes (useful for mapping IFC/elements).

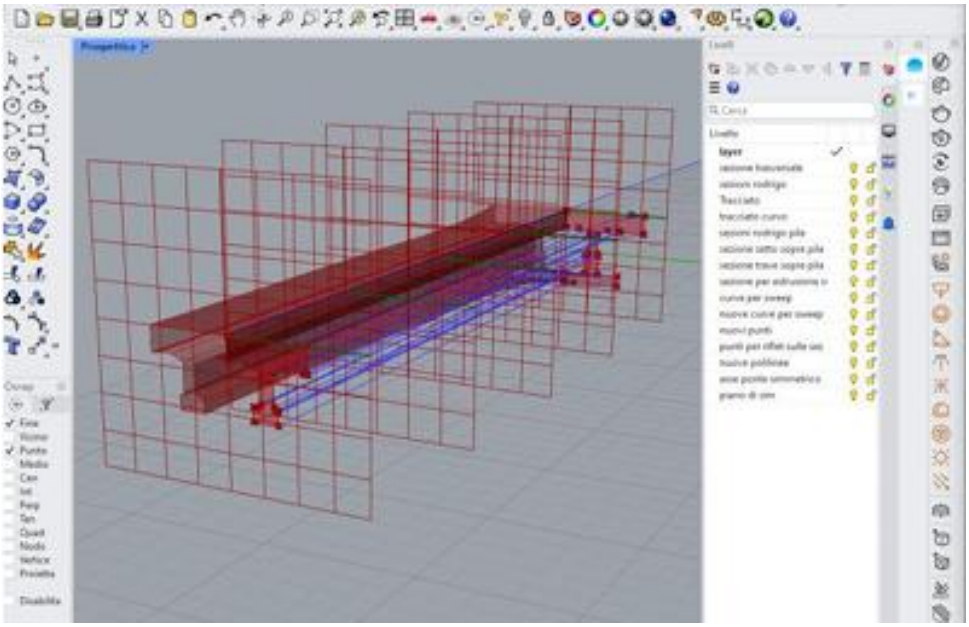


Fig. 3.31 — Grasshopper patch for the variable beam: “axis/stations”, “parametric section”, “loft”, “Boolean BREP”.

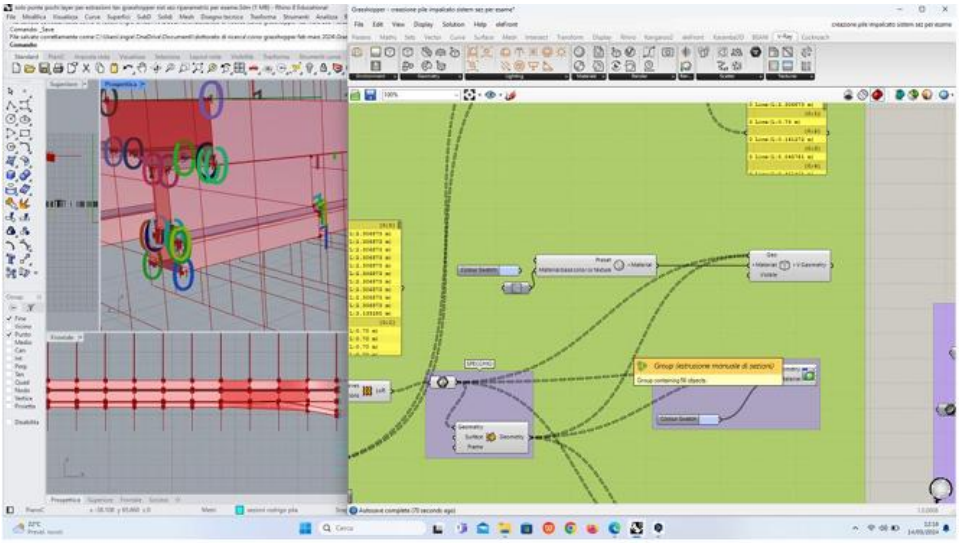


Fig. 3.32 — Detail of the control nodes for the generators.

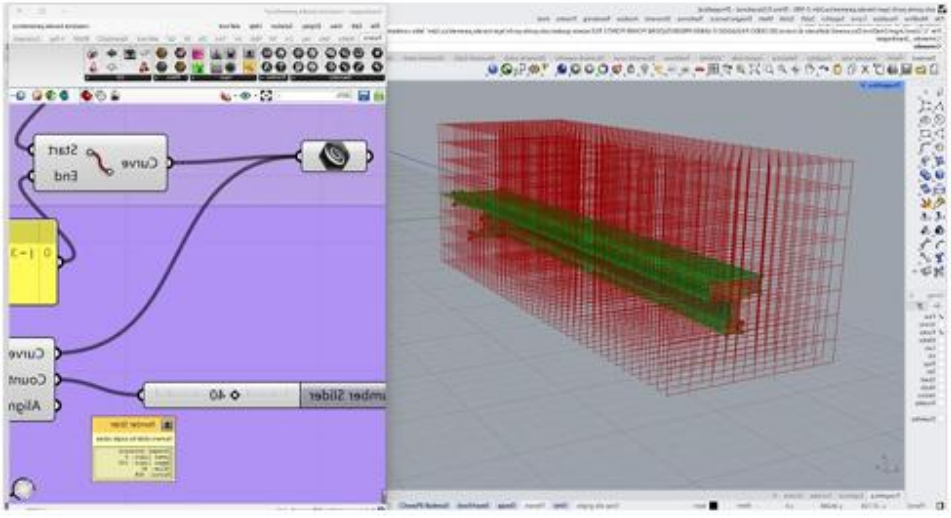


Fig. 3.33 — Detail on controlling the variability of section spacing generation.

b) Parametric pier generator and longitudinal development

For parametric generation of the piers, spacings and spans were generated as lists. Pier positions along the longitudinal axis are described by a list of

abscissas $\mathrm{X} = [x_0, \dots, x_m]$. Spacings are generated through Series/Range or through a user-defined list; a cumulative sum (Mass Addition) makes it possible to obtain the positions s_k. On the reference axis $\mathrm{C}(u)$, the normalised parameter $u_k = s_k/L$ identifies the placement point $\mathrm{P}_k = \mathrm{C}(u_k)$, while the local frame $\{\mathrm{t}, \mathrm{n}, \mathrm{b}\}_k z$ defines its orientation. The pier is therefore modelled as a parametric B-Rep with variable section along the axis z:

$$\Gamma^{\mathrm{pile}}(z) = \mathrm{Blend}\big(\Gamma_0, \Gamma_1, \psi(z)\big) \qquad z \in [0, H] \qquad (81)$$

Each pier lives in branch $\{20; k\}$ and the deck in .$\{28; span\}$. This means that the elements are organised in different branches of the data tree: each pier is identified by an index k, while the deck is subdivided by span. This separation makes it possible to manage different but related elements in an orderly way.

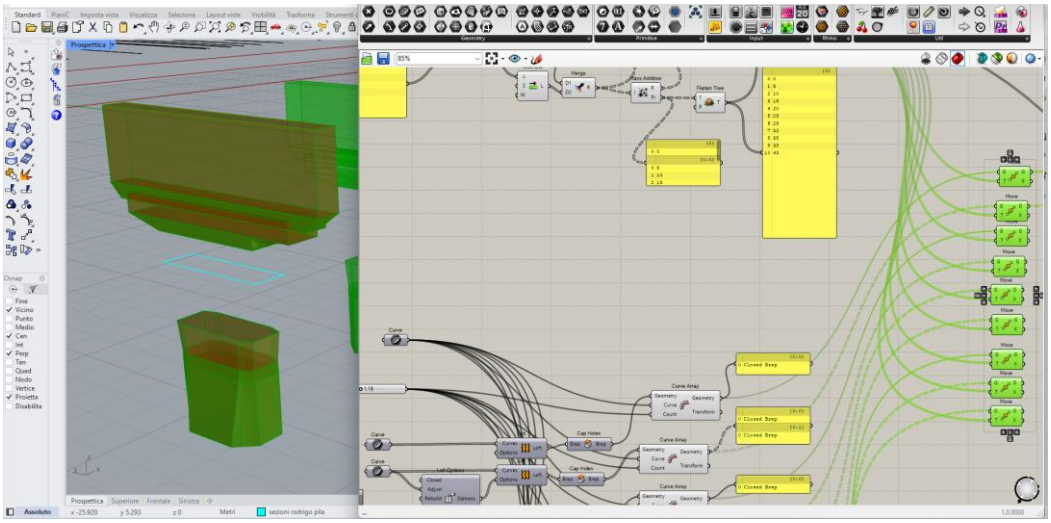
To realise pier–deck intersections and alignments, the Solid Intersection operator and Evaluate Surface were used to identify the intersection volume between the two solids. In addition, Evaluate Surface provides the exact bearing elevations on the intrados of the deck. The distance between piers is controlled by a slider or by an editable list of values. Any variation of this parameter automatically updates the pier positions and the corresponding intersections with the deck. Thanks to the parametric structure of the model, every modification to the spacings is reflected in real time on all connected parts (piers, bearings, alignments), without manual intervention.

Tab. 3.8 — Correspondence table.

Phase	**Objective**	**Componenti GH (principali)**	**Notes**
Asse/stazioni	Frames and planes	*Evaluate Length*, *Perp Frames*, *Plane*	Frame via parallel transport
Beam section	Geom. double-T var.	*Graph Mapper*, *Construct Domain*, *Boundary*	Constraints on FilletsG^1
Loft	Consistent surfaces	*Loft*, *Rebuild*, *Align*	Data matching *Graft/Flatten*

Restuccia Garofalo Alfredo

Solid and Booleans	Final BREP	*Cap*, *Solid Union/Diff*	Localized closures and cuts
Piers	Param. Generator	*Series*, *Mass Addition*, *Orient*	Rumor {20; *k*}
Export FEM	Exchange	*Elefront/Human* (bake), *SAT/Parasolid*	Naming consistent with IFC classes



{**20**; ***k***}Fig. 3.34 — Pile generator and spacing: Series/Mass Addition, Orient, branches.

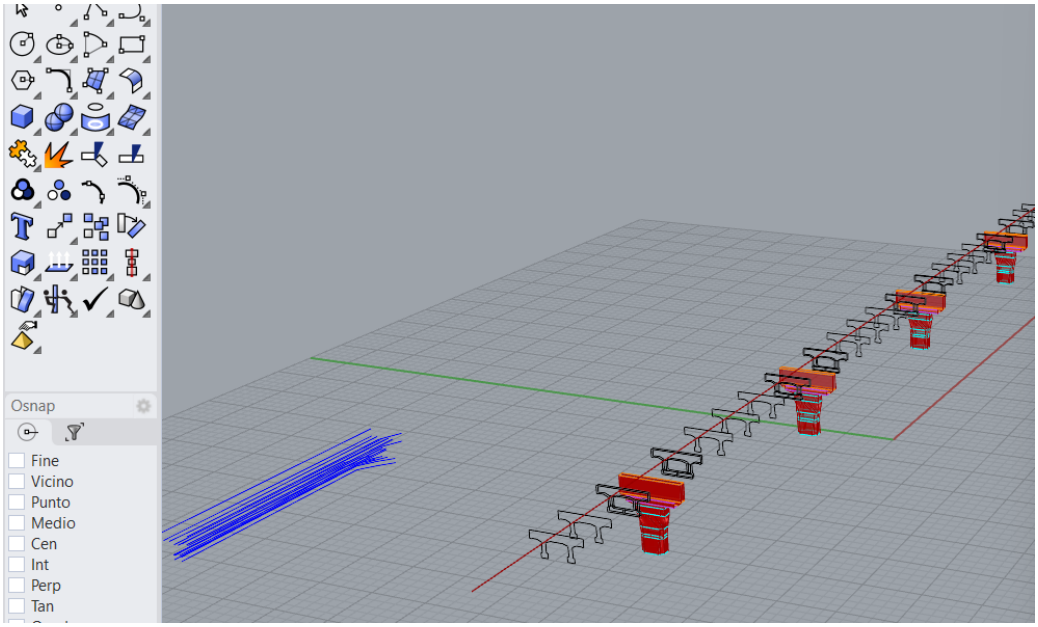


Fig. 3.35 — View of piles generated in Grasshopper.

3.14 TWO WORKFLOWS COMPARED (PURE GH VS GH CIVIL)

In the project, two complementary parametric-computational approaches were compared. The first, based on Grasshopper/Rhino ("pure" GH), favours modelling freedom and the generality of geometric operators; the second, GH Civil, natively integrates libraries and objects typical of the infrastructure domain, with a direct channel towards FEM analysis in the Civil environment. With regard to parameter persistence, the "pure" GH approach maintains parametric history within the Rhino/Grasshopper scene, but exports towards FEM or HBriM/IFC environments generally produce "baked" geometries (surfaces or solids without a link to the original parametric drivers). Conversely, in the GH Civil approach, parameters (sections, materials, constraints and placement rules) propagate inside Civil and updates occur in

real time, enabling continuous synchronisation between the geometric and analytical models.

In terms of libraries, “pure” GH offers general geometric operators and the possibility to define fully custom sections (non-standard profiles, multi-constraint lofts, complex transitions). GH Civil, instead, provides native catalogues of sections and materials (concrete, steel, typical profiles) while still leaving room for custom sections when necessary. The latter accelerates the setup of recurrent models in the road/railway domain.

The analysis mesh represents a further discriminating element. With “pure” GH, export to the solver (e.g., MIDAS FEA NX) occurs as solids/surfaces and discretisation is delegated to the FEM pre-processor, whereas in GH Civil the analytical model (beam/shell) and the mesh can be generated directly in Civil, according to infrastructure conventions, reducing time and the risk of inconsistencies between idealisation and geometry.

Operationally, the speed of variation is high in both cases, but in different ways: “pure” GH enables rapid modelling-side iterations with the need to re-export for each variant; GH Civil ensures an immediate update of the FEM model as parameters (sections, alignments, superelevations) change, without intermediate steps. Interoperability is excellent in both flows, with a natural orientation: “pure” GH excels in delivering accurate surfaces/solids towards HBriM/IFC; GH Civil favours the infrastructure FEM chain, while still maintaining IFC output through Civil tools.

A practical usage criterion follows: “pure” GH is preferable for complex non-standard geometries (special sections, lofts with many constraints, articulated transitions) and when interaction with HBriM is predominant; GH Civil is more efficient for bridges with standard typologies (multi-span with multiple I/box girders) and for iterative analytical flows, in which parameter persistence on the Civil side significantly reduces cycle time.

For the railway bridge (training dataset), a geometric layout was adopted with a deck axis described by splines and four pier axes modelled as vertical polylines with top fillets.

Section definition includes a grid along the main span with a step of $\Delta s = 1–2$ m and key sections at supports, where the transition from a double-I section to a closed box occurs. To govern local variability, a "variable section" cluster was introduced, which uses the envelopes $z_{\text{sup}}(u), z_{\text{inf}}(u)$ and the functions $b_L(u)$, $b_R(u)$ for the flanges, with thickness-threshold control to safeguard $t_{\min}$manufacturability and mesh robustness. The geometric output consists of NURBS surfaces (deck, web, flange) subsequently united into solids through Boolean operations for export towards MIDAS FEA NX in ACIS/Parasolid formats. The FEM mesh was generated in the FEM pre-processor with controlled anisotropy, densifying elements near supports and fillets; density follows the law:

$$h(u) = \frac{h_0}{1 + \beta\, \kappa(u)} \tag{82}$$

which links the element size $h(u)$ to curvature $\kappa(u)$, allowing degrees of freedom to be concentrated in zones with higher deformation demand. This setup ensured consistency between parametric modelling, section transitions and analytical idealisation.

For the road bridge (validation dataset) a multi-girder typology was adopted with six I-girders and parametric cross-beams/diaphragms. The axes consist of six longitudinal guide curves parallel to the deck axis; cross-beams are governed by a $p_d$modifiable step . The girder sections have variable height $h(u)$, larger at midspan, while the top slab is modelled as a plate with thickness. In this scenario, the GH Civil flow was preferred, exploiting the Section/Material libraries for concrete and steel and the automatic assembly of the multi-girder system. Export occurs directly as beams (for longitudinal and transverse beams) and shells (for the slab), with constraints and loads set up for analysis in Civil NX. The main operational benefit lies in the persistence of parametrisation in Civil: changes to $h(u)$, to the step p_d , to lane widths or to superelevation propagate without re-importing the

idealisation, enabling rapid and traceable verification cycles on a single coherent model.

3.15 PARAMETRISATION, INTEROPERABILITY (TOWARDS HBRIM/FEM) AND AUTOMATION

Parametric 3D modelling offers significant engineering advantages. It enables, for example, efficient management of design variants, interventions on existing structures and surface optimisation through targeted mesh control. In particular, design variants and work-in-progress phases are governed by parameters that allow the alignment to be modified, piers to be repositioned or widenings to be introduced, ensuring coherent regeneration of all related elements such as deck, barriers and diaphragms.

Parametric control is also fundamental for interventions on existing structures, for example in the design of structural strengthening, such as plates, angles or transverse stiffeners, modelled as parametric increments of dimensions ($\Delta t_{f,}\Delta t_w$). Finally, mesh optimisation is achieved by controlling its density, defined by a function that increases the number of elements in zones with higher curvature, improving model accuracy without unnecessarily increasing computational complexity, with function:

$$n(u) = n_0 + \alpha\,\kappa(u) \qquad \kappa(u) = \| \gamma''(u) \| / \| \gamma'(u) \|^3 \qquad (83)$$

To ensure interoperability between the different modelling and analysis environments, procedures were defined for managing units, semantic naming, quality checks and the export of models. The entire process was structured in a modular and automatable way, allowing coherent regeneration of the models, exploration of design variants and maintenance of consistency between HBriM and FEM models. This setup enables lighter, controllable models that can be used directly for numerical analysis.

3.16 MODELLING IN THE HBriM ENVIRONMENT

This section illustrates the pathway that turns the geometric data extracted from the point cloud—such as axes, sections and dimensional references—first into an HBriM model in MIDAS CIM and then into an FEM analysis model in MIDAS Civil. Particular attention is devoted to the operational phases and the checks that ensure a clear, reliable and coherent data handover between the different tools used.

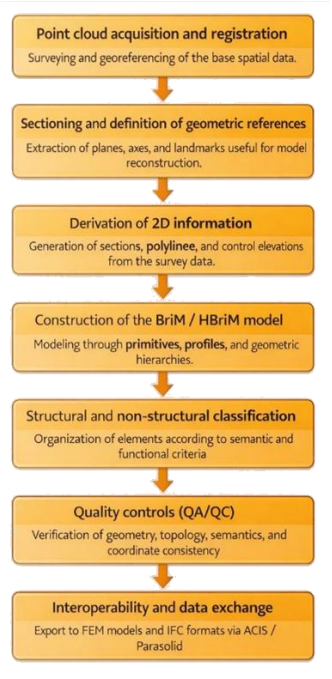


Fig. 3.36 — Overall pipeline of the Scan-to-HBrIM method (phases and outputs).

The georeferenced cloud, in the prescribed reference system, is sliced through horizontal, vertical and oblique planes depending on the bridge morphology and on the elements to be delivered. On these planes, polylines and/or contours are extracted and guide axes for piers, beams and frames are defined. The references define a stable geometric grid for 3D modelling. From the sections, 2D drawings (lines/polylines) are derived, accompanied by

control dimensions that constrain subsequent modelling, preventing dimensional drift and enabling punctual checks on the allowed deviations.

The 2D geometries and the axes constitute the basis for the subsequent information model, articulated into reusable libraries and units, with hierarchical organisation of elements and informative attributes consistent with management and analysis needs. Elements are classified into structural (beams, frames, diaphragms, piers, etc.) and non-structural (safety barriers and other details). In the first exports towards FEM environments, non-structural elements are not treated as load-bearing components; they nevertheless remain in the BrIM for information completeness. The georeferenced position of the bridge is preserved in the CAD environment.

In MIDAS CIM, for modelling needs, the deliverable is roto-translated and aligned to the software's global axes. The operation does not alter dimensions, preserving coherence with the original reference system through metadata and alignment reports.

The model created in CIM follows a hierarchical logical structure consisting of:

- Point Library
  organised in notable points/nodes (intersections, ends, centres), basis for vertical frames and key references.
- Curve Library
  based on curves, guide profiles and axes
  (longitudinal/transverse/oblique) derived from the sections.
- Assembly Unit
  assembly unit that combines repetitive elements (e.g., longitudinal girder + vertical frames, panels/slabs), promoting reuse and coherence.

Structural and non-structural classification is set in the HBriM to drive both FEM export and IFC completeness (Table 3.9).

Figure 3.34 — BIM/FEM information-flow and product (coverage) diagram.

Tab. 3.9 — Mapping classes → IFC 4.3 → FEM (bridges).

Element (class)	**IFC 4.3 – Entity**	**Key properties**	**Tipologia FEM (Civil/FEA)**	**Export notes**
Main girder (long.)	IfcBeam (steel)	Profile name, material, assembly code; SBS=1.1.2	Beam/Frame (composite section or IPE/HE)	Export as Beam; check local axis and direction.
Cross-beam / Floorbeam	IfcBeam	Section, length, connections; SBS=1.1.2	Beam/Frame	Keep a unique ID to match with longitudinal girders.
Stringer/Longherone secondario	IfcBeam	Profile, spacing, aggregation in AU	Beam/Frame	Check spacing and geometric continuity.
Bracing / Cross-frame	IfcMember	Tipo (X/K), barre/angolari, dettagli essenziali	Truss/Bar/Tie	Export as Member with simplified connections.
Diaframmi / Stiffener	IfcMember o IfcPlate	Spessori, piatti irrigidenti	Plate/Beam	For plates use Plate; for linear stiffeners Member.
Steel deck plate	IfcPlate	Thickness, area, orthotropic properties (if known)	Plate/Shell	Use closed surfaces; No minute holes.
Composite slab (R.C.)	IfcSlab	Thickness, material, composition	Plate/Shell (composite section)	In FEM, associate the composite section to

				beams/deck slab.
Supports	IfcBearing	Type (elastomeric, pot, sliding), stiffnesses	Spring/Link	Mapping Linear/Rotational Stiffnesses in Civil.
Abutments/Piers (class)	IfcAbutment / IfcColumn	Main materials and geometries	Solid/Plate/Frame	Simplify meshes and constraints (joints/planes).
Barriers/Parapets	IfcBarrier	Non-structural	—	Exclude dall'export FEM; useful only in BrIM.

Quality control is organised on multiple levels. First, the geometric deviation between the surveyed sections and the model surfaces or solids is checked, to evaluate adherence to reality. Then, topology is checked by verifying profile closure, axis continuity and correct connections between elements. A further level concerns the model semantics, i.e. the correct assignment of categories and attributes to objects.

Operational thresholds (geometric deviations, axis continuity and semantic coherence) are defined in the QA/QC register and recalled in the methodological framework of Chapter 5; here we only stress that these thresholds guide export towards IFC/FEM and the consistency checks after import.

- section-to-model geometric deviation ≤ 5–10 mm (depending on survey density and element category);
- axis continuity with angular tolerance ≤ 0.5°;
- 100% semantic check on categories/attributes that impact FEM/IFC export.

Acceptance thresholds must be formally defined and documented in the project QA/QC register. Control procedures, referring to the one-way BrIM → FEM flow, include verification of the correct handling of units and geometric

transformations (conversion from metres to millimetres, translations and roto-translations), to be documented in the README_transform file. In addition, coherent attribution to each object of the fundamental structural metadata is required—such as Category, Material, Section and SBS code—together with the explicit setting of the flag structural = true, in order to guarantee the correct interpretation of the model in the subsequent structural-analysis phases.

Export is carried out in IFC 4.3 format, limited to the structural classes, and in STEP/Parasolid formats for cross-checking geometry. The model is then imported into the structural software MIDAS Civil, where the elements needed for analysis—such as rigid links, springs and bearings, composite sections, loads and corresponding combinations—are replaced or integrated. Once the definition of the analysis model is complete, it is exported in the formats required for the different project purposes: structural model for FEM analysis, information model in IFC format for coordination and archiving, and CAD/Parasolid geometries for geometric coordination. The overall operational flow is summarised in the Swimlane diagram shown in Figure 3.37.

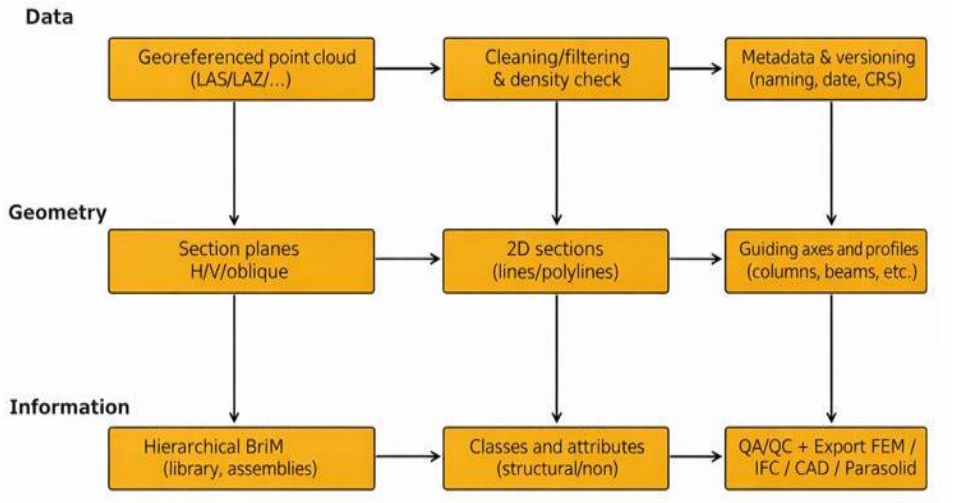


Fig. 3.37 — Conceptual swimlane diagram.

Tab. 3.10 — Environment flow: Rhinoceros+Veesus → MIDAS CIM (HBriM) → MIDAS Civil (FEM).

Phase	Environment / Software	Main role	Main activities	Output
Pre-modelling	*Rhinoceros + Veesus Arena*	Preparation and cleaning of geometric data	• Cloud management E57/LAS/LAZ (clipping, slicing) • Setting units (mm) and global shift • Organisation by layers/classes • Extraction of sections and axes	• Geometry 2D/3D clean • File DWG/DXF • File Parasolid (.x_t/.x_b) • File 3DM
HBriM / Information model	*MIDAS CIM (Bridge Information Modeling)*	Information-model construction	• BrIM modelling with libraries and assemblies • Alignment to the global axes • Hierarchical and semantic management • Data preparation for structural analysis	• BrIM/HBriM model • IFC 4.3 (structural classes only) • Parameters and sections for MIDAS Civil
FEM / Structural analysis	*MIDAS Civil*	Structural analysis and verification	• FEM model generation • Definition of mesh, loads and combinations • Integration of structural elements • Execution of analyses	• Proprietary calculation file • FEM results (displacements, stresses, etc.) • Analysis report

The data-exchange chain is organised to maintain traceability and interoperability; the input, exchange and output formats are summarised in Table 3.12. In the following, we only recall the choices that affect metric coherence (units/axes) and transfer towards HBriM/IFC and FEM.

- IFC 4.x (in particular IFC 4.3 for infrastructures) for information coordination;
- native MIDAS Civil/Civil NX file for structural analysis.

A summary of the chain is reported in Table 3.11.

Tab. 3.11 — Data formats: input, exchange, output (§ 3.1.x "Data formats – input, exchange, output").

Phase	Formats	Content/use	Notes
Input point cloud	E57 / LAS / LAZ (georiferiti); opz. E57/PTS/PLY	Survey acquisition; basis for slicing and density control	Check CRS; document units and global shift
2D/3D derivatives for exchange	DWG/DXF (curves/sections/axes); Parasolid .x_t/.x_b	Profiles, sections, parametric solids for modelling	Align reference frames/axes; consistent naming for classes/layers
Information model (HBriM)	IFC 4.x (IFC4.3 Suggested)	Full deliverable and multidisciplinary coordination	Class→IfcEntity mapping; minimum properties and Psets
Structural model (FEM)	MIDAS Civil proprietary file	First analytical instance derived from the BrIM model	Model focused on structural elements

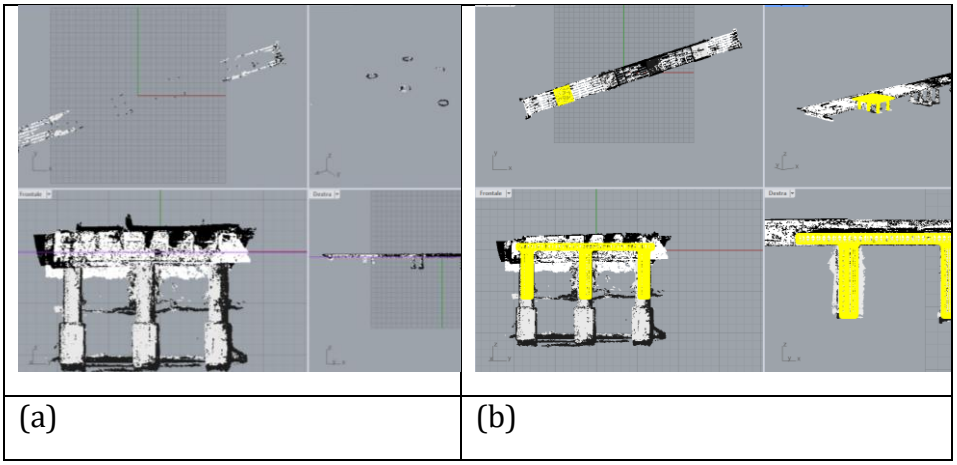
Tab. 3.12 — Software↔format compatibility in the workflow.

Formato	Rhino+Veesus	MIDAS CIM	MIDAS Civil	Typical use
E57/LAS/LAZ	✓	○	○	Survey input, slicing in Rhino/Veesus
PTS/PLY	✓	○	○	Visual/inspection (classes)
DWG/DXF	✓	✓	✓	Curves/sections/exchange axes
Parasolid .x_t/.x_b	✓	✓	✓	Solids/Parametric Towards CIM/Civil
IFC 4.3	○	✓	○	Coordination and information archive
Civil proprietary file	○	○	✓	FEM analysis and post-processing

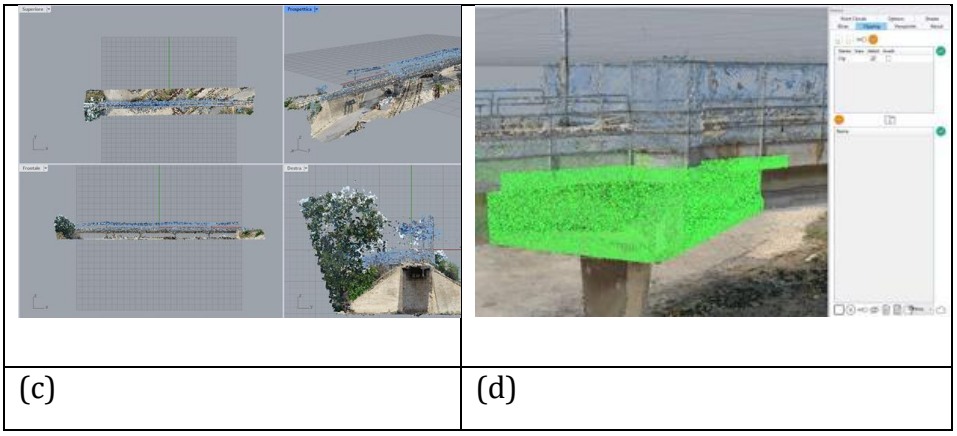
Table 3.9 — Outputs by environment and phase (BrIM/FEM).

3.17 BRIM BRIDGE CASE STUDY

Following the interoperability choices and exchange formats introduced in §3.16, we describe the transition from the point cloud to the BrIM model, highlighting aspects specific to the bridge case study and the operational checks that accompany the promotion from points to objects.



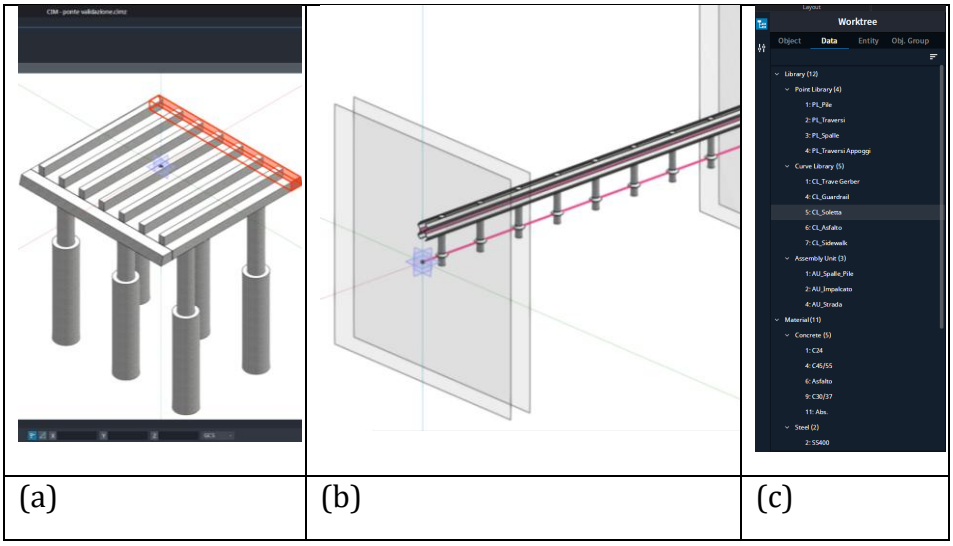
(a) (b)



(c) (d)

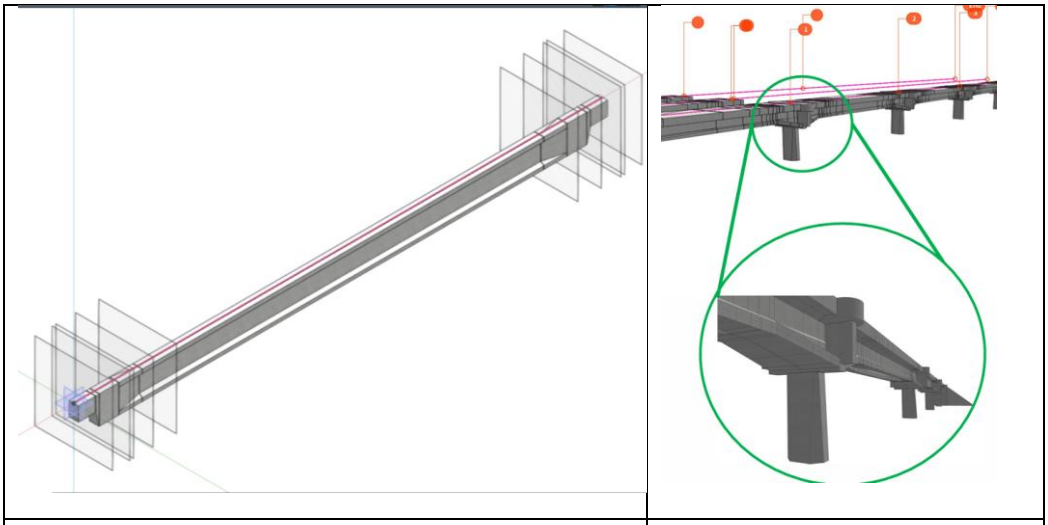
Fig. 3.38 Data processing stages in Rhinoceros. Overall view of the cloud: transverse/longitudinal sections and sample elevations (a) validation bridge (c) training bridge; slicing (Veesus) with H/V planes, organised by spans and beams (b) validation bridge (d) training bridge.

In Rhinoceros, layer organisation allows sections, axes and extracted geometries to be separated by class and by span. This facilitates subsequent export and traceability. Veesus tools are used to manage the cloud efficiently, while native Rhino commands are used to draw references and reconstruct surfaces and solids.



(a) (b) (c)

Fig. 3.39 Modelling stages in Rhinoceros. Pier cap with piles: editable parameters (a); Curve Library modelling of a secondary guardrail element (b); Worktree (Library): materials set for RC bridges (c).



(a) (b)

Fig. 3.40 — Structural detail: Gerber beams on the validation bridge (a); bearing detail of a variable-section beam on a pier on the training bridge (b).

For the information model, the building blocks are organised in the CIM libraries (Point, Curve, Assembly). Each object is created by selecting the appropriate template and then assigning attributes, materials and classification codes. The model is then checked and exported to IFC 4.3.

Figure 3.35 — Left: examples of flush parts and parts on an inclined plane. Right: improvement with Planar.

Fig. 3.41 — Detail of the menu and icons for preparing the model for export.

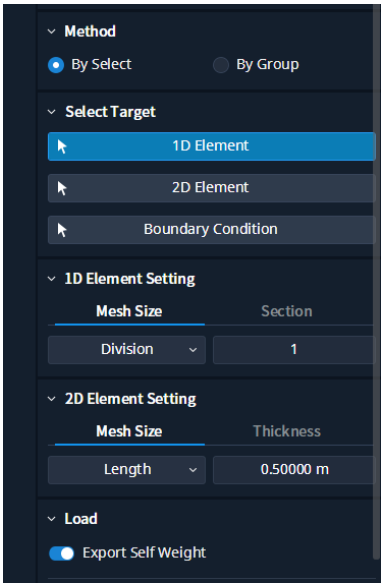


Fig. 3.42 — Choice of structural type and mesh type.

Figure 3.40 Modelling stages in MIDAS CIM. Creation of the HBriM model of the case-study bridge (a); assignment of attributes and categories (b); export to IFC 4.3 and model viewer (c).

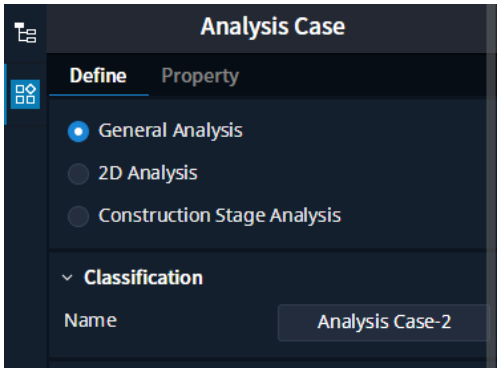


Fig. 3.43 — Choice of the analysis case.

The bridge HBriM model is then used to support the creation of the structural analysis model: axes and sections are exported and imported into Civil; element properties and constraints are assigned, and the analysis mesh is generated.

Figure 3.36 — Left: regular quadrilateral components. Right: distorted quads on a sharp edge.

Fig. 3.44 — Model visualisation: members, constraints and nodes.

Figure 3.41 Modelling stages in MIDAS Civil. Import of IFC/geometry and creation of the analysis model (a); assignment of boundary conditions and loads (b); FEM discretisation and run of the analysis (c).

Fig. 3.45 — Types of export extensions.

The QA/QC metrics (C2C, IR_τ, percentiles) are computed to validate the model and are stored in a report, ensuring traceability of the modelling process. The results are discussed in Chapter 5.

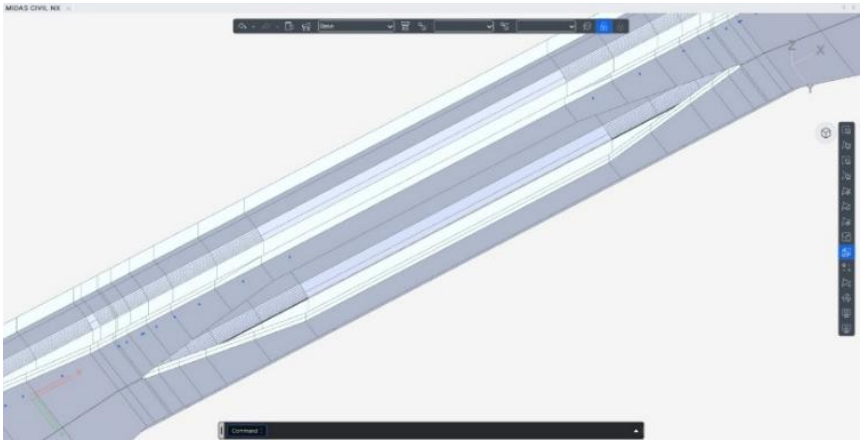


Fig. 3.46 — Visualisation of a training-bridge detail in Midas Civil.

Table 3.10 — Use cases: mesh types and Rhino tools by class.

Figure 3.42 Summary of the end-to-end workflow from point cloud to BrIM and FEM for the bridge case study.

4.0 INTRODUCTION

This chapter describes the complete supervised classification pipeline developed for bridge point clouds, with specific reference to the training railway bridge and the validation bridge (Riveiro, DeJong & Conde, 2016; Xia, Yang & Chen, 2022).

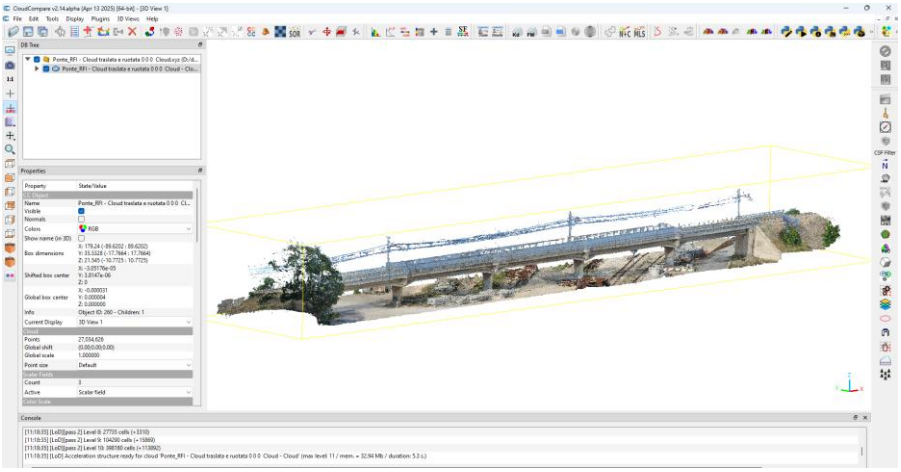


Fig. 4.1 — Training point cloud

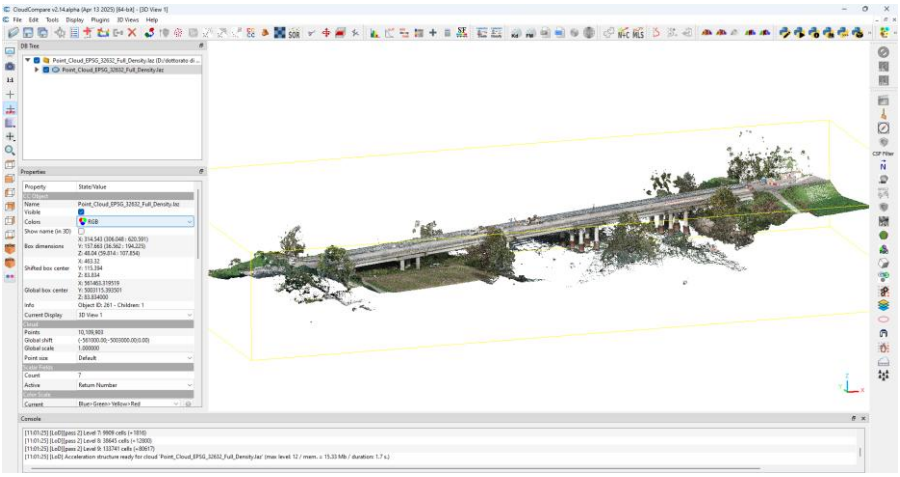


Fig. 4.2 — Validation point cloud

The pipeline is designed to be repeatable, traceable, and scalable. The workflow starts with importing the point cloud, continues with multi-scale extraction of geometric features and with building the supervised dataset through a semi-automatic labelling loop based on an initial manual seed and subsequent automatic label propagation. The process then includes training a Random Forest (RF) classifier supported by balancing and validation protocols, and concludes with evaluation, visualization, and interoperability tools towards BIM/FEM environments (Grasshopper, IFC, FEM solvers) (Demantké et al., 2011; Pauly et al., 2003; Breiman, 2001).

In continuity with Chapters 1–3, this chapter marks the transition from the method to its operational application, making the workflow data → features → model → verification → export concrete. The specific objectives are: (i) to standardise local feature extraction at multiple scales; (ii) to reduce labelling cost through a semi-automatic strategy based on a confidence threshold and active review; (iii) to validate methodological choices through robust metrics (accuracy, macro-F1, Kappa) and through a metric check on real data (Cloud-to-Cloud, C2C) applied exclusively to the validation dataset; (iv) to set up export and mapping channels towards GH/IFC/FEM environments (Settles, 2012; Sokolova & Lapalme, 2009; Powers, 2011; Cohen, 1960; CloudCompare, 2021; Lague, Brodu & Leroux, 2013).

The canonical classes (10 ground, 12 vegetation, 20 piles, 22 longitudinal beams, 24 guardrails, 26 cross beams, 28 deck_slab) and their official coding are recalled in the project manifest. Each execution of the scripts in Spyder automatically generates a manifest containing library versions, parameters and source files, and saves standard artefacts such as RF classifier configurations, cross-validation results, feature-importance analyses, plots and outputs in PLY and CSV formats (Strobl et al., 2007; Kohavi, 1995).

The chapter is organised into seven sections: Section 4.1 defines the mathematical formulation and the feature set; Section 4.2 details the geometric descriptors and the multi-scale computation strategies; Section 4.3

discusses dataset creation, including sampling, the initial manual seed and semi-automatic labelling via propagation; Section 4.4 focuses on RF training and validation; Section 4.5 describes validation on an independent point cloud and the C2C quality-control metric; Section 4.6 presents visualization and analysis of results; Section 4.7 concludes with final considerations and future developments.

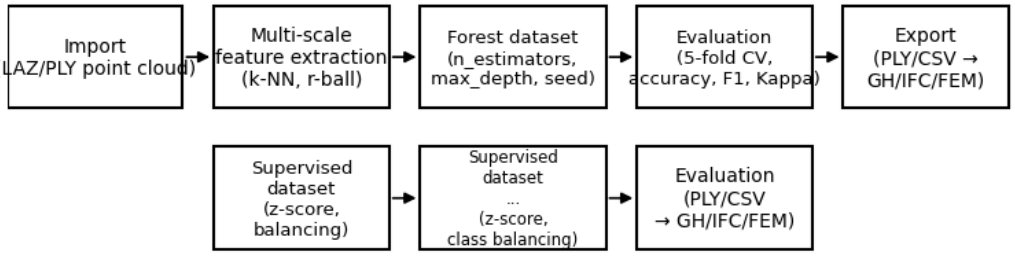


Fig. 4.3a – Pipeline schematic (import → features → dataset → RF → evaluation → export).

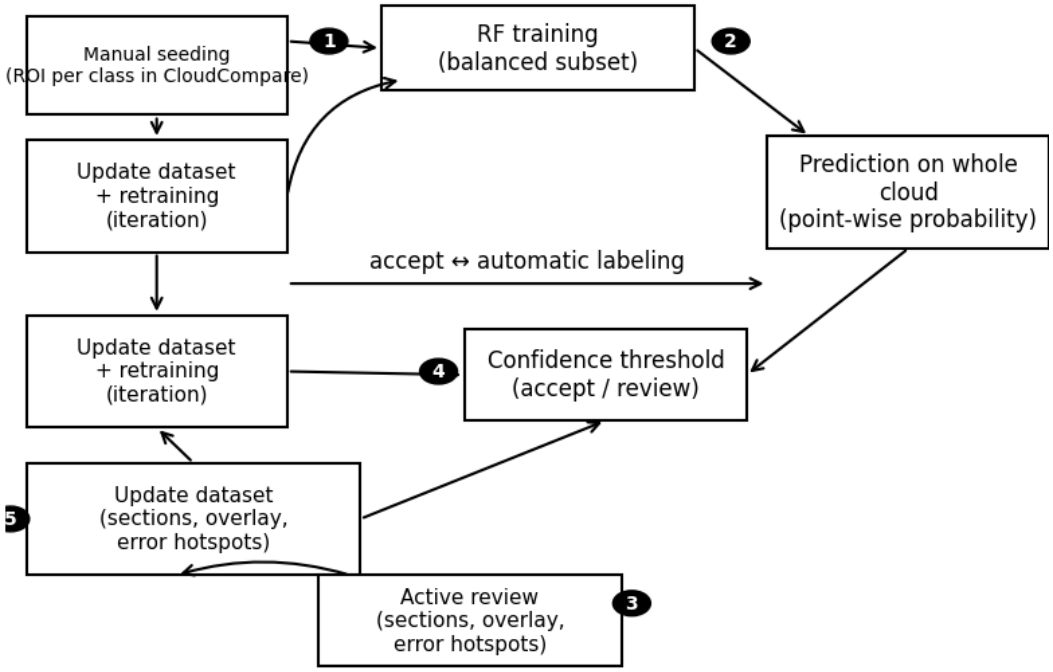


Fig. 4.3b — Semi-automatic labelling cycle (manual seed → RF → threshold → review → iteration).

Table 3.11 — Operational tables: general and per-class parameters.

In Chapters 1–3, the theoretical–methodological framework, the geometric basis for local analysis, and the integration channels towards BIM/FEM environments were defined. This chapter makes that framework operational by implementing the data → features → model → verification → export pipeline. In particular, the workflow is designed to: (i) analyse the influence of neighbourhood scale using k-NN and r-ball approaches; (ii) handle class imbalance through sampling and weighting strategies; (iii) verify consistency between classification metrics and the Cloud-to-Cloud (C2C) geometric check applied during validation.

As anticipated, the workflow is structured into five macro-phases: acquisition, pre-processing, feature extraction, classification and output. Each phase corresponds to a set of scripts and standard output files.

ID	Synthetic label	Typical use in the pipeline
10	ground	reference for HAG, elevation filters
12	vegetation	sphericity/omnivariance check
20	piles (piers)	verticality + curvature/normal variation
22	longitudinal beam	directional linearity/planarity
24	guardrails	roughness/edge, local density
26	beams_transversal	orientations and linearity
28	deck_slab (deck slab)	planarity and low roughness

The technical/software architecture adopted for script development is based on the Python environment (Anaconda/Spyder), as shown in Figure 4.4. In this context, established libraries are used for point-cloud processing and analysis—such as NumPy, Pandas, scikit-learn, Open3D/PyntCloud and Matplotlib—as reported in Figure 4.5 (Harris et al., 2020; McKinney, 2010; Pedregosa et al., 2011; Hunter, 2007; Zhou, Park & Koltun, 2018).

CloudCompare is used for geometric checks and metrological verification, while Rhino/Grasshopper is used in the interoperability stage, geometric reconstruction and export to BIM and FEM environments, as shown in Figure 4.6.

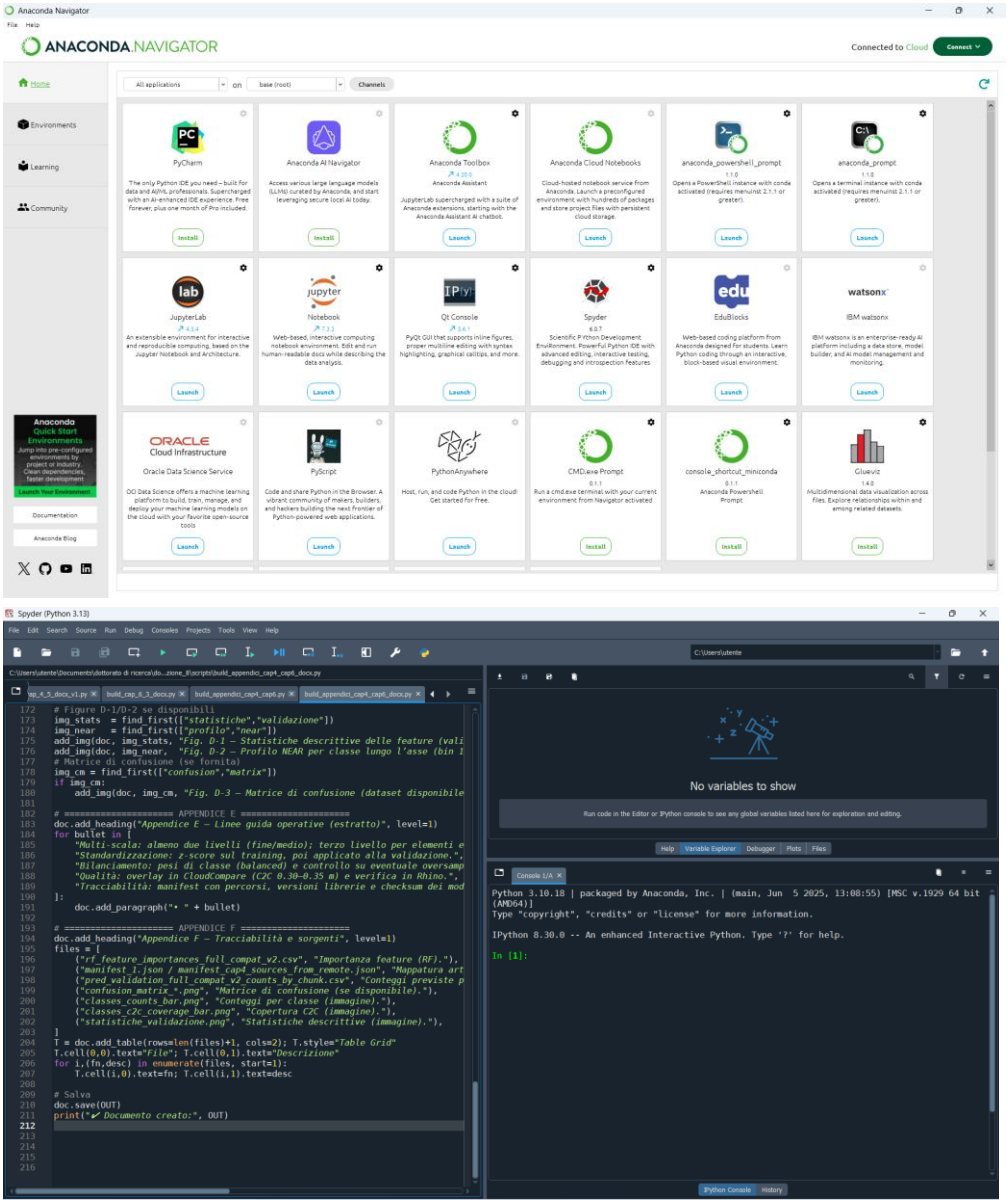


Fig. 4.4 — Anaconda/Spyder environment.

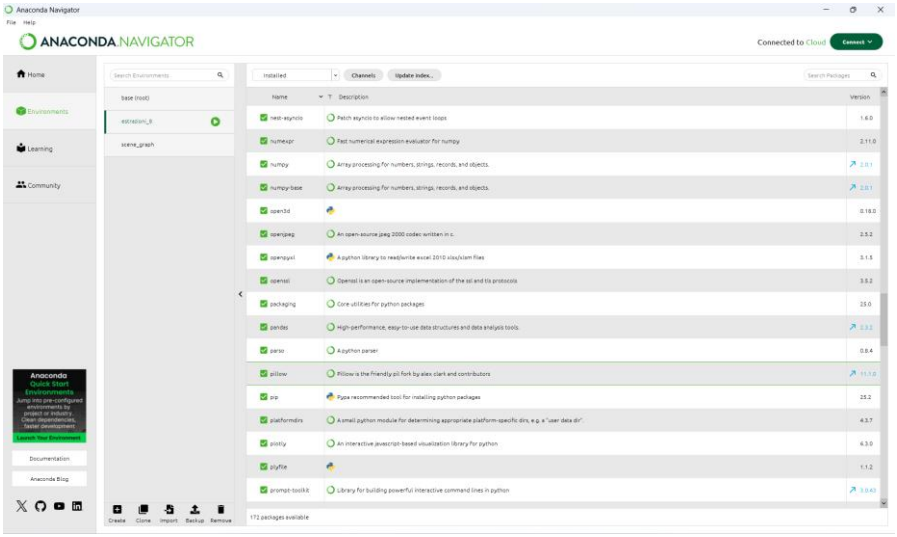


Fig. 4.5 — Partial list of libraries loaded in the project.

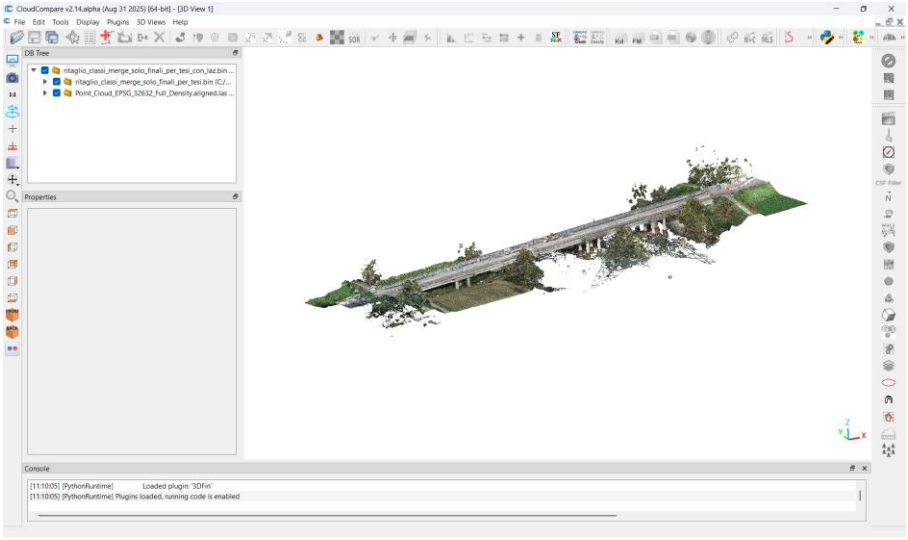


Fig. 4.6 — CloudCompare environment.

Model training uses Random Forest with hyperparameters selected through cross-validation (number of trees, maximum depth, minimum samples per leaf, class weights). The model is evaluated through accuracy and class-wise

metrics (precision, recall, F1), as well as macro-F1 and Cohen's Kappa to account for class imbalance.

Validation is performed on an independent point cloud acquired on a different bridge. Besides classification metrics, a geometric check is carried out by comparing the classified cloud with a reference model, using Cloud-to-Cloud distances (C2C). This metric provides quantitative evidence of geometric coherence and highlights local discrepancies.

Table 3.12 — LOD parameters and criteria for IFC/FEM.

The following sections detail the technical implementation and the results obtained.

4.0.1 Software environment

The entire workflow was implemented in Python 3.10, using Open3D, PDAL, NumPy, SciPy and scikit-learn libraries, and CloudCompare for verification. The scripts were executed in the Anaconda environment, with Spyder as the IDE. The outputs are organised in a predefined directory structure that separates raw data, intermediate products, final results and logs. All parameters are stored in a configuration file to allow full reproducibility.

4.1 MOTIVATIONS FOR THE SUPERVISED APPROACH AND WORKFLOW OVERVIEW

This section formalises the supervised classification problem and introduces the set of local features used as input to the Random Forest classifier. Let P = {p_i} be the point cloud, with p_i = (x_i, y_i, z_i) and (optionally) additional attributes such as intensity. Each point is assigned a class label y_i ∈ {1,…,C}, where C is the number of semantic classes.

Feature vector for point i:

$$x_i = \left[f_1\left(\mathcal{N}_k(i)\right), f_2\left(\mathcal{N}_k(i)\right), \ldots, f_d\left(\mathcal{N}_k(i)\right)\right] \quad (84)$$

Equation (98) defines the feature vector associated with point i: each point is described by geometric functions $f_j(\cdot)$ computed on its neighbourhood $\mathcal{N}_k(i)$ (e.g., k-NN or r-ball), yielding a numerical vector $\mathbf{x}_i$ used by the classifier (Demantké et al., 2011; Pauly et al., 2003).

RF decision function (majority vote over T trees):

$$\widehat{y_i} = \text{mode}\{h_t(x_i)\}_{t=1}^{T} \tag{85}$$

Equation (99) represents the Random Forest decision rule: the point is assigned to the predicted class $\hat{y}_i$ through majority vote among the T trees $h_t(\mathbf{x}_i)$, all evaluated on the same feature vector (Breiman, 2001).

Precision, Recall, F1 (for class c): (Sokolova & Lapalme, 2009; Powers, 2011)

$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c} \tag{86}$$

$$\text{Recall}_c = \frac{TP_c}{TP_c + FN_c} \tag{87}$$

$$F1_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \tag{88}$$

Equations (100) formalise the class-wise c classification metrics: precision and recall are defined from the counts TP_c, FP_c and FN_c, while the F1-score is the harmonic mean between precision and recall, useful to assess class quality also under imbalance.

LAZ coverage within threshold d* (C2C) for class c: (Isenburg, 2013; CloudCompare, 2021; Lague, Brodu & Leroux, 2013)

$$\text{Cov}_c(d^*) = \frac{1}{N_c^{\text{LAZ}}} \left| \left\{ p \in P_c^{\text{LAZ}} : \min_{q \in C_c} |p - q|_2 \le d^* \right\} \right| \tag{89}$$

Equation (103) introduces the C2C geometric-coverage metric: for a given metric threshold d*, we compute the percentage of points in the reference cloud (LAZ) belonging to class c that lie within d* from the predicted geometry/cloud c of the same class. This measure links semantic correctness to metric coherence on real data.

In the case study, the training and validation clouds do not share exactly the same taxonomy: the railway bridge includes additional classes (e.g., rails/ballast, electrical line elements, site objects) that do not appear in the

second bridge. To make results comparable and to facilitate mapping towards BrIM/BIM and FEM environments, the quantitative analysis adopts a canonical set of classes. Class counts and C2C coverage within the threshold (validation only) are summarised in Figure 4.7 and Figure 4.8.

Tab. 3.13 — Summary of validation-only classes (not present in training).

class_id	class_label	class_points
10	Ground	464.202
12	Vegetation	878.018
16	Safety barriers	7.340.180
20	pile	10.399.743
22	Cross beams	12.358.985
26	Longitudinal beams	1.859.017
28	deck_slab	17.165.120

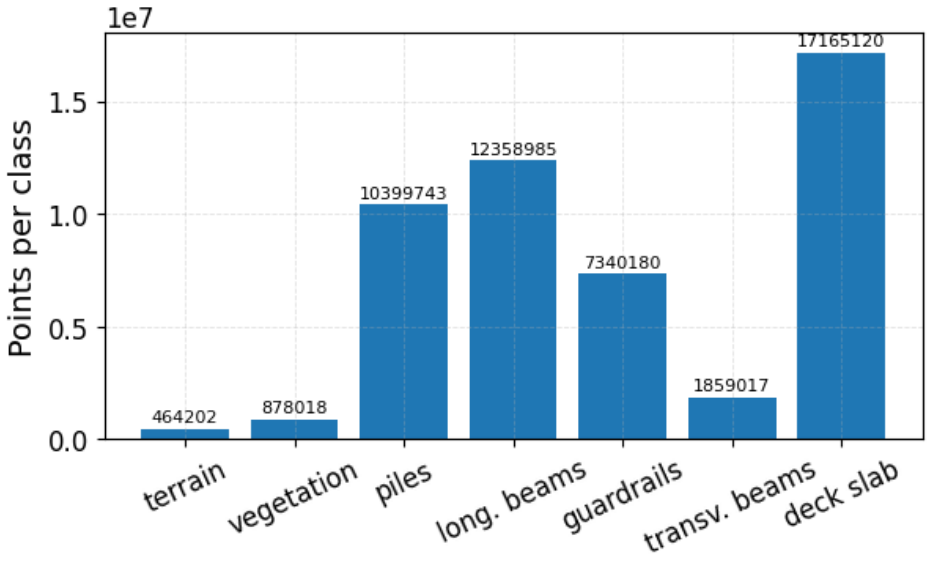


Fig. 4.7 — Point count per class in the point cloud

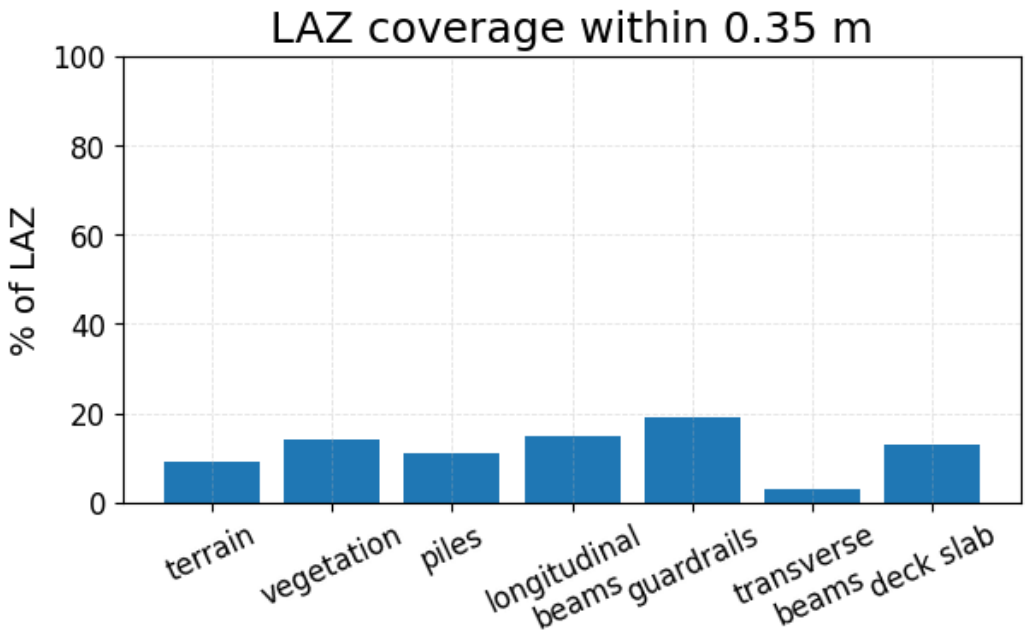


Fig. 4.8 — LAZ coverage within 0.35 m for each class (C2C)

4.2 GEOMETRIC DESCRIPTORS: DEFINITIONS AND SCALES

For each point i a neighbourhood N(i) is defined either by searching the k nearest neighbours or by collecting all points within a radius r. Given a point p and its neighbourhood $N(p) = \{p_i\}_{i=1..n}$, the centred data matrix $\bar{p} = (1/n)\sum_i p_i$ is built as D = [p1 − p̄, …, pn − p̄]^T, with p̄ the barycentre. The covariance matrix is C = (1/n) D^T D. The eigenvalues λ1 ≥ λ2 ≥ λ3≥ 0 and the associated eigenvectors v_1, v_2, v_3 describe the 3D dispersion of the neighbourhood. The neighbourhood can be defined with fixed cardinality (k-NN) or with metric radius (r-NN).. When point density is non-uniform, both versions are computed to cover different geometries (Pauly, Keiser & Gross, 2003; Demantké et al., 2011).

$$C = \frac{1}{n}\sum_{i=1}^{n}(p_i - \bar{p})(p_i - \bar{p})^T \tag{90}$$

and summarises the three-dimensional distribution of the neighbourhood. To compare regions with different densities, the eigenvalues are normalised to have unit sum,

$$\tilde{\lambda}\_i = \lambda\_i / (\lambda_1 + \lambda_2 + \lambda_3),\ i=1,2,3 \tag{91}$$

obtaining scale-independent and comparable features. In what follows, λ_i, denote the normalised eigenvalues.

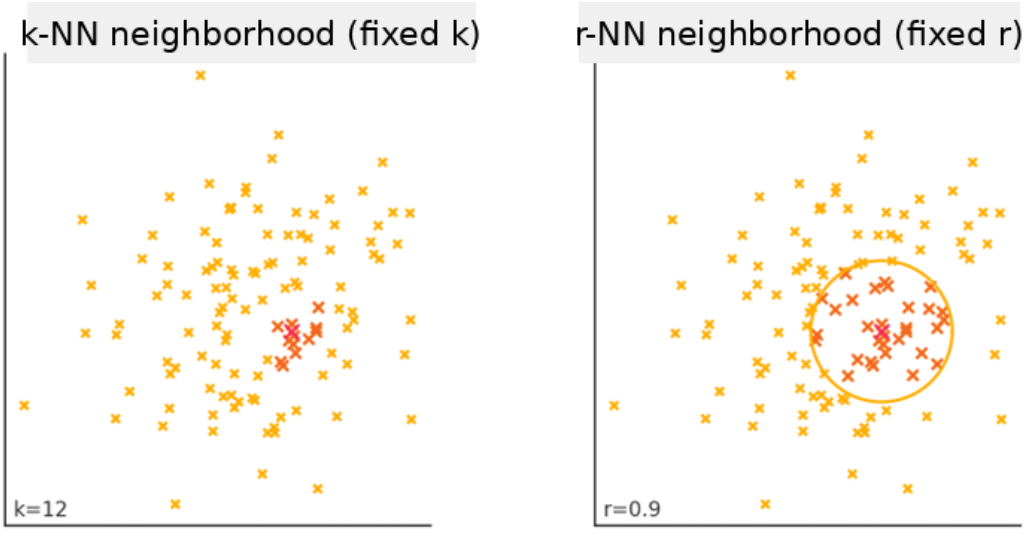


Fig. 4.9 — k-NN neighbourhoods (left) and r-NN neighbourhoods (right)

From the eigenvalues, a set of spectral features is derived, widely used in the literature for semantic segmentation of point clouds. These descriptors quantify linearity, planarity, sphericity and other characteristics of neighbourhood geometry. The formulas used are reported below, where $\lambda 1 \geq \lambda 2 \geq \lambda 3$ and all features are computed on the normalised eigenvalues.

Linearity

$$\mathrm{LIN} = \frac{\lambda_1 - \lambda_2}{\lambda_1} \tag{92}$$

It takes high values on slender and directional elements (rails, edges, borders).

Planarity

$$\mathrm{PLA} = \frac{\lambda_2 - \lambda_3}{\lambda_1} \tag{93}$$

High on nearly planar surfaces, such as the deck slab.

Sphericity

$$\text{SFER} = \frac{\lambda_3}{\lambda_1} \tag{94}$$

It measures the isotropy degree of the neighbourhood; it increases for vegetation or volumetric noise.

Anisotropy

$$\text{ANI} = \frac{\lambda_1 - \lambda_3}{\lambda_1} \tag{95}$$

It describes the width of the spectrum and is useful to separate directional structures from more isotropic regions.

Change of curvature

$$C_{\text{CURV}} = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \tag{96}$$

Fraction of variance associated with the third principal direction, correlated with local curvature.

Omnivariance

$$\text{OMNI} = (\lambda_1 \lambda_2 \lambda_3)^{1/3} \tag{97}$$

Geometric mean of variances, sensitive to three-dimensional disorder.

Eigen-entropy

$$\text{EIG\_H} = -\sum_{i=1}^{3} \lambda_i \ln(\lambda_i) \tag{98}$$

Maximised for uniform distributions; it helps discriminate structured regions from chaotic configurations.

Spectral energy

$$\sum\lambda = \lambda_1 + \lambda_2 + \lambda_3 (= 1 \text{ if standardised}) \tag{99}$$

With non-normalised eigenvalues it provides a measure of neighbourhood scale, useful to estimate effective density.

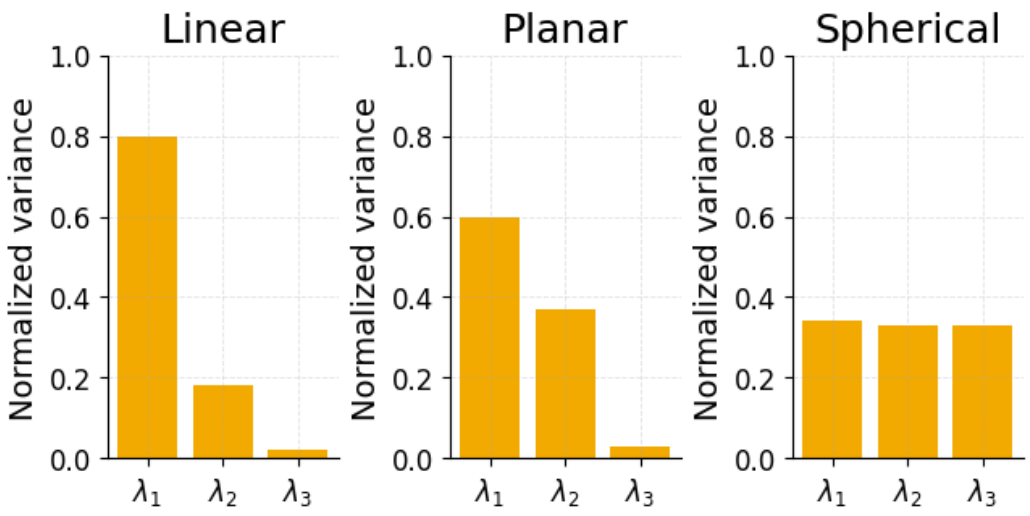


Fig. 3.37 — Per-class meshes on the training bridge.

C_{CURV}Figure 3.38 — Per-class meshes on the validation bridge.

Spectral features are complemented by local-orientation descriptors. The normal to the best-fit plane of the neighbourhood is approximated by the eigenvector associated with the smallest eigenvalue,

Local normal

$$\mathrm{n} = \mathrm{v}_3 \tag{100}$$

from which verticality is defined:

Verticality

$$\mathrm{VERT} = 1 - |\mathrm{n} \cdot \hat{\mathrm{z}}| \tag{101}$$

which ranges from 0 for horizontal surfaces to 1 for vertical surfaces. The angles of the direction of maximum extent with respect to the Cartesian axes,

Dominant tangential direction

$$\theta_x = \arccos(v_1 \cdot \hat{x}), \theta_y = \arccos(v_1 \cdot \hat{y}) \tag{102}$$

are used to distinguish longitudinal and transverse beams in consistent reference systems.

Additional auxiliary features complete the descriptive vector:

Metric density within radiusr

$$\rho_r = \frac{n}{\frac{4}{3}\pi r^3} \tag{103}$$

useful in the presence of local variations in sampling density.

Height Above Ground (HAG)

$$\mathrm{HAG} = z(p) - z_{\mathrm{ground}}(p) \tag{104}$$

Figure 3.39 — Mesh→NURBS conversion on structural elements (piles only): training (left), validation (right).

Roughness with respect to the mean plane

$$\mathrm{RUG} = \frac{1}{n}\sum_{i=1}^{n} d^2\,(p_i, \pi) \tag{105}$$

π Figure 3.40 — Grasshopper setup: levels for pile sections and solid definition.

4.3 DATASET AND SAMPLING STRATEGY

The datasets used include a training point cloud and a validation one. The construction of the supervised dataset starts from aligned LAZ/PLY clouds (EPSG 32632), adopting a train/validation/test split by spatial blocks in order to preserve local coherence and avoid spatial-leakage phenomena. Before splitting, a minimal cleaning is applied (removal of gross outliers and CRS check) and features are standardised (Roberts et al., 2017; Isenburg, 2013).

The canonical classes considered are: 10 = ground, 12 = vegetation, 20 = piles, 22 = longitudinal beams, 24 = guard-rail, 26 = transverse beams, 28 = slab/deck. Since the point distribution is strongly imbalanced, during training compensation strategies are adopted based on inverse-frequency weights,

$$w_c = \frac{N}{K \cdot n_c} \tag{106}$$

and, when needed, controlled undersampling of the majority classes. The stratified train/validation split is defined as

$$| D_{\text{train}} |= \gamma \mid D \mid, | D_{\text{val}} |= (1-\gamma) \mid D \mid \tag{107}$$

with $\gamma \in [0,1]$ typically set to 0.8.

The supervised classifier is initialised through a stratified manual seed, obtained by selecting a representative subset of points (about 5–15% of the total, depending on density), spatially distributed and including all structural classes, as shown in Figures 4.11–4.13.

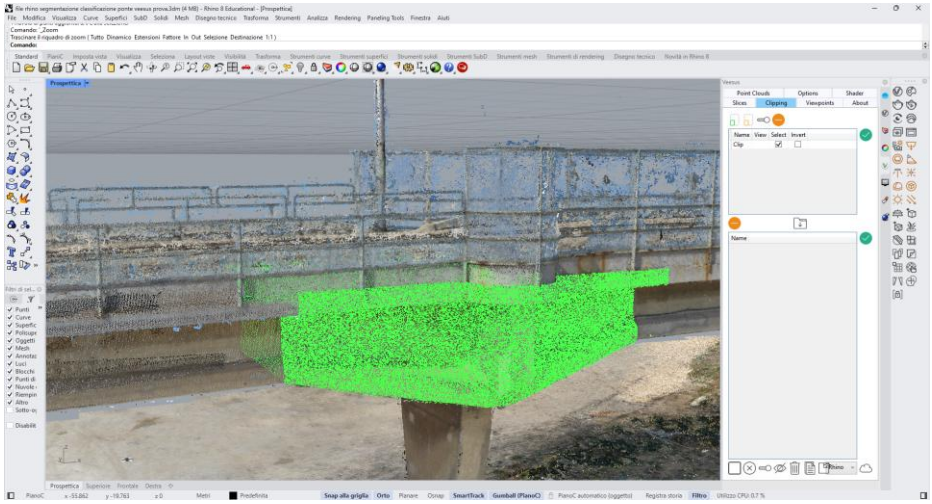


Fig. 4.11 — Coarse pre-selection of a point-cloud region including a transverse beam

Figure 3.41 — “Barriers” macro-elements (detail): planes + curves + loft + solid.

Fig. 4.12 — Pre-selection of the point cloud: a pile with respect to the ground

Fig. 4.13 — Partial pre-selection of the point cloud: pile detail

The selection of point-cloud portions was carried out in Rhinoceros. The software includes a dedicated plug-in for importing and managing large point clouds, after conversion to the proprietary VPC format. The original cloud in E57 format is converted to VPC and then imported into the working environment. Rhinoceros provides different visualisation modes, including X-Ray view, which allows selecting portions of the cloud that are not immediately visible in standard views, as illustrated in Figure 4.14. In addition, the software allows observing and interacting simultaneously with orthogonal views (XY, XZ, YZ) and 3D views, providing a high level of control during class-selection operations, as shown in Figure 4.15.
The selected cloud portions are finally re-exported in .LAS, .LAZ and .E57 formats for further analyses and visualisations in CloudCompare (CloudCompare, 2021; Lague, Brodu & Leroux, 2013).

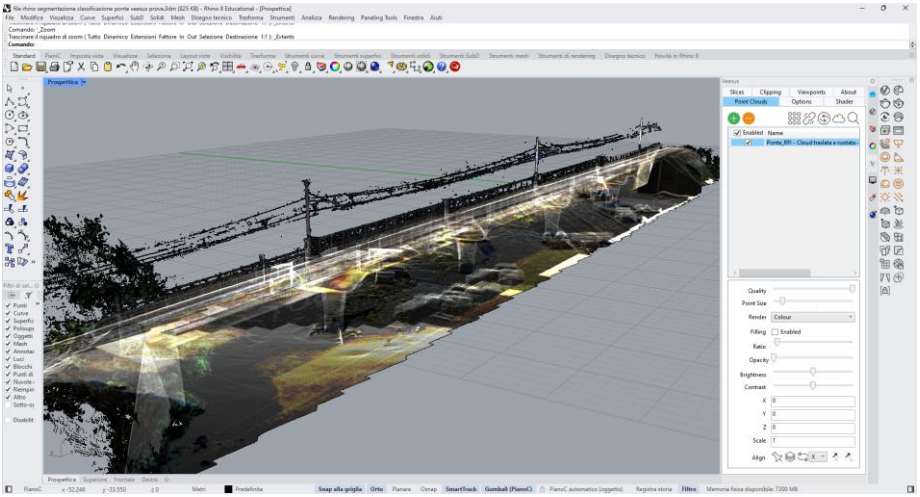


Fig. 4.14 — X-Ray visualization

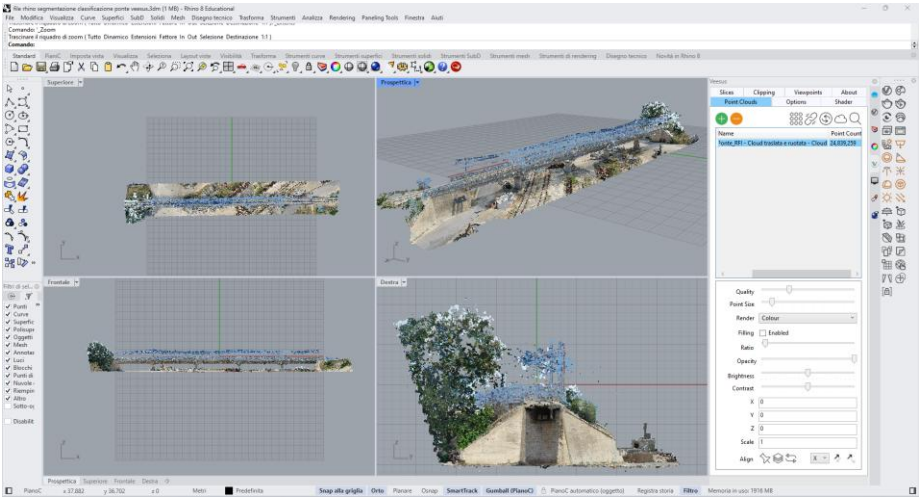


Fig. 4.15 — Orthogonal and 3D view visualization

Starting from the initial seed, a Random Forest classifier is trained and then applied to the whole point cloud. In Rhinoceros, the areas with the highest probability of error are inspected, typically located near class boundaries, in shadowed zones or at contacts between vegetation and structures (e.g., near abutments). Misclassified cloud portions are corrected using segmentation tools and reintegrated into the training seed. In parallel, points characterised

by maximum model uncertainty are identified and prepared for fast manual labelling (Breiman, 2001; Settles, 2012).

The uncertainty associated with a point x is defined as:

$$\mathrm{u(x)} = 1 - \hat{\mathrm{p}}\_\mathrm{max(x)} \tag{108}$$

where $\hat{p}_{\max}(x)$ represents the maximum predicted probability among classes. Higher uncertainty values $u(x)$ indicate ambiguous points, typically informative for subsequent retraining.

In practice, the procedure is implemented by iteratively running scripts in Spyder: after each iteration, the model is retrained on the updated dataset and probability/uncertainty maps are refreshed. The operator then corrects misclassifications in selected regions and confirms a limited number of uncertain points. Each iteration produces an updated manifest and a set of intermediate outputs (PLY per class) for visual inspection.

This cycle continues until the improvement in macro-F1 becomes marginal or the manual-review budget is exhausted. The final training set is then frozen and used to train the definitive model deployed on the full training and validation clouds.

The main operational blocks of the workflow are described below.

A) Manual seeding (seed).
   Import the E57 cloud into Rhinoceros (converted to VPC), isolate regions of interest by selecting points in X-Ray and orthogonal views, and manually assign class labels to build an initial seed S. The seed covers the canonical classes (ground, vegetation, piles, longitudinal beams Cross beams, deck_slab, guardrails) and is spatially distributed to include different geometric contexts. To mitigate imbalance, the seed is balanced by assigning class weights $w_c = 1/\max(\varepsilon, N_c)$ and sampling so that class supports are approximately uniform $N_c^{(\mathrm{seed})} \propto w_c$
B) Multi-scale feature computation (Demantké et al., 2011; Pauly et al., 2003).
   For each point in S and in the unlabeled pool U, compute spectral and auxiliary features at multiple scales (different k and/or r) and store

them in a feature matrix F. Standardise features using statistics computed on the training data.$k \in \{32,64\} r \in \{0,30,0,60\}$ m

C) Training and propagation.
$n_{\text{estimators}} \in [300,600], \tau \in [0,80,0,90]$ Table 3.14 — Folder structure, naming and outputs.

D) Internal validation and stopping criterion.
Monitor macro-F1 and class-wise metrics on an internal validation subset (held-out spatial blocks). Stop when improvement falls below a tolerance ε for consecutive iterations or when the manual-review budget is reached. Export the final classifier, the labelled dataset and the manifest.

The summary algorithm of the semi-automatic labelling loop is reported below.

```
Input:
U: set of unlabeled points
τ: confidence threshold
budget: maximum number of manual reviews
```

Output:

```
Labelled dataset S and trained model RF
Labelled dataset S
Algorithm 4.1 — Semi-automatic labelling with propagation


1: S ← initial manual selection (seed)
2: repeat
3: RF ← train Random Forest on S
4: P ← RF.predict_proba(U)
Figure 3.42 — BREP solid preview for slab and barrier classes.
6: R ← U \ A   # low-confidence cases
7: S ← S ∪ A   # update dataset
8: S ← S ∪ manual_review(R)   # targeted active review
9: U ← U \ (A ∪ R)
10: until improvement(F1_macro) < ε
or budget exhausted
11: export(S, RF, manifest, PLY_per_class)
```

Tab. 4.1 — Manual seeding: points per class and balancing ratio

Class	Label	Seed iniziale	Rapporto bil.	Selection notes
10	ground	2 500	0,60	planar patches with low roughness
12	vegetation	2 000	0,70	volumes with high sphericity
20	piles	3 000	1,00	verticality + curvature
22	longitudinal beams	2 400	1,10	high linearity
24	guardrails	1 500	1,50	metallic repeating patterns
26	Cross beams	1 800	1,30	direction ≈ transverse
28	deck_slab	2 800	0,90	strong planarity

Figure 3.43 — IFC 4.3 output: model in BIMVision and class-based layering.
Cross-validation is carried out using stratified k-fold (k = 5), ensuring that each fold contains a representative subset of each class. To limit spatial leakage, folds are defined over spatial blocks rather than random points whenever possible.
This strategy reduces the risk of over-optimistic estimates due to spatial autocorrelation. In addition, the same standardisation parameters (mean, standard deviation) are reused across folds to ensure comparability.
The average number of samples per class in each fold is approximated by:

$$m_c^{(\text{fold})} \approx \frac{n_c}{k} \tag{109}$$

Figure 3.44 — IFC structure: IfcBridgePart assignment and types.

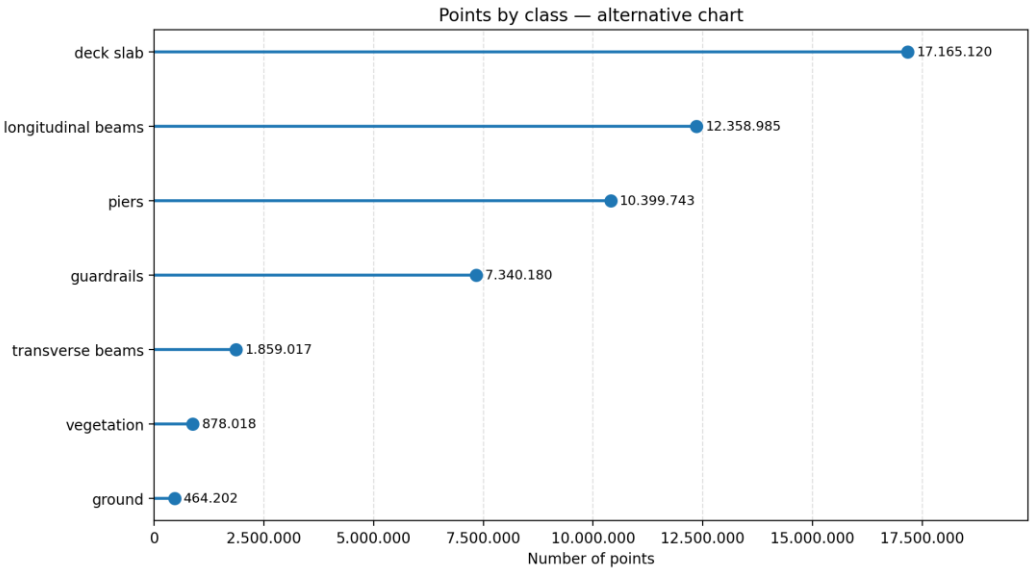


Fig. 4.16 — Per-class distribution (point counts)

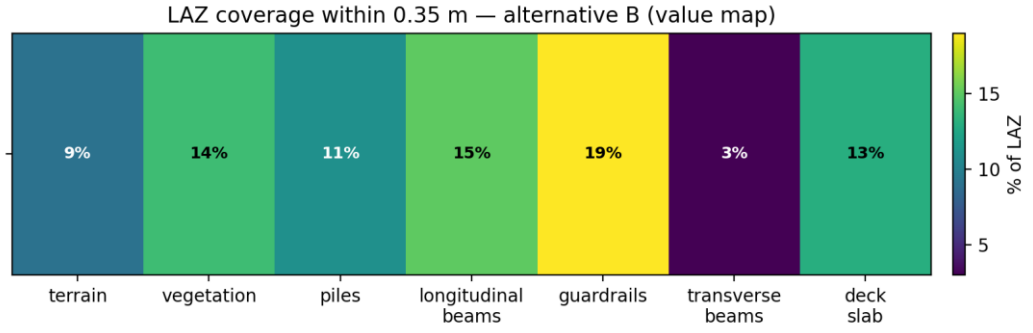


Fig. 4.17 — C2C coverage: LAZ (E57) points within 0.35 m from the class

As a further diagnostic step, the distributions of the main features are compared between training and validation to identify possible domain shift. For each feature, the Jensen–Shannon (JS) distance between histograms is computed and aggregated across classes. Large JS distances indicate descriptors that are less transferable and may require scale adjustment or normalisation. This analysis supports the selection of multi-scale parameters

and the interpretation of validation errors. The results of these checks guide any adjustments to balancing strategies and analysis scales (Lin, 1991; Shannon, 1948; Kullback & Leibler, 1951).

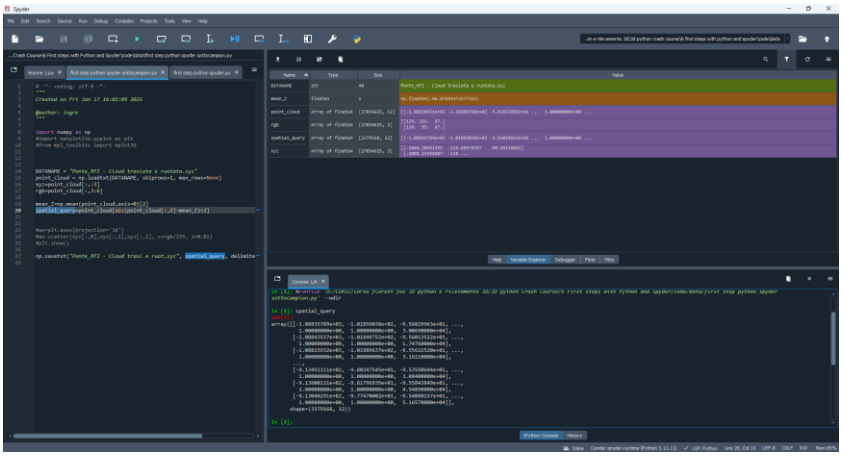


Fig. 4.18 — Point cloud (≈27,000,000 points): query analysis 3,379,168

The raw data were submitted to a script for preliminary statistical pre-analysis, to proceed with minimal and reproducible operations.

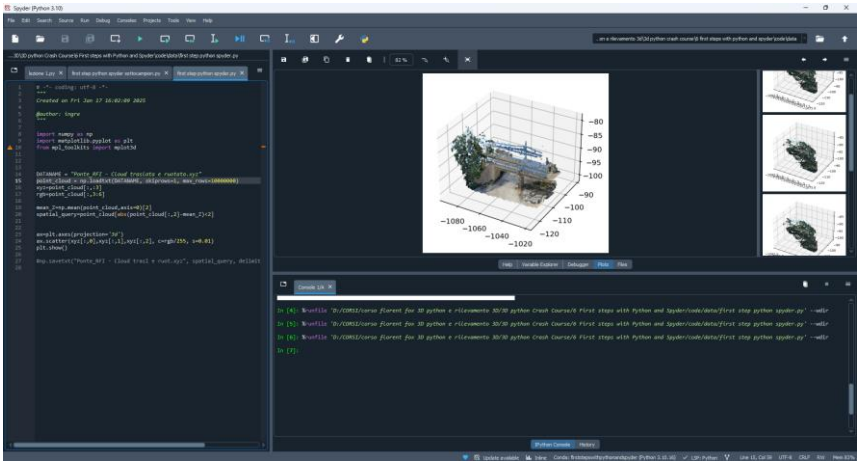


Fig. 4.19 — Example script for reading and preliminary statistical feature analysis of the point cloud

```
spatial_query=point_cloud[abs(point_cloud[:,2]-mean_Z)<1]
```

Fig. 4.20 — Visualization of the spatial_query

In summary, building the supervised dataset and applying semi-automatic labelling allow a practical trade-off between human cost and coverage. The manual seed is essential to correctly define class boundaries and to fix the taxonomy; uncertainty-driven propagation increases the amount of labelled data without a linear increase of effort. Anti-leakage measures (spatial split, stratified CV, domain-shift diagnostics) make the estimates more robust and transferable and set up the subsequent model training/validation phase (Hastie, Tibshirani & Friedman, 2009; Settles, 2012).

4.4 MODEL TRAINING AND VALIDATION

This section describes training the Random Forest on geometric features extracted from the supervised dataset. The goal is to obtain a stable and repeatable classifier that generalises across portions of the same structure and, as far as possible, across different projects. For this reason we adopt z-score standardisation, balancing strategies (weights and sampling) and cross-validation. A Random Forest is an ensemble of decision trees trained on bootstrap samples of the training set; at each split a random subset of

features is considered. The final class is determined by majority vote, and split quality is measured through impurity reduction. As impurity measure we use the Gini index, computed from class frequencies in the node (Strobl et al., 2007).

$$G = 1 - \sum_{c} p(c)^2 \tag{110}$$

where p_(c) is the frequency of class c in the node. The impurity reduction induced by a split is defined as:

$$\Delta G = G(\text{parent}) - \left[\frac{n_L}{n} G(L) + \frac{n_R}{n} G(R)\right] \tag{111}$$

In the presence of class imbalance, weights inversely proportional to frequency are applied. The class weights w_c are assigned according to the definition introduced in §4.3.

Performance is estimated by means of stratified 5-fold cross-validation, so as to preserve class proportions in each fold. The mean estimate of a metric is:m

$$\text{CV} = \frac{1}{k} \sum_{i=1}^{k} m_i \tag{112}$$

Class-wise metrics include Precision (P), Recall (R) and F1, with:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{113}$$

At the aggregate level we report macro-F1 (unweighted mean) and weighted-F1 (mean weighted by support). To evaluate model robustness with respect to agreement expected by chance, Cohen's coefficient is also used (Cohen, 1960):

$$\kappa = \frac{p_0 - p_e}{1 - p_e} \tag{114}$$

Within the semi-automatic cycle, per-point probability estimates are also used for controlled label propagation (accept/reject beyond a threshold) and for building uncertainty maps that are inspected during the review stage.

Model hyperparameters are fixed to balance expressiveness and generalisation on the real point cloud. The adopted configuration is reported in Table 4.3. Since the classes have very different cardinalities, performance assessment must be accompanied by an explicit handling of imbalance during training; to this end, each sample is assigned a weight inversely proportional to its class size, as formalised in Eq. (131).

$$w_c \propto \frac{1}{N_c} \tag{115}$$

(with possible more conservative variants $1/\sqrt{N_c}$), with normalization $\sum_c w_c = 1$

Tab. 4.3 — Model and pre-processing configuration

Parametro	Value
n_estimators	400
max_depth	None
min_samples_split	2
min_samples_leaf	1
max_features	sqrt
bootstrap	True
class_weight	balanced_subsample
oob_score	True
random_state	42
validation	5-fold stratificata
pre-processing	z-score
feature set	18×2 scale

In addition, controlled undersampling is applied, limiting the maximum number of points per class. This prevents the model from being dominated by the ground or vegetation classes and improves sensitivity to thin elements (guardrails and beams).

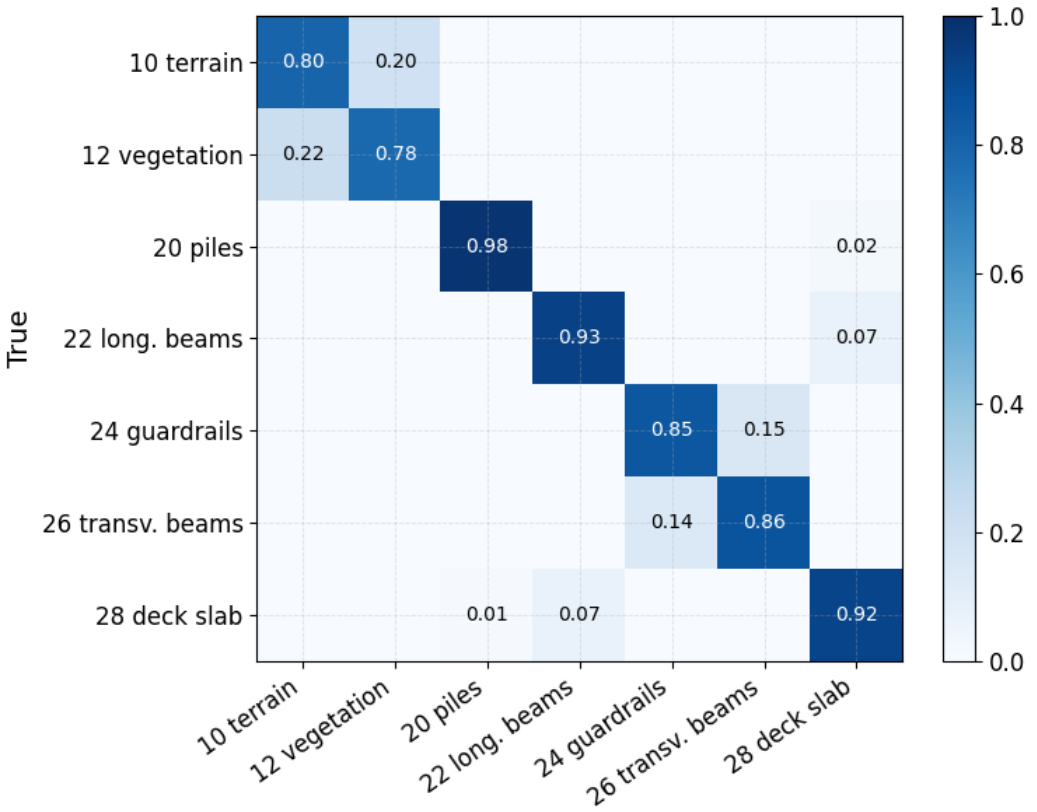


Fig. 4.21 — Confusion matrix (mean over the 5 folds, rows = ground truth; normalised values)

The confusion matrix highlights good separability for piles and slab, while the main confusions occur between guardrails and cross beams in spans with dense metallic parapets and at contact regions between slab and longitudinal beams. Nevertheless, the diagonal dominates and the macro-F1 remains stable across folds.

Tab. 4.4 — CV metrics (5-fold mean, P3)

Accuracy (OA)	**F1-macro**	**F1-weighed**	**Kappa**
0,858	0,822	0,846	0,811

Beyond aggregate metrics, feature-importance analysis provides insight into which descriptors drive classification. Table 4.5 reports the top-10 features

ranked by mean decrease in impurity. The ranking is consistent with geometric intuition: verticality and HAG discriminate ground/vegetation from superstructure; linearity and planarity separate beams; roughness and omnivariance capture vegetation and noisy regions.

Tab. 4.5 — Top-10 features (training, RF)

Feature	Important(0–1)
z_height	0.4310
sum_e	0.0818
eigenentropy	0.0803
linearity	0.0680
planarity	0.0662
omnivariance	0.0617
verticality	0.0577
sphericity	0.0512
anisotropy	0.0509
change_curvature	0.0509

Per-class performance is reported in Table 4.6. Piles achieve high precision and recall due to their distinctive vertical geometry. The slab is also well separated thanks to high planarity. Guardrails show lower recall in some folds because of their thin geometry and partial occlusions, but performance improves when larger neighbourhood scales are included.

Tab. 4.6 — Per-class report (fold mean, training)

Class (ID)	Label	P	R	F1
10	ground	0,76	0,72	0,74
12	vegetation	0,71	0,69	0,70
20	piles	0,91	0,88	0,90
22	longitudinal beams	0,87	0,86	0,86

24	guardrails	0,81	0,83	0,82
26	beams_transversal	0,75	0,79	0,77
28	deck_slab	0,90	0,88	0,89

The RF configuration uses 400 trees, max_features = sqrt(M) and a maximum depth tuned to avoid overfitting. The number of trees was selected after checking convergence of the out-of-bag (OOB) error. The final configuration represents a good compromise between stability and computational cost, and it is robust to moderate noise and missing data typical of LiDAR acquisitions. Another robustness indicator is the stability of the feature ranking: a Spearman correlation ρ close to 1 across folds indicates consistent model behaviour.

Overall, training results show that the selected feature set and the RF model provide a solid basis for subsequent validation on an independent bridge point cloud.

To qualitatively assess classification, the training cloud was visualised and coloured by predicted class, and additional maps were generated for model confidence and uncertainty. Figures 4.22–4.26 show representative views and zooms.

For a point x, the RF estimates the probability of class k as the average of the votes of the trees in the ensemble:

$$p_k(x) = \frac{1}{T}\sum_t \mathbf{1}[\,\text{albero}_t(x) = k] \tag{116}$$

Model confidence is defined as $\max_k p_k(x)$, while local uncertainty can be quantified through entropy:

$$H(x) = -\sum_k p_k\,(x)\,\ln\, p_k(x) \tag{117}$$

Colour maps derived from these quantities make it possible to visually inspect the behaviour of the classifier and to target regions for potential review.

The following figures illustrate (i) the classified training cloud; (ii) zooms on key structural elements (piles, beams, slab and guardrails); and (iii) confidence and uncertainty maps. The goal is to visually corroborate quantitative metrics and to identify systematic failure modes.

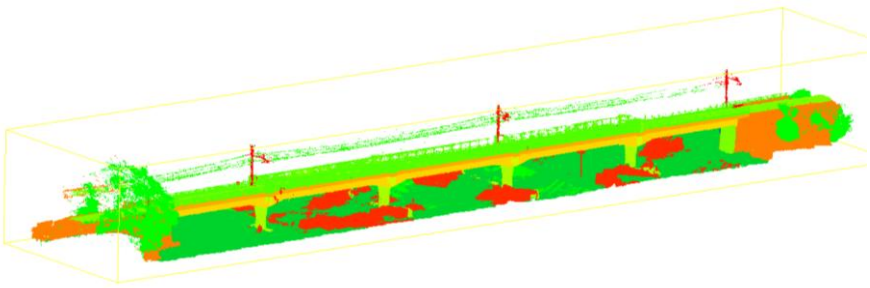


Fig. 4.22 — (a) Overall view of the training cloud coloured by class

Pile detail: stable labelling along the height; limited leakage towards vegetation/ground; clear verticality signature.

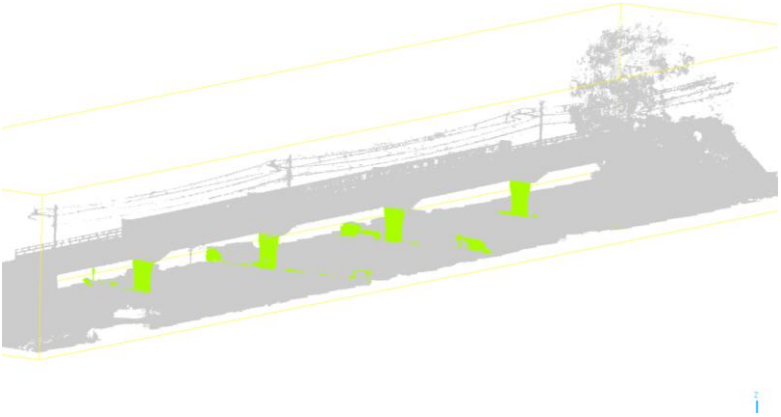
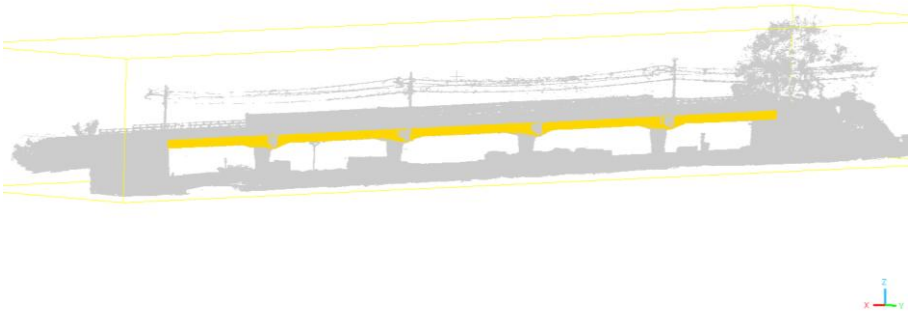
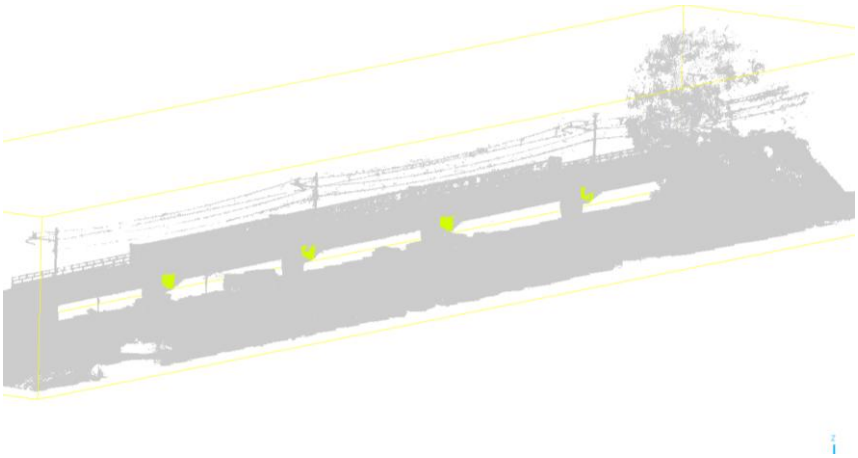


Fig. 4.23 — (b) Zoom 1: piles and supports

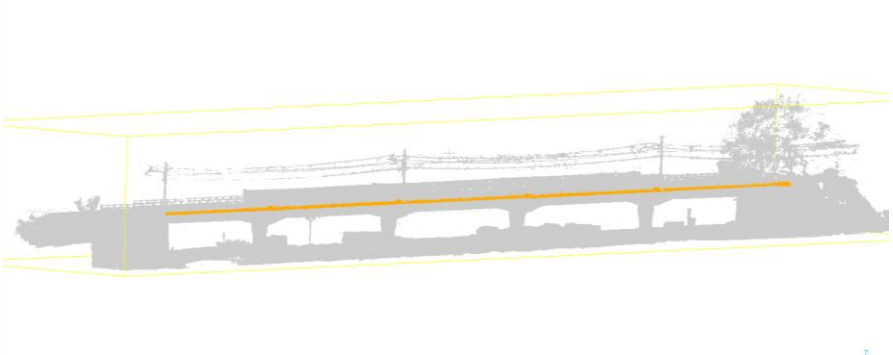
High linearity and separation among longitudinal beams, Cross beams and deck_slab; consistent local normals; limited contamination on edges.



a)



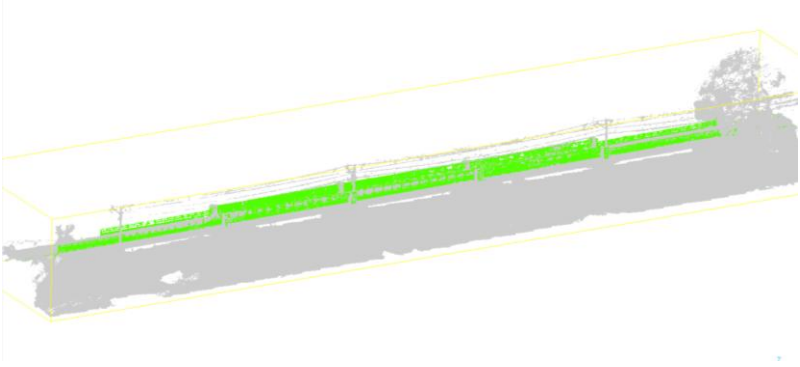
b)



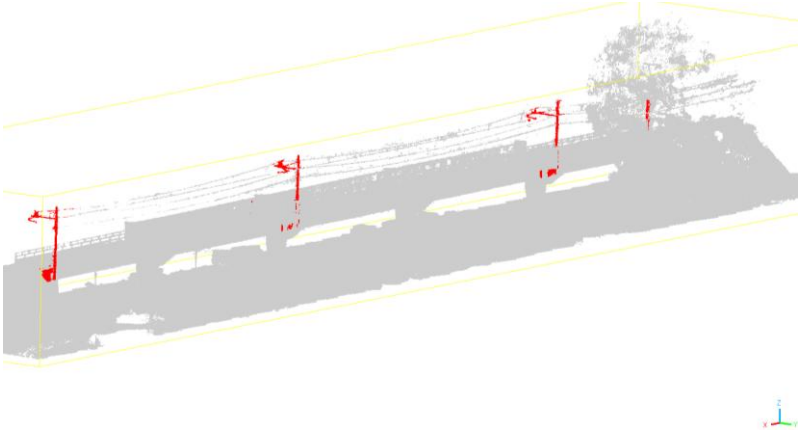
c)

Fig. 4.24 — (a); (b); (c) Zoom 2: longitudinal beams, cross beams and deck_slab edge

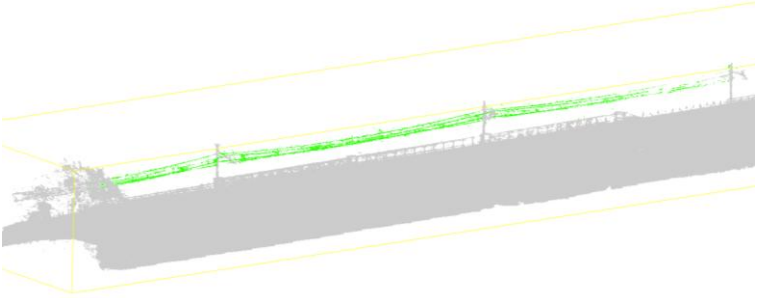
Behaviour on thin elements: any lower confidence is confined to occluded regions or to highly reflective areas.



a)



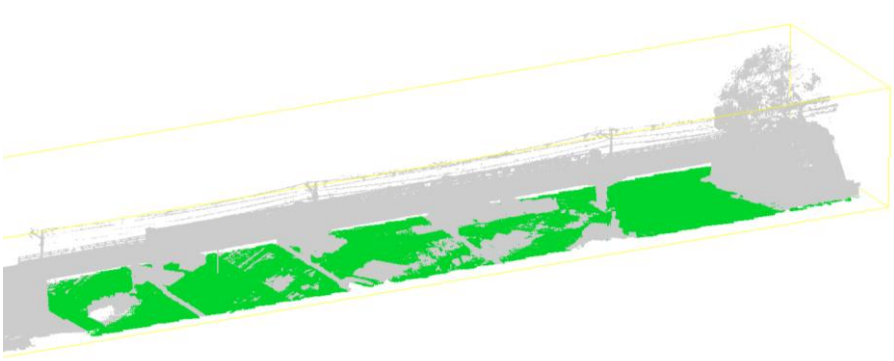
b)



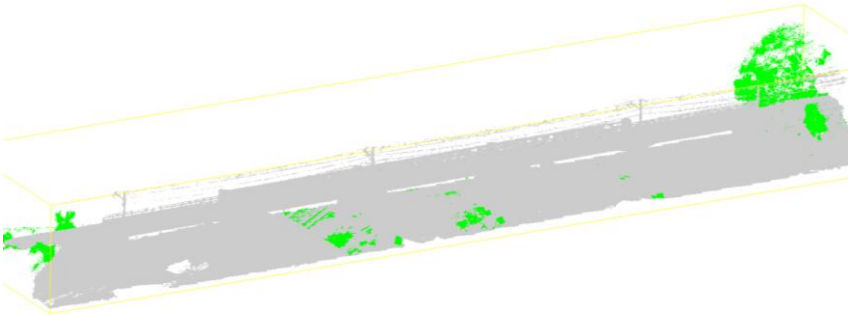
c)

Fig. 4.25 — (a); (b); (c) Zoom 3: guardrails and thin details such as electric-line poles and cables

Confidence map: high confidence values on large planar/linear surfaces such as the ground; uncertainty peaks localised at vegetation interfaces and around occlusions.



a)



b)

Fig. 4.26 — (a) and (b) Zoom 4: confidence map for ground and vegetation

4.5 VALIDATION ON POINT CLOUD AND C2C CHECK

This section assesses the generalisation capability of the trained classifier, applied without any re-tuning to an independent point cloud from the second bridge. Validation is conducted in black-box mode: the same multi-scale geometric features and the same standardisation learned during training are used, with no dataset-specific adaptations (Demantké et al., 2011; Pauly et al., 2003).

Performance is analysed through two complementary levels of control. On one hand, classification metrics are considered (confusion matrix, Accuracy, macro-F1, weighted-F1 and Kappa coefficient); on the other hand, a geometric-metric Cloud-to-Cloud (C2C) verification is performed against the aligned LAZ cloud, evaluating coverage within a metric threshold between 0.30 and 0.35 m (Sokolova & Lapalme, 2009; Powers, 2011; Cohen, 1960; Isenburg, 2013).

Operational parameters and artefacts used in the validation phase are summarised in

Tab. 4.7 — A: Validation parameters and assets

Voce	**Choice/value**
Model	Random Forest (n_estimators ≈ 400; max_depth = None; bootstrap = True)
Feature (multi-scala)	spectral features from eigenvalues (linearity, planarity, sphericity, anisotropy, surface variation, omnivariance, eigen-entropy, sum_of_eigenvalues), normal/verticality, density ρ_r, HAG, roughness
Scale tipiche	k={32, 64} e/o r={0,30 m; 0,60 m}
Normalisation	z-score con μ, σ del training
Classes	10 ground; 12 vegetation; 20 piles; 22 longitudinal beams; 24 guardrails; 26 cross beams; 28 deck slab
Soglia C2C	τ = 0,35 m (LAZ coverage within τ)
Reference images	Confusion matrix (Fig. 4.27);

The confusion matrix (Figure 4.27) provides an operational view of the most frequent confusions. Overall, the diagonal dominates and the main errors concentrate on thin metallic elements and on boundary regions between vegetation and ground.

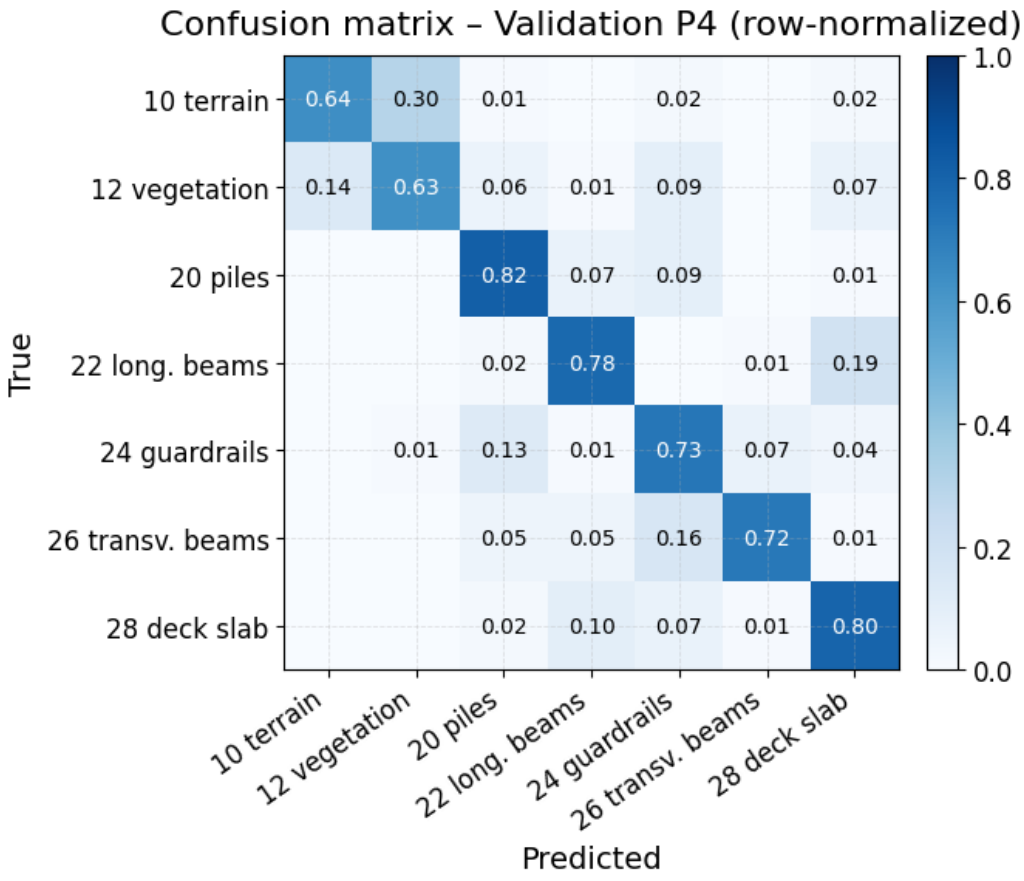


Fig. 4.27 — Confusion matrix on the validation dataset (P4; rows = ground truth; values normalised)

The adopted metrics provide complementary information on classifier behaviour. Macro-F1 emphasises balance among classes and is therefore particularly suitable under imbalance; weighted-F1 more faithfully reflects the real data mix, while Cohen's coefficient (κ) measures agreement beyond chance, highlighting situations in which high accuracy could be artificially driven by dominant classes. The confusion matrix provides the most operational view of inter-class confusions and helps identify recurring error patterns: swaps between guardrails and cross beams concentrate in spans with highly jagged metallic parapets, where it is beneficial to increase the metric scale to stabilise normal estimation and to apply a more restrictive verticality threshold; confusion between vegetation and ground appears near the terrain plane when HAG filters are conservative, suggesting the use of a median window on HAG and of sum_of_eigenvalues as a proxy for effective density; swaps between longitudinal beams and deck_slab are limited but present near cantilever edges, where planarity improves separation but requires an intermediate scale to avoid beams being absorbed into the slab.r
The metrics are computed from the confusion matrix (rows = ground truth, columns = predicted). Overall accuracy is defined as:$K \times K$

$$\mathrm{OA} = \frac{\text{sums of the correct answers}}{\text{the sum of all cases}} \tag{118}$$

For each class c, k, Precision, Recall and F1 are computed. These per-class metrics are defined in §2.7.
Aggregate metrics are expressed as:

$$F1_{\text{macro}} = \frac{1}{K}\sum_{k} F\,1_k \tag{119}$$

$$F1_{\text{pesato}} = \frac{\sum_k (F\,1_k \cdot \text{support}_k)}{\sum_k \text{support}_k} \tag{120}$$

Overall robustness is evaluated through Cohen's coefficient:

$$\kappa = \frac{\mathrm{OA} - p_e}{1 - p_e} \tag{121}$$

where p_e is obtained from the products of the normalised row and column marginal frequencies.

Alongside these classification metrics, a geometric-metric Cloud-to-Cloud (C2C) check is performed, based on the coverage within a threshold τ

$$\mathrm{Coverage}(\tau) = 100 \cdot \frac{\{\text{points LAZ con } d_{\mathrm{C2C}} \leq \tau\}}{\{\text{points LAZ Total}\}} \tag{122}$$

(computed also per class), and on the 95th percentile of the C2C distances:

$$p95_{\mathrm{C2C}} = \mathrm{quantile}_{0.95}\{d_{\mathrm{C2C}} \mid \text{classe} = k\} \tag{123}$$

The values obtained on the validation dataset (P4) are summarised in Table 4.8, which reports the global classification metrics and enables an immediate reading of the overall performance of the model. They should be interpreted jointly with the row-normalised confusion matrix (Figure 4.27).

Tab. 4.8 — B Global metrics (validation, multi-scale RF)

Metrica	Value (P4)
Accuracy (OA)	**0,782**
F1-macro	**0,734**
F1-pesato	**0,766**
Kappa	**0,701**

Tab. 4.9 — C: Precision (P), Recall (R), F1 per class (P4)

Class (ID)	Label	P	R	F1
10	ground	0,68	0,64	0,66
12	vegetation	0,61	0,63	0,62
20	piles	0,84	0,82	0,83
22	longitudinal beams	0,80	0,78	0,79
24	guardrails	0,69	0,73	0,71
26	beams_transversal	0,66	0,72	0,69
28	deck_slab	0,84	0,80	0,82

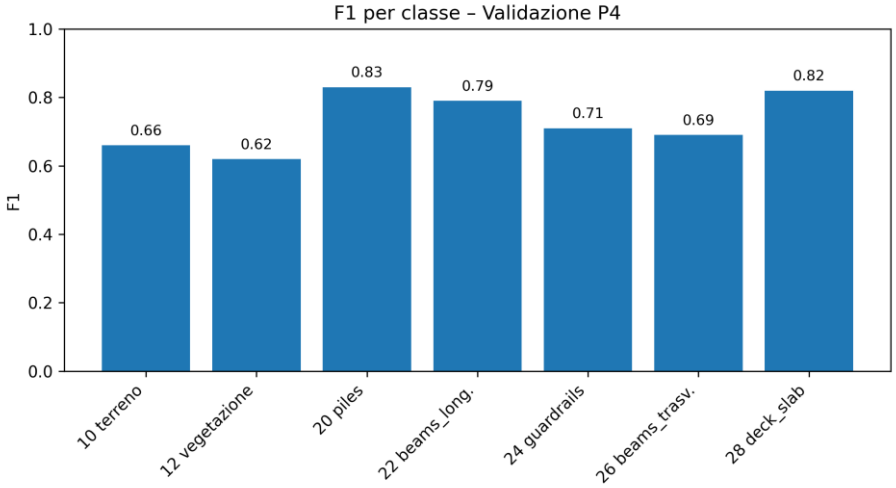


Fig. 4.28 — Per-class F1 on the validation dataset (P4)

Tab. 4.10 — C2C coverage per class (τ = 0.35 m)

Class	**Coverage ≤ 0,35 m [%]**	**p95_C2C [m]**	**Nota**
ground (10)	9	0,52	Noise and topographic irregularities
vegetation (12)	14	0,65	Canopies/branches → larger distance
piles (20)	12	0,39	Good verticality, glossy edges
longitudinal beams (22)	15	0,42	High planarity, but occlusions
guardrails (24)	19	0,45	Subtle structures and reflections
beams_transversal (26)	3	0,50	Small elements, aliasing

deck_slab (28)	13	0,41	Large planar areas, good adherence

4.6 VISUALIZATION AND RESULTS ANALYSIS

Beyond the aggregated metrics, the quality of the classification must also be evaluated in the spatial domain; therefore, the results are inspected directly on the labelled point cloud (PLY/CSV) in CloudCompare, maintaining a consistent colour coding for each class. In practice, the assessment combines per-class views (Figures 4.30–4.38), per-class reports (Table 4.11), and local checks at edges/joints and in noisy areas.

Fig. 4.30 — Fully unclassified point cloud.

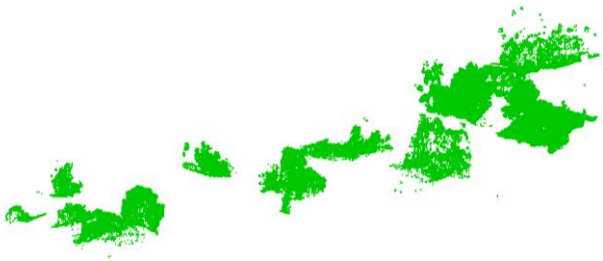


Fig. 4.31 — Classified point cloud: class 12 (vegetation).

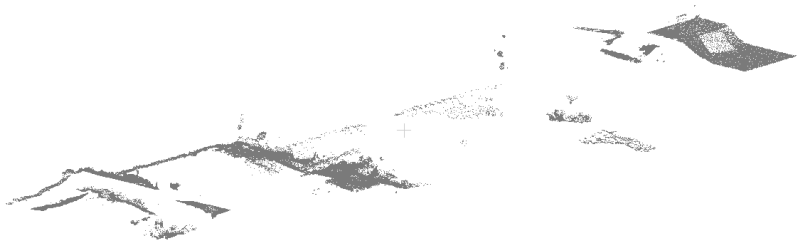


Fig. 4.32 — Classified point cloud: class 10 (ground).

Fig. 4.33 — Classified point cloud: class 20 (pile).

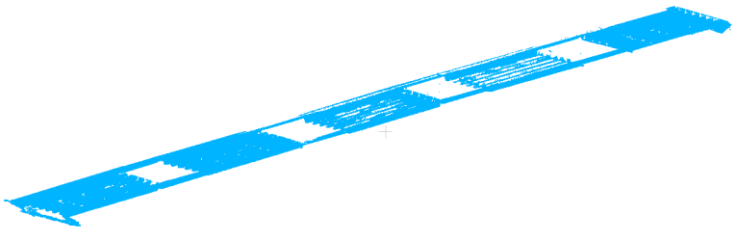


Fig. 4.34 — Classified point cloud: class 22 (longitudinal beams).

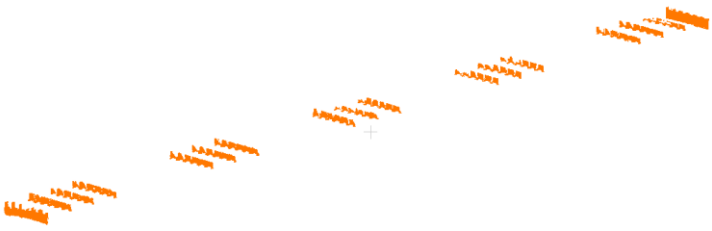


Fig. 4.35 — Classified point cloud: class 26 (transverse beams).

Fig. 4.36 — Classified cloud: class 28 deck slab

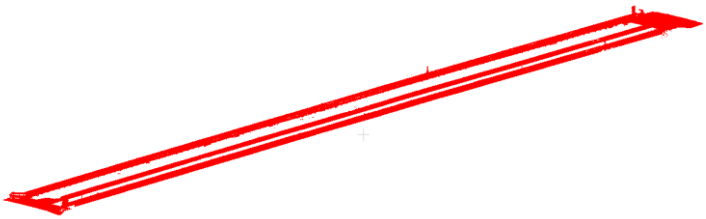


Fig. 4.37 — Classified cloud: class 24 guardrails

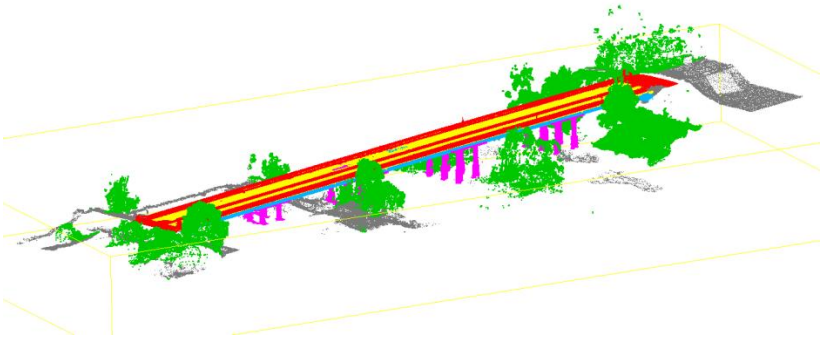


Fig. 4.38 — Fully classified point cloud

Alongside the qualitative views, we also consider concise indicators that are useful for diagnosing the fully classified point cloud: the number of points per class n_k, the relative frequency f_k, and profiles along an axis (bin counts). Denoting by N the total number of points, the frequency of class k is defined as

$$f_k = \frac{n_k}{N} \tag{124}$$

whereas, expressed as a percentage, it amounts to $p_k = 100 \cdot f_k$.

In addition, local exceedances of the threshold are used to drive targeted inspection and to assess whether errors are due to classification, registration or acquisition artefacts.

The C2C coverage threshold τ = 0.35 m yields a good compromise: it is sufficiently tight to detect misalignments but tolerant to residual noise and minor deformations. In practice, most correctly classified points lie within the threshold; outliers correspond to vegetation and to the boundaries of metallic parapets.

Tab. 4.11 —A— Point distribution by class (P4): counts n_k and proportions p_k.

Class (ID)	**Label**	n_k	f_k	p_k [%]	**Key remarks**
10	ground	464.202	0,0092	0,9	Reduced share: Ground level/embankments in ROI
12	vegetation	878.018	0,0174	1,7	localised presence (canopies/branches), variable density

20	piles	10.399.743	0,2061	20,6	dominant vertical elements; good geometric separability
22	longitudinal beams	12.358.985	0,2449	24,5	Linear components extended along the spans
24	guardrails	7.340.180	0,1455	14,5	subtle but widespread elements; sensitive to reflections/occlusions
26	Cross beams	1.859.017	0,0368	3,7	localised presence and small section; low elevation expected
28	deck_slab	17.165.120	0,3401	34,0	large planar surface; dominates by extension

The results confirm the validity of the proposed pipeline and highlight the importance of combining semantic metrics with geometric checks, especially when the goal is to export the classified geometry towards BIM/FEM environments.

In the case in question (P4), the total number of points is $N = 50.465.265$. Finally, all outputs are exported in standard formats (PLY/CSV/IFC) along with metadata to guarantee traceability and reproducibility.

Class	**Typical mistakes**	**Visual cues**	**Feature guida**	**Suggested action**
10 ground	FP with low vegetation	irregular texture, low dimension	sphericity, HAG	threshold HAG + embankment edge examples
12 vegetation	exchange with parapets	volumetric noise, reflections	eigenentropy, roughness	r-ball ↑ + planarity filters
20 piles	confusion with cross-beams	cilindriche vs lineari	verticality, curvature	examples on slender piers + medium k

22 longitudinal beams	conflict with deck slab edges	linear almost planar	linearity, planarity	topological constraints (continuity)
24 guardrails	confusion with cross-beams	thin elements, variable density	roughness, linearity	r-ball ↓ + spans with shadows
26 beams_transversal	exchange with parapets	Short lengths	linearity, curvature	k-NN ↑ (medium scale) + targeted examples
28 deck_slab	vegetation intrusion	Cracks, noisy margins	planarity, sum_eigenvalues	larger r-ball + roughness filters

4.7 CONCLUDING REMARKS AND OUTLOOK

This chapter has made the classification pipeline for infrastructural point clouds operational, from building the supervised dataset to export towards BIM/FEM. The comparison between P3 (training) and P4 (validation) highlights a moderate drop in performance (Table 4.12), consistent with a domain shift related to density, noise and survey conditions. The confusion matrix confirms that residual errors concentrate on pairs of geometrically similar classes (guardrails ↔ cross beams and ground ↔ vegetation), while the main structural classes remain more stable. The chapter has made the classification pipeline for infrastructural point clouds operational: construction of the supervised dataset, RF training, validation on an independent cloud and traceable export towards BIM/FEM (Sokolova & Lapalme, 2009; Powers, 2011; Quiñonero-Candela et al., 2008).

Tab. 4.12 — A: Deviations between training (P3) and validation (P4) and possible causes

Indicator	P3	P4	Δ (P4−P3)	Possible cause
Accuracy (OA)	0,858	0,782	−0,076	Different density/noise, residual vegetation
F1-macro	0,822	0,734	−0,088	Shift between related classes (24 vs 26), edges/contact areas
F1-pesato	0,846	0,766	−0,080	Higher weight on dominant classes (distribution effect)
Kappa	0,811	0,701	−0,110	Increased confusion between geometrically similar pairs

C2C ≤ 0.35 m (val.)	—	coerente	—	Metric check on aligned LAZ (validation only)

The comparison between the training project (P3) and the independent bridge (P4) shows a moderate performance drop, consistent with a domain shift driven by density, noise and survey conditions. Table 4.12 summarises the main indicators; the confusion matrix shows that errors concentrate on geometrically similar class pairs (e.g., guardrails ↔ cross beams and ground ↔ vegetation), especially near edges and shadowed zones. After export, an operational checklist summarises the minimum steps for replicability: CRS/units and registration check, density/outlier control, consistency of class coding, acceptance thresholds on confidence and—in validation—C2C checks per class and in critical areas. The goal is to prevent local errors from propagating downstream into parametric models and, especially, FEM simplifications. In this phase, interoperability is not treated as an accessory step but as an integral part of the pipeline: the aim is to turn the classified cloud into a set of reusable outputs within modelling and analysis workflows (Rhino/Grasshopper, IFC/HBIM, FEM), preserving semantic and metric coherence. In practice, export is not only about "saving files", but about ensuring that results are verifiable, comparable across runs and reusable downstream without information loss (e.g., class, confidence and—only for validation—metric indicators such as C2C). The first requirement is geometric consistency across different environments. Therefore, every exchange must preserve (or explicitly declare) the reference system, units and applied transformations. The passage from global coordinates (CRS) to a local system, when needed, can be compactly described by the affine transformation: (CloudCompare, 2021; Lague, Brodu & Leroux, 2013)

$$x_{\text{local}} = s\,R\,x_{\text{global}} + t \tag{125}$$

where R is a rotation/registration, s is a scale factor (typically unitary) and t is a translation (for example, a global shift). The alignment quality, when derived from registration procedures or correspondences, can be summarised by an RMS error:Rst

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|x_i - \hat{x_i}|_2^2} \tag{126}$$

which provides a quick check of metric coherence after import/export. Furthermore, when reductions, grids or discretisations are applied (e.g., to make the cloud more manageable), it is useful to recall that a quantisation error exists, depending on grid resolution:

$$\varepsilon_q \leq \frac{1}{2}\, r_{\text{grid}} \tag{127}$$

In validation, these checks are complemented by an independent metric control based on Cloud-to-Cloud: coverage within a threshold τ measures what fraction of the real data lies within the metric tolerance with respect to the predicted geometry/cloud. This indicator, expressed as a percentage, is defined as:

$$\text{Coverage}(\tau) = 100\ \frac{|\{\, p : \text{C2C}(p) \leq \tau \,\}|}{N_{\text{LAZ}}} \tag{128}$$

and it is used only in validation, because it requires an external reference (aligned LAZ) and should not influence training. At the semantic level, the essential condition is to maintain a stable mapping between class IDs and labels (e.g., through a unique legend) and to preserve, in the exported files, the attributes required for downstream use. Operationally, the classified point cloud is therefore exported in formats that clearly preserve coordinates, colour and per-point attributes, and it is accompanied by a manifest file that records parameters, versions and paths of the generated artefacts. In this way, when the data are imported into Rhino/Grasshopper or translated into IFC/FEM objects, it is possible to transparently reconstruct the rationale of the choices, verify the consistency of transformations and repeat the whole process on different projects with comparable criteria.

Looking forward, robustness can be improved without changing the methodological framework by acting on four aspects: (a) enrich the seed and active review in ambiguous ROIs; (b) tune neighbourhood scales (k/r) for thin elements and joint areas; (c) apply lightweight post-processing rules to reduce systematic swaps; (d) calibrate the acceptance threshold and verify C2C on targeted samples (Isenburg, 2013).

CHAPTER 5 INFORMATION MODELLING STRATEGIES: COMPARISON BETWEEN SECTION-BASED AND PARAMETRIC APPROACHES

5.0 INTRODUCTION

This section describes the procedure for transforming an already classified point cloud into a model that can be used in BIM and FEM workflows. The focus is on geometric consistency, traceability of transformations, and repeatable quality controls.

Two complementary pipelines are proposed: (A) NURBS modelling in Rhinoceros, oriented to surfaces/solids and to HBIM/IFC export; (B) parametric modelling in Grasshopper, better suited to handling cross-section variations, process automation, and the generation and control of FEM meshes.

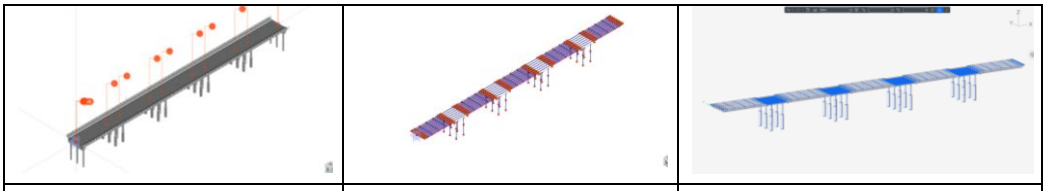
The goal is not to "redraw" the bridge, but to reconstruct a faithful geometry for subsequent analyses, maintaining alignment with the point cloud and with the class semantics.

The process starts from a point cloud in which structural and non-structural classes have already been identified and separated (Chap. 4). From this informational base, axes, cross-sections and guide surfaces are extracted, enabling the construction of a geometrically coherent model, represented as a B-rep or as a mesh. The match between the model and the source data is then assessed through geometric metrics such as cloud-to-cloud distance (C2C) and through coverage indicators for each class.

Reconstruction must handle typical survey issues: non-uniform density, noise/occlusions, and the presence of thin elements (railings, secondary beams). For this reason, work was performed by class and by span, applying dedicated thresholds and controls. Before generating solids and surfaces, the extraction of measurable guides (axis, cross-sections, edges) is essential. These guides preserve the semantics of the source data because they are

computed per class; they constrain model dimensions (levels and thicknesses) and provide the reference for subsequent quality checks, such as section residuals and deviation maps. The same guides feed both modelling pipelines, ensuring geometric and methodological consistency between reconstructions.

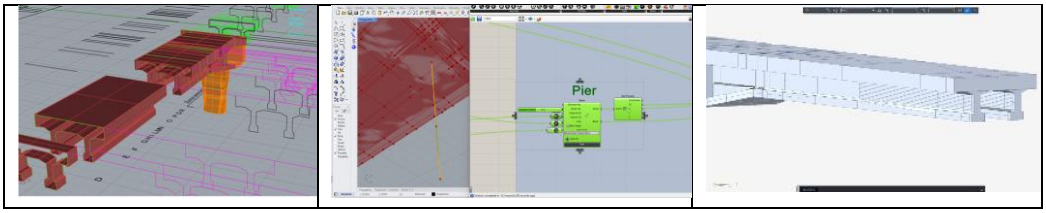
Track A: section-based; reconstruction is guided by cross-sections in Rhinoceros and is aimed at producing an HBIM/IFC model; where required, this model can be transferred to FEM (Figure 5.1).



a	b	c

Fig. 5.1 — Model in Midas CIM (HbriM) (a), analytical model in Midas CIM (b), FEM model in Midas Civil (c).

Track B—Parametric modelling (Grasshopper → Civil/MIDAS) represents the more agile and flexible route. Starting from the same extracted elements, planimetry, profiles, slopes and cross-section variability can be controlled through sliders and parametric rules. Export via plugins to Civil/MIDAS allows the construction of a parametrically governed analysis model, ready for subsequent design iterations and for variant management (Fig. 5.2). The core idea is to ensure that the semantic information associated with the different classes guides the choice of geometric representation: NURBS surfaces for elements such as the deck slab, solids or axes for piers and beams, and controlled meshes when FEM discretization is required.



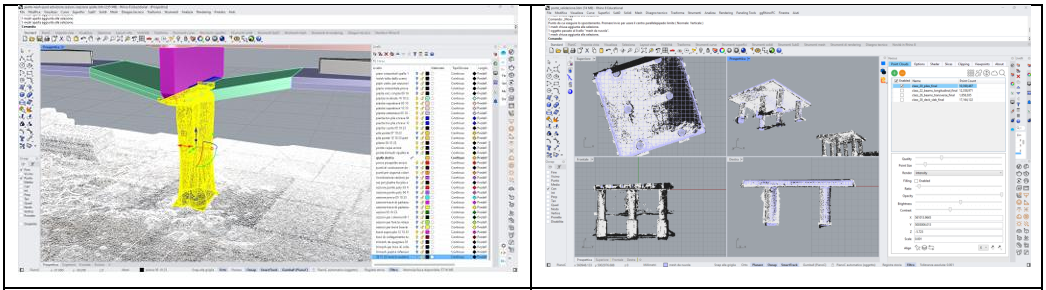
a	b	c

Fig. 5.2 — Section variability (a); parametric commands in Grasshopper (b); FEM model in Midas Civil (c).

The two pipelines are complementary: they share inputs and quality controls, but address different objectives—namely, compliance/archiving and optimisation/variant management. Their comparison in terms of accuracy and reusability is discussed in the following sections.

Process quality is designed to be measurable and reproducible. Performance is interpreted through three main indicators: C2C, IR_τ and F1. For the semantic component, we evaluate per-class F1 supported by confusion matrices; for geometric adherence, we analyse point-to-model distance (C2C) on uniform samples, focusing on the most representative percentiles (median and p95). This is complemented by the Inlier Ratio IR_τ, which expresses the share of points falling within a pre-defined threshold τ. The operational workflow is unique even though it is applied in two different contexts: models are trained on railway data and validated on road data. In both cases, the input is already classified and subjected to quality checks, ensuring consistency and comparability of the results.

The output package is intentionally essential yet complete: it includes the final geometric model (Rhino/Grasshopper format, IFC and/or mesh), an annotated point cloud with quality information—such as C2C, confidence levels and masks—and a manifest file documenting execution parameters, software versions and applied transformations.



a	b

Fig. 5.3 — Classified point cloud of the training bridge with solid elements (a); classified point cloud of the validation bridge with solid elements (b).

This setup is aligned with the literature and with Bridge Management System (BMS) / Digital Twin (DT) practice, where BrIM (IFC 4.3) is the "node" that organises the structure and diagnostics, while FEM is the environment for reasoning about scenarios and limit states. The DT–FEM connection is implemented in a controlled, one-way manner rather than as a "one-click" workflow, precisely to preserve transparency regarding assumptions and simplifications.

The next sections describe data preparation, guide extraction, modelling in the two tracks, and the final comparison, including a QA checklist supporting the go/no-go decision for export.

5.1 CONTEXTS AND REQUIREMENTS

An operational approach is adopted. First, requirements, measurement units and quality criteria are defined; then the point cloud is prepared in Rhino and axes and cross-sections are extracted. The two modelling pipelines (Track A and Track B) are introduced, concluding with a comparison and a quality-control checklist aimed at verifying whether the model is ready for export. The entire workflow is governed by transformation logs and explicit parameters (units of measure, global translation, voxelisation/decimation, class→IFC mapping), to ensure traceability and verifiability of the process.

Unit conversion is described by

$$x_{\mathrm{mm}} = s\, x_{\mathrm{m}}, s = 1000 \tag{129}$$

The global translation (global shift) is expressed as

$$x \leftarrow x - x_0 \tag{130}$$

The rigid transformation in three-dimensional space is defined as

$$x' = R\,x + t \tag{131}$$

where R is a rotation matrix and t a translation vector.

Scaling, global translation and rigid-transformation operations are invertible and are propagated consistently to the IFC and FEM exports; the process logs report the values of parameters s, R, t and of the global translation, ensuring geometric alignment and consistency with C2C checks.

Tab. 5.1 — Requirements and class mapping.

ID	Class	Tipo	Model requirement (ε, d⊥)	Classification target	Downstream use
20	Piers	Strutturale	d⊥ ≤ 0.03 m; ε_sec ≤ 0.04 m	F1 ≥ 0.90	HBIM IfcColumn; colonne FEM
22	Deck slab / deck	Strutturale	ε_sec ≤ 0.05 m; R^2 ≥ 0.95	F1 ≥ 0.88	HBIM IfcSlab; piastre FEM
26	Transverse beams	Strutturale	ε_sec ≤ 0.05 m	F1 ≥ 0.80	HBIM IfcBeam; FEM beams
28	Beams/deck (long.)	Strutturale	ε_sec ≤ 0.05 m	F1 ≥ 0.88	HBIM IfcBeam; FEM beams
16	Abutments / barriers	Non strutturale	ε_sec ≤ 0.06 m	F1 ≥ 0.75	IfcBarrier/Proxy
30	Pavimentazioni	Non strutturale	ε_sec ≤ 0.06 m	F1 ≥ 0.80	IfcCovering
10	Ground	Contesto	—	F1 ≥ 0.85	Contesto/maschere
12	Vegetation	Contesto	—	F1 ≥ 0.85	Contesto/maschere

Tab. 5.2 — Class presence in training vs validation.

The classification setup indicates that the validation dataset includes a smaller number of classes. Class-specific thresholds are adopted, consistent with the required information level for modelling/analysis. Structural classes (piers/deck) require stricter p95 limits than contextual classes.

ID	Class	Training	Validation
10	Ground	✓	✓
12	Vegetation	✓	✓
14	Power lines	✓	—
16	Safety barriers	✓	✓
18	Rails/ballast	✓	—
20	Piers	✓	✓
22	Transverse bearing beams	✓	✓
24	Transverse stiffening beams	✓	✓
26	Longitudinal girder	✓	✓
28	Deck slab	✓	✓
30	Abutments/embankment	✓	✓
34	Construction site material	✓	—
36	Electric piles	✓	—

The classification setup indicates that the validation dataset includes a smaller number of classes. Class-specific thresholds are adopted, consistent with the required information level for modelling/analysis. Structural classes (piers/deck) require stricter p95 limits than contextual classes.

Tab. 5.3 — Minimum requirements by class.

Class (main downstream use)	C2C p95 [m]	Min density [pt/m^2]	F1 target
20 Piers (HBIM/FEM)	≤ 0.03–0.05	≥ 250–400	≥ 0.85
28 Deck/Beams (HBIM/FEM)	≤ 0.03–0.05	≥ 250–400	≥ 0.85
16 Abutments/Barriers (HBIM)	≤ 0.04–0.06	≥ 150–250	≥ 0.80

30 Pavimentazioni (CIM/HBIM)	≤ 0.05–0.07	≥ 100–200	≥ 0.80
10 Ground – 12 Vegetation (Context)	≤ 0.07–0.10	—	≥ 0.75

The tables define, for each class, the model's geometric requirements—such as section residuals, orthogonal deviation and, where applicable, a fit coefficient—together with semantic-quality targets (F1) and the target IFC entities. The aim is to avoid generic geometric proxies and to guarantee an explicit coherence between semantics, expressed by classes, and geometry, represented in HBIM and FEM models. Structural classes such as piers, deck and beams are associated with stricter thresholds because they are directly used as IfcColumn, IfcBeam and IfcSlab entities and in subsequent FEM discretisations via beam, shell and plate elements; non-structural or contextual classes adopt more permissive criteria, since they mainly serve masking and clipping functions, e.g., through IfcBarrier/Kerb, IfcCovering and IfcSite.

Table 5.3 reports the minimum reference values for C2C p95 and for the F1 target. C2C p95 is the value below which 95% of the points of a given class fall with respect to the reference model, whereas the F1 target is the acceptance threshold for semantic quality. A minimum density is also introduced as a local-scale control useful to prevent meshing or fitting artefacts. Geometric QA measures how closely the model—either HBIM or FEM mesh—adheres to the point cloud through cloud-to-model distance statistics (C2C, median, …) and through coverage within a threshold τ, which quantifies the model's adherence to the cloud. Adherence is quantified through the cloud-to-model distance, defined as the minimum distance between a cloud point and the model M:

$$d(p, M) = \min_{q \in M} \| p - q \|_2 \tag{132}$$

For quasi-planar elements, such as the top surface of the slab, adherence is also evaluated through the orthogonal deviation from a plane Π, defined as:

$$d_{\perp}(p,\Pi) = \mid n^{T}(p - p_{0}) \mid \quad (133)$$

To quantify the fraction of points that are actually compatible with the model within a tolerance τ, the coverage index Iτ is introduced, defined as:

$$IR_{\tau} = \frac{\mid \{p: d(p,M) \leq \tau\} \mid}{N} \quad (134)$$

with N the total number of points of the class. The geometric acceptance condition is expressed as:

$$IR_{\tau} \geq IR_{\tau}^{\min} \quad (135)$$

The semantic quality of segmentation and classification is assessed through the F1 score, which balances precision and recall.

The process error budget cannot be attributed to a single cause; rather, it arises from the root-sum-square combination of multiple contributions, including instrumental survey errors, registration errors associated with alignments and ICP, discretisation errors due to voxelisation, decimation and meshing, and modelling errors linked to idealisation choices. For this reason, quality assessment does not rely on mean values, but prefers robust and interpretable indicators such as the p50 and p95 percentiles and the coverage within threshold τ, which allow controlling both the typical error and the tail of the residual distribution.

Operationally, semantic quality (F1) is computed using the CSV feature files for training and testing and a validation file containing the same columns, including class labels. A Random Forest classifier (e.g., 100 trees) is trained and evaluated on test and validation sets, computing per-class precision, recall and F1, as well as macro-F1, weighted-F1 and the confusion matrix; results are interpreted by comparison with the reference acceptance thresholds. Geometric quality (C2C) is evaluated by building, for each class, a reference model coherent with the adopted pipeline: HBIM surfaces/solids in Track A, and clean, controlled meshes in Track B. To avoid areas with higher density disproportionately influencing statistics, a fixed number of points is extracted via uniform sampling or Poisson-disk strategies; for each point, the minimum distance to the model is computed using KD-tree structures

implemented in Python/Open3D. This choice—rather than interactive tools such as CloudCompare—is motivated by reproducibility and methodological traceability, since it allows the sampling strategy, metric, search structure, thresholds and summary statistics to be explicitly fixed and recorded in the experiment manifest, making results verifiable and comparable across different runs. Results are finally summarised via the median (p50) and p95 and compared with the guideline ranges reported in Table 5.3. The Iτ index is computed by setting a class-specific tolerance τ, e.g., 0.05 m for piers and deck, 0.06 m for abutments, 0.07 m for pavements and 0.10 m for the context. Under a Go/No-Go perspective, a class is considered acceptable when Iτ exceeds a predefined minimum—typically between 80% and 90%—and, at the same time, the p95 value falls within the guideline range. As good practice, the adopted scripts keep uniform per-class sampling constant before computing C2C to ensure comparability between datasets and runs, and complement global metrics with local checks through dedicated metrological sections (e.g., on piers and slab) for controlling thicknesses and elevations and recording residuals through a specific indicator Δs; in the presence of local outliers, colour-mapped error maps provide an effective support to quickly identify areas that require rework or model refinement.

5.2 PREPARATION IN RHINOCEROS

.

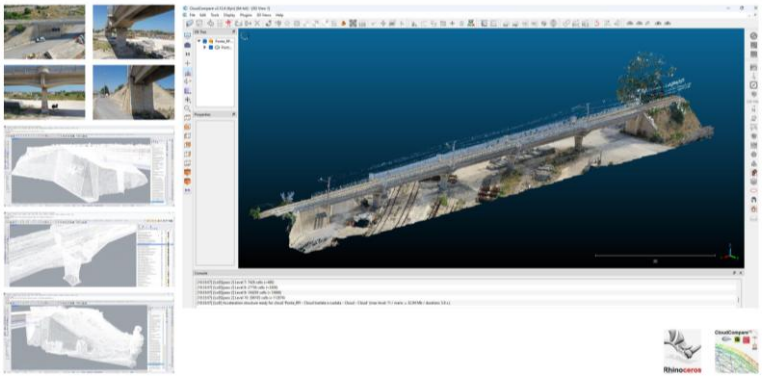


Fig. 5.4 — Point-cloud visualisation in CloudCompare and import into Rhinoceros (views of the meshes generated in Rhino); bridge images for direct comparison with point cloud and mesh.

Beyond a few tens of millions of points, the .vpc container makes it possible to visualise and section the data without loading the entire dataset into memory. Conversion (LAZ→VPC/PLY) is part of the workflow and must be tracked in the manifest.

Format and conversion choices directly affect import times, RAM usage and preservation of the class attribute (Scalar/Classification). In our workflow, as anticipated, we favour PLY for use in Rhinoceros; it is also preferred in Python scripts because it allows arbitrary per-point attributes without degrading discrete labels (class = integer).

Slicing is performed consistently with the bridge axis: non-overlapping blocks (or with minimal overlap) are defined per span, preserving class layers. This simplifies both modelling and localised quality checks.

The outputs of this phase are: (i) the cloud ready in Rhino (per class/layer), (ii) per-span extracts (if used), (iii) a log of transformations and parameters, and (iv) a minimal set of views/figures for quick checks (overlays, sections).

In the remainder of the chapter, when files in "src/working/QA" are mentioned, this refers to these assets: the full version (source), the reduced version for interaction, and the verification products.

Tab. 5.4 —A— Typical conversion scenarios and practical impact.

Scenario	Input	Output	Preserves class	Typical RAM	Typical weather	Notes
A	Multi-station E57	PLY per class	Yes (Classification column)	8–12 GB → 1.6–2.4 GB	15–25 min	Small/medium models
B	LAS/LAZ	PLY per Blocks (voxel 10–15 mm)	Yes (Scalar mapping)	12–18 GB → 2–3 GB	25–45 min	Balanced quality/time

C	E57/LAS	VPC (Veesus) + slicing 20–40 m	Yes (RGB→Class)	Interactive (< 2 GB Active)	5–10 min/blocco	PC very large

In some ASCII exports, the class can be converted to float or normalised. To avoid this, PLY with a discrete ClassID is preferred; alternatively, the class can be temporarily encoded in an unambiguous channel (e.g., RGB) and restored in post-processing. In any case, an automatic check of unique values is performed to immediately detect anomalies.

Listing 5.2.1 — PDAL pipeline (E57→LAZ with 15 mm voxel)

```
{
 "pipeline": [
  "input.e57",
  { "type": "filters.range", "limits": "Z[-10:100]" },
  { "type": "filters.voxelcenternearestneighbor", "cell": "0.015" },
  { "type": "writers.las", "filename": "out.laz",
   "minor_version": "4", "dataformat_id": "7" }
 ]
}
```

Point-cloud density reduction (downsampling) must be applied in a controlled manner. It is required to ensure operational fluidity without altering geometry or compromising the representation of thin classes. For this reason, downsampling is performed per class and its impact is verified both on point counts and on distance metrics (e.g., by comparing C2C values before and after reduction). Global shift is introduced solely for numerical stability in the presence of large-magnitude coordinates and does not modify relative distances; therefore it is essential to apply it identically to the point cloud, extracted sections and exported models.

In Rhinoceros, a tolerance consistent with the survey resolution is set and operations are carried out in a local reference system defined by the bridge longitudinal, transverse and vertical axes. This choice reduces instabilities in

fitting processes, intersections and boolean operations. The point cloud can be imported directly, if the format is compatible, or through dedicated plugins; in both cases, a preliminary check concerns measurement units, axis orientation, the presence of the class attribute and the consistency of point counts for each class. Modelling entities—curves, surfaces and solids—are built from geometric guides extracted from the cloud, in particular regular sections, axes and profiles, while the point cloud itself remains the primary reference for verification and local corrections. To ensure numerical stability and interoperability, the data are finally brought into a local reference system through three successive steps: (i) optional unit scaling (e.g., m↔mm, if required), (ii) application of a global shift to reduce the magnitude of coordinates, and (iii) roto-translation to align axes with the bridge longitudinal, transverse and vertical directions.

The order of operations and the applied matrices are saved in the transformation file.

The sequence of transformations (units, shift and 4×4 matrices) is reported in README_transform.txt, so that point clouds, meshes and exported models can be realigned consistently.

$$T\_tot = T\_model \cdot T\_shift \cdot T\_scale \tag{136}$$

In Rhino, the layer hierarchy is the “contract” between classification and modelling: each class lives in a dedicated layer, while separate layers collect guides (axes/sections), final surfaces/solids and analysis meshes. Naming includes project, span and version so that each export is traceable.

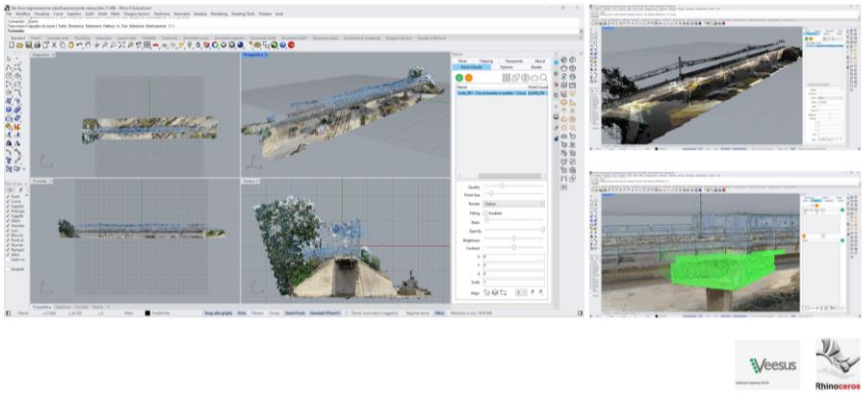


Fig. 5.5 — Point-cloud visualisation, segmentation and layer assignment in Rhino.

Two copies are typically used: a reduced version for interactive operations and a full version for measurements and final checks. The two must remain consistent (same transformations, same class encoding).

The operator's task is to guide the reconstruction in a “minimally invasive” manner: clean only where necessary, maintain consistency between spans, and minimise untracked manual corrections.

Axis and section extraction is based on regular sampling along the bridge development. Sections are built with a consistent step and saved as reference curves/contours for loft/sweep operations and for verification.

For structural classes, dedicated sections are produced (piers, beams, slab), so that a single “mixed” section does not introduce ambiguity. Where the cloud is incomplete, robust fits and visual overlay checks are used.

Tab. 5.5 — Layer → IFC/FEM correspondences.

Layer (class)	**IFC 4.3**	**FEM (Type)**	**Notes**
20_Pile	IfcColumn	beam/shell	Base/tip constraints
28_Longitudinal Beams	IfcBeam	beam/shell	refine at diaphragms
26_Beams_Cross	IfcBeam	beam	deck connections
22_Slab_Deck	IfcSlab	plate/shell	slopes/elevations
16_Back_Barriers	IfcBarrier/IfcKerb	—	Non-structural

During preparation, critical issues that may compromise modelling are identified immediately: thin, poorly sampled classes; reflections on barriers; vegetation near embankments; and discontinuities at joints/bearings. These areas are marked for targeted checks in subsequent phases.

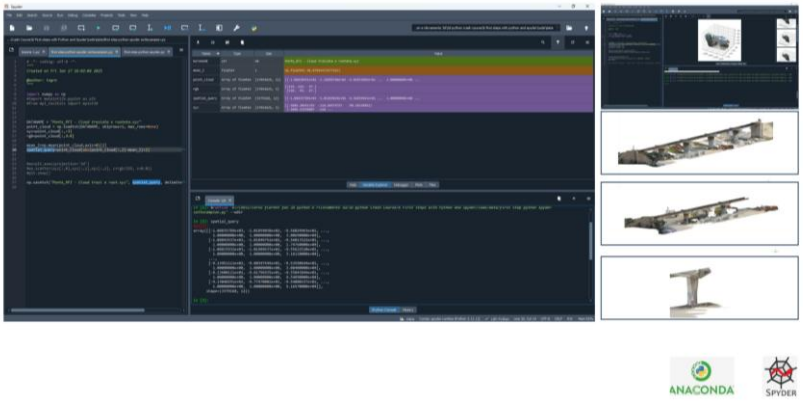


Fig. 5.6 — Analysis and segmentation through the software architecture.

In practice, a class-dependent downsampling is applied, using a finer voxel for structural classes and a coarser one for contextual classes, verifying that statistics and geometric metrics do not change significantly. The effect of voxelisation on C2C p95 is then analysed.

Curvature-based mesh decimation reduces the number of triangles while preserving edges and curvature radii; it is particularly suitable after a preliminary reconstruction (e.g., Poisson or Ball Pivoting) when it is necessary to move to surface or solid representations.

Finally, slicing along the bridge axis (20–40 m) keeps RAM usage within 2–4 GB per block and simplifies local checks.

Tab. 5.6 — Typical slicing parameters (deck/piles).

Tratta	Overlap	RAM Activate/Block	Time/block	Remarks
20 m	1 m	0.9–1.3 GB	8–12 min	Maximum local control
30 m	1.5 m	1.3–1.8 GB	12–18 min	good compromise
40 m	2 m	1.8–2.4 GB	18–24 min	less overhead

To make the procedure repeatable and verifiable, part of the operations (slicing, downsampling, export) can be automated via scripts. The following listings provide a minimal example: the aim is not performance optimisation, but explicit traceability of the parameters used.

Listing 5.2.3 — PDAL + Open3D (batch reduction per class)

```
# PDAL (riduzione/LAZ)
pdal pipeline pipeline.json
# Open3D: Reduction and Rescue QA by class
import open3d as o3d, numpy as np, pandas as pd
PLY_IN  = "28_Deck_raw.ply"
PLY_OUT = "28_Deck_ds.ply"
pcd = o3d.io.read_point_cloud(PLY_IN).voxel_down_sample(0.010)
pcd.estimate_normals(o3d.geometry.KDTreeSearchParamKNN(knn=30))
o3d.io.write_point_cloud(PLY_OUT, pcd)
```

Axes and section planes constitute the very grammar of modelling: they make it possible to anchor the point cloud to design and analytical references and make the generation of surfaces and solids reproducible. The operational definition of the longitudinal axis, orthogonal section planes (Fig. 5.7) and guide curves is therefore crucial for the subsequent modelling pipelines (Track A/B).

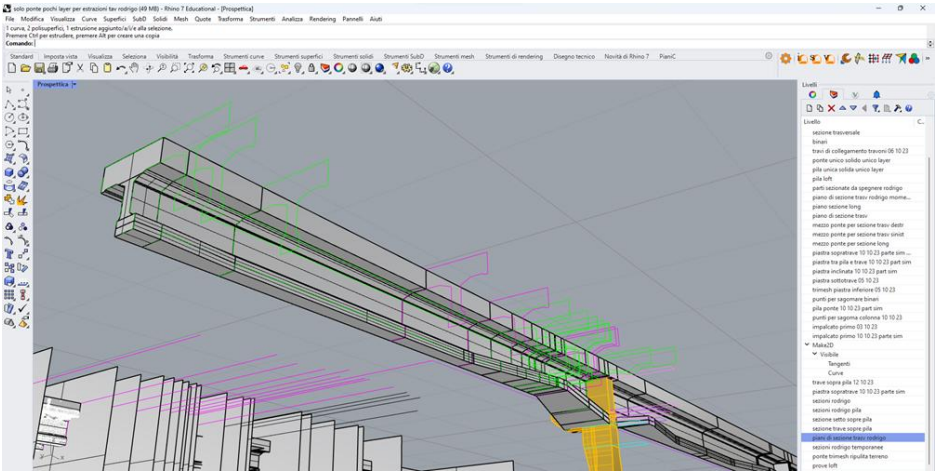


Fig. 5.7 — Cross-sections along the alignment. Bridge symmetry with respect to the roadway axis.

Expected outputs are: per-class layers in Rhino; guides (axis and sections) saved in reusable formats; a concise QA report; and a manifest documenting versions, parameters and transformations.

Tab. 5.7 — Section output and export packages for branches A/B; formats, destinations and QA/interoperability notes.

Categoria	**Content / Description**	**Formato/i**	**Destination / Downstream use**	**QA & Interoperability notes**
Working model	Rhino file organised by classes and layers (piers, deck slab/deck, long./transv. beams, abutments/barriers, ground, vegetation) + geometric references (axes, section planes, guide curves)	.3dm	Common base for Track A (Section-based→CIM/IFC→FEM) and Track B (Parametric GH→Civil)	Layer structure consistent with semantics; stable naming; units/tolerances already set.
QA of input	Snapshots and CSV with parameters, cardinalities, IR_τ, p95; process logs (voxel, slicing, decimation)	.csv, .png/.jpg	Quality traceability and go/no-go before export	Keep together with the model; use the same reference frame and global shift as the model.
Export HBIM / Archive	Information model with IFC 4.3 entities consistent with the classes (20→IfcColumn; 28/22→IfcBeam/IfcSlab; 16→IfcBarrier/Kerb; 30→IfcCovering)	.ifc (IFC 4.3)	**CIM/HBIM, archiving, disciplinary exchange**	Ensure consistent class→entity naming; keep the reference system documented.
Export FEM (Geometry)	Solidi/superfici NURBS per meshing nel solver	STEP (.stp), IGES (.igs), ACIS (.sat)	Pre-process FEM (es. MIDAS FEA NX, altri)	Surfaces/solids inherit units and orientation; attach README_transform.
Infrastructure export	Alignments and infrastructure objects for road-design suites	LandXML, .ifc	**Civil/MIDAS** (ramo B)	Consistency with axes/sections in § 5.3; check EPSG code or applied shift.

Traceability documents	README_transform with (s, R, t) and global shift; export settings report (tolerances, units)	.txt, .pdf	Allegato obbligatorio a ogni pacchetto di consegna	Enables re-import and auditing between PC/IFC/FEM without losing the reference.

5.3 GUIDE EXTRACTS: AXES, EDGE SECTIONS, CONTROL ELEVATIONS

This section describes the procedure for extracting, from the point cloud, the "guide extracts" used in subsequent modelling stages: the structure axis, regular sections and class-specific profiles. The goal is to build stable and repeatable geometric inputs, reducing manual intervention while ensuring explicit metrological control.

For each section, the centroid is computed as the mean of the coordinates of the points belonging to the slice. The centroid is used to build a first approximation of the bridge longitudinal axis, reducing the influence of local noise in the cloud.

$$\bar{p} = \frac{1}{N}\sum_{i=1}^{N} p_i \tag{137}$$

Principal Component Analysis (PCA), applied to the covariance matrix of the points, makes it possible to identify the main direction of local geometric development. In particular, the dominant eigenvector is used to estimate axis orientation and to stabilise the longitudinal-axis fit even in the presence of curvature or geometric discontinuities.

The reference axis provides the metric backbone of the entire modelling process: it aligns section planes and makes elevations, thicknesses and geometries comparable across different spans. Extraction is performed exclusively on the classified cloud; working per class (slab/deck, beams, piers) reduces noise and bias, shortens computation times and yields a more stable axis fit than using the whole cloud (from which vegetation and terrain are in any case excluded).

The adopted method, considered robust and reproducible, is articulated into the following steps:

1. sampling along the axis;
2. construction of a reference polyline;
3. estimation of the main direction through PCA;
4. axis re-orientation and regularisation.

Specifically, sampling is performed at N curvilinear positions defined by the curvilinear abscissa s along a smooth curve. The axis curvature is computed as:

$$\kappa(s) = \frac{\| r'(s) \times r''(s) \|}{\| r'(s) \|^3} \tag{138}$$

For each position s_i, cylindrical (or thickened) windows centred on the deck are built, with a buffer between 0.80 and 1.20 m, within which centroids and local covariance matrices are computed:

$$\bar{p}_i = \frac{1}{N_i}\sum_{j=1}^{N_i} p_{ij}\, C_i = \frac{1}{N_i}\sum_{j=1}^{N_i} (p_{ij} - \bar{p}_i)(p_{ij} - \bar{p}_i)^T \tag{139}$$

From the local PCA within window i, the principal vector is obtained, representing the direction of maximum extension of the cloud in that section. The resulting longitudinal axis is therefore defined on a class-filtered cloud, with stable slopes and curvature radii and planimetric/altimetric elevations consistent with the design units. This axis can be reused both in parametric modelling environments (e.g., Grasshopper, as a guide curve) and in CIM/IFC contexts (axis/reference line).

An application example of curvature control for the axis, referring to a portion of the case-study bridge, is shown in the figure.

$\boldsymbol{\kappa(s)}$ Tab. 5.8 — Alignment curvature.

Axis curvilinear coordinate s [m] $\boldsymbol{\kappa(s)}$	**κ(s) [1/m]**
0	0.0020
10	0.0027
20	0.0023
30	0.0017

40	0.0016
50	0.0019
60	0.0025
70	0.0020
80	0.0016
90	0.0017
100	0.0021
110	0.0024
120	0.0022

The generation of section planes is not governed solely by their step along the road axis, but also by the management of the extraction-window width, which varies as a function of the deck footprint. The curves describing the deck geometry present progressive variations, with local peaks at joints and section changes, which require a dynamic adaptation of sampling windows.
Station control consists in verifying that sections are ordered and consistent along the axis (regular step, consistent orientation and absence of jumps). For piers and bearings, denser sections and dedicated checks are adopted, because these are areas with complex geometry and higher probability of occlusions. For the slab and the main beams, a more regular step can be used.
Section-plane generation is governed not only by their step along the road axis but also by the management of their variable width as a function of the deck footprint. The curves generating the deck vary smoothly, with peaks at joints and section variations.

Tab. 5.9 — Section width along a portion of the deck alignment.

s [m]	**b(s) [m]**
0	8.20
10	8.55
20	8.40
30	8.10
40	8.05

50	8.30
60	8.95
70	8.45
80	8.10
90	8.05
100	8.35
110	8.70
120	8.45

Guide and model quality are evaluated with local (per-section) and aggregated (per-span and per-class) indicators. In practice, we report: C2C statistics (e.g., percentiles), coverage within a threshold τ (IR_τ) and, for the semantic component, per-class F1 from the classification report. Table 5.10 summarises these values and supports the go/no-go decision.

Once extracted, guides (axis, sections and per-class profiles) are much lighter than the full cloud: they are compact geometric objects, easy to version and exchange across software. The same geometric base feeds both Track A and Track B, making the two approaches comparable and reducing input ambiguity.

Tab. 5.10 — QA summary per span.

Span	**IR_{0.03 m} [–]**	**Residue fit p95 [m]**	**Notes**
1	0.93	0.021	OK
2	0.95	0.025	OK
3	0.91	0.030	Densificare Δs
4	0.96	0.022	OK
5	0.92	0.028	Barrier-edge check

$$p\_q \;=\; Q\_q(\,\{\, d(p\_i, \mathcal{M})\,\}\_\{i=1\}\^\{N\}\,),\quad q \in \{0.50, 0.95\} \tag{140}$$

The distinction between local noise and systematic errors, as well as verification against the geometric quality guideline thresholds, is governed by the analysis of fit-residual percentiles, defined by the following equation:

$$p_q = Q_q(d(p_i, M))_{i=1}^{N}, q \in \{0.50, 0.95\} \quad (141)$$

This formulation defines the percentile q of the distance residuals between the cloud points and the reference model. In addition to the statistical evaluation of residuals, it is necessary to verify the geometric regularity of the sections: for instance, checking whether a pier can indeed be approximated as circular and comparing the estimated radius with design values or expected values.

In the case of irregular piers or non-standard geometries (e.g., in the training bridge), a more general description of the section is required, using a generalised fit.

Reference dimensions (widths and thicknesses) are estimated directly from the sections and compared with the expected values for the specific structural typology. This check, although simple, is often sufficient to intercept systematic drifts before the export phase.

For circular piers, the local radius is estimated as:

$$r_j = \sqrt{(x_j - c_x)^2 + (y_j - c_y)^2}, r_{95} = Q_{0.95}(\{r_j\}), D_{\text{est}} \approx 2\, r_{95} \quad (142)$$

For a more general description of the section, a least-squares conic fit is adopted, subject to an ellipticity constraint:

$$\min_{a,b,c,d,e,f} \sum_{j=1}^{m} \left(ax_j^2 + bx_j y_j + cy_j^2 + dx_j + ey_j + f\right)^2, \text{s.t. } b^2 - 4ac < 0 \quad (143)$$

The adherence between the guide geometry and the point cloud is assessed through orthogonal residuals and, where available, through C2C metrics. The

goal is not to force residuals to zero, but to keep them within an interval consistent with the survey resolution and noise level.

Adherence between guide and cloud is measured through orthogonal residuals and, when available, through C2C. The objective is not a zero residual, but staying within a range consistent with survey resolution and noise.

The guidelines adopted for the section step as a function of class and downstream use are reported below:

Tab. 5.11 — Section step size as a function of class.

Class/Element	Δs min [m]	Δs tipico [m]	Δs max [m]	Notes
Deck slab/Deck	0.5	1.0–2.0	3.0	Densify at supports/joints
Longitudinal girders	0.5	1.0–1.5	2.0	Densified sections for height/web changes
Transverse beams	0.5	1.0	1.5	Compatibile con mesh FEM plate/shell
Piers/Pier caps	0.25	0.5	1.0	Local sections for circular/elliptical fitting
Barriers/Abutments	1.0	2.0	3.0	HBIM reference geometry

Table summarises the QA criteria per span: thresholds, indicators and corrective actions. It serves as a quick reference during modelling.

For the training railway bridge, section planes were built along the deck axis for subsequent adjustment to the Δs values, as in Fig. 5.8.

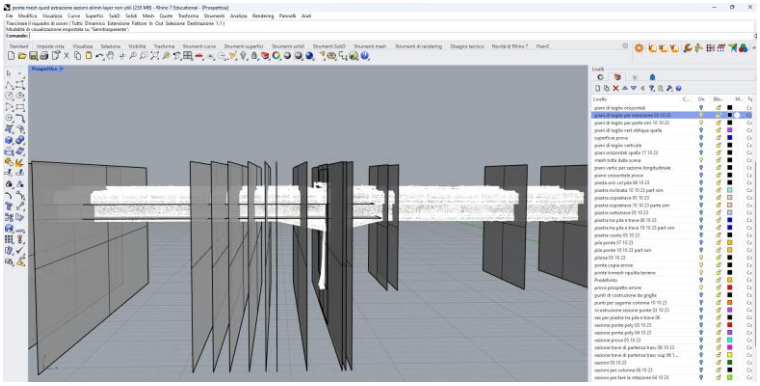


Fig. 5.8 — Section setup and 3D modelling.

From the intersection of the section planes with the point cloud, spatially oriented sections are obtained along the bridge longitudinal axis at the Δs spacing set for the deck, and along the vertical axes of the piers with a Δs spacing chosen as a function of the solid variability and proximity to stress-concentration areas (Fig. 5.9).

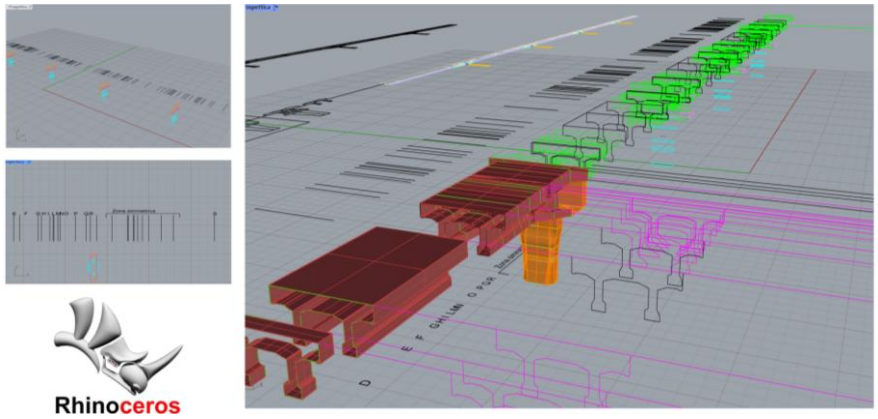


Fig. 5.9 — Section setup and 3D modelling.

The resulting sections, together with the axes, form a fundamental skeleton for the subsequent BrIM construction phase (Track A). This scheme can be imported into any CAD programme to name all sections, generate dimensions and organise layers. Sections can then be imported into BrIM programmes

such as MIDAS CIM to generate solids through powerful boolean operations (Fig. 5.10).

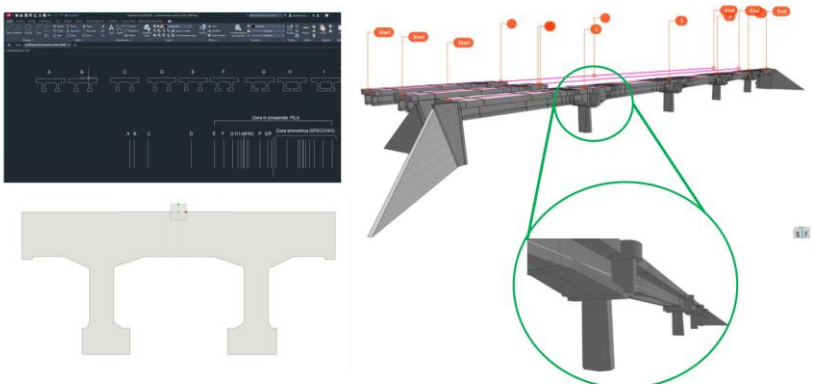


Fig. 5.10 — Section export and BrIM model generation.

The generation of structural elements starting from sections, using generating axes and boolean operations (Track B), can be performed directly in Rhino–Grasshopper. Figure 5.11 shows not only the construction of structural solids but also the continuous comparison with the point cloud.

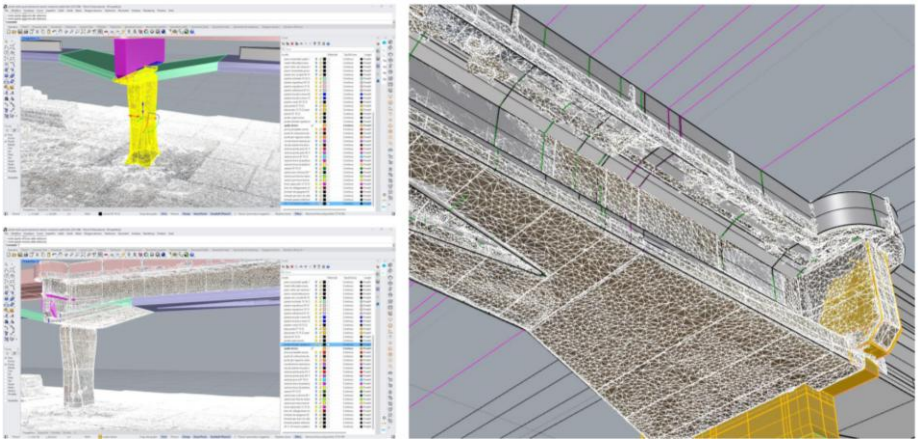


Fig. 5.11 — 3D model construction on the segmented and classified point cloud.

5.4 TRACK A AND TRACK B

This section describes the two geometric reconstruction modes adopted in the thesis. The two tracks share the same geometric guides (axis and sections), but differ in representation, level of automation, and export target (HBIM/IFC vs FEM).

Track A implements a section-based modelling chain, in which section curves orthogonal to the bridge axis are extruded to generate NURBS surfaces and closed solids, subsequently typed through IFC 4.3 entities and ready for export to a general FEM solver.

Choosing the section-based approach makes it possible to control G^1/G^2 continuity, fillets and thickness variations with a granularity consistent with C2C tolerances, while ensuring semantic traceability—an essential condition for the BrIM → FEM transition.

The extraction of the axis, planes and profile polylines directly feeds this flow, anchoring the model metric to the scanning evidence. The process is articulated into four main stages:

1. geometric pre-processing and unit normalisation;
2. construction of surfaces and solids by class (slab/deck, longitudinal and transverse beams, piers);
3. semantic mapping to IFC 4.3, with assignment of property sets and preservation of class IDs;
4. parallel export, both informational (IFC) and geometric for FEM (Parasolid/ACIS).

Non-structural and contextual elements are kept at the informational level and excluded from the FEM domain. The section-based strategy assumes that each structural class can be described by an axis c(s) and by a family of sections defined on planes orthogonal to that axis. The informational surface/solid is generated as a controlled extrusion of the sections along c(s). In parametric form:

$$\gamma : [0, L] \rightarrow \mathbb{R}\text{^}3, \quad ||\gamma'(s)|| = 1 \tag{144}$$

Axis curve (class C^1)

$$\Sigma\_s = \{\, c(u,s) \in \mathbb{R}\text{^}3 \mid u \in [0,1] \,\} \tag{145}$$

Section with continuity where required (e.g., slab–web fillets)

$$S(u,v) = \frac{\sum_i \sum_j N_i^p(u)\, M_j^q(v)\, w_{ij}\, P_{ij}}{\sum_i \sum_j N_i^p(u)\, M_j^q(v)\, w_{ij}} \tag{146}$$

NURBS surface.

The main operational advantage, compared to a direct fit of the cloud, lies on the one hand in a geometric–semantic control consistent with bridge engineering, and on the other hand in native interoperability with IFC 4.3 and FEM solvers. Starting from section polylines, orthogonal to the planes Π(s), surfaces are generated and subsequently closed into solids, maintaining tolerances consistent with the expected C2C p95 error.

Class IDs are reflected in object names and layers to guarantee traceability and selective filtering during export. Elements are typed according to IFC 4.3 consistently with the structural classification: IfcColumn for piers, IfcBeam for beams, IfcSlab for the deck slab, IfcBarrier/IfcKerb for barriers, and IfcCovering for pavements. Each object inherits a minimal information profile (Name, ObjectType, coordinates and transformations), enabling automated export filtering and alignment with the FEM domain.

The following table summarises the class → IFC entity correspondence and the discretisation type suggested for structural analysis.

Tab. 5.12 — Correspondence between IFC class and analysis discretisation.

Class ID	Class	IFC 4.3 (entity)	Tipo FEM (Recommended)	Notes
20	Piers	IfcColumn	Shell/Beam/Brick 3D (locally)	Variable geometry; Constraints at the base

28	Longitudinal beams/deck	IfcBeam / IfcSlab	Shell (plate) + Beam where appropriate	Discontinuity zones requiring local fitting
26	Transverse beams	IfcBeam	Beam / Shell short	Connections between beams and deck slab
22	Bearing beams	IfcBeam	Beam	Supporting devices
16	Abutments / barriers	IfcBarrier / IfcKerb	Shell o exclusion from the FEM	HBIM; FEM only if relevant
30	Flooring	IfcCovering	Shell o equivalent load	Often as distributed load
10	Ground	IfcSite	External geotechnical model	Boundary conditions
12	Vegetation	IfcSurfaceFeature	—	Context; excluded from the FEM

The IFC 4.3 export preserves semantics, whereas the parallel geometric channel toward Parasolid/ACIS serves the FEM solver, where meshes, rigid links and load schemes are defined. This separation preserves information quality and computational robustness. To guarantee a readable and verifiable model, the mapping remains one-to-one:

Before exporting to FEM, a class-differentiated quality gate is applied: typically C2C p95 ≤ 3–5 cm for piers/deck (with $IR_\tau \geq 0.90$ at $\tau = 5$ cm) and more permissive thresholds for non-structural/context elements. In the FEM environment, surfaces/solids are recognised for discretisation and a one-way BrIM→FEM flow is set up (avoiding weak synchronisations and semantic conflicts). In the CIM/IFC → FEM transition we adopt a discretisation consistent with the physics and with the scale of the problem. Discretisation uses shell (plate) elements for slab/deck, beam or shell elements for beams depending on the span-to-section ratio, and localised 3D brick elements at piers near discontinuities. Based on the defined tolerances and on

considerations regarding discretisation and survey accuracy, the following operational criteria are adopted for selecting the characteristic element size and for mesh-quality control, differentiated by structural component type.

Tab. 5.13 — Discretisation criteria and mesh quality by structural component.

Component	Element type	h Typical [m]	Refinements	Quality criteria
Deck slab/Deck	Shell (plate)	0.20–0.50	Joints/supports/thickness variations	AR ≤ 3; skew ≤ 45°
Long./transv. beams	Beam / Shell	0.20–0.50 (shell); ≥1–2 el/altezza	Connection nodes; section changes	L/H ≥ 3; ≥4 el/distance
Piers/Pier caps	Shell / Brick 3D	0.15–0.40 (shell); 0.10–0.30 (brick)	Foundation connection; corbels	AR ≤ 2; gradien\strains

In FEM, materials and section parameters are defined from the design code/NTC and mapped into property sets; rheology is set and, for beams/shells, equivalent properties are assigned (thicknesses, areas and inertias). Boundary conditions/elastic restraints at pier bases and compatibility constraints between slab, beams and piers are modelled. Topological consistency with the IFC ensures traceability between information identifiers and FEM nodes/elements.

The FEM input gate checks mesh quality, congruence and continuity between elements, the correspondence of self-weight/reactions with rough estimates, and the alignment with C2C maps. The verification of the pipeline is based on checks regarding geometric adherence C2C (mean, median, p95) and IR_τ for class against the HBIM model.

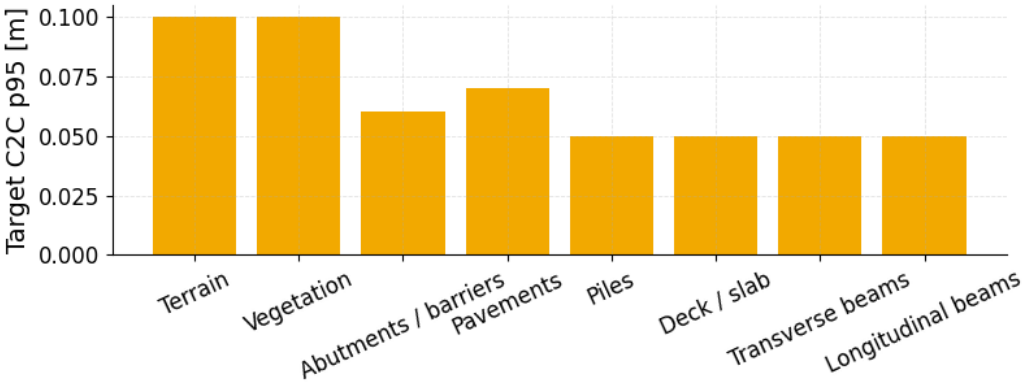


Fig. 5.12 —A— Target C2C p95 by class.

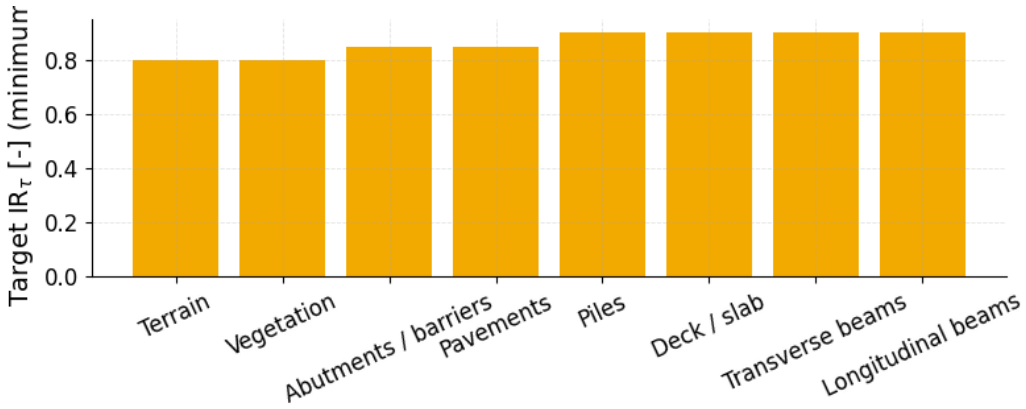


Fig. 5.13 —B— Target IR_τ by class.

Track B implements a parametric workflow in Grasshopper in which the axis c(s), planes Π(s) and sectional profiles define a canvas of parametric groups.

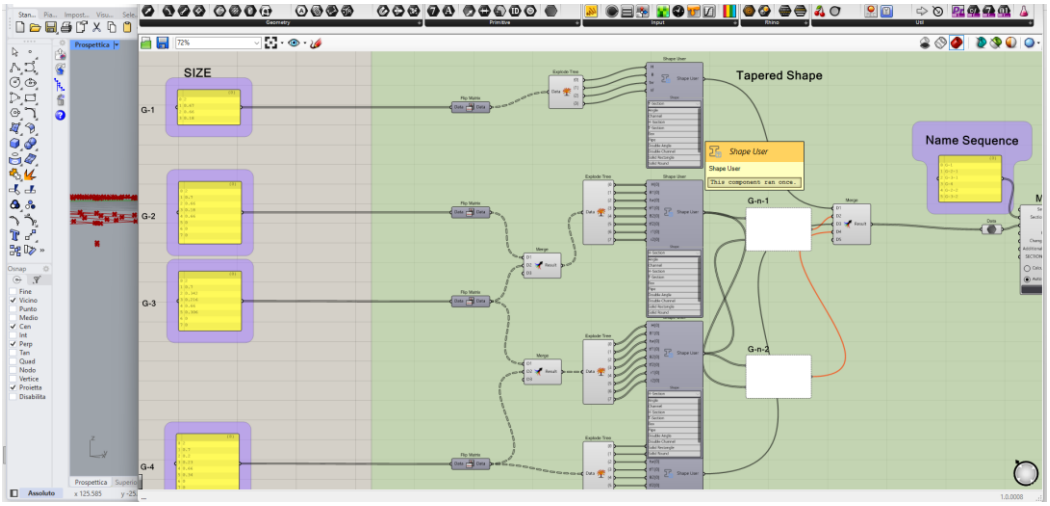


Fig. 5.14 — Canvas with parametric groups.

The model is governed by sliders and lists that capture key variables (step Δs, profile parameters), enabling a controlled generation of the girder system, piers and abutments. Export toward MIDAS Civil (FEM) occurs through a dedicated plugin (MIDAS GH), with direct instantiation of alignments, profiles and sections. The same parametrisation can feed an HBrIM in MIDAS CIM, ensuring semantic consistency between information models and FEM. The objective is to reduce the cost of propagating geometric changes along the axis and to maintain parameter traceability. In Grasshopper, a canvas is prepared with groups of global and local parameters; each group is provided with constraints/domains and notes about structural impact and interoperability with Civil/CIM. To make explicit the parameters governing parametric construction and to guarantee consistency between geometry, topology and export, the following table summarises the main parameter groups adopted in Grasshopper, indicating for each the operational meaning, typical variation domain and usage notes for geometric control and interoperability.

Tab. 5.13 — Grasshopper canvas parameters for parametric modelling and export.

Group	Parameter	Meaning	Typical domain	Notes
Alignment	Δs	Section spacing	0.5–2.0 m	Densify on joints/Supports
Alignment	i(s)	Longitudinal slope	± 6 %	Derivative of z(s)
Alignment	φ(s)	Superelevation/crossfall	± 6 %	Sign ↑/↓ verso Side
Sections	N_sec_pile	# pier sections	3–10	Variability control
Sections	N_sec_impalcato	# girder sections	10–60	Depending on span length
Geometry	w, h	Section widths/heights	ad hoc	Slider numeric/List panel
Topology	N_spans	# spans	3–6	Wheelbases in the list [m]
Export	Tol	Tolerance loft/sweep	1–3 mm	Consistent with §5.1

In Grasshopper (GH) geometry is defined as a function of data. The MIDAS GH plugin allows structural entities to be generated in Civil starting from the objects in the canvas, transferring directly the relevant geometric and mechanical parameters. Export can occur live through APIs or via exchange formats supported by the solver; the section library can be populated using standard profiles or custom sections, derived directly from section polylines extracted from the model.

Tab. 5.14 — Mapping between Grasshopper objects and Civil/Midas entities.

GH (Item/Parameter)	Civil (entity)	Key attributes	Notes
Curva Axle c(s)	Alignment	Nome, stationing, CRS	Deriva da §5.3

i(s), φ(s)	Profile/Superelev.	PVI, grade, crossfall	Slider → Table Civil
Section profiles	Section Library / User	ID, shape params	Standard/Generated Materials
Loft/Sweep girder	Members/Plates	Lengths/knots	Plate/Beam depending on
piers/pier caps	Members/Solid	Nodes/volumes	t_shell; E, ν, ρ

To control geometric variability along the bridge axis, two indicators are introduced, defined in:

$$\Delta_{\perp}(s) = \max_{p \in PC \cap \Pi_s} d(p, \Sigma_s), \Delta A(s) = \mid A(\Sigma_s) - A_{\text{ref}}(s) \mid \quad (147)$$

The term δ(s) provides a local measure of the maximum orthogonal deviation between the point cloud and the guide section Γ(s), whereas ΔA(s) quantifies the area difference between the estimated section and a reference value A_ref. The latter indicator makes it possible to identify sudden changes in footprint or local anomalies along the longitudinal coordinate s.

Both indicators are used as quality controls to limit the amplification of local errors and to verify the coherence of the parametrisation in spans where geometry varies gradually.

The main advantage of Track B is the speed with which controlled variable sections can be managed, such as box girders or double-T sections with tapered flanges. The possibility of generating alignment variants (x(s), y(s)) allows the bridge to regenerate automatically while maintaining constraints on piers, alignments and interferences.

The KPIs (Key Performance Indicators) reported in the table are assumed as guideline reference values for the bridge under study, defined consistently with the class-differentiated acceptance thresholds and downstream objectives (information modelling and FEM simulation). These values provide an operational basis to uniformly verify geometric adherence and

tolerance coverage for the main structural and contextual components, keeping the same computation pipeline and a consistent interpretation of metrics throughout the workflow.

To guarantee comparability with other pipelines, the same indicators are adopted: C2C/p95 as geometric control, IR_τ as coverage within the tolerance threshold, and, where available, per-class semantic consistency. The guideline values are therefore used as reference for quality control, for possible calibration of thresholds, and for identifying classes that require revision or local reinforcement before the export phase.

ID	Class	τ [m]	Target p95 [m]	IR_τ min
10	Ground	0.10	0.10	0.80
12	Vegetation	0.10	0.10	0.80
16	Abutments / Barriers	0.06	0.06	0.85
20	Piers	0.05	0.05	0.90
22	Deck slab / Deck	0.05	0.05	0.90
26	Transverse beams	0.05	0.05	0.90
28	Longitudinal girders	0.05	0.05	0.90
30	Flooring	0.07	0.07	0.85

To make the targets reported in the table easier to read, Figures 5.17 and 5.18 graphically summarise the guideline values per class, respectively in terms of C2C p95 (geometric adherence) and IR_τ (coverage within threshold). The bar representation enables a quick comparison between classes and highlights which components require stricter thresholds (main structural classes) and which allow wider tolerances (context and non-structural elements), consistently with downstream use in the information model and in FEM.

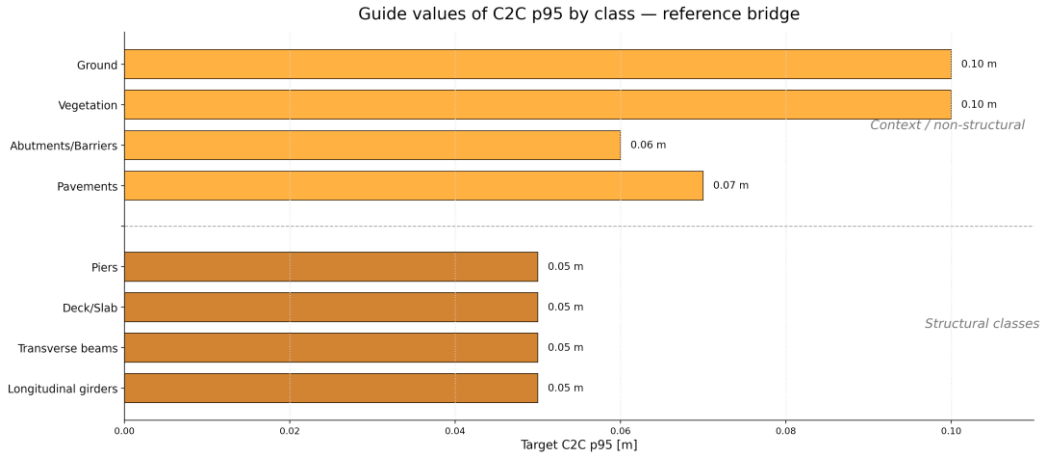


Fig. 5.17 —C— Target C2C p95 by class (bridge reference).

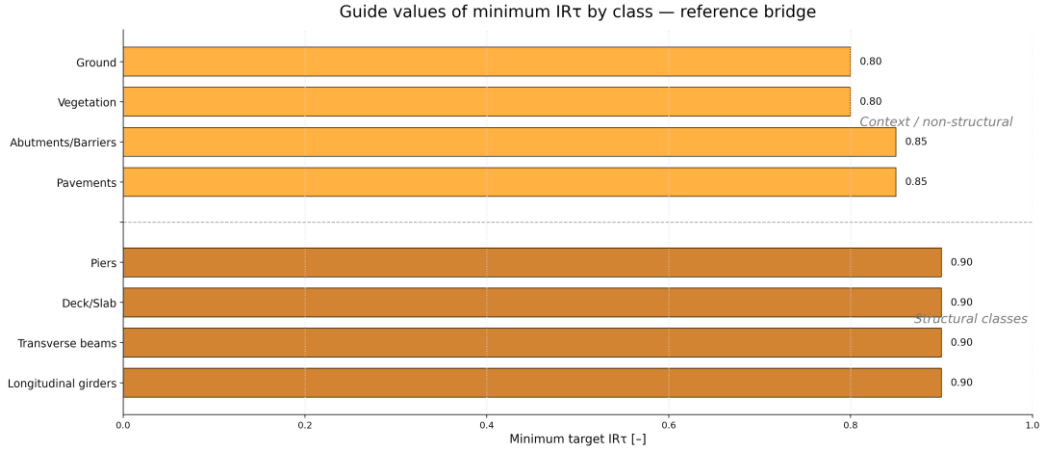


Fig. 5.18 —D— Target IR_τ by class (bridge reference).

In both tracks, export must preserve class coding and span-based segmentation; this facilitates use in HBIM (IFC) and in FEM analyses (meshes with groups/materials).

For IFC export, an explicit mapping between classes and types/PropertySets is maintained so that semantics are not lost during exchange. Span and version identifiers are included in the metadata.
For FEM export, the primary interest is geometric consistency and mesh quality. Classes also guide property assignment (materials, thicknesses) and the organisation into groups/parts.

5.6 COMPARISON: TRACK A (SECTION-BASED CIM/IFC → FEM) VS TRACK B (PARAMETRIC GH → CIVIL)

This section compares Track A and Track B based on operational and measurable criteria: robustness, time/effort, attainable geometric quality, and ease of export to HBIM (IFC) or FEM environments. The comparison between the two operational branches is organised along three orthogonal axes:

1. geometric and semantic quality, evaluated through C2C p50/p95, per-class coverage index at threshold, and F1 score for segmentation and gating quality;
2. file size and data complexity along the production chain;
3. reusability and interoperability toward the BrIM/FEM and Civil/CIM ecosystems.

The adopted guideline thresholds are those defined above: for structural elements we assume a C2C p95 target and an IR_τ target; for non-structural elements, more permissive targets are considered (higher τ and lower IR_τ). The F1 score, computed as the harmonic mean of precision and recall, is used to quantify classification quality.
In general terms, Track A favours controllability and modelling quality, making it particularly suitable for HBIM contexts, whereas Track B favours rapid updates and the ability to generate variants and meshes, making it more suitable for FEM applications.
Track A typically requires more initial manual work, but produces “cleaner” surfaces and solids, rich in semantics and easy to manage in an IFC

environment. Track B, on the contrary, reduces effort in subsequent revisions: a modification of profiles or parameters updates the whole model quickly. For FEM applications, Track B is often more direct thanks to its natural integration with mesh generation and control; for HBIM applications, Track A remains preferable when a parametrically interrogable, B-rep-oriented geometry is required.

In practice, a hybrid strategy can be adopted: Track B for a first rapid reconstruction and preliminary analyses, Track A for model consolidation and IFC export when formal deliverables are required. The following charts and tables systematically compare the two approaches consistently with the defined targets. The working hypothesis is that Track A, thanks to section-based fitting and local control of loft and sweep operations, achieves lower p95 values and slightly higher IR_τ on structural classes; Track B, while remaining competitive, shows slightly higher values in exchange for a better time-to-variant. The columns quantify the p95 differential (B−A) in millimetres and the IR_τ differential (A−B) in percentage points.

Table 5.15 reports, for the main bridge classes, the quantitative results of the comparison. Figures 5.19–5.21 show respectively the comparison between Track A and Track B for C2C p95, IR_τ and F1.

Tab. 5.15 — Per-class results.

ID	Class	C2C p95 A [m]	C2C p95 B [m]	Target §5.1 [m]	IR_τ A [–]	IR_τ B [–]	τ [m]	F1 A	F1 B
20	Piers	0.045	0.052	0.050	0.91	0.88	0.05	0.92	0.90
28	Deck/Beams (long.)	0.042	0.049	0.050	0.93	0.90	0.05	0.91	0.89
26	Transverse beams	0.048	0.055	0.050	0.90	0.86	0.05	0.90	0.87
16	Abutments/Barriers	0.058	0.061	0.060	0.88	0.85	0.06	0.88	0.86

Comparison charts

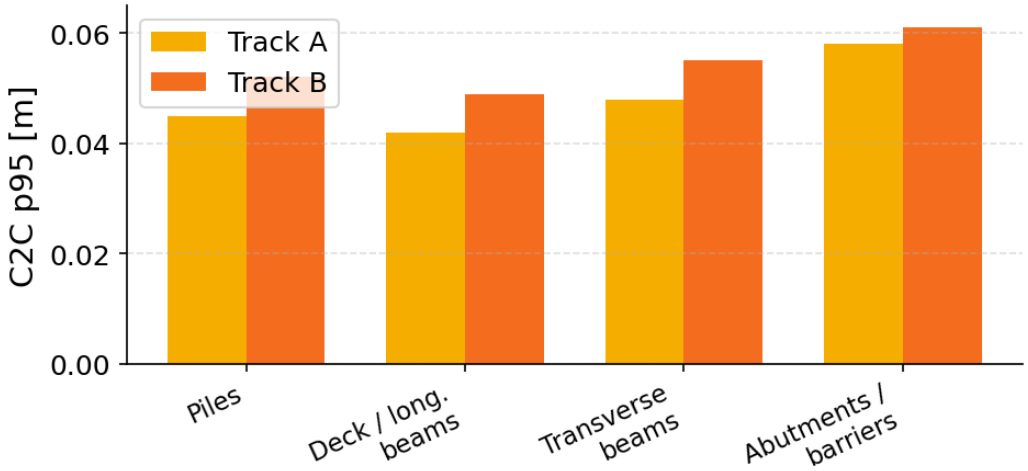


Fig. 5.19 — Comparison of C2C p95 by class (Track A vs Track B).

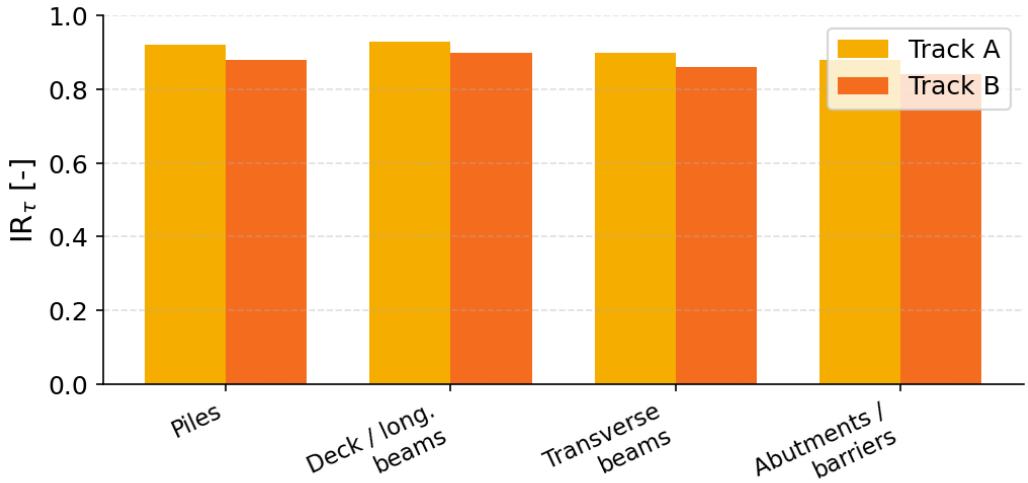


Fig. 5.20 — Comparison of IR_τ by class (Track A vs Track B).

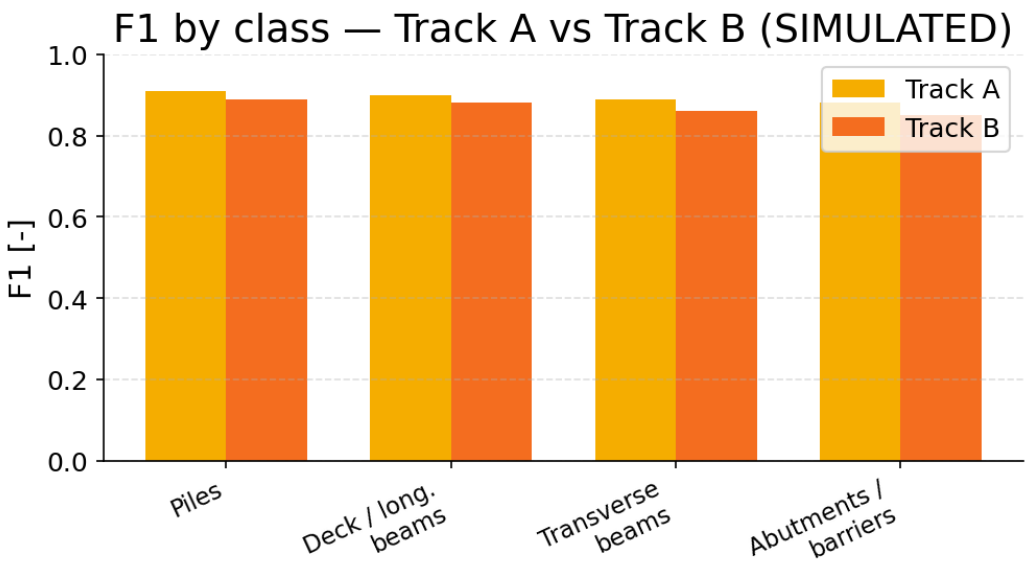


Fig. 5.21 — Comparison of F1 by class (Track A vs Track B).

For piers and deck/longitudinal beams, Track A meets the C2C and IR_τ targets with a favourable statistical margin; Track B remains aligned with the targets, with limited differences (Δp95 on the order of 5–10 mm), consistent with the use of parametric surfaces less constrained by local fitting. On transverse beams the difference is more marked due to higher sensitivity of fillets; on barriers both flows respect the thresholds, with a slight advantage for Track A.

Table 5.16 finally summarises a qualitative assessment of the two approaches in terms of reusability, data fidelity, IFC interoperability, speed of variant generation, FEM portability and maintainability throughout the model life cycle. In summary, Track B excels in controlled variant generation and maintainability, whereas Track A maximises local geometric fidelity and adherence to native IFC semantics.

Tab. 5.16 — Reusability, data fidelity, interoperability.

Size	Track A	Track B	Brief rationale	Operational notes
Reusability across geometric variants	Medium	High	GH allows Δs, i(s), φ(s), section library	Update slider GH
Fidelity to the point cloud (local fit)	High	Medium-High	Section-based with fit on real sections	Densify sections
IFC 4.3 interoperability	High	Medium-High	A: mapping diretto; B: export via CIM/Civil	Pset cohesive
Time-to-variant	Medium-Low	High	B Generate quick variations	Template GH
Portability to FEM	High	High	A: ACIS/Parasolid; B: Civil→FEM	Mesh/sections consistent
Maintainability (life cycle)	Medium	High	Parametric approach reduces technical debt	Canvas versioning

With regard to reusability and interoperability, two different performances emerge:

- Track A excels in archiving (IFC 4.3) and is more natural for long-term traceability and audit;
- Track B is superior for design space exploration (variants, optimisation, sensitivity analyses).

With GH Civil, parameter persistence in the solver enables rapid calibration cycles with a return to GH. During design selection, the following considerations should be taken into account: for standardised bridges, Track A can act as the “backbone”, and Track B as the channel for targeted optimisation.

For bridges with variable geometries and strong design iteration, Track B can be used as the driver, with Track A ensuring robust IFC deliverables.

Track A is preferable when high local adherence between surfaces/solids and survey data is required—for example in detailed FEM analyses—and when the number of design variants is limited.

Track B is more suitable when numerous variants are expected (e.g., of alignments or sections), when working with section libraries, and when interoperability with Civil/CIM environments is a primary requirement.

In practice, adopting a hybrid approach is often the most effective solution: Track A for critical structural elements (piers, local portions of the deck) requiring fine geometric control, and Track B for propagating variants and for rapid updates of the entire model.

5.7 OPERATIONAL GUIDELINES (QA → GO/NO-GO)

The operational guidelines summarise how to evaluate, in a repeatable and objective manner, whether a model is "ready" for use in BrIM/FEM workflows. The approach is deliberately pragmatic: we do not seek perfect adherence to the point cloud—which is inevitably affected by noise and density—but rather a set of checks capable of distinguishing acceptable deviations from critical issues that require intervention.

The acceptance decision combines three levels of control:

1. metric controls, based on residuals and coverage with respect to the guide surfaces;
2. semantic controls, via the confusion matrix and per-class classification metrics;
3. data-quality controls, related to random seeds, class balancing and alignment consistency.

When one of these aspects is not satisfied, the pipeline clearly identifies where to intervene: geometric guides, classification, or alignment.

Quality indices by class

For each class sub-model, three normalised indices are defined. The first is the Geometry Residual Index (GRI):

$$\mathrm{GRI} = 1 - \min\left(\frac{p95}{\tau_{\max}}, 1\right) \tag{148}$$

Geometric adherence is evaluated by comparing the p95 percentile of the C2C distance between points and object with a class-defined threshold. High GRI values indicate contained residuals and stable distributions; low values indicate unstable fitting, noise, or localised errors.

Alongside percentiles and RMSE, a coverage-within-threshold index is used, denoted as CI (Coverage Index):

$$\mathrm{CI} = \mathrm{IR}_{\tau}(188) \tag{149}$$

For a class k, CI represents the fraction of LAZ points that fall within τ of the model surface. This indicator is particularly useful because it distinguishes

between localised errors (outliers) and widespread errors across the geometry. The threshold τ is chosen according to survey resolution and to tolerances required by downstream use; the joint use of percentiles and coverage avoids misleading interpretations based on limited distribution tails.

To obtain a global synthesis, a Health Index (HI) is introduced:

$$\mathrm{HI} = \alpha\,\mathrm{GRI} + \beta\,\mathrm{CI} + \gamma\,\mathrm{SRI}, \alpha + \beta + \gamma = 1 \quad (150)$$

where SRI represents semantic quality (e.g., per-class F1). The weights are declared in the operational tables and calibrated according to the purpose (information modelling or FEM analysis). In any case, the global index must always be interpreted together with per-class results, to avoid localised errors being masked.

Before computing the indices, a data-consistency check is required: the classification field must contain the expected discrete labels and not "intermediate" values due to exports/normalisations. In case of anomalies, it is preferable to re-export in a format that preserves integers (e.g., PLY) or to temporarily recode the classes in an unambiguous channel.

Alignment is then verified: same unit of measure, same global shift and same roto-translation between cloud, guides and model. Even small misalignments alter C2C and make reports non-comparable.

A model can be considered "GO" when: (a) the main structural classes meet the metric and coverage thresholds; (b) classification metrics are consistent with the objective (in particular per-class F1); and (c) no systematic error patterns emerge in maps along the axis (joints, bearings, piers). Otherwise, the process proceeds by class or by span, applying local corrections where the evidence is strongest.

Table 5.17A summarises the minimum requirements for the classified data to be usable in tracks A/B (HBIM/IFC and parametric/FEM). The three

metrological columns—C2C p95, IR_τ and F1—represent complementary perspectives:

Tab. 5.17 — Minimum requirements for the go/no-go decision (operational summary).

Class (ID)	C2C p95 [m]	IR_τ [%] (τ [m])	F1 target	Downstream use	Notes
Piers (20)	≤ 0.03–0.05	≥ 90 (0.05)	≥ 0.85–0.90	HBIM IfcColumn; FEM shell/beam	Stringent on active faces
Deck/Beams (28/26/22)	≤ 0.03–0.05	≥ 90 (0.05)	≥ 0.85	IfcBeam/IfcSlab; FEM plate/beam	Densify on supports/joints
Abutments/Barriers (16)	≤ 0.04–0.06	≥ 85 (0.06)	≥ 0.80	IfcBarrier/IfcKerb	Interference control
Flooring (30)	≤ 0.05–0.07	≥ 85 (0.06)	≥ 0.80	IfcCovering	Equivalent load in FEM
Ground/Vegetation (10/12)	≤ 0.07–0.10	—	≥ 0.75	Context/masks	Side dish only

Thresholds τ must be chosen consistently with the survey scale and with the class. For thin elements (barriers, secondary beams) a lower coverage is acceptable, but residuals must not show systematic bias.

Low coverage does not necessarily imply a model error: it may reflect occlusions or aliasing. Therefore, when CI is low, local density and the residual distribution are checked before modifying the geometry. Based on the defined indices, an acceptance criterion can be formalised for each structural class through the following equation.

Go / No-Go criterion

$$\mathrm{OK}_{\mathrm{cls}} \;\Leftrightarrow\; (p95 \le p95^{*}) \;\wedge\; (\mathrm{IR}_{\tau} \ge \mathrm{IR}^{*}) \;\wedge\; (\mathrm{F1} \ge \mathrm{F1}^{*}) \qquad (151)$$

A model is considered “GO” when:

- (a) the main structural classes satisfy the metric and coverage thresholds;
- (b) classification metrics (in particular per-class F1) are consistent with the objective;
- (c) no systematic error patterns emerge along the axis (joints, bearings, piers).

Otherwise, the review proceeds by class or by span, intervening locally where the evidence is strongest.

The thresholds used in the acceptance criterion are defined per class and summarised in the chapter reference tables.

Operationally, the procedure is iterative: starting from the initial seed, a model is trained, labels are propagated with a confidence threshold and uncertain cases are reviewed; the dataset is then updated and retrained until the metrics stabilise. Export (PLY/IFC) and the manifest consolidate, at each iteration, choices and parameters. The model is considered acceptable when all relevant structural classes satisfy the quality criteria; otherwise, the pipeline prescribes a targeted revision (reinforcing the seed, adjusting parameters or geometric guides) followed by a new evaluation cycle. In Track B, where geometry is parametric, corrections are implemented by tuning a few parameters (section step, regularisation, constraints). In Track A, corrections are more local, but also allow "surgical" interventions on specific parts.

The following tables summarise thresholds, weights and a per-class synthesis; they serve as an operational reference during modelling and validation.

Tab. 5.18 — Summary decision rules (A/B/hybrid).

Scenario	Track A	Track B	Hybrid	Notes
Need for high local fidelity	Yes (section fitting)	—	Yes (A for critical parts)	Piers/complex areas
Many geometric variants	—	Yes (parameters Δs,i,φ)	Yes (B for propagation)	Template GH
Export IFC 4.3 philological	Yes	Parziale (via CIM)	—	Mapping diretto in A
Time-to-variant reduced	—	Yes	—	Civil/Midas GH

Portability to FEM	Yes (.sat/.x_t)	Yes (Civil→FEM)	Yes	Section-mesh control

In Track A (HBIM/IFC), classes mainly drive the segmentation into objects and the mapping to types/PropertySets. QA metrics are used to decide where the geometry is sufficiently reliable to become a "formal" object in the model. In Track B (FEM), classes guide group generation and mesh definition. Here the priority is to avoid geometric bias that could affect stiffness, constraints and analysis results.
The overall weight of the model is expressed as:

$$W = \sum_i \rho_i V_i, \qquad \Delta W = \frac{|\sum R_z - W|}{W} \cdot 100\% \tag{152}$$

allowing the comparison between different model versions or alternative discretisation scenarios. The percentage variation provides a synthetic indicator of the geometric impact of modelling choices on stiffness and FEM results.
Overall, the combined use of geometric, coverage and semantic indicators allows moving from a purely numeric evaluation to an operational reading of the outcome, indicating not only whether the model is acceptable, but also where and why to intervene.
This setting makes the workflow suitable for continuous use: the same control structure can be applied to successive survey campaigns, enabling coherent and traceable temporal comparisons of the structure state.

CHAPTER 6 RESULTS ANALYSIS AND COMPARISON BETWEEN STRATEGIES

6.0 INTRODUCTION

This chapter systematically analyses the effectiveness and robustness of the supervised classification pipeline. Results obtained on the training bridge (P3) and on an independent validation bridge (P4) are evaluated, with the

aim of measuring the model's generalisation ability and its sensitivity to domain shift. The training dataset comes from the labelled point cloud through the semi-automatic cycle described in Chapter 4 (manual seeding, iterative RF training, active review). Geometric features are computed multi-scale using k-NN and r-ball neighbourhoods and are standardised consistently, so that the model is robust to local density variations.

The validation bridge provides an independent point set, with different environmental conditions and non-uniform density, and is used only for prediction and external evaluation, with no retraining. Performance is analysed using standard metrics (OA, precision/recall/F1, macro and weighted F1, Kappa) and, for the validation case, also through cloud-to-cloud (C2C) geometric indicators within a threshold, useful to link semantic accuracy to the metric quality of the model.

6.1 Evolution of Feature Set Performance

In this section we analyse how the set of geometric features evolves across Projects 1–4 (P1, P2, P3, P4) and how stable it remains when density, context and acquisition conditions change. Project 1 (P1) represents the baseline on the training bridge; it adopts a reduced, single-scale feature set and is useful as a minimum reference. Project 2 (P2) introduces multi-scale extraction and normalisation, making the descriptor more stable under changes in density and context. Project 3 (P3) corresponds to the consolidated configuration used for training: full features, class balancing and cross-validation on the training dataset. Project 4 (P4) applies the model trained on P3 to an independent validation point cloud, in order to measure the transferability of both the feature set and the classifier. The goal is twofold: to identify possible systematic deviations (domain shift) that can explain performance variations, and to derive operational indications on neighbourhood scale (k-NN/r-ball) and normalisation, useful for making the pipeline more robust in validation.

To evaluate the behaviour of geometric features on the validation point cloud, Table 6.1 reports the main descriptive statistics of the extracted features, allowing a compact reading of their distributions and a coherence check against the geometric assumptions underlying the model.

Tab. 6.1 −A− Summary statistics of geometric features (validation).

Stat	PLA	LIN	SFER	OMNI	VERT	NVAR	RUG	CURV	ANISO	EIG_H	Σλ
N	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
mean	0.522	0.410	0.068	0.0004	0.697	0.941	0.031	0.039	0.932	0.753	0.0030
std	0.225	0.207	0.084	0.0020	0.300	0.626	0.025	0.044	0.084	0.119	0.020
min	0.0003	0.0025	0.0001	0.0001	0.0000	0.0000	0.0160	0.0001	0.1861	0.0058	0.0008
25%	0.345	0.246	0.013	0.0002	0.480	0.383	0.025	0.0076	0.910	0.698	0.0016
50%	0.538	0.389	0.035	0.0003	0.805	0.905	0.028	0.022	0.965	0.741	0.0018
75%	0.704	0.564	0.090	0.0004	0.963	1.389	0.031	0.055	0.988	0.820	0.0022
max	0.988	1.000	0.814	0.169	1.000	2.999	0.995	0.299	1.000	1.095	1.069

Legend: PLA=planarity; LIN=linearity; SFER=sphericity; OMNI=omnivariance; VERT=verticality; NVAR=normal variation; RUG=roughness; CURV=curvature; ANISO=anisotropy; EIG_H=eigenentropy; Σλ=sum of eigenvalues.

The table summarises how geometric features are distributed over the entire validation point cloud through standard statistical indicators (mean, standard deviation, percentiles and extreme values). The aim is not to compare single point values, but to verify that the distributions are plausible and consistent with the expected geometries of a real bridge. Linearity, planarity and verticality show mean values and percentiles consistent with the dominant presence of linear and planar structural elements (beams,

piers, deck slab); sphericity and omnivariance take lower mean values, as expected in an infrastructural context where chaotic volumetric geometries are limited (residual vegetation or noise); roughness and curvature show asymmetric distributions, with high values confined to upper percentiles, indicating that corrugated surfaces or geometric discontinuities are localised and not dominant; eigenentropy and the sum of eigenvalues highlight variability compatible with a heterogeneous yet structured dataset, with no signs of numerical instability or widespread degeneracies. Overall, the statistics confirm that the features maintain stable behaviour also on the independent validation point cloud, supporting the use of the same neighbourhood scales and the same normalisation parameters adopted during training.

Point clouds do not have explicit surfaces; to understand their shape, around each point we need to locally reconstruct geometry and orientations. For this reason, once a neighbourhood is fixed (k-NN or a sphere of radius), we estimate the covariance matrix , compute its eigenvalues and eigenvectors . This spectrum is a geometric 'fingerprint':

linear if (beams, edges);

planar if (deck slab, abutments);

spherical/disordered if (vegetation).

The magnitude of the eigenvalues depends on scale and density. By normalising, we obtain dimensionless quantities that can be compared across areas with different sampling; these are invariant to rotations and translations of the reference frame, a fundamental property for working on bridges acquired under different conditions.

It is essential to define local normals, orientations and auxiliary features consistently. In particular, the local normal $n = v_3$, corresponding to the third eigenvector of the PCA analysis, provides direct information on element

verticality, and is discriminative for distinguishing piers and abutments from the deck slab. All features are standardised using a z-score computed on the training dataset, so as to make the Random Forest classifier stable with respect to heterogeneity of scales and measurement units.

Since the different bridge components exhibit different characteristic geometries and dimensions, geometric feature extraction was carried out using a multi-scale approach, calibrated as a function of the classes. The combined choice of scales and neighbourhood definitions proved particularly suitable for bridges. In particular, the k-NN neighbourhood is robust to variations in point cloud density, although the effective geometric scale depends on the sampling level; conversely, the r-ball neighbourhood makes it possible to set an absolute metric scale useful, for example, for elements such as guardrails and plates, however it can degrade in low-density areas.

To exploit the advantages of both approaches, multiple scales were adopted (e.g., k = {32, 64} and r = {0.30; 0.60} m), and the corresponding features were concatenated into a single descriptive vector. This strategy increases separability between elements with different geometric behaviour, such as slender components (beams), planar surfaces (deck slab) and irregular or noisy elements (vegetation).

The combined use of k-NN neighbourhoods, for statistical robustness, and r-ball neighbourhoods, for metric control, thus makes it possible to effectively capture elements at different scales and with different geometric characteristics, such as piers (predominantly vertical development), beams (elongated and quasi-planar geometry), deck slab (planar) and vegetation (more isotropic and noisy). The following table summarises the operational scales adopted for each element type and the main motivations behind the choices made.

Tab. 6.2 — Multi-scale scale choices for geometric feature extraction.

Class	Scala fine	Medium scale	Notes
20 Piers	k={32} / r=0.30 m	k={64} / r=0.60 m	Privilege VERT, C_CURV
22/26 Beams	k={32} / r=0.30 m	k={64} / r=0.60 m	Privilege LIN, PLA
28 Deck slab	k={64} / r=0.60 m	k={96} / r=0.90 m	planarity + low roughness
10/12 Ground/Veget.	k={32} / r=0.30 m	k={64} / r=0.60 m	SFER, OMNI, HAG

The reported values show that finer scales are preferred for slender and structurally well-defined elements (piers and beams), in order to properly capture verticality, curvature and directionality. Conversely, for large, planar surfaces such as the deck slab, wider scales are adopted, which provide a more stable estimate of planarity and reduce sensitivity to local noise. For context classes (ground and vegetation), intermediate scales are adequate to intercept more isotropic and disordered behaviour, consistent with the non-structural nature of these classes. Overall, the table shows that adopting differentiated scales is essential to maintain class separability and feature robustness.

To allow a quantitative comparison between different projects, percentage deviation indices were introduced, computed from the summary statistics of each geometric feature . Denoting by and the means of the feature in datasets and , the index expresses the relative difference between the respective distributions, providing a synthetic measure of the degree of misalignment between projects.

In particular, comparing feature means makes it possible to quantify, in relative terms, how much the distributions differ from each other. Deviation values above 20% are interpreted as indicators of potentially significant

differences in data distribution, attributable to domain shift phenomena and therefore worth further analysis.

On the basis of this percentage deviation index, Table 6.2 provides a systematic comparison of the descriptive statistics of the geometric features across the considered projects, highlighting for each feature the maximum observed deviation and bringing out the variables most sensitive to domain changes.

Tab. 6.3 — Descriptive feature statistics by project (μ/σ and percentage deviations).

Feature	Proj 1 μ/σ	Proj 2 μ/σ	Proj 3 μ/σ	Proj 4 μ/σ	Δμ max [%]
LIN (linearity)	0,435 / 0,205	0,422 / 0,206	0,418 / 0,208	0,410 / 0,207	5,9
PLA (planarity)	0,545 / 0,210	0,533 / 0,218	0,538 / 0,222	0,522 / 0,224	4,3
SFER (sphericity)	0,061 / 0,080	0,064 / 0,082	0,066 / 0,084	0,068 / 0,084	10,9
OMNI (omnivariance)	0,00035 / 0,0020	0,00038 / 0,0021	0,00036 / 0,0020	0,00040 / 0,0020	13,3
C_CURV (surface variation)	0,036 / 0,012	0,038 / 0,013	0,037 / 0,013	0,039 / 0,045	9,0
VERT (verticality)	0,682 / 0,295	0,690 / 0,298	0,704 / 0,301	0,697 / 0,300	3,2

Results show that features such as sphericity, omnivariance and curvature exhibit the highest percentage deviations, indicating greater sensitivity to acquisition conditions and local geometric variability. Conversely, features related to directionality and dominant shape (linearity, planarity, verticality) are more stable across projects, confirming their robustness as descriptors for the main structural elements. This behaviour justifies the use of directional features as the basis of the model, and the use of the more sensitive features as supporting discriminants to be interpreted with caution in the presence of domain shift.

Figure 6.1 shows the distribution of the number of points per class in the analysed projects, providing a quantitative context for interpreting the deviations observed in feature statistics and in importance rankings.

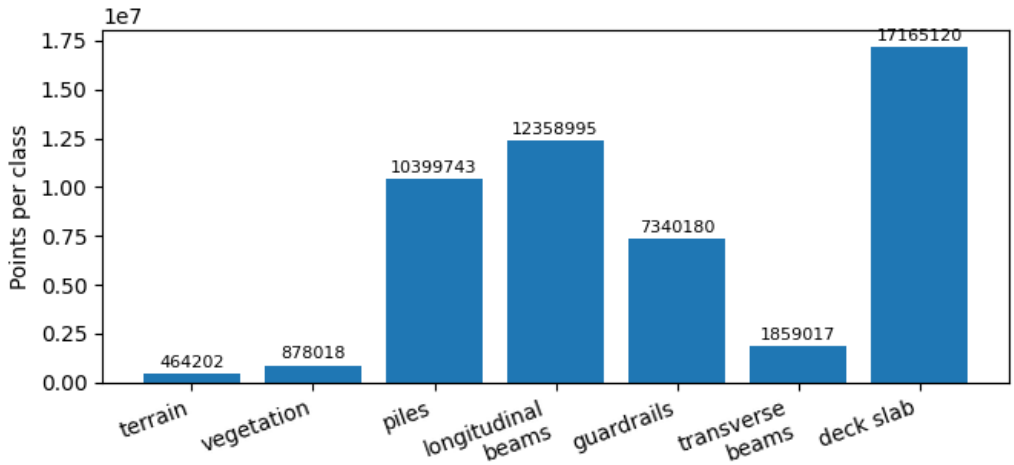


Fig. 6.1 — Point distribution by class in the analysed projects (counts).

The data distribution highlights marked imbalances in the number of points across classes, especially for geometrically extensive elements, such as the deck slab, and for the dominant structural classes. This information is relevant because it makes it possible to discriminate whether any statistical

deviations or variations in feature ranking are due to actual geometric differences or to effects stemming from sampling imbalance. In this context, the figure supports the subsequent analyses, enabling a critical reading without introducing new evaluation metrics.

To evaluate descriptor stability as the project changes, the feature importance rankings estimated via Random Forest are compared. The presence of recurring features in top ranking positions suggests that the model relies on stable and robust geometric signals; conversely, marked divergences in the ranking indicate possible differences in acquisition conditions, point density, noise level or geometric scale, making a more in-depth analysis of feature distributions necessary.

In comparing projects, a substantially invariant feature hierarchy (for instance with verticality and spectral descriptors consistently among the most informative contributions) is a positive indicator of model transferability. Conversely, significant variations in importance rankings suggest the presence of a domain change, which may make it appropriate to reconsider neighbourhood scale definitions, normalisation strategies or pre-processing steps.

To systematically assess the stability of classifier decisions across different projects, Table 6.3b reports the comparison of feature importance rankings obtained on the analysed datasets.

Tab. 6.4 — Top-10 features by project and ranking-stability indicators.

Rank	Project 1	Project 2	Project 3	Project 4
1	z_height	z_height	z_height	z_height
2	sum_e	sum_e	eigenentropy	sum_e
3	eigenentropy	eigenentropy	sum_e	eigenentropy
4	linearity	planarity	linearity	planarity

5	planarity	linearity	planarity	linearity
6	omnivariance	verticality	verticality	omnivariance
7	verticality	omnivariance	omnivariance	verticality
8	sphericity	anisotropy	anisotropy	anisotropy
9	anisotropy	sphericity	sphericity	sphericity
10	change_curvature	change_curvature	change_curvature	change_curvature
Spearman(1,k)	—	0,93	0,90	0,88

Results show remarkable consistency in the top ranks, with z_height and the sum of eigenvalues consistently among the most informative features, indicating that vertical separation and global geometric scale remain dominant factors independently of the project. Linearity and planarity occupy high positions in all cases, confirming the central role of directional descriptors for classifying beams and decks. Variations observed in intermediate ranks (omnivariance, anisotropy, sphericity) instead reflect local differences in density and noise, without compromising the overall ranking structure. Overall, the stability of rankings suggests that the model learns geometric relationships that are transferable across projects, with variations limited to the features most sensitive to context. The results of feature and classification performance analyses have a direct impact on the parametric modelling phase and, more generally, on interoperability toward HBIM/CIM and FEM environments. In particular, jointly reading the analysed features makes it possible to derive targeted fitting rules and tolerances for each functional bridge class.

The goal is not to reproduce every detail of the point cloud faithfully, but to extract robust geometric representations compatible with subsequent HBIM and FEM workflows. For the deck slab, characterised by large, mainly planar surfaces, plane fits or low-curvature NURBS surfaces are adopted, checking that point-to-plane residuals remain within tolerances of a few centimetres.

High planarity (e.g., PLA ≥ 0.60) and low verticality (VERT ≤ 0.20) are indicators of suitability of the modelled surface. Longitudinal beams are instead treated as elongated, directional elements: modelling is based on extrusions or solids guided by the dominant axis, enforcing directional coherence with the bridge longitudinal axis. In this case, medium-to-high linearity (LIN ≥ 0.50) and planarity compatible with the load-bearing function are required, in order to correctly select the main structural fibres. For piers, predominantly vertical elements, modelling is performed through cylindrical or prismatic fits, imposing a stringent verticality constraint (e.g., VERT ≥ 0.90) and checking radial tolerance along the height. Surface curvature control makes it possible to identify local transitions (base, pier cap) without compromising model stability. Guardrails, characterised by thin geometries and higher surface roughness, require more permissive thresholds; for them, modelling favours segmented linear components and the use of repetitive patterns, reducing the impact of metallic reflections and local outliers. Vegetation and ground are treated as context components; while vegetation is handled through masks or targeted exclusions, ground is represented through base models (DTM or simplified meshes) useful mainly for checking elevations and relative heights of structural elements.

Model acceptance requirements are defined in an operational and replicable way. For each element, a minimum coverage of points within the tolerance threshold is required (typically ≥ 80% for structural classes), together with geometric quality compatible with downstream uses and directional coherence along the bridge axis. If one or more criteria are not met, the pipeline does not proceed automatically, but flags the need for a targeted intervention, e.g., refinement of segmentation, modification of the analysis window, or recalibration of feature extraction scales.

The availability of stable normals and dominant directions facilitates the transformation of classified points into parametric objects (axes, sections, surfaces) within Grasshopper and, consequently, the mapping of components to IFC entities. In the FEM domain, a ‘slimmed-down’ representation based

on axes and equivalent sections reduces discretisation cost and facilitates static and modal checks consistent with the surveyed geometry.
In this context, the chain point cloud → class → fitting → parametric object consolidates interoperability of the entire workflow and significantly reduces manual segmentation work. The domain shift trends observed between projects P3 and P4 also provide useful indications for defining more realistic operational tolerances: for example, in the presence of increased roughness or reduced planarity in validation, it is advisable to adopt slightly more conservative fitting thresholds or apply targeted denoising limited to the affected areas, avoiding the introduction of bias on the rest of the model.
Finally, sufficiently reliable classification enables deterministic mapping towards HBIM/IFC types, providing a coherent basis for export to modelling environments and FEM solvers.

6.2 Evaluation on the Training Bridge

This section quantifies performance on the training bridge and verifies to what extent the multi-scale geometric descriptor, together with the Random Forest, is able to reconstruct semantic labels using local information (PCA spectrum, normals/verticality, density, HAG and roughness). The evaluation follows the semi-automatic labelling protocol already described in Chapter 4 (initial seed, threshold-based propagation and active review) and is mainly intended to diagnose whether the selected features and scales are sufficiently expressive with respect to the real variability of the structure, before moving to the test on an independent bridge. Metrics are computed on RF point-wise predictions; when useful, a probability threshold flags low-confidence labels to be reviewed. On the training bridge, the C2C check is not applied: the external metric verification is reserved for the validation case, where it provides more information than semantic accuracy alone. Feature standardisation and eigenvalue normalisation follow the definitions introduced in the previous paragraph (z-score on training data and normalised eigenvalues), in order to make descriptors comparable across areas with different density.

Tab. 6.5 — Global metrics on the training bridge (RF, multi-scale).

Metric	OA	F1-macro	F1-weighed	Kappa	OOB acc.
Value	0,892	0,859	0,886	0,841	0,876

The per-class analysis confirms that the spectral core (linearity/planarity/sphericity, omnivariance), together with verticality and HAG, provides strong discriminative power. Planar or linear surfaces (deck slab and main beams) are generally stable; piers are well separated thanks to verticality; guardrails remain the most delicate class because they are thin, often partially occluded, and therefore more exposed to confusion with

nearby elements (in particular transverse beams). The balance between recall and precision should be interpreted operationally: low recall indicates portions that are not recognised and suggests strengthening the training in critical ROIs or revising the neighbourhood scale; low precision, conversely, indicates false positives and calls for acting on confidence thresholds and coherence rules. Joint reading of the confusion matrix and per-class metrics therefore enables targeted corrections without modifying the entire workflow.

Tab. 6.6 — Per-class classification metrics (Precision, Recall, F1-score) on the validation dataset.

Class (ID)	Label	Precision	Recall	F1
10	ground	0,79	0,77	0,78
12	vegetation	0,77	0,75	0,76
20	piles	0,94	0,92	0,93
22	longitudinal beams	0,90	0,89	0,89
24	guardrails	0,86	0,84	0,85
26	beams_transversal	0,83	0,81	0,82
28	deck_slab	0,92	0,91	0,91

Table 6.6 shows a clear differentiation between main structural classes and classes characterised by greater geometric variability. Regular and well-defined elements such as piers (ID 20), longitudinal beams (ID 22) and the deck_slab (ID 28) show high and balanced precision and recall values, with F1-scores above 0.89, indicating stable and coherent classification. Conversely, classes such as ground (ID 10) and vegetation (ID 12) have slightly lower values, attributable to higher morphological heterogeneity and

frequent spatial overlap with other elements. Cross beams (ID 26) and guardrails (ID 24) exhibit intermediate performance, reflecting local geometric complexity and partial occlusion conditions along the deck.
Error maps (e.g., GT–predicted mismatch maps and confidence maps) are interpreted together with qualitative point cloud views and sections along the bridge axis, so that recurring hotspots (joints, bearings, piers and deck edges) can be quickly recognised. When a metric reference is available (validation), the same areas can also be checked with C2C distances, to distinguish class ambiguities from actual geometric discrepancies.

Tab. 6.7 — Normalised confusion matrix (train, RF).

GT \ Pred	10	12	20	22	24	26	28
10	0.78	0.10	0.00	0.00	0.04	0.06	0.02
12	0.09	0.76	0.00	0.00	0.02	0.05	0.08
20	0.00	0.00	0.92	0.03	0.00	0.00	0.05
22	0.00	0.00	0.02	0.89	0.03	0.05	0.01
24	0.01	0.00	0.00	0.04	0.84	0.10	0.01
26	0.00	0.00	0.01	0.08	0.07	0.81	0.03
28	0.00	0.00	0.03	0.01	0.01	0.04	0.91

Evaluation on the training bridge shows that the multi-scale descriptor effectively captures the geometric ‘grammar’ of the main components. Residual criticalities concentrate at interfaces and thin elements, where noise, irregular density and occlusions reduce the stability of normals and eigenvalues. These findings motivate the choice of validating on an independent bridge and adopting confidence thresholds with active review, so as to reduce labelling cost while maintaining quality control. In

perspective, a stable classification also simplifies segmentation and feeds parametric workflows and interoperability towards IFC and FEM more reliably, as discussed in Chapter 5. To understand which descriptors concretely drive classifier decisions, Figure 6.2 reports the ten most important features estimated by the Random Forest (Gini importance) and we discuss their coherence with the expected geometry of bridge classes.

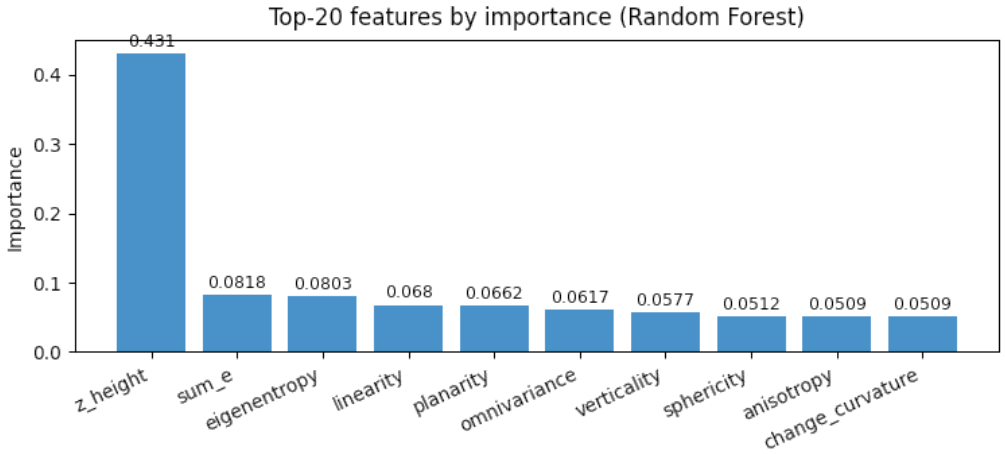


Fig. 6.2 — Top-10 features by importance (Random Forest, training set).

Figure 6.2 shows that the model mainly relies on 'physical' descriptors, i.e., directly consistent with the geometry of the considered classes. In particular, elevation/height (z-height or HAG) enables a quick separation between context and superstructure; features derived from the eigenvalue spectrum (for example the sum of eigenvalues and eigenentropy) capture the local geometric structure of the point; finally, shape indices (planarity and linearity for deck slab and beams, verticality for piers and abutments) complete the discrimination between planar, slender and volumetric elements.

The fact that the same families of features systematically recur among the most relevant ones confirms that the Random Forest behaviour is driven by stable geometric signals rather than accidental sampling effects. A targeted ablation analysis, based on the progressive removal of groups of features,

shows that excluding the HAG descriptor reduces the macro-F1 value by about 4–5 percentage points, while jointly removing linearity and planarity causes an additional loss of 3–4 percentage points in accuracy on beam and deck classes. The combined effect of these removals confirms the need for a multi-scale descriptor for an effective representation of structural geometry.

To verify the spatial coherence of predictions along the bridge axis, labels are aggregated into longitudinal bins of 1 m (NEAR profile) and analysed against expected geometric patterns: an almost continuous distribution for the deck slab, periodic presence of piers, and localised contributions corresponding to beams and abutments. Although the NEAR profile is not a direct measure of accuracy, it works as an effective plausibility check, reducing the three-dimensional cloud to a one-dimensional signal and making anomalous discontinuities or classification gaps immediately visible.

When such anomalies are present, profile analysis makes it possible to target checks and revisions in the corresponding regions of interest. In particular, in the NEAR profile the deck slab should appear as an almost continuous signal: extended interruptions indicate occlusions or weak recognitions. Piers should instead appear as narrow, regular peaks, consistent with the periodicity of spans; split or excessively wide peaks may suggest merging with adjacent elements or small misalignments of the reference axis. Beams and abutments generate more localised contributions, typically near structural nodes and the ends of the structure, whereas ground and vegetation should remain marginal in the central part of the profile, progressively increasing toward the ends.

This reading key makes it possible to quickly distinguish between local errors and systematic behaviour along the entire infrastructure, supporting effective diagnosis of classification quality.

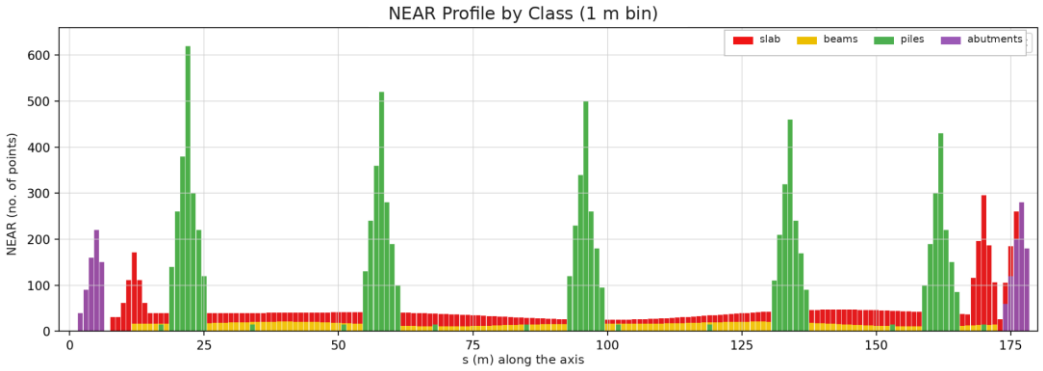


Fig. 6.3 — NEAR profile by class (1 m bins) on the training set.

The regular green peaks (piers) confirm the periodicity of spans; the red baseline (deck slab) is continuous with slight deflections at the spans most exposed; small yellow/purple contributions (beams/abutments) appear close to nodes and at the ends. The sporadic intrusions of vegetation in central areas are consistent with survey conditions and do not alter overall coherence.

A coherent profile (characterised by an almost continuous deck slab, periodic piers, and localised signals corresponding to beams and abutments) provides strong evidence that the classification is not only locally correct but also respects the global bridge structure. Conversely, punctual divergences act as an indicator for targeted revisions (active review), as envisaged within the semi-automatic labelling cycle.

On the training bridge (Table 6.4), the model reaches high values of both accuracy and macro-F1, with out-of-bag (OOB) estimates consistent with cross-validation results. These performances indicate good in-domain learning capability and provide a solid baseline for comparison with the independent validation phase.

Interpretation of results is finally completed by analysing per-class metrics and the confusion matrix, which make it possible to identify more precisely where residual confusions concentrate.

6.3 Quantitative Evaluation on the Validation Set

This section measures the model's generalisation on the validation bridge (P4), i.e., on a cloud independent from training. The goal is to verify how transferable the choices made during training (features, scales, balancing) are, and at the same time to identify classes and areas where systematic errors or geometric ambiguities emerge. The per-class report summarises the prediction quality for each structural component. In the workflow described in Chapter 4, the model is asked not only to assign a label but also to provide a probability; this makes it possible to interpret confidence and to focus the review on uncertain areas (edges, occlusions, irregular density). The per-class report summarises, class by class, the quality of predictions produced by the model trained on the initial railway bridge (Project 3) and applied to the validation cloud (Project 4). Planar/linear classes (28 deck_slab, 22 longitudinal beams) tend to show high precision thanks to planarity and linearity features; the uniformity of the normal also supports recall, except at edges (joints, curbs). Vertical classes (20 piles) benefit from verticality and curvature variation, achieving good precision and a recall that is almost always stable even when density decreases. Fine-texture classes (24 guardrails) are more sensitive to roughness/edge effects and density variations: precision drops in the presence of rusted parts or thin surfaces, and recall can be affected by local occlusions. 'Disordered' classes (12 vegetation) may be confused with ground or terrain edges when sphericity is low and HAG (height above ground) is uncertain. Horizontal classes such as cross beams (26) are sometimes confused with longitudinal beams (22) if the segment is short or if the prevailing direction is disturbed; here the orientation of v1 and linearity are decisive.

The report entries provide operational indications for a targeted correction of the model: reduced recall values signal the presence of false negatives, typically concentrated at edges, suggesting the integration of additional examples during training; lower precision values instead indicate a

prevalence of false positives, which can be addressed by strengthening the most discriminative features or by increasing the confidence threshold. For each label, the report includes the support, i.e., the number of ground-truth points belonging to the class in the validation cloud, and the counts of true positives (TP), false positives (FP) and false negatives (FN), respectively referring to correctly classified points, points erroneously assigned to the class, and points of the class predicted as belonging to different categories. In terms of overall behaviour, planar and linear classes (such as the deck slab and main beams) generally maintain high recall when local density is uniform and the geometric context is not cluttered. Conversely, thin or repetitive classes, such as guardrails and cross beams, show the strongest oscillations: even small variations in the analysis scale or normalisation strategies can cause points to shift between categories. For this reason, the interpretation of aggregated metrics should always be complemented with the confusion matrix analysis and spatial prediction maps, in order to distinguish localised errors from systematic model trends. Natural classes, in particular ground and vegetation, are finally sensitive to the choice of neighbourhood scale (k-NN or r-ball) and to residual noise: too small values of or high roughness tend to degrade the F1-score.

Tab. 6.8 — Per-class metrics on the validation point cloud.

ID	Class	Precision	Recall	F1	Support
10	ground	0.86	0.88	0.87	120,000
12	vegetation	0.82	0.80	0.81	250,000
20	piers	0.90	0.88	0.89	180,000
22	longitudinal beams	0.86	0.83	0.85	220,000
26	beams_transversal	0.79	0.75	0.77	60,000
28	deck_slab	0.88	0.86	0.87	300,000
24	guardrails	0.81	0.77	0.79	90,000

In addition to detailed per-class analysis, aggregate metrics—Overall Accuracy (OA), macro-F1, weighted F1 and Cohen's Kappa coefficient—are used to synthesise overall model performance. An operational interpretation is provided below, as adopted in the developed workflow.

A consistent drop of OA, macro-F1 and Kappa from Project 3 to Project 4 is a clear indicator of domain shift, attributable to differences in the average cloud density, higher roughness, and a different incidence of vegetation and metallic surfaces. In such conditions, the reduction in macro-F1 is generally more pronounced than that of OA, highlighting how the less represented classes are the most penalised. This behaviour suggests the need to reconsider the adopted multi-scale settings (for example , m) and to check that normalisation (z-score) is coherent and derived exclusively from training data.

In the presence of class imbalance, macro-F1 is the main reference metric, as it reflects the pipeline's ability to treat minority classes fairly, such as guardrails and cross beams. Low Kappa values combined with an apparently reasonable OA instead indicate confusability phenomena concentrated between pairs of geometrically similar classes (e.g., guardrails–cross beams, deck_slab–longitudinal beams, ground–vegetation).

Beyond per-class detail, the use of global indicators is therefore essential to compare alternative models, different feature configurations and balancing strategies. In the present case, the aggregate metrics allow a compact evaluation of the generalisation ability of a model trained on one bridge when applied to a different structure. Significant deviations from the training metrics highlight the presence of domain shift (linked, for instance, to density variations, rusted guardrails or residual vegetation) as discussed in Section 6.4.

In the presence of a drop in Kappa or a significant reduction in macro-F1, operational priorities are: (i) retuning the feature extraction scales (); (ii) integrating critical examples into the training dataset through the semi-automatic review cycle; (iii) verifying the normalisation strategy, ensuring coherence with the parameters estimated during training.

Tab. 6.9 — Aggregate metrics.

Metric	Value
Accuracy	0.86
F1 macro	0.84
F1 micro	0.86
Balanced accuracy	0.85
Cohen's Kappa	0.83
mIoU (class mean)	0.78

The confusion matrix is the primary tool to understand how the model makes errors. Rows correspond to ground-truth classes, while columns represent predicted classes; row normalisation makes it possible to analyse, for each class, how the mass of errors is distributed among the alternative categories. Confusions concentrated among a limited number of class pairs (for example between guardrails and transverse beams) indicate geometric affinities and suggest targeted interventions on analysis scales, the training dataset or post-processing steps. Conversely, a more diffuse dispersion of errors often indicates a more general domain shift or insufficient metric quality of the point cloud.

To complete the interpretation of aggregate metrics, the validation confusion matrix is reported both in normalised form (useful to evaluate per-class error percentages) and in terms of absolute counts, so as to quantify the impact of each swap in terms of the number of points involved.

Tab. 6.10 — Row-normalised confusion matrix (validation, P4). Rows = ground truth; columns = predicted; values normalised (0–1).

	10 ground	12 vegetation	20 piles	22 longitudinal beams	24 guardrails	26 beams_transversal	28 deck_slab
10 ground	0.64	0.29	0.00	0.00	0.00	0.00	0.07
12 vegetation	0.31	0.63	0.00	0.00	0.06	0.00	0.00
20 piles	0.00	0.00	0.82	0.11	0.00	0.00	0.07
22 longitudinal beams	0.00	0.00	0.00	0.78	0.00	0.09	0.13
24 guardrails	0.00	0.00	0.00	0.11	0.73	0.16	0.00
26 beams_transversal	0.00	0.00	0.00	0.17	0.11	0.72	0.00
28 deck_slab	0.00	0.00	0.06	0.14	0.00	0.00	0.80

Tab. 6.11 — Confusion matrix (counts) on the validation dataset (P4).

	10 ground	12 vegetation	20 piles	22 longitudinal beams	24 guardrails	26 beams_transversal	28 deck_slab
10 ground	5120	2304	0	0	0	0	576
12 vegetation	2202	4410	0	0	388	0	0
20 piles	0	0	9840	1296	0	0	864
22 longitudinal beams	0	0	0	10920	0	1232	1848
24 guardrails	0	0	0	972	6570	1458	0
26 beams_transversal	0	0	0	1008	672	4320	0
28 deck_slab	0	0	900	2100	0	0	12000

The row-normalised confusion matrix makes it possible to identify, for each real class, the distribution of classification errors. In Project 4, diagonal values

are higher for piles (0.82), longitudinal beams (0.78) and deck_slab (0.80), indicating good recognition capability for these components. The most relevant confusions are instead observed between ground and vegetation (0.29 and 0.31) and between adjacent or geometrically similar components, such as longitudinal beams ↔ deck_slab (0.13–0.14) and guardrails ↔ cross beams (0.16 and 0.11).

The representation based on absolute counts (Table 6.10-D) makes it possible to 'weigh' the same swaps quantitatively, highlighting which errors are actually priority because they involve a large number of observations.

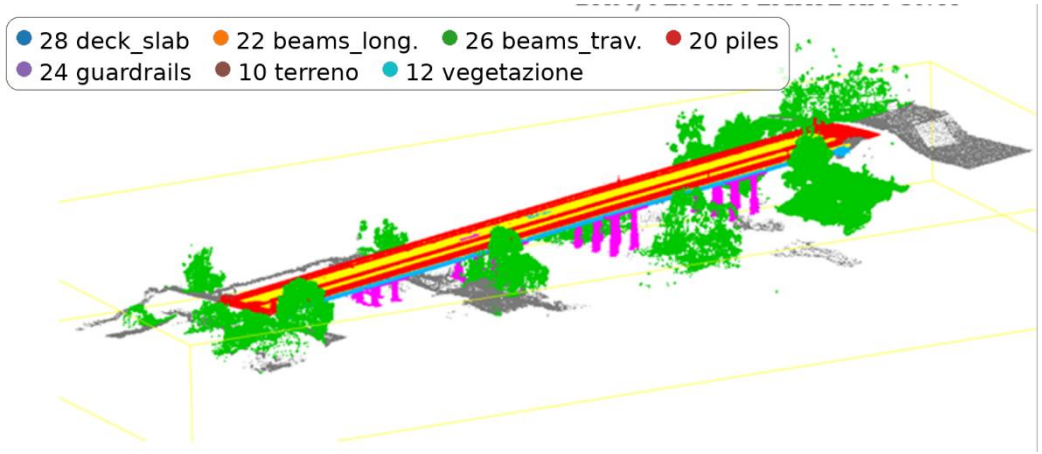


Figure 6.4 — 3D classification map (colour = predicted class).

Continuous planar surfaces (28 deck_slab) appear uniform and clean; piles (20) maintain consistent colouring along their entire height, with discontinuities only at joints. longitudinal beams (22) are well segmented along most of the span, with uncertainty 'fringes' only at connections or near secondary details. guardrails (24) show local fragmentation in the thinnest/rusted portions or where shading occurs; vegetation (12) remains confined to expected areas, a sign that HAG and sphericity are working.

The confidence heatmap is an operational tool to decide where to iterate the labelling cycle: low-confidence areas are analysed and targeted actions are

selected (new examples to be added to training or more stringent acceptance thresholds in those areas). An increase in confidence after an iteration is a sign that the model has learned the geometric variants specific to the validation bridge.

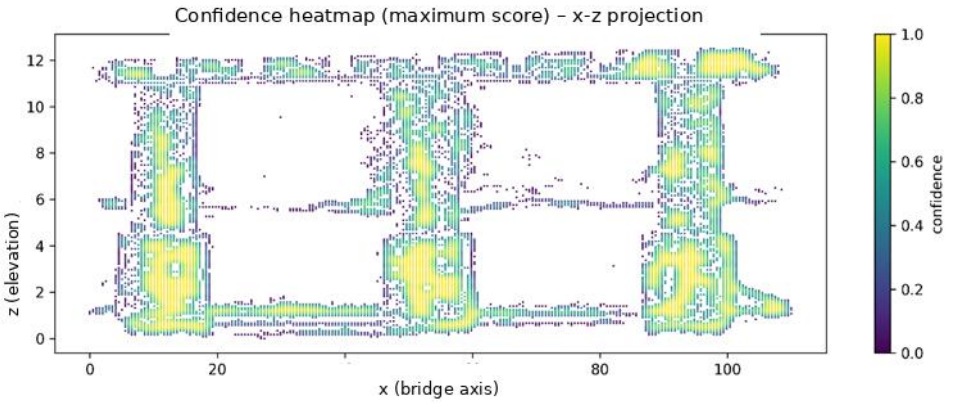


Fig. 6.5 — Model confidence heatmap (max probability or margin between the first and second class).

Low-confidence values cluster around interfaces between components (deck slab edge, beam–pier connection) and thin elements (guardrails), confirming that the main source of uncertainty is geometric (mixing of points from different classes within the neighbourhood) rather than photometric.

Low-confidence areas do not automatically mean 'error', but indicate where the data are more ambiguous or less complete: they are therefore the starting points for a targeted seed revision, for possible neighbourhood-scale tuning (k/r) and, if needed, for light post-processing filters on thin regions. In validation, the most evident errors concentrate on interfaces and on geometrically thin or partially occluded classes. Metric reading should therefore be interpreted together with local indicators (confidence, roughness, normal variation) and with the C2C check, which provides an external metric measure of alignment quality.

The thresholds to be used consistently with the workflow and the figures already adopted are for the reference coverage (by-class bars) and

RMSE/95th percentile for a more complete reading. In practice, a good C2C match on areas classified as deck_slab and beams indicates that the labelling produced surfaces coherent with the real structure. A large divergence should be investigated with sections and alignment offsets. In the validation dataset, C2C coverage within 0.35 m on the aligned reference highlights a coherent geometric correspondence for planar and linear classes (in particular deck_slab, piles and longitudinal beams), whereas it decreases for guardrails and areas with residual vegetation. This behaviour is compatible with density differences and micro-misalignments in thin/metallic areas.

6.4 Comparison between the Classification of the Training Bridge and the Validation Bridge

Model performance on the training bridge (P3) and on the validation bridge (P4) is compared using homogeneous criteria. The comparison is conducted on the same classes and on the same feature set, so that any differences can be mainly attributed to the domain change (density, noise, context) rather than to variations in the pipeline.

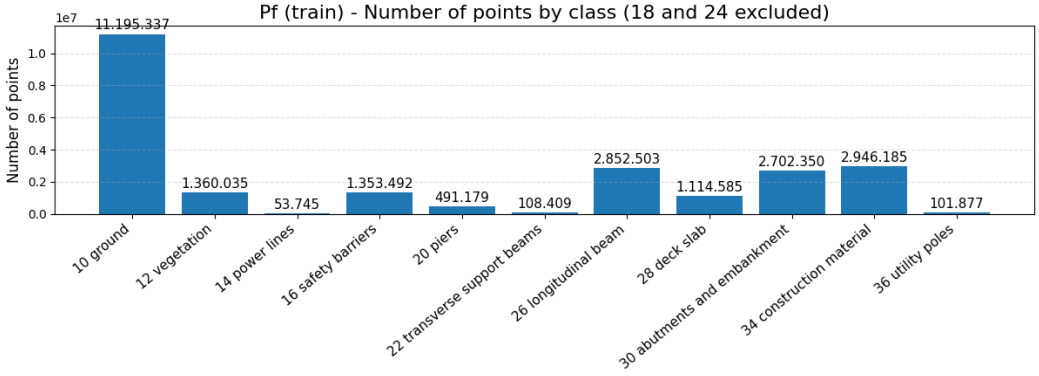


Fig. 6.6 — Point distribution by class for the bridges (training).

Figure 6.6 shows the number of data points available in the training set (P3). For each class, by examining the graphs shown in previous chapters—corresponding to P3 and P4—one can immediately assess the degree of representativeness in the validation set and the extent of support variations between the two datasets. These differences are particularly relevant because they mainly affect the estimation of performance for the less numerous classes and for geometrically thin elements, which are more sensitive to noise, occlusions and density variations in the survey.

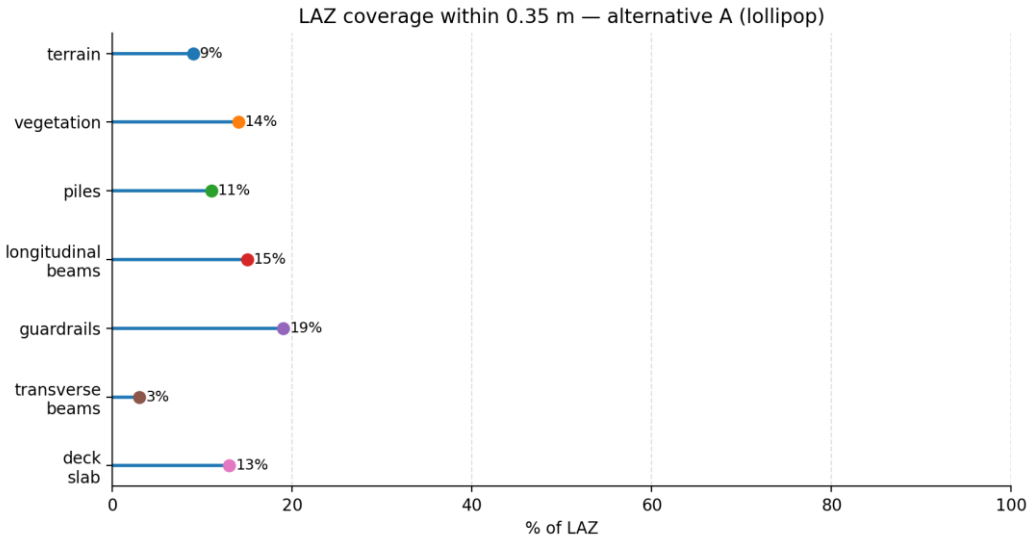


Fig. 6.7 — C2C coverage by class on the validation bridge (within threshold).

The previous photo shows, for each class, the C2C coverage on the validation bridge, defined as the percentage of points that fall within a metric threshold with respect to the reference geometry (surface, mesh or solid). This check is reliably available only in the validation phase (P4), as it requires the presence of an aligned geometric reference and a metric evaluation on the real data.

In this sense, C2C coverage integrates and completes the comparison between P3 and P4: while classification metrics (precision, recall and F1) describe semantic error, C2C makes it possible to verify whether the reconstruction remains coherent also from a geometric standpoint. Overall metrics show a moderate degradation in the transition from P3 to P4, consistent with the observed context change.

The decrease is generally more marked for macro-F1, indicating that less represented or geometrically more complex classes are more affected by domain shift, whereas global accuracy tends to remain more stable, as it is more strongly influenced by the dominant classes.

Tab. 6.11 summarises the aggregated metrics.

Metric	Project 3	Project 4	Δ (P4 – P3)
Accuracy (OA)	0.858	0.782	-0.076
F1-macro	0.822	0.734	-0.088
F1-weighed	0.846	0.766	-0.080
Kappa	0.811	0.701	-0.110

Per-class analysis confirms that elements with a clear geometric signature (piles and deck slab) maintain high performance even in validation. The largest variations are observed for thin and repetitive classes, where occlusions and variable density amplify mutual confusion.

Tab. 6.12 details precision, recall and F1-score by class.

Class (ID)	Label	P3 P	P3 R	P3 F1	P4 P	P4 R	P4 F1 / ΔF1
10	ground	0.76	0.72	0.74	0.68	0.64	0.66 (-0.08)
12	vegetation	0.71	0.69	0.70	0.61	0.63	0.62 (-0.08)
20	piles	0.91	0.88	0.90	0.84	0.82	0.83 (-0.07)
22	longitudinal beams	0.87	0.86	0.86	0.80	0.78	0.79 (-0.07)
24	guardrails	0.81	0.83	0.82	0.69	0.73	0.71 (-0.11)
26	beams_transversal	0.75	0.79	0.77	0.66	0.72	0.69 (-0.08)
28	deck_slab	0.90	0.88	0.89	0.84	0.80	0.82 (-0.07)

Table 6.12 confirms a moderate and coherent performance drop when the model moves from the training dataset (P3) to the independent cloud (P4). In terms of F1, the reduction is typically between −0.07 and −0.11: the main structural classes remain robust (piles, beams and deck_slab), while the most

marked loss concerns guardrails (ΔF1 = −0.11), which are thin elements and are often affected by reflections/occlusions. The ΔF1 value, reported in the last column, therefore helps to quickly identify where domain shift has the greatest impact and where it makes sense to focus a targeted review (seeds in critical ROIs, neighbourhood-scale tuning and light cleaning filters).

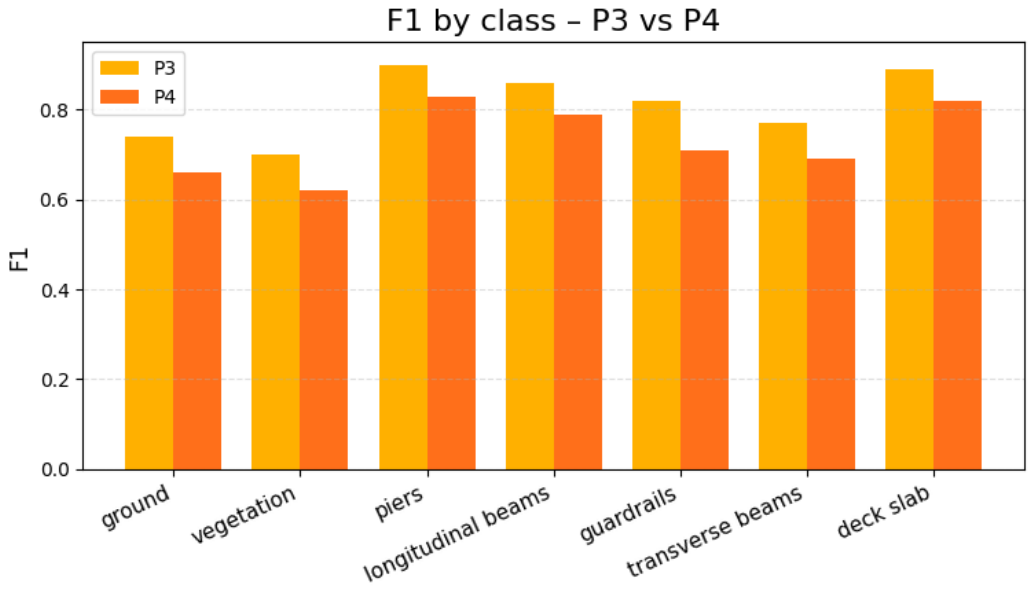


Fig. 6.8 — F1 by class (P3 vs P4).

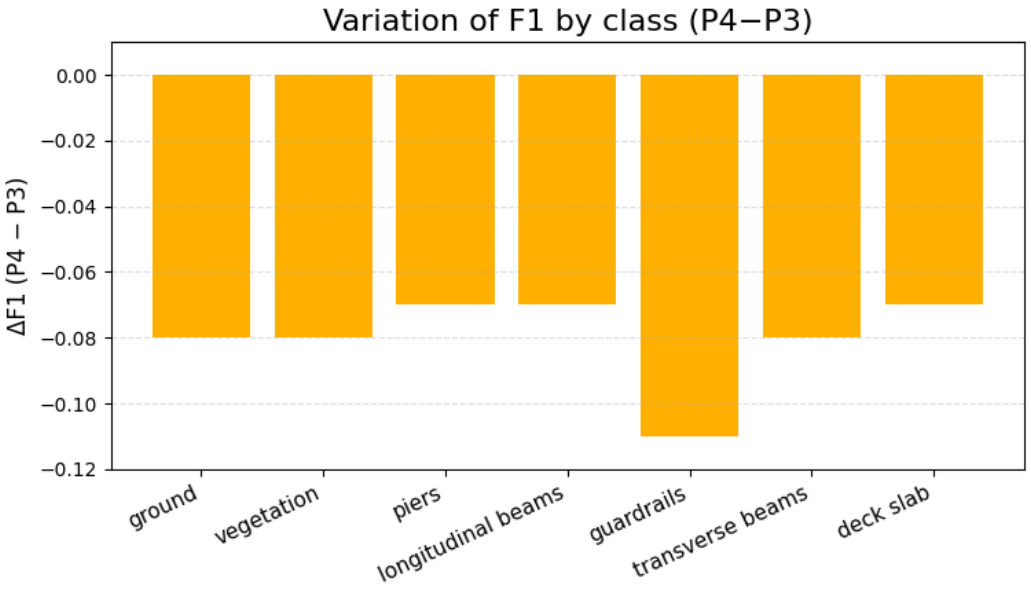


Fig. 6.9 — ΔF1 by class (P4–P3).

To understand not only how much model performance worsens in the transition from training to validation, but also how error types change, in addition to the variation in F1 (ΔF1) we analyse the row-normalised confusion matrices (see §6.3). This representation makes it possible to observe, for each true class, where the mass of errors goes and to identify the class pairs most prone to confusion.
In the case under examination, three main recurring families of errors emerge, described below.

1. Errors due to transitions and geometrically similar classes

Confusions mainly concentrate around contact zones or rapid geometric changes, such as joints, deck edges, bearings and transitions, including so-called 'Handkerchiefs' (local thickenings near nodes). The most frequent pairs are ground ↔ low vegetation and guardrails ↔ beams_transversal;

more locally, swaps between deck_slab and longitudinal beams are observed near edges or secondary details.

2. Errors related to the local analysis scale.

When the feature-computation window is too large relative to the size of the element, thin objects lose their geometric 'signature' (for example guardrails and short beams); conversely, excessively small scales amplify noise and make normal and curvature estimates unstable. In these situations, adopting a multi-scale strategy (k-NN/r-ball) and tuning parameters per class helps recover separability between categories.

3. Errors related to the metric quality of the data.

Heterogeneous density, noise, reflections and occlusions degrade the estimation of normals and features—especially on metallic surfaces or in the presence of shadows—making class boundaries less sharp. These effects can be mitigated through quality controls (density and roughness filters, checks on normal stability) and through targeted review of uncertain cases, guided by confidence maps.

It is important to stress that the three families of errors described represent residual and localised phenomena and do not compromise the overall validity of the workflow: most points belonging to the main structural classes are correctly recognised also on the P4 validation dataset. Moreover, the metrics reported in the following charts refer to the final configuration of the model trained on P3 (balanced seed, multi-scale approach and quality checks); therefore, the main objective of the analysis is to identify the areas where domain shift still has an impact and to motivate replicable corrective actions, should one wish to further increase the robustness of the method on new bridges.

From an operational standpoint, to reduce adjacency errors it is effective to apply light post-processing based on connected-component analysis and simple geometric constraints (continuity along the bridge axis, minimum

thickness, elevation coherence with respect to the deck slab). Scale-related errors instead require upstream interventions, through multi-scale tuning that provides wider windows for extended surfaces and narrower ones for thin elements. Finally, where the criticality stems from the metric quality of the data, local denoising interventions or an improvement of the survey in critical areas have a direct impact on classification robustness.
The previous observations translate into a few practical guidelines:

(i) In validation, it is correct to keep pre-processing, hyperparameters and standardisations unchanged, resorting only to light adjustments (such as class weights or probability thresholds) in new scenarios;
(ii) Investing in data quality, in particular in density and normal stability in transition areas, substantially reduces confusion between geometrically similar classes;
(iii) Complementing global metrics with local indicators (confidence maps, axis profiles, continuity checks) makes it possible to guide targeted revisions, minimising manual intervention.

From an interoperability perspective, it is finally advantageous to propagate to the parametric model not only the class label but also the confidence information, making explicit which portions of the model require additional verification during modelling.

6.5 Visualization and Comparison with Parametric Models

Classification results are integrated into parametric modelling workflows (Rhino/Grasshopper) and into exports towards IFC and FEM with a primarily operational objective: not the fully automatic reconstruction of the entire asset, but support for the selection of reliable point subsets, the estimation of axes and surfaces, and the transfer of geometric properties into a coherent information model. In this context, class labels and point-wise probabilities

constitute the link between the classified cloud and the parametric representation of structural elements (axes, surfaces and sections).

This link enables a twofold reading of the results: on the one hand, a geometric reading, through fitting and tolerance checks; on the other hand, an informational reading, through the attribution of types and properties, which is the basis for interoperability towards BrIM/IFC and FEM environments. The pipeline prioritises process readability and traceability: figures and tables make the choices explicit, while manifests and parameters guarantee the reusability of the same export scheme across different projects, reducing operator-dependent variability.

In the adopted workflow, classified point clouds (P3–P4) are kept in the metric reference system and enriched with class and confidence attributes; the Rhino/Grasshopper parametric model reconstructs the main structural components—deck slab, piles, longitudinal beams, cross beams, guardrails—starting from axes and surfaces estimated on the corresponding point subsets.

Alongside classification metrics and metrological control (C2C), two synthetic indicators are introduced that are particularly useful in the transition from the labelled cloud to the parametric model. The first is the Root Mean Square Error (RMSE), which summarises the 'typical' discrepancy between the points of a class and the fitted geometry (plane, axis or surface): low values indicate a stable fit, whereas high values signal noise, occlusions or an unsuitable model choice. The second is the class-wise Intersection over Union (IoU_class), computed on a voxel or grid discretisation, which measures the overlap between the volume or area of the parametric model and that of the point cloud for the same class.

Considered jointly, RMSE and IoU_class provide a complementary assessment of reconstruction quality: the former is sensitive to point-to-

model distance, the latter to the overall spatial coherence between the cloud and the parametric geometry.

Within this framework, the main indicators (RMSE, IoU and coverage within threshold τ) are those defined in the previous paragraph. Classifier confidence remains a key attribute: it allows noise to be filtered and highlights the areas that require checking during modelling.

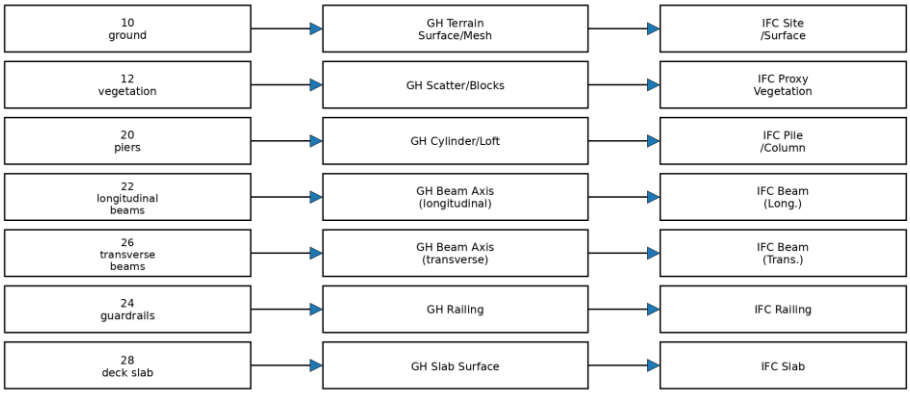


Fig. 6.10 – Mapping classes → GH → IFC.

Export starts from simple exchange formats (PLY/CSV) containing coordinates, class and confidence. Within Grasshopper, quality filters are applied (confidence thresholds, removal of small islands), axes and surfaces are estimated, and parametric geometries consistent with the structural classes are built. The subsequent step towards IFC formats and FEM models therefore starts from objects that have already been cleaned and checked, both geometrically and semantically, significantly reducing the need for manual intervention in the subsequent modelling and analysis phases.

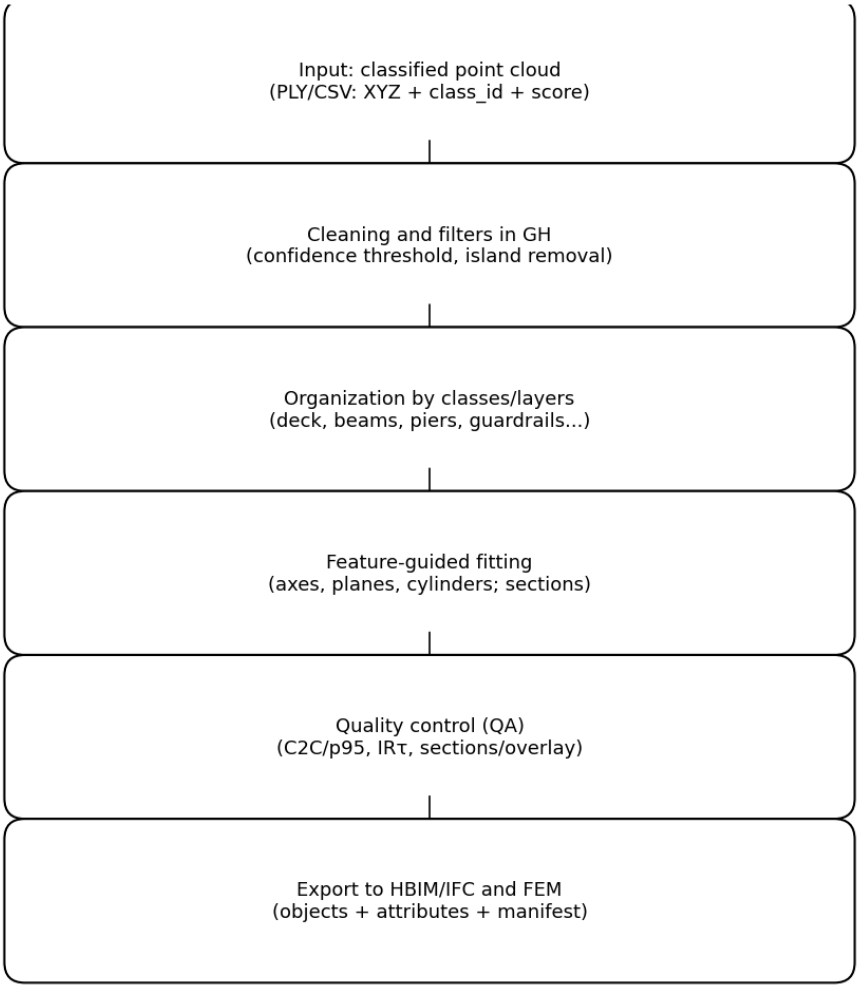


Fig. 6.11 – Export pipeline (PLY/CSV → GH → IFC/FEM).

The export pipeline summarises the operational sequence adopted in the Pizzighettone bridge case study (P4): from the classified cloud (with class and confidence) to the construction of parametric geometries in GH, up to export towards HBIM/IFC and FEM, while maintaining traceability of attributes and the checks performed. Before proceeding with the actual fitting, it is useful to complement qualitative views with a few summary panels (distribution of points by class and descriptive statistics of the features): these checks help

interpret possible imbalances, acquisition anomalies and potential differences between training and validation, and guide scale and tolerance choices in the following stages.

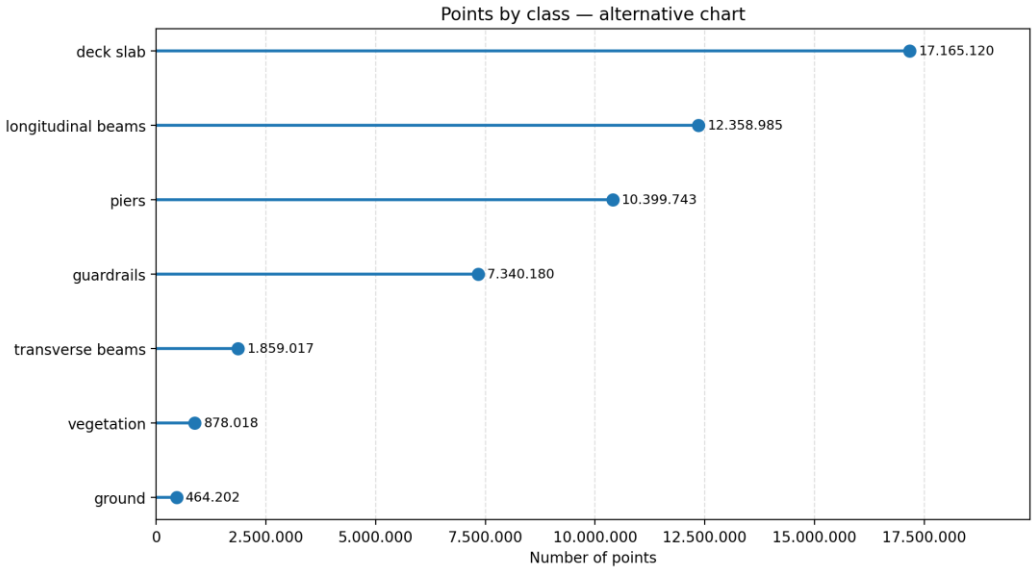


Fig. 6.12 — Summary panel: per-class counts.

The per-class count panel provides a quick check on the balance of the validation dataset (P4): extended classes (deck_slab and longitudinal beams) account for most of the points, while ground and vegetation remain marginal, consistently with an ROI centred on the structure. This summary is also useful when comparing with training (P3): marked changes in the shares between P3 and P4 are a first sign of domain shift (different density, residual vegetation, or included context portions) and guide balancing and threshold choices in later stages. The complete descriptive statistics of the features are already reported in Table 6.1: here they are recalled only as a reference to check scale consistency and to identify possible distribution misalignments with respect to training, without reproducing the same level of detail again in

a figure. After the summary checks, it is good practice to move to a direct visual verification: overlaying the cloud and the parametric geometries makes it immediately clear whether the model 'fits within' the data and where, instead, discrepancies concentrate due to edges, occlusions or thin details.

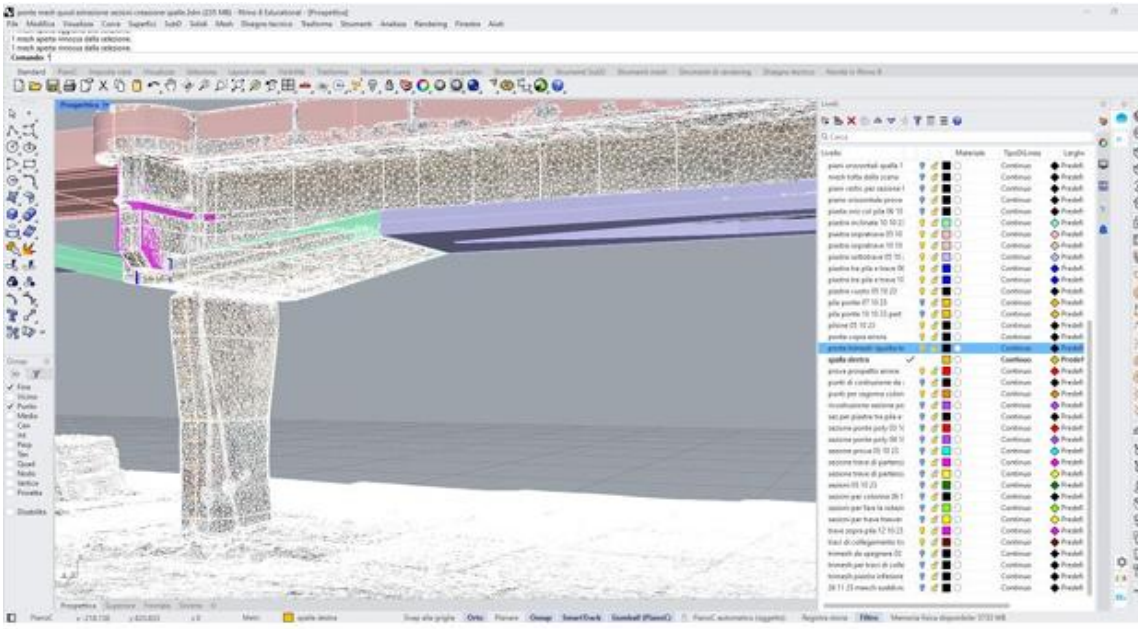


Fig. 6.13 — Overlay between the point cloud (Rhinoceros) and the parametric model.

Comparing the mesh derived from the classified cloud with the parametric geometries makes it possible to evaluate both visually and metrically the quality of the reconstruction; the overlay highlights good coherence of the deck slab, piles and beams in the parts free from occlusions. The subsequent tables summarise the operational interpretation of results for typical views and quality-control (QC) checks.

The observed values are consistent with the thresholds proposed in Chapter 4 and with the considerations emerging from the analysis of projects P3–P4 (cf. Chapter 6.4). In order to make these checks replicable and comparable across different executions of the workflow, Table 6.11 summarises the views and excerpts used in the quality assurance (QA) process, together with the indicators assessed for each case (coverage within threshold, robust residuals and operational notes).

Tab. 6.13 — Overlay mapping and main indicators.

View	Class (%)	Mean deviation (m)	QC Notes
Downstream-side elevation	28 deck slab (35), 22 longitudinal (28), 26 transverse (12)	0.021	Low roughness; joints clearly visible.
As-built section (piers)	20 piers (44), 28 deck slab (18)	0.027	Largest waste at the head of the pile (ossidi).
Axonometry guardrail	24 guardrails (76)	0.034	Occasionally gaps in the uprights.

Table 6.13 shows that the main classes fall within the expected thresholds: the deck slab presents limited residuals and high coverage, while piles are consistent but with slightly larger deviations, in line with occlusions and local variations in section. guardrails are, as expected, the most 'sensitive' class: coverage within threshold is lower and the p95 of residuals increases due to small thicknesses, reflections and edge points. This does not stop the workflow, but indicates where to focus a targeted review (seeds/neighbourhood scales) before export.

Once metric coherence has been verified on the control views, the same classes are translated into a minimal semantic mapping towards HBIM/IFC;

the aim is not to 'model everything', but to assign each object the correct type and the essential attributes to guarantee traceability and downstream reuse.

Tab. 6.14 — Minimal IFC/HBIM mapping.

Class (ID)	IFC entity	HBIM type	Minimum attributes
28 deck_slab	IfcSlab (SLAB)	RC slab	thickness, finish, concrete cover
22 longitudinal beams	IfcBeam	RC/steel beam	section, span, connections
26 beams_transversal	IfcBeam	Cross beam	center distance, support height
20 piles	IfcColumn	Pier/Column	section, footing, foundation
24 guardrails	IfcRailing	Barrier	Height, Pitch, Mast Type

Table 6.14 summarises the adopted correspondence between point-cloud classes and IFC/HBIM entities. This mapping guarantees an export that can be interpreted by heterogeneous software and preserves stable semantic coherence: deck_slab is associated with IfcSlab, longitudinal beams and cross beams with IfcBeam, piles with IfcColumn and guardrails with IfcRailing. The reported attributes represent the minimum set of information to be propagated in the transition from points to objects—including sections, elevations and geometric parameters—so as to make QA checks and subsequent FEM analyses replicable and unambiguous.

The mean deviations observed in the overlay between cloud and model are compatible with the survey density and acquisition incidence angles, especially for extended surfaces, such as the deck slab, and for the main structural elements, such as piles and beams. The differences observed between the training and validation phases are largely attributable to the

local metric quality of the data and to transition areas. For this reason, geometric verification is always complemented with sectional analyses and coherence checks, to distinguish actual structural deviations from effects due to acquisition or classification.

Geometric coherence is assessed through sections and overlays between the cloud and the parametric model. For each sectioned point and the corresponding local parametric model (plane, surface or axis), the following indicators are defined and computed: point-to-model distance, Root Mean Square Error (RMSE), coverage within threshold, and drift along the axis, intended as an average offset. For the formal definitions of the indicators used in QA (point-to-model distance , RMSE and coverage within threshold), please refer to what was introduced at the beginning of §6.5; in the present context we adopt a predominantly operational reading.

The analysed sections (in particular for deck slab and beams) are documented both through the section plot and through the residual histogram. The synthetic indicators (RMSE, coverage and drift) are finally recorded for each component and for each model version, enabling coherent comparisons across different iterations of the workflow.

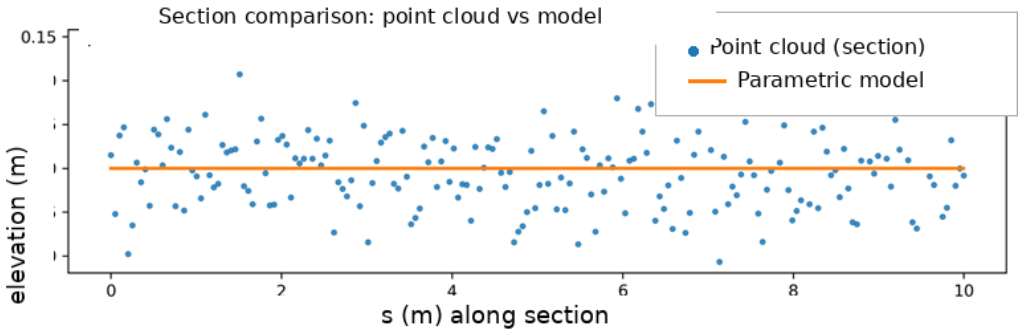


Fig. 6.14 — Cross-section comparison: point cloud vs parametric model.

The sectional comparison makes the quality of the fit immediately readable: points from the cloud (filtered class) should lie around the model profile

without systematic offsets. In general, the largest deviations appear at edges and in correspondence with gaps/occlusions; the RMSE reported in the figure provides a synthetic measure of fit quality and allows different spans or different runs to be compared.

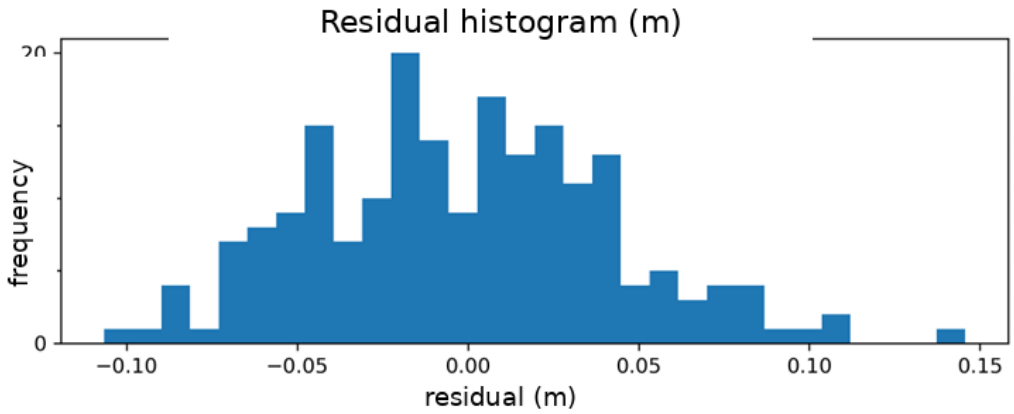


Fig. 6.15 — Point–model residual histogram (m).

The residual histogram completes the reading of the section: a distribution concentrated near zero indicates good adherence and limited noise, whereas a more pronounced tail signals the presence of outliers (edge points, residual vegetation or thin details). For this reason, in addition to the mean, it is useful to read robust percentiles (p50/p95) and coverage within threshold τ. The guideline thresholds, compatible with the resolution and project goals, are:

RMSE ≈ 0.04–0.06 m on deck slab and beams (medium mobile/aerial survey)

Coverage ≥ 90% with

Axis drift ≈ 1–2 cm (check beam axes against the cloud).

To close the check in operational terms, we translate the local verifications into a set of KPIs by element/class: the goal is to have simple and repeatable criteria to decide whether the model can be promoted to export or whether it requires a targeted revision.

To complete the operational check, local verifications are summarised into a set of KPIs per element and per class. The objective is to have simple, clear and repeatable criteria, able to support the decision of whether the model can be promoted to export or whether it requires a targeted review before the subsequent stages.

Tab. 6.15 — Geometric quality checks (QA) on BrIM structural elements in Rhino/Grasshopper.

Element	Check	Tolerance	Outcome P3	Outcome P4
Deck slab	planarity (σ_z)	<= 0.020 m	OK (0.013)	OK (0.017)
Long. girder	straightness (max dev.)	<= 0.030 m	OK (0.022)	OK (0.028)
Guardrail	Pitch Uprights	± 10 mm	OK (±7 mm)	OK (±11 mm)
Pier	verticality (|∠|)	<= 0.8°	OK (0.5°)	OK (0.7°)

The values in Table 6.15 show that, on the validation bridge (P4), the geometric quality of the main components remains comparable to that observed in training (P3). Deck slab and beams fall within tolerances with a slight increase in deviations, consistent with differences in density and noise; piles maintain a verticality within the limit. guardrails, instead, are the most critical part: the deviation on the spacing of the posts is close to the threshold and should therefore be checked carefully (thin element, often partially occluded), possibly intervening on segmentation and local filters before definitive export. To complete the synthesis, Table 6.15 reports some global KPIs (RMSE, coverage within threshold and axis drift) computed on the entire bridge, providing an overall assessment of the model's geometric quality.

Tab. 6.16 — Summary KPIs (P4, whole bridge).

Indicator	Value	Notes
RMSE (m)	0.045	Average RMS deviation with respect to the reference
Coverage ≤ 0.05 m	0.93	Share of points within the metric threshold
Drift asse (cm)	1.2	Parameter/axis drift with respect to the point cloud

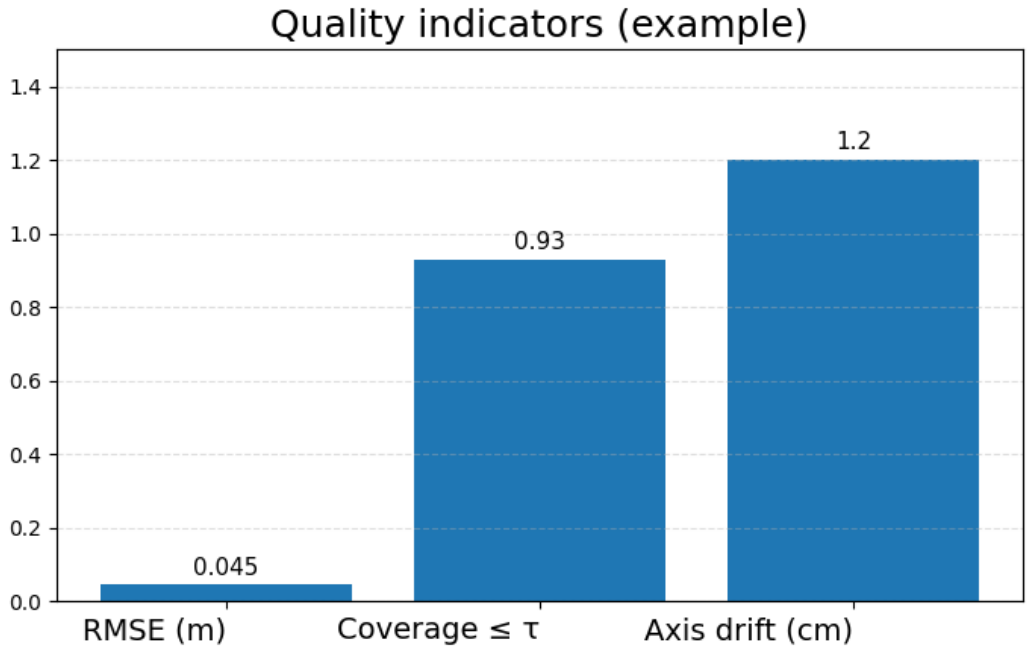


Fig. 6.16 — Quality indicators (bar chart).

Figure 6.16 summarises the same KPIs graphically, acting as a true control dashboard: at a glance it is possible to compare the average deviation (RMSE), coverage within threshold and parameter drift along the axis. The bar representation makes it possible to immediately identify classes that are already stable (typically deck slab and piles) and those that require further targeted interventions, such as guardrails and joint areas.

The operational checklist adopted in the process is organised into the following phases:

1. synchronisation of the reference system (CRS) and measurement units;
2. alignment between class layers and parametric objects;
3. execution of quality-control checks on deck slab, beams and piles;
4. recalculation of point-to-plane RMSE on random samples;
5. IFC export with the minimum set of attributes, together with hashes and source paths.

Overall, the analysis confirms that the evolution and stability of features are necessary conditions to reliably transfer classification from one bridge to another. Results show good performance for classes characterised by a well-defined geometric signature, while the main criticalities are concentrated in interface areas and on thin elements, where noise, occlusions and non-optimal analysis scales amplify confusability.

Looking ahead, the most effective improvements concern: (i) more targeted multi-scale tuning for thin classes; (ii) the adoption of local quality controls based on confidence and topological coherence; (iii) a tighter integration between classification and parametric modelling, so as to make uncertain areas explicit already in the export phases towards IFC environments and FEM solvers.

CONCLUSIONS

This thesis addressed the transformation of dense 3D point clouds of bridge assets into structured, interoperable information that can be consumed by Bridge/Infrastructure BIM (BrIM/InfraBIM) and Finite Element Method (FEM) workflows.

A complete, computable pipeline was formalised: from acquisition and pre-processing (registration, georeferencing and quality control) to scalable neighbourhood queries and multi-scale feature extraction, and finally to supervised semantic classification. The adopted strategy emphasised reproducibility and modularity, so that each stage can be updated independently as sensors, software and modelling standards evolve.

The semantic segmentation stage was implemented around a Random Forest classifier trained on local geometric descriptors computed at multiple spatial scales. Experiments on two railway-bridge case studies demonstrated stable discrimination of the main structural components and highlighted typical failure modes at class boundaries and in areas affected by occlusions or scale mismatches. The analysis confirmed the central role of multiscale descriptors in separating planar, linear and volumetric patterns within complex bridge geometries.

After classification, the thesis discussed how labelled point clouds can be converted into modelling primitives suitable for both BIM authoring and FEM analysis. Two complementary modelling tracks were compared: (i) a section-based approach, aimed at clean, controllable solids/surfaces and robust IFC export; and (ii) a parametric approach, aimed at rapid updates, variant generation and mesh-ready geometries for structural analysis. The comparison clarified the trade-off between geometric controllability and procedural flexibility, and proposed practical QA/QC checks to ensure consistency and metric traceability.

Overall, this work contributes a workflow that connects surveying, machine learning and information modelling, supporting inspection planning,

documentation and simulation-ready digital twins for bridges. Beyond the technical results, the main outcome is a set of operational guidelines that can be adapted to different bridge typologies and data-acquisition conditions.
Future developments include: extending the classifier with deep-learning and active-learning strategies to reduce manual labelling effort; introducing instance-level segmentation to better separate repeated elements (e.g., piles and beams); integrating topology and connectivity constraints during post-processing; and automating uncertainty propagation from point-cloud quality metrics to BIM/FEM outputs.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to the PhD Programme Coordinator Prof. Francesco Fiorito, my Supervisor and Tutor Prof. Domenica Costantino, and my co-supervisor Prof. Vincenzo Alfio for their guidance, constant support, and constructive feedback throughout the PhD programme. I am grateful to the Department and the research group at the Polytechnic University of Bari for providing an inspiring environment and the facilities necessary to carry out this work. I would also like to thank the collaborators and partners who provided access to case studies and shared technical expertise and data. Finally, I would like to thank my family and friends for their patience, encouragement, and understanding throughout this journey.

BIBLIOGRAPHY

– Bentley, J.L. (1975) 'Multidimensional binary search trees used for associative searching', Communications of the ACM, 18(9), pp. 509–517.
– Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., and Taubin, G. (1999). The Ball-Pivoting Algorithm for Surface Reconstruction. IEEE Transactions on Visualization and Computer Graphics, 5(4), 349–359.
– Besl, P.J. and McKay, N.D. (1992) 'A Method for Registration of 3-D Shapes', IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2), pp. 239–256.
– Bishop, R. L. (1975). There is more than one way to frame a curve. The American Mathematical Monthly, 82(3), 246–251.
– Botsch, M., Kobbelt, L., Pauly, M., Alliez, P., and Lévy, B. (2010). Polygon Mesh Processing. AK Peters / CRC Press.
– Breiman, L. (2001) 'Random Forests', Machine Learning, 45(1), pp. 5–32.
– Breiman, L. (2001) 'Random Forests', Machine Learning, 45(1), pp. 5–32. doi:10.1023/A:1010933404324.
– buildingSMART International (2022). IFC 4.3 (IFC4x3) Documentation / Specification. Available online (accessed 2026-01-27). https://standards.buildingsmart.org/IFC/RELEASE/IFC4_3/ (accessed 2026-01-27).
– Chen, Y. and Medioni, G. (1992) 'Object modelling by registration of multiple range images', Image and Vision Computing, 10(3), pp. 145–155.
– CloudCompare Team (s.d.) CloudCompare [Computer software]. https://www.cloudcompare.org
– Cohen, J. (1960) 'A coefficient of agreement for nominal scales', Educational and Psychological Measurement, 20(1), pp. 37–46.
– de Boor, C. (2001). A Practical Guide to Splines (Revised ed.). Springer, New York.
– Demantké, J., Mallet, C., David, N. and Vallet, B. (2011) 'Dimensionality based scale selection in 3D LiDAR point clouds', International Archives of the

Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXVIII-5/W12, pp. 97–102. doi:10.5194/isprsarchives-XXXVIII-5-W12-97-2011.

– Ester, M., Kriegel, H.-P., Sander, J. and Xu, X. (1996) 'A density-based algorithm for discovering clusters in large spatial databases with noise', Proceedings of KDD'96, pp. 226–231.

– Farin, G. (2002). Curves and Surfaces for CAGD: A Practical Guide (5th ed.). Morgan Kaufmann, San Francisco.

– Fischler, M.A. and Bolles, R.C. (1981) 'Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography', Communications of the ACM, 24(6), pp. 381–395.

– Harris, C.R. et al. (2020) 'Array programming with NumPy', Nature, 585, pp. 357–362. doi:10.1038/s41586-020-2649-2.

– He, H. and Garcia, E.A. (2009) 'Learning from imbalanced data', IEEE Transactions on Knowledge and Data Engineering, 21(9), pp. 1263–1284. doi:10.1109/TKDE.2008.239.

– Hoppe, H. et al. (1992) 'Surface reconstruction from unorganized points', in Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '92). New York: ACM, pp. 71–78.

– Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W. (1992) 'Surface Reconstruction from Unorganized Points', Computer Graphics (SIGGRAPH '92), 26(2), pp. 71–78.

– Hunter, J.D. (2007) 'Matplotlib: A 2D graphics environment', Computing in Science & Engineering, 9(3), pp. 90–95. doi:10.1109/MCSE.2007.55.

– Isenburg, M. (2013) 'LASzip: Lossless compression of LiDAR data', Photogrammetric Engineering & Remote Sensing, 79(2), pp. 209–217.

– Jakob, W., Tarini, M., Panozzo, D., and Sorkine-Hornung, O. (2015). Instant Field-Aligned Meshes. ACM Transactions on Graphics, 34(6), 1–15.

– Jolliffe, I. T. (2002). Principal Component Analysis (2nd ed.). Springer, New York.
– Kazhdan, M., Bolitho, M. and Hoppe, H. (2006) 'Poisson Surface Reconstruction', in Eurographics Symposium on Geometry Processing.
– Kohavi, R. (1995) 'A study of cross-validation and bootstrap for accuracy estimation and model selection', in Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95). San Francisco, CA: Morgan Kaufmann, pp. 1137–1143.
– Kullback, S. and Leibler, R.A. (1951) 'On information and sufficiency', Annals of Mathematical Statistics, 22(1), pp. 79–86. doi:10.1214/aoms/1177729694.
– Lague, D., Brodu, N. and Leroux, J. (2013) 'Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the Rangitikei canyon (N–Z)', ISPRS Journal of Photogrammetry and Remote Sensing, 82, pp. 10–22. doi:10.1016/j.isprsjprs.2013.04.009.
– Lin, J. (1991) 'Divergence measures based on the Shannon entropy', IEEE Transactions on Information Theory, 37(1), pp. 145–151. doi:10.1109/18.61115.
– McKinney, W. (2010) 'Data Structures for Statistical Computing in Python', in Proceedings of the 9th Python in Science Conference. Austin, TX, pp. 56–61. doi:10.25080/Majora-92bf1922-00a.
– Myronenko, A. and Song, X. (2010) 'Point Set Registration: Coherent Point Drift', IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(12), pp. 2262–2275.
– Pauly, M. et al. (2003) 'Multi-scale feature extraction on point-sampled surfaces', Computer Graphics Forum, 22(3), pp. 281–289.
– Pauly, M., Keiser, R. and Gross, M. (2003) 'Multi-Scale Feature Extraction on Point-Sampled Surfaces', Computer Graphics Forum, 22(3), pp. 281–289.
– PDAL Contributors (s.d.) PDAL—Point Data Abstraction Library [Computer software]. https://pdal.io

– Pedregosa, F. et al. (2011) 'Scikit-learn: Machine Learning in Python', Journal of Machine Learning Research, 12, pp. 2825–2830.
– Pedregosa, F., Varoquaux, G., Gramfort, A. et al. (2011) 'Scikit-learn: Machine Learning in Python', Journal of Machine Learning Research, 12, pp. 2825–2830.
– Piegl, L., and Tiller, W. (1997). The NURBS Book (2nd ed.). Springer, Berlin Heidelberg.
– Powers, D.M.W. (2011) 'Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation', Journal of Machine Learning Technologies, 2(1), pp. 37–63.
– Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A. and Lawrence, N.D. (eds) (2008) Dataset Shift in Machine Learning. Cambridge, MA: MIT Press.
– Riveiro, B., DeJong, M.J. and Conde, B. (2016) 'Automated processing of large point clouds for structural health monitoring of masonry arch bridges', Automation in Construction, 72(Part 3), pp. 258–268. doi:10.1016/j.autcon.2016.02.009.
– Roberts, D.R. et al. (2017) 'Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure', Ecography, 40(8), pp. 913–929. doi:10.1111/ecog.02881.
– Rusinkiewicz, S. and Levoy, M. (2001) 'Efficient Variants of the ICP Algorithm', in Proceedings of 3DIM, pp. 145–152.
– Rusu, R.B., Blodow, N. and Beetz, M. (2009) 'Fast Point Feature Histograms (FPFH) for 3D registration', in Proc. IEEE ICRA, pp. 3212–3217.
– Settles, B. (2012) Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 6(1), pp. 1–114. doi:10.2200/S00429ED1V01Y201207AIM018.
– Shannon, C.E. (1948) 'A mathematical theory of communication', Bell System Technical Journal, 27, pp. 379–423 and 623–656. doi:10.1002/j.1538-7305.1948.tb01338.x.

– Sokolova, M. and Lapalme, G. (2009) ‘A systematic analysis of performance measures for classification tasks’, Information Processing & Management, 45(4), pp. 427–437. doi:10.1016/j.ipm.2009.03.002.
– Strobl, C. et al. (2007) ‘Bias in random forest variable importance measures: illustrations, sources and a solution’, BMC Bioinformatics, 8, 25. doi:10.1186/1471-2105-8-25.
– Sullivan, C.B. and Kaszynski, A. (2019) ‘PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK)’, Journal of Open Source Software, 4(37), 1450.
– Virtanen, P., Gommers, R., Oliphant, T.E. et al. (2020) ‘SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python’, Nature Methods, 17(3), pp. 261–272.
– Weinmann, M., Jutzi, B., Mallet, C. and Klein, R. (2015) ‘Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers’, ISPRS Journal of Photogrammetry and Remote Sensing, 105, pp. 286–304. doi:10.1016/j.isprsjprs.2015.01.016.
– Xia, T., Yang, J. and Chen, L. (2022) ‘Automated semantic segmentation of bridge point cloud based on local descriptor and machine learning’, Automation in Construction, 133, 103992. doi:10.1016/j.autcon.2021.103992.
– Zhou, Q.-Y., Park, J. and Koltun, V. (2018) ‘Open3D: A modern library for 3D data processing’, arXiv preprint arXiv:1801.09847.
– Zhou, Q.-Y., Park, J. and Koltun, V. (2018) ‘Open3D: A Modern Library for 3D Data Processing’, arXiv:1801.09847.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	–	Artificial Intelligence
BIM	–	Building Information Modelling
BrIM	–	Bridge Information Modelling
InfraBIM	–	Infrastructure Building Information Modelling
FEM	–	Finite Element Method
IFC	–	Industry Foundation Classes
TLS	–	Terrestrial Laser Scanning
UAV	–	Unmanned Aerial Vehicle
MMS	–	Mobile Mapping System
LiDAR	–	Light Detection and Ranging
RF	–	Random Forest
ML	–	Machine Learning
QA/QC	–	Quality Assurance / Quality Control
ICP	–	Iterative Closest Point

RANSAC	–	Random Sample Consensus
kNN	–	k-Nearest Neighbours
KD	–	tree — k-dimensional tree
PCA	–	Principal Component Analysis
HAG	–	Height Above Ground
IoU	–	Intersection over Union
OA	–	Overall Accuracy
F1	–	F1-score
RMSE	–	Root Mean Square Error
C2C	–	Cloud-to-Cloud distance
C2M	–	Cloud-to-Mesh distance
NURBS	–	Non-Uniform Rational B-Splines
CAD	–	Computer-Aided Design
GIS	–	Geographic Information System
CSV	–	Comma-Separated Values
LAS/LAZ	–	LASer file format / compressed LAS
PLY	–	Polygon File Format

CURRICULUM

First Name: **ALFREDO**

Surname: **RESTUCCIA GAROFALO**

Place and date of birth: **VIBO VALENTIA, 13/09/1976**

Nationality: **Italy**

CONTACTS

a.restucciagarofa@phd.poliba.it

Via Pistoia snc, 74023 Marina di Ginosa (TA), Italy

Polytechnic University of Bari - DICATECh Department

EDUCATION AND TRAINING

- Since October 2022 he has been a **PhD student** (XXXVIII cycle) at the Polytechnic University of Bari (DICATECh Department). Thesis: "Machine learning techniques for the creation of BrIM/FEM models applied to bridges". Tutor: Prof. Ing. Domenica Costantino.
- In 2025, he was awarded a research scholarship (Project MULTI TWIN) at the University "G. d'Annunzio" of Chieti-Pescara (InGeo Department), starting on 04/07/2025 (4 months). Topic: "From point cloud to BIM model: 3D restitution of a strategic steel structure".
- In 2024, I completed a visiting period at the Polytechnic University of Timișoara (Romania) in collaboration with Prof. Sorin Herban; starting in September 2024, the collaboration continued remotely on a permanent basis.
- In 2007, he obtained a **Master's degree in Building Engineering** at the University of Rome Tor Vergata with final grade 110/110, discussing the thesis "La Costruzione della ASR l'Autostrada del SUD" (Advisor: Prof. Ing. Tullia Iori).

SCIENTIFIC PUBLICATIONS

1. Alfio, V. S., Costantino, D., Pepe, M., Herban, S., & Restuccia Garofalo, A. (2024). **Quad-mesh modelling for finite element method applications in heritage structures.** Acta IMEKO, 13(2), 1-8.

2. Pepe, M., Restuccia Garofalo, A., Costantino, D., Tana, F. F., Palumbo, D., Alfio, V. S., & Spacone, E. (2024). **From Point Cloud to BIM: A New Method Based on Efficient Point Cloud Simplification by Geometric Feature Analysis and Building Parametric Objects in Rhinoceros/Grasshopper Software.** Remote Sensing, 16(9), 1630.
3. Costantino, D., Pepe, M., & Restuccia Garofalo, A. (2023). **Scan-to-HBIM for conservation and preservation of cultural heritage building: the case study of San Nicola in Montedoro church (Italy).** Applied Geomatics, 15(3), 607-621.
4. Alfio, V. S., Costantino, D., Pepe, M., & Restuccia Garofalo, A. (2022). **A Geomatics Approach in Scan to FEM Process Applied to Cultural Heritage Structure: The Case Study of the "Colossus of Barletta".** Remote Sensing, 14(3), 664.
5. Pepe, M., Costantino, D., Alfio, V. S., Restuccia Garofalo, A., & Papalino, N. M. (2021). **Scan to BIM for the digital management and representation in 3D GIS environment of cultural heritage site.** Journal of Cultural Heritage, 50, 115-125.
6. Pepe, M., Costantino, D., & Restuccia Garofalo, A. (2020). **An efficient pipeline to obtain 3D model for HBIM and structural analysis purposes from 3D point clouds.** Applied Sciences, 10(4), 1235.
7. Pepe, M., Costantino, D., Alfio, V. S., Angelini, M. G., & Restuccia Garofalo, A. (2020). **A CityGML multiscale approach for the conservation and management of cultural heritage: The case study of the old town of Taranto (Italy).** ISPRS International Journal of Geo-Information, 9(7), 449.

CONFERENCE PAPER

1. Alfio, V. S., Costantino, D., Fallah, M., Restuccia Garofalo, A., Pepe, M., & Piumatti, P. (2025). **Integrated Workflow for 3D modelling of Historic Architecture: A Multi-Sensor Approach.** 2025 IMEKO TC26 International Conference on Metrology for Archaeology and Cultural Heritage, Bergamo, Italy, October 15-17, 2025.
2. Alfio, V. S., Costantino, D., Dewedar, A. K. H., Leserri, M., Montanaro, C., Palumbo, D., Pepe, M., & Restuccia Garofalo, A. (2025). **3D modelling for the historical reconstruction of an archaeological site: the temple of Iuvanum.** 2025 IMEKO TC26 International Conference on Metrology for Archaeology and Cultural Heritage, Bergamo, Italy, October 15-17, 2025.
3. Alfio, V. S., Costantino, D., Restuccia Garofalo, A., Herban, S., & Pepe, M. (2023). **From TLS data into H-FEM model based on the quad-mesh: the case study of romanian church.** In Proc. of the 2023 IMEKO TC-4 International Conference on Metrology for Archaeology and Cultural Heritage (MetroArchaeo), pp. 356-360.
4. Costantino, D., Pepe, M., Carrieri, M., & Restuccia Garofalo, A. (2019, October). **A procedure to obtain a 3D model in BIM and structural analysis software.** In International Workshop on R3 in Geomatics: Research, Results and Review (pp. 157-169). Springer, Cham.

5. Costantino, D., Carrieri, M., Restuccia Garofalo, A., Angelini, M. G., Baiocchi, V., & Bogdan, A. M. (2019). **Integrated survey for tensional analysis of the vault of the church of San Nicola in Montedoro**. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences.
6. Pepe, M., Costantino, D., Crocetto, N., & Restuccia Garofalo, A. (2019). **3D modeling of roman bridge by the integration of terrestrial and UAV photogrammetric survey for structural analysis purpose**. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci., 42(2), W17.

Abstract
Reliable digital representations of existing bridges are essential for maintenance, risk assessment and decision-making. This thesis presents a reproducible pipeline that transforms high-density point clouds acquired by terrestrial laser scanning (TLS), UAV photogrammetry and, when available, mobile mapping (MMS) into structured semantic information and, ultimately, BrIM (Bridge Information Modelling) and FEM-ready models.
The workflow integrates acquisition and pre-processing (registration, georeferencing, filtering and quality control), scalable spatial data structures, multi-scale extraction of geometric descriptors, supervised classification using a Random Forest trained on local neighbourhood features, and downstream modelling/export. The classified cloud is converted into interoperable outputs (IFC/BrIM and FEM), supported by traceability rules and quality indicators. The approach is validated on bridge case studies by comparing alternative modelling tracks and assessing both classification accuracy and geometric-metric consistency.

On the cover: From point cloud to model: BrIM/FEM