POLITECNICO DI TORINO

Master's Degree in Building Engineering



Politecnico
di Torino

Master's Degree Thesis

From the 3D metric survey to a living digital shadow through IoT integration of real-time data. The case study of the Cavour Canal water bridge

Supervisors
Prof. Francesca MATRONE
Prof. Luca BARBIERATO
Prof. Marco ZERBINATTI

Candidate
Saketh CHINTHAKINDI

22nd September 2026

Summary

The preservation of 19th-century architectural and civil engineering works, such as masonry bridges, is essential to ensure structural safety and the conservation of their historical value. Ageing and prolonged exposure to environmental actions progressively threaten these infrastructures, making planned maintenance a sustainable alternative to invasive restoration. In this context, digital replicas integrated with Internet of Things (IoT) technologies enable a proactive, data-driven approach to asset management. This paper presents a methodology for monitoring and preserving hydraulic historical structures, specifically the Cavour Canal bridge, a historic masonry water bridge within a wider network in northern Italy. A 3D metric integrated survey generated point clouds and orthophotos as the basis for the scan-to-HBIM workflow and for subsequent Historic Building Information Modelling (HBIM), supporting structural analyses and maintenance activities. The HBIM is enhanced through the integration of real-time data from continuous monitoring sensors. The resulting digital shadow supports an interactive, web-based management platform that can detect critical conditions and issue real-time alerts, improving the resilience and long-term management of historical infrastructures.

Acknowledgements

To my parents and friends. special thanks to prasrita for supporting me

Table of Contents

List of Tables VI

List of Figures VII

1 Introduction 1
- 1.1 Objective . . . 1
- 1.2 The case study . . . 2
  - 1.2.1 Current condition of the bridge . . . 2

2 Data acquisition & Data processing 4
- 2.1 Data acquisition . . . 5
  - 2.1.1 Site planning . . . 5
  - 2.1.2 Global Navigation Satellite Systems (GNSS) & Global Positioning System (GPS) . . . 6
  - 2.1.3 Total Station . . . 7
  - 2.1.4 Unmanned Aerial Vechiles (UAV) . . . 8
  - 2.1.5 Terrestrial Laser Scanner (TLS) . . . 10
  - 2.1.6 Mobile Mapping System (MMS) . . . 11
- 2.2 Data Processing . . . 13
  - 2.2.1 Manual Registration . . . 13
  - 2.2.2 Georeferencing and indexing . . . 20
  - 2.2.3 Noise cleaning and filtering . . . 25
- 2.3 Results . . . 29
  - 2.3.1 Terrestrial Laser Scanner-Leica Cyclone Register 360 Plus . . . 29
  - 2.3.2 Unmanned Aerial Vehicle-Agisoft Metashape professional . . . 31
  - 2.3.3 Post processing-Cloud Compare . . . 36

3 Scan to BIM 44
- 3.1 Transition from BIM to Historical BIM (HBIM) . . . 45
  - 3.1.1 Building Information Modelling (BIM) . . . 45
- 3.2 Semantic breakdown of the bridge . . . 47
  - 3.2.1 Structural understanding and historical features . . . 47
  - 3.2.2 Semantic segmentation driven by Structural Health Monitoring . . . 48
  - 3.2.3 Division strategy and sensor driven modelling logic . . . 48
- 3.3 From .e57 to Autodesk ReCap Pro . . . 49
- 3.4 Scan to BIM execution using Autodesk Revit . . . 52
  - 3.4.1 Setting the environment . . . 52

3.4.2 Executing the divisions . . . 58
3.4.3 Modeling non geometric elements: Terrain . . . 67

4 Revit and IoT Integration 69
4.1 IoT sensors . . . 69
4.1.1 IoT sensor placement on bridge . . . 69
4.1.2 Sensor modelling in Autodesk Revit . . . 72
4.2 Autodesk Revit-IoT architecture . . . 77
4.2.1 Device layer . . . 77
4.2.2 Concentrator layer . . . 78
4.2.3 Middleware layer . . . 79
4.2.4 Visualisation layer in Autodesk Revit & IoT integration . . . 81

5 Unity and IoT integration 83
5.1 Digital Twin vs Digital Shadow . . . 83
5.1.1 Digital Twin in AEC industry . . . 84
5.2 Introduction to unity game engine . . . 85
5.2.1 Unity game engine and autodesk revit interoporability . . . 85
5.2.2 Unity Reflect for BIM integration . . . 86
5.2.3 Custom autodesk revit plugin for metadata export . . . 87
5.2.4 Integration with in unity game engine . . . 89
5.3 Unity game engine-IoT architecture . . . 91
5.3.1 Device layer . . . 92
5.3.2 Middleware layer . . . 92
5.3.3 Application layer . . . 94
5.3.4 Visualisation layer in Unity game engine (WebGL Build) and IoT integration 97
5.3.5 Build deployment and dashboard architecture . . . 97
5.3.6 Engine to browser communication architecture . . . 100
5.3.7 System visualisation & interactive workflows . . . 101

6 Experimental Results 104
6.1 Visualisation in Autodesk Revit using Dynamo . . . 104
6.2 Web platform visualisation using Unity game engine . . . 107
6.2.1 Home page . . . 108
6.2.2 WebGIS page . . . 109
6.2.3 BIMmodel page . . . 110
6.2.4 Monitoring page . . . 111
6.2.5 Settings page . . . 112

7 Conclusions 113

8 Future Devleopments 115

Bibliography 117

List of Tables

2.1 Technical specifications comparison between DJI Mavic 3 Pro and Autel EVO II Pro V3. 9
2.2 Technical specifications of the Leica RTC360 Terrestrial Laser Scanner (TLS). 11
2.3 Technical specifications of the Stonex X70GO SLAM Laser Scanner. 13

4.1 Real time IoT sensor telemetry data from the Cavour Canal water bridge lato monte CO_8204_1. 72
4.2 Real time IoT sensor telemetry data from device lato valle CO_8204_2. 72

5.1 Autodesk Revit element metadata mapping structure for BIM-to-web platform integration. 88

List of Figures

1.1 Aerial photo of Ponte Canale Cavour . . . 2
1.2 Issues of Ponte Canale Cavour . . . 2
1.3 Water infiltration and efforescence . . . 3
1.4 Vegetation growth and hydraulic scour risk . . . 3
1.5 Team DIRECT Logo . . . 3

2.1 Data acquisiton on Canale Cavour . . . 5
2.2 Site planning with Team DIRECT . . . 5
2.3 GNSS working principle image [4] . . . 6
2.4 GNSS/GPS system on side deck of the bridge. . . . 7
2.5 Total station principle [7] . . . 7
2.6 left: Total station on ground picking checker points from face, right: total station on side deck of the bridge. . . . 8
2.7 UAV photogrametrical working principle [10] . . . 8
2.8 Unmanned Aerial Vechile (UAV) Autel evo II pro 3. . . . 9
2.9 Total station principle [11] . . . 10
2.10 Terrestrial Laser Scanner under the bridge arch. . . . 10
2.11 Mobile Mapping System working principle . . . 11
2.12 Mobile Mapping System on deck of the bridge . . . 12
2.13 Import and visual inspection . . . 14
2.14 Coarse alignment top view . . . 15
2.15 Coarse alignment from 3d view . . . 15
2.16 Bundle optimisation error . . . 16
2.17 Manual virtual marking . . . 16
2.18 Setting for align photos . . . 17
2.19 Tiepoints info after aligning photos . . . 18
2.20 Imported reference markers from .txt file . . . 19
2.21 Reference coordinae system settings used in Metashape . . . 19
2.22 Total Station reference coordinates imported . . . 20
2.23 Target matching with help of total station photos . . . 21
2.24 Error assessment of group 1 **??** . . . 21
2.25 Error assessment of group 1 **??** . . . 22
2.26 Error assessment of group 2 **??** . . . 22
2.27 Manual marker pinning . . . 23
2.28 check markers in reference settings **??** . . . 23
2.29 Error of check markers in reference panel **??** . . . 24
2.30 Point cloud generating settings . . . 25

2.31 Point cloud selection tool . . . 26
2.32 Point cloud after cleaning the noise . . . 26
2.33 Point cloud segmentation tool . . . 27
2.34 Point cloud with noise . . . 27
2.35 Point cloud after cleaning the Noise . . . 28
2.36 Point cloud after cleaning the Noise . . . 29
2.37 Group 2 Overall bundle report and target error . . . 29
2.38 Group 1 Overall bundle report and target error **??** . . . 30
2.39 Group 1 & group 2 link errors **??** . . . 31
2.40 DEM settings in build DEM workflow . . . 32
2.41 (a) DEM of north facade, (b) DEM of south facade . . . 33
2.42 Orthomosaic settings in build orthomosaic workflow . . . 34
2.43 (a) Orthomosaic of north facade, (b) Orthomosaic of south facade . . . 35
2.44 Automated ICP registration assessment . . . 36
2.45 Automated ICP registration assessment . . . 37
2.46 Point pair method for roation and translation . . . 38
2.47 Align info final rms . . . 38
2.48 Subsampling using spatial method settings . . . 39
2.49 Subsampling using spatial method . . . 39
2.50 Subsampling using spatial method . . . 40
2.51 Cleaning of LiDAR point cloud . . . 41
2.52 Cloth Simulation Filter settings . . . 41
2.53 Rasterize Kriging settings . . . 42
2.54 Rasterize Kriging output . . . 43

3.1 Illustration of Scan to BIM workflow . . . 44
3.2 Historical acquaduct bridge parts (Source: Scotland's Oldest Bridge) . . . 47
3.3 Illustration of .e57 file to .rcp file workflow . . . 49
3.4 Noise cleaning . . . 50
3.5 volumetric clipping . . . 51
3.6 Unifying scan data into single scan . . . 51
3.7 Recap volumetric clipping . . . 52
3.8 Autodesk Revit 2024 . . . 52
3.9 New project in metric architecture model template . . . 53
3.10 Metric project unit settings . . . 53
3.11 Point cloud import . . . 54
3.12 extracting base point from cloud compare to Autodesk Revit . . . 55
3.13 Setting up base point coordinates . . . 55
3.14 Elevation reference levels from north elevation of point cloud . . . 56
3.15 Elevation reference levels from north elevation of point cloud . . . 57
3.16 Elevation reference levels from north elevation of point cloud . . . 57
3.17 Green dashed line perfect horizontal line following red line is with slope of the deck from longitudinal section . . . 58
3.18 Temporary cubes are created to setup working plane . . . 59
3.19 Draw tool in cross section following point cloud . . . 59
3.20 Architecture wall 350mm wall type properties . . . 60
3.21 Temporary cubes are created from section view . . . 61

3.22 Right Generic family template . . . 61
3.23 Generic family template . . . 62
3.24 Isolated cutwater element as loadable .rfa . . . 63
3.25 Main Pier Body by architecture wall element . . . 63
3.26 Pier caps modelled as In-place mass . . . 64
3.27 Pier modelling divisions . . . 64
3.28 Spandrel wall slope . . . 65
3.29 Spandrel wall & arch curve following point cloud . . . 66
3.30 Spandrel wall & arch curve following point cloud . . . 66
3.31 Spandrel wall & arch curve following point cloud . . . 67
3.32 Create toposolid using import .csv . . . 68

4.1 IoT sensors installation . . . 70
4.2 IoT sensors installation . . . 70
4.3 IoT sensors installation . . . 71
4.4 IoT sensors installation . . . 71
4.5 Crack meter & its properties . . . 73
4.6 Inclinometer & its properties . . . 74
4.7 Temperature sensor & Data logger & its properties . . . 74
4.8 Datalogger & its properties . . . 75
4.9 Autodesk Revit Family, Sensor Parameter properties . . . 76
4.10 IoT-Revit integration architecture . . . 77
4.11 IoT-BIM integration architecture . . . 81

5.1 Digital Twin vs Digital Shadow . . . 83
5.2 Digital twin in AEC industry (Source: e verse learn) . . . 84
5.3 Illustration of interoperability . . . 85
5.4 Unity Reflect 360 overview (Source: Autodesk) . . . 86
5.5 Autodesk Revit plugin visulisation . . . 87
5.6 Left Autodesk Revit, Custom plugin exicuted & Right destination folder exported both .fbx and .json files . . . 89
5.7 Illustration of workflow using BIMElementManager.cs . . . 89
5.8 IoTUnity game engine architecture . . . 91
5.9 IoT-Unity game engine workflow . . . 97
5.10 Unity game engine, Build profile settings page . . . 98
5.11 Unity game engine, Build profile exported files . . . 98
5.12 Build WebAssembly Binary & Javascript loader files . . . 99
5.13 Distinct user interface . . . 100
5.14 Unity game engine, C# scripts used in the project . . . 100

6.1 Dynamo script to retrieve data from a web request and set the data as parameters. 104
6.2 Automatic change of element colour based on the temperature parameter in the Autodesk Revit family. . . . 105
6.3 Dynamo script showing the connection of String, Web request, Python script and List flatten nodes. . . . 106
6.4 Dynamo script showing the connection of List get item at index and Set parameter by name nodes. . . . 106

6.5 Dynamo script showing the connection of the Python script and its inputs to change the colours of the selected element in the model. 107
6.6 digital shadow web platform 108
6.7 digital shadow web platform, Home page 108
6.8 digital shadow web platform, WebGIS page 109
6.9 digital shadow web platform, BIMmodel page 110
6.10 digital shadow web platform, Monitoring page 111
6.11 digital shadow web platform, Settings page 112

Chapter 1

Introduction

The safeguarding and enhancement of 19th century architectural and civil engineering works, such as masonry bridges, are crucial to preserve not only the engineering operas of their, time but also the historical significance embedded in their design and construction. As they age, exposure to environmental factors, such as weathering and natural wear, puts their structural integrity at risk, making preservation efforts crucial.

As emerged, planned maintenance plays a critical role in ensuring the long term preservation of such infrastructures, offering a proactive alternative to costly and disruptive restorations. This approach, facilitated by the realization of a digital replica integrated with Internet of Things (IoT) data [1, 2], not only ensures that the bridges remain functional and safe, but also protects their architectural stability, protecting them for future generations while minimizing the impact on the surrounding environment.

This contribution thus describes the methodology set up to preserve an example of this historical heritage. In particular, the preliminary results of the initial data acquisition campaign on the case study, the *Ponte Canale Cavour* (Cavour water bridge), which allowed for obtaining 3D integrated point clouds and orthophotos as the basis for the scan to Historic Building Information Modelling (HBIM) modelling and further analysis, with the definition of a digital shadow. In particular, attention was paid to the integration of real time data into the HBIM model, which came from continuous sensors installed on the bridge for its structural monitoring via IoT devices. The core idea, in fact, is to create an interactive web based platform for managing and maintaining the case study, achieved by using IoT sensors installed on the bridge.

1.1 Objective

The main objective is therefore to develop a web based model capable of detecting and alerting the management authority in the event of critical situations. In the past, similar infrastructures have suffered structural failures due to heavy rainfall and flooding, resulting in the accumulation of logs and debris around the bridge piers, which led to their collapse. The availability of a digital replica that can function as a "traffic light" system, indicating in real time significant variations or critical values detected by the sensors, is of vital importance for the effective management of this type of asset. Giving the user a visual platform to understand the data coming from the sensors through a web platform. the end user can fully control the alerts and information that he wants to look into.

1.2 The case study

1.2.1 Current condition of the bridge

The Cavour Canal is a hydraulic engineering project built between 1863 and 1866, primarily to support rice cultivation in northern Italy. When it was inaugurated in 1869, the canal irrigated a surface area of approximately 200,000 hectares, with a flow rate of 110 m^3/sec for a total length of 82 km [3]. Among the infrastructure of this massive opera, water bridges enable the canal's water to flow over rivers. An example is the Ponte Canale Cavour (Cavour bridge canal). This bridge canal, necessary to cross the Dora Baltea river, is the first (1864) and largest in Italy (Figure 1.1).



Figure 1.1: *Aerial photo of Ponte Canale Cavour*

ISSUES

Despite a general retrofit intervention carried out in 1998, some degradation phenomena undermining some piers is underway, in addition to water infiltrations that accelerate deterioration of the structure.





Figure 1.2: *Issues of Ponte Canale Cavour*

Water Infiltration and Efflorescence: White mineralization (efflorescence) on the intrados (underside) indicates water leaking through the arch barrel from the deck drainage system, leaching lime and salts from the masonry binder. (Figure:1.3)

Mortar Loss and Joint Degradation: Continuous moisture exposure has eroded binder joints between the bricks, causing significant mortar wash out and open joints. (Figure:1.4)

Vegetation Growth and Hydraulic Scour Risk: Plant root systems are expanding within the open joints, accelerating mechanical weathering. Concurrently, the riverbed positioning shows exposed footings vulnerable to hydrodynamic scour and pier undermining. (Figure:1.4)



Figure 1.3: *Water infiltration and efforescence*



Figure 1.4: *Vegetation growth and hydraulic scour risk*

During my time with Team DIRECT (Politecnico di Torino), I had the opportunity to contribute directly to the 3D metric survey of the historic Ponte Canale Cavour. As part of the field team, I helped execute the integrated surveying strategy, which combined terrestrial 3D laser scanning with aerial and ground photogrammetry. the data acquisition parts were divided in two seasons one in june 2024 and march 2025. My role involved setting up survey target networks, operating scanning equipment across challenging positions on the structure, and acquiring high resolution spatial data. Experiencing the full pipeline from on site sensor placement to managing raw point cloud alignment gave me practical insight into how non destructive geomatic techniques support structural health monitoring and digital twin development for historical infrastructure.

DIRECT
DISASTER RECOVERY TEAM

Figure 1.5: *Team DIRECT Logo*

Chapter 2

Data acquisition & Data processing

The Scan to BIM process starts with the following steps:

a) Site planning

b) Data Acquisition

- Total Station and GNSS/GPS
- Unmanned Aerial Vechiles photogrammetry
- Terrestrail Laser Scanner
- Mobile Mapping System

c) Data Processing

- Registration
- Georeferencing and indexing
- Noise cleaning and filtering

The documentation of the historic bridge required the use of multiple survey techniques to ensure complete geometric coverage of the structure. Due to the complexity of the bridge geometry and accessibility constraints, a combination of Unmanned Aerial Vehicle photogrammetrical survey, Mobile Mapping System (MMS) scanning, and Terrestrial Laser Scanning (TLS) was employed. Each technique was selected based on its suitability for capturing specific parts of the bridge and generating accurate three dimensional data for subsequent HBIM development.



Figure 2.1: *Data acquisiton on Canale Cavour*

2.1 Data acquisition

2.1.1 Site planning

Prior to data acquisition, a detailed site analysis was conducted along the canale cavour corriodr to define survey extents, required level of detail, and accessibility constraints. Given the linear nature of the canal and the presence of heritage masonry structures affected by surface deterioration, a hybrid survey strategy was adopted.

A topographical control network was established using a total station, providing stable reference points for terrestail laser scanning, photogrammetric, and Mobile mapping data sets. Differntial leveling was carried out to define reliable height benchmarks, ensuring vertical consistency across the surveyed sections and supporting subsequent geometric and hydraulic interpreations (Figure 2.2)



Figure 2.2: *Site planning with Team DIRECT*

2.1.2 Global Navigation Satellite Systems (GNSS) & Global Positioning System (GPS)

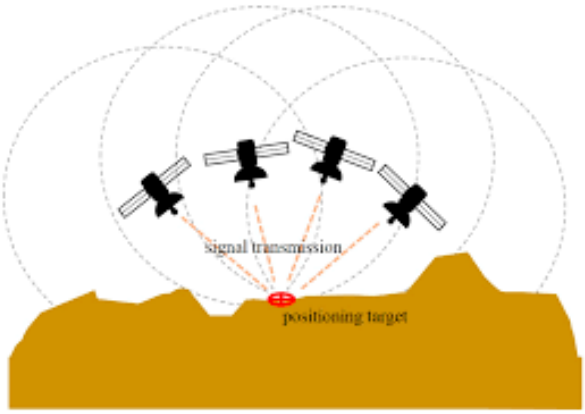


Figure 2.3: *GNSS working principle image [4]*

Global Navigation Satellite Systems (GNSS) encompass global satellite networks including the United States GPS, Russia's GLONASS, the European Union's Galileo, and China's BeiDou that provide absolute 3D positioning within global geodetic reference frames such as WGS 84 [5]. Unlike terrestrial instruments, a GNSS receiver calculates its position on Earth using radio frequency trilateration. The receiver tracks signals broadcast from multiple satellites at known orbital positions and computes ranges based on the precise time delay of the incoming signals. Mathematically, a minimum of four satellites is required to simultaneously solve for three spatial coordinates (X, Y, Z) and correct for internal receiver clock bias [6].

Although standalone GNSS yields meter level positioning due to atmospheric and orbital errors, high precision research uses differential carrier phase techniques like Real Time Kinematic (RTK) or Post Processed Kinematic (PPK) [5]. By applying corrections from a base station, these methods mitigate atmospheric interference to achieve centimeter level accuracy (1 cm2 cm). While GNSS efficiently covers vast areas without requiring ground level line of sight, its performance depends on clear sky visibility and suffers degradation under dense canopy, around buildings (multipath interference), or underground [6].

A GNSS/GPS receiver combines satellite based positioning with real time kinematic (RTK) technology to deliver high precision 3D geodetic coordinates (X, Y, Z) directly in global or national reference frames. It provides continuous site wide spatial positioning without requiring line of sight between survey stations, eliminating the need for manual coordinate conversions or long baseline traverses across open ground. (Figure 2.4)



Figure 2.4: *GNSS/GPS system on side deck of the bridge.*

For this project, To ensure high dimensional fidelity and prevent model distortion, a hybrid georeferencing methodology was implemented. Primary site control was established using a GNSS/GPS receiver positioned on the unobstructed bridge deck to capture absolute global coordinates (X, Y, Z). A Total Station was then referenced to these control points to accurately measure target markers distributed across the structure, including the obscured arch intrados.The measured coordinates were subsequently transformed into a Local Coordinate System (LCS) to optimize computational efficiency during processing.

2.1.3 Total Station

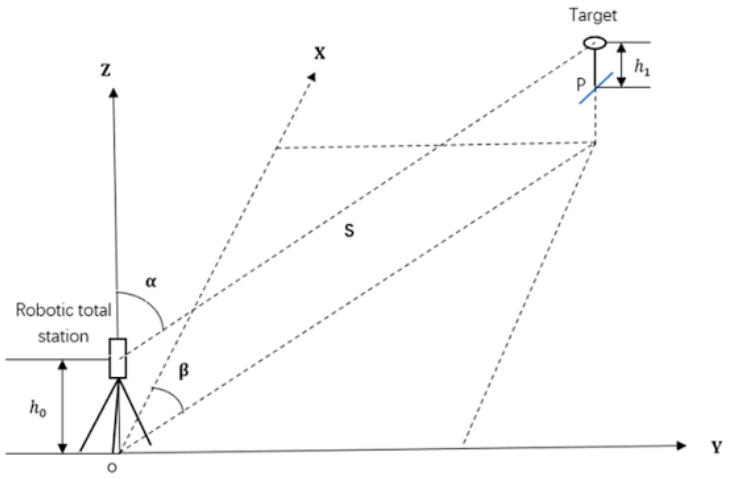


Figure 2.5: *Total station principle [7]*

A Total Station is a terrestrial surveying instrument that integrates an electronic digital theodolite with an Electronic Distance Meter (EDM) to compute positions within a local polar coordinate system [8]. The instrument operates by emitting a modulated optical or infrared light beam toward a target typically a glass retroreflective prism or a solid surface and determining distance through carrier phase shift or time of flight measurements [9]. Internal optical encoders simultaneously record the horizontal and vertical angles relative to the instrument's setup position. By combining these distance and angular parameters, the system uses basic trigonometry to derive spatial offsets $(\Delta X, \Delta Y, \Delta Z)$ relative to a known reference point.

A total station combines high precision prism surveying and 3D laser scanning into a single device. It captures dense 3D point clouds directly tied to exact project coordinates, eliminating the need for a separate laser scanner or manual scan alignment. (Figure 2.6)



Figure 2.6: *left: Total station on ground picking checker points from face, right: total station on side deck of the bridge.*

2.1.4 Unmanned Aerial Vechiles (UAV)

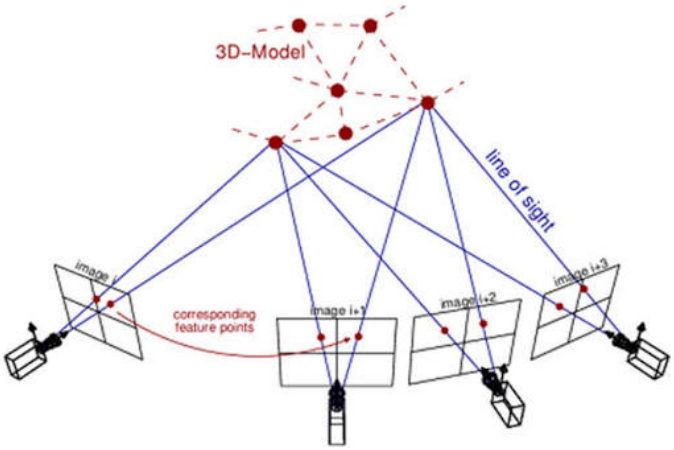


Figure 2.7: *UAV photogrametrical working principle [10]*

Unmanned Aerial Vehicle (UAV) photogrammetry captures overlapping aerial images to create 2D maps and 3D digital models. The aircraft flies a grid path taking photographs with high overlap (70% – 80%). Software detects matching keypoints across photos using Structure from Motion (SfM) algorithms, reconstructing camera positions and generating a 3D point cloud, Digital Elevation Models (DEMs), and orthomosaics.

Unmanned Aerial Vehicles, commonly known as drones, have become an effective tool for data acquisition in surveying and heritage documentation. Unmanned aerial vehicle provide access to elevated and difficult to reach areas while capturing high resolution imagery suitable for photogrammetric processing. (Figure: 2.9)



Figure 2.8: *Unmanned Aerial Vechile (UAV) Autel evo II pro 3.*

For this project, a Unmanned aerial vehicle survey was carried out to document the bridge façades and upper sections, particularly areas that were difficult to capture from ground level. Two Unmanned aerial vehicle instruments, the DJI Mavic 3 Pro and Autel EVO II Pro V3, were used to collect the aerial imagery. For the bridge deck, an approximate flight altitude of 40 m was maintained, with an 80% overlap between consecutive images. The flight was conducted following a horizontal U shaped path to provide consistent coverage of the deck and upper structure.

For the bridge façades, the Unmanned aerial vehicle was maintained at an approximate distance of 20 m from the façade, again with an 80% overlap between consecutive images. Both horizontal and vertical U shaped flight paths were used to capture the façades from different directions and to improve coverage of the structural details. The Unmanned aerial vehicle imagery was then combined with the terrestrial survey data to provide a more complete and detailed photogrammetric representation of the bridge. (Figure: 2.8)

Table 2.1: *Technical specifications comparison between DJI Mavic 3 Pro and Autel EVO II Pro V3.*

Feature / Specification	**DJI Mavic 3 Pro**	**Autel EVO II Pro V3**
Takeoff Weight	958 g	1,191 g
Main Camera Sensor	4/3 inch CMOS (20 MP)	1 inch CMOS (20 MP)
Additional Sensors	1/1.3" CMOS (48 MP) + 1/2" CMOS (12 MP)	Single camera system
Shutter Type	Electronic	Electronic
Aperture Range	$f/2.8$ $f/11$ (Main lens)	$f/2.8$ $f/11$
Max Flight Time	Up to 43 minutes	Up to 40 minutes
Max Wind Resistance	12 m/s	12 m/s
Obstacle Sensing	Omnidirectional (APAS 5.0)	Omnidirectional (12 visual sensors)
GNSS Support	GPS + Galileo + BeiDou	GPS + GLONASS + Galileo + BeiDou
RTK Support	No (Enterprise series required)	Optional (Separate RTK module)

2.1.5 Terrestrial Laser Scanner (TLS)

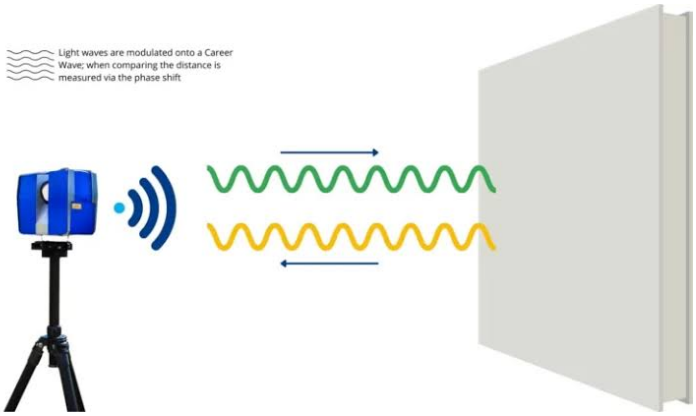


Figure 2.9: *Total station principle [11]*

A Terrestrial Laser Scanner (TLS) is an active ground based sensor that acquires dense 3D point clouds. The device emits laser pulses toward an internal mirror that rotates vertically while the scanner rotates $360°$ horizontally. [12, 13] Distances (d) are determined via electro optical time of flight or waveform digitising, while optical encoders record zenith (α) and azimuth (θ) angles to derive relative 3D coordinates $(\Delta X, \Delta Y, \Delta Z)$:

$$\Delta X = d \cdot \sin(\alpha) \cdot \cos(\theta)$$
$$\Delta Y = d \cdot \sin(\alpha) \cdot \sin(\theta)$$
$$\Delta Z = d \cdot \cos(\alpha)$$

Integrated cameras map true color RGB and intensity data onto the geometric point cloud. Providing millimeter level accuracy, TLS is widely applied in building information modeling (BIM), deformation monitoring, and heritage preservation.



Figure 2.10: *Terrestrial Laser Scanner under the bridge arch.*

Terrestrial Laser Scanning (TLS) is a high precision surveying technique that captures millions of three dimensional points by measuring the distance between the scanner and surrounding objects using laser pulses. TLS is particularly effective for documenting complex geometries and areas requiring high levels of accuracy.

For the historic bridge, Terrestrial Laser scanner Leica RTC 360 scanner was used to survey the underside of the arches and other areas that required detailed geometric documentation.

These sections were difficult to capture adequately using photogrammetry alone due to limited visibility and lighting conditions. The high density point clouds generated by TLS provided accurate geometric information of the arch structures and ensured complete coverage of the bridge geometry.(Figure: 2.10)

Table 2.2: *Technical specifications of the Leica RTC360 Terrestrial Laser Scanner (TLS).*

Feature / Specification	TLS Leica RTC360
Measurement Principle	High speed Time of Flight (Waveform Digitising technology)
Data Acquisition Rate	Up to 2,000,000 points / second
Operating Range	0.5 m to 130 m
3D Point Accuracy	1.9 mm @ 10 m, 2.9 mm @ 20 m, 5.3 mm @ 40 m
Field of View (FoV)	$360°$ (Horizontal) / $300°$ (Vertical)
Laser Safety Class	Class 1 (Invisible wavelength: 1550 nm)
Integrated Camera System	36 MP 3 camera system (432 MPx raw data for HDR spherical image)
Inertial Tracking / Alignment	Visual Inertial System (VIS) using real time IMU tracking for targetless registration
System Weight	5.35 kg (Nominal, excluding battery)
Ingress Protection	IP54 (Dust / splash water resistant)

2.1.6 Mobile Mapping System (MMS)

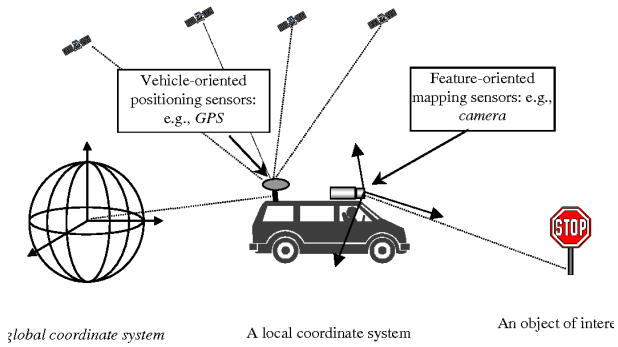


Figure 2.11: *Mobile Mapping System working principle*

Mobile Mapping Systems (MMS) integrate LiDAR sensors, camera systems, Inertial Measurement Units (IMUs), and spatial positioning technology (such as GNSS or SLAM algorithms) onto a moving platform. Unlike static terrestrial scanners that require fixed setups on tripods, an MMS allows operators to continuously capture dense 3D point clouds and imagery while walking or driving, drastically reducing field survey times. (Figure: 2.11)



Figure 2.12: *Mobile Mapping System on deck of the bridge*

A Mobile Mapping System (MMS) integrates laser scanners, cameras, and positioning sensors mounted on a mobile platform to rapidly collect spatial data while moving through the survey area. MMS technology enables efficient acquisition of large datasets and is particularly suitable for linear infrastructures such as roads, tunnels, and bridges.

In this study, Stonex X70GO SLAM laser scanner was employed to capture the geometry of the bridge deck. The system enabled rapid data collection across the entire deck surface while maintaining adequate point density and geometric accuracy. The resulting point cloud provided detailed information regarding the deck geometry and surrounding structural elements, contributing to the overall digital documentation of the bridge.(Figure: 2.12)

Table 2.3: *Technical specifications of the Stonex X70GO SLAM Laser Scanner.*

Category	Feature / Specification	Detail
LiDAR Sensor	Scanning Range	0.1 m 70 m
	Scan Rate	200,000 points/second
	Field of View (FoV)	360° (Horizontal) / 7° to $+52^\circ$ (Vertical)
	Laser Class & Wavelength	Class 1 (Eye safe) / 905 nm
System Performance	Relative Accuracy	Up to 6 mm
	Operating Modes	SLAM Mobile Mode & Static "X Whizz" Mode
	Processing	Real time onboard processing & Post processing via GOpost
Imaging System	Color RGB Camera	12 MP (1 inch CMOS sensor, 4000×3000 px)
	Visual SLAM Camera	12 MP (for real time tracking stability)
Physical & Power	System Weight	925 g (without battery) / 1.45 kg (with battery)
	Dimensions	364.5 mm $\times$ 173.8 mm $\times$ 170 mm
	Operating Time	1.5 hours per battery
	Protection Rating	IP54 (Dust and splash water resistant)
Storage & Modules	Internal Memory	512 GB SSD
	Optional Add ons	RTK GNSS Module for global coordinate georeferencing

2.2 Data Processing

2.2.1 Manual Registration

Objective of this scans are to capture and document the target structure with high geometric fidelity using Terrestrial Laser Scanning (TLS) and UAV photogrammetry. Both survey techniques achieved a precision and absolute error of under 2 cm, ensuring full compliance with accuracy requirements for producing architectural and engineering deliverables at scales of 1:100 and 1:200.

Registration is the process of combining and aligning multiple point clouds that are collected from different positions or scans. Since each scan may have a slightly different position and orientation, they cannot be directly combined without alignment. During registration, common or overlapping features between the point clouds are identified and used to bring them into the same coordinate frame. This helps create a single, continuous and accurate point cloud of the surveyed area or object.

Techniques such as Iterative Closest Point (ICP) can be used to improve the alignment

between overlapping point clouds. In simple terms, registration makes sure that different scans fit together correctly before further processing or analysis.

To construct a high fidelity 3D point cloud of the bridge, a multi sensor data fusion approach was implemented, combining Terrestrial Laser Scanning (TLS) and Unmanned Aerial Vehicle (Unmanned aerial vehicle (UAV)) photogrammetry. Because these two capture techniques operate on vastly different spatial geometries and scales, individual data registration and cross sensor integration were executed through specialized software pipelines.

- Leica Cyclone REGISTER 360
- Agisoft Metashape
- CloudCompare

Terrestrial Laser Scanning workflow (Leica Cyclone REGISTER 360)

Terrestrail Laser Scanner (TLS) dataset, captured via a Leica RTC360. Provides a high density structural details of the bridge intrados and piers. However, due to line of sight obstructions inherent to arch geometries.

The Leica ScanStation P50 TLS was selected for this survey due to its high scanning speed and precision, with a capacity of up to 1 million points per second and a range of more than 1 km. Its high measurement accuracy, 360° horizontal field of view, and ability to capture detailed 3D point clouds made it particularly suitable for documenting the complex geometry of the bridge, including the arches, piers, and intrados.

Following data acquisition, the collected scans were processed using Leica Cylcone 360 Plus. The following workflow steps were then followed:

1. Import & visual inspection: The raw scan setups were imported into Leica Cyclone REGISTER 360, where individual setup locations were analyzed for spatial distribution and preliminary link connections. (Figure: 2.13)

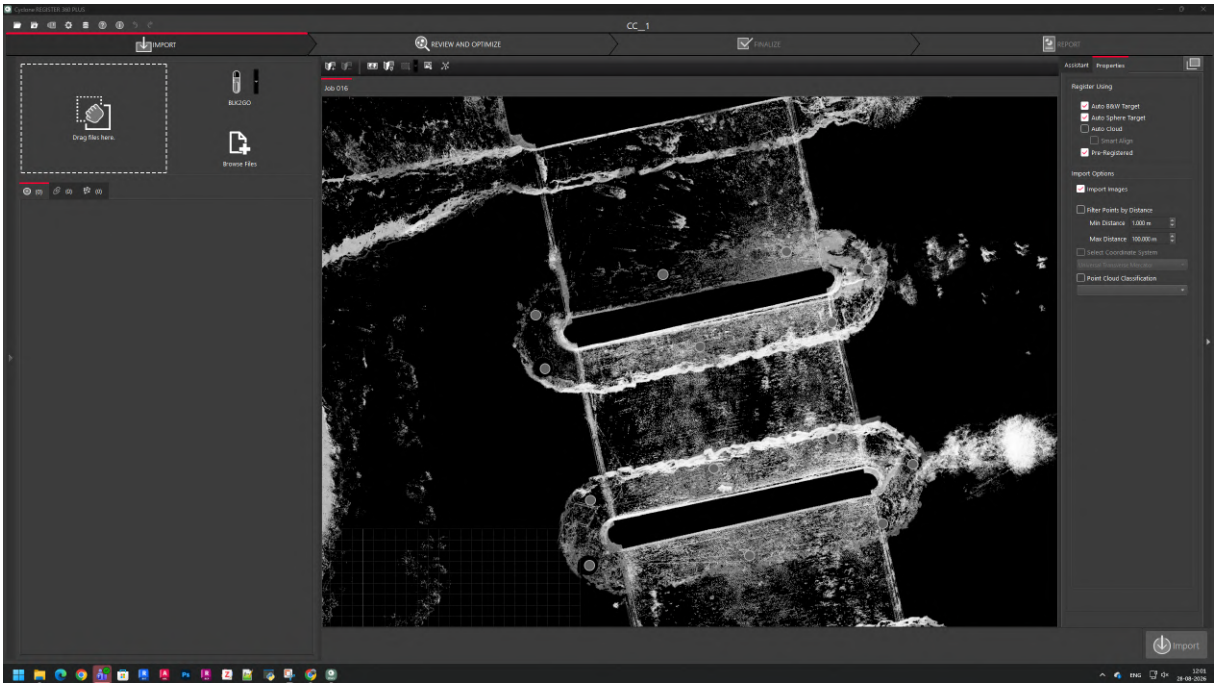


Figure 2.13: *Import and visual inspection*

2. Coarse Alignment (Manual translation & rotation): Prior to automated processing, setups with low visual overlap or complex geometries were manually rotated and translated using the visual alignment tool to place corresponding structural features into approximate proximity. both from top view, side view and 3d view. (Figure: 2.14 & 2.16)

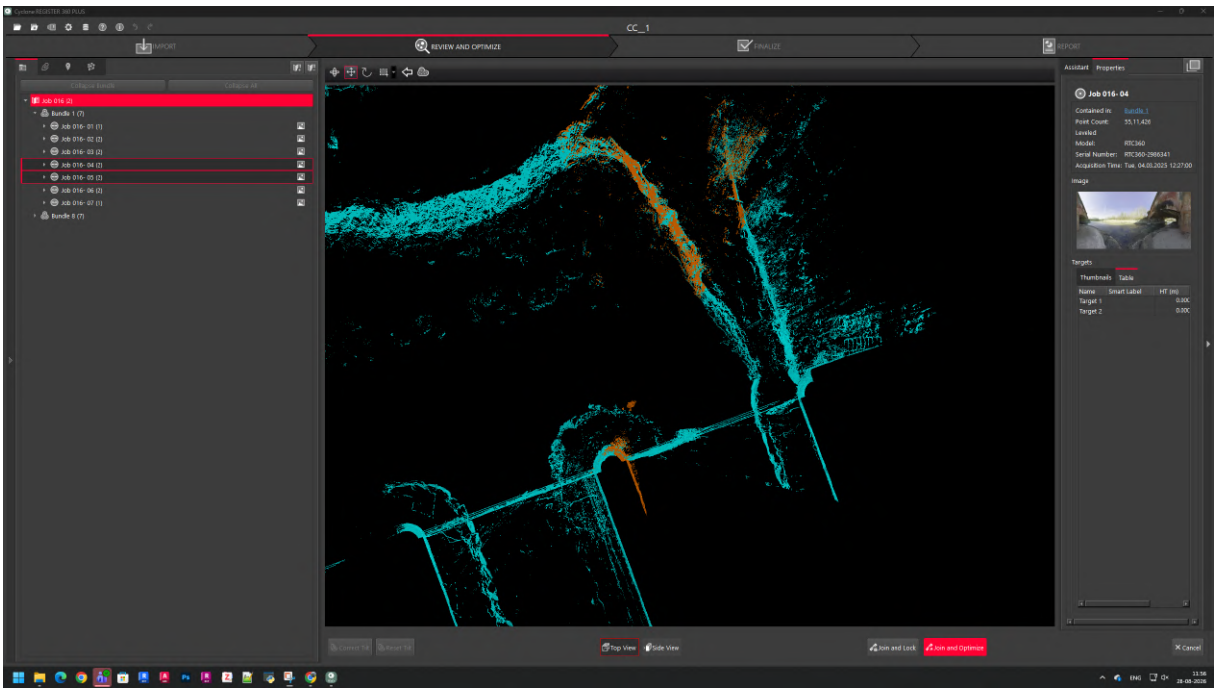


Figure 2.14: *Coarse alignment top view*

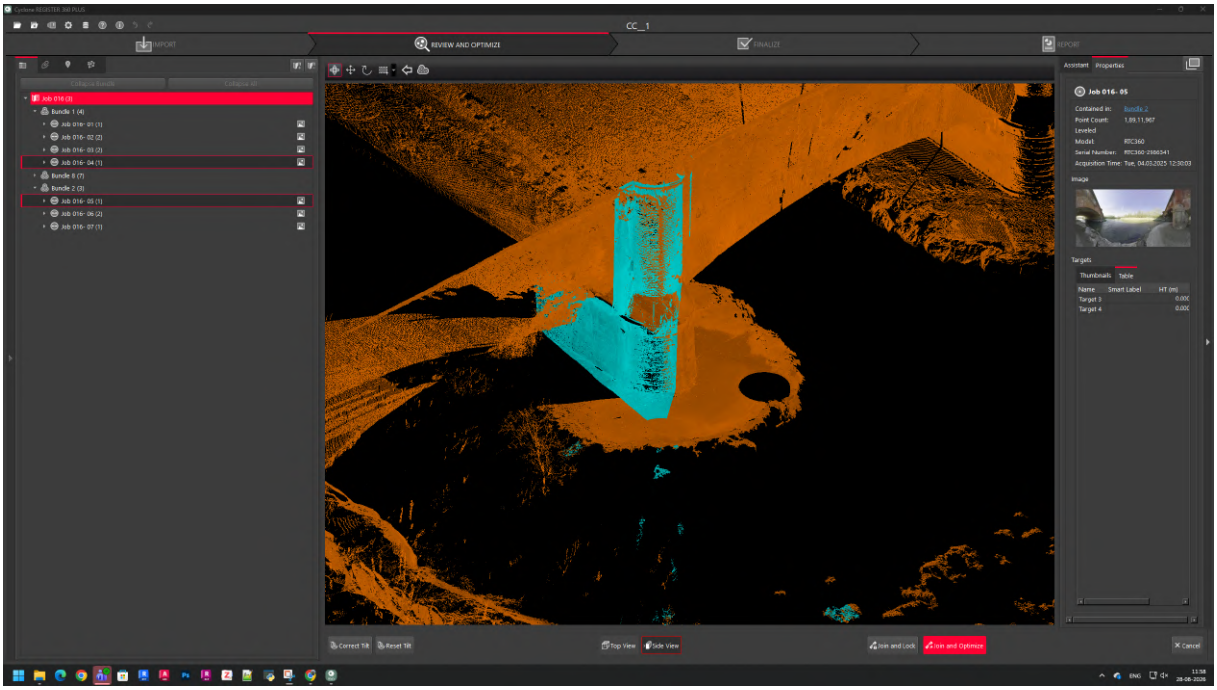


Figure 2.15: *Coarse alignment from 3d view*

3. Smart align & bundle optimization: Once roughly oriented, the SmartAlign algorithm was executed. This cloud to cloud matching process leverages Iterative Closest Point (ICP) variations to calculate optimal transformation matrices, minimizing distance residuals across overlapping point clouds. (Figure: 2.17) During this stage, the alignment was optimized with the objective of achieving a registration error of less than 2 cm, ensuring that the individual TLS scans were accurately combined into a consistent point cloud dataset.

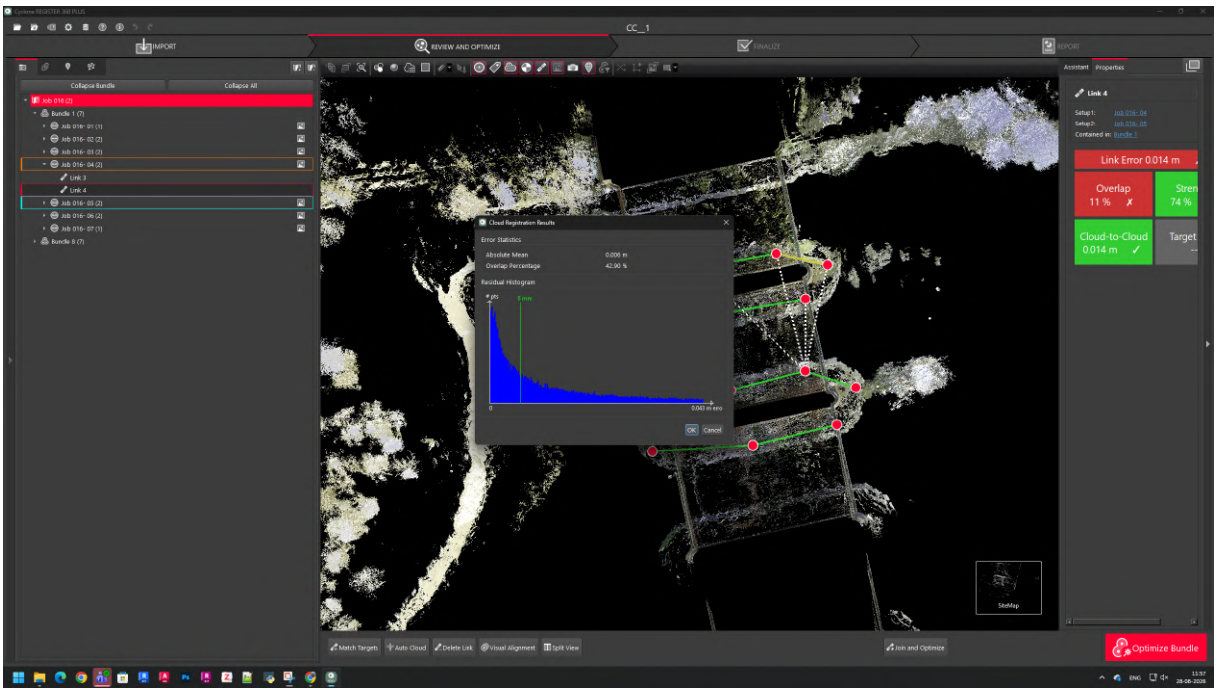


Figure 2.16: *Bundle optimisation error*

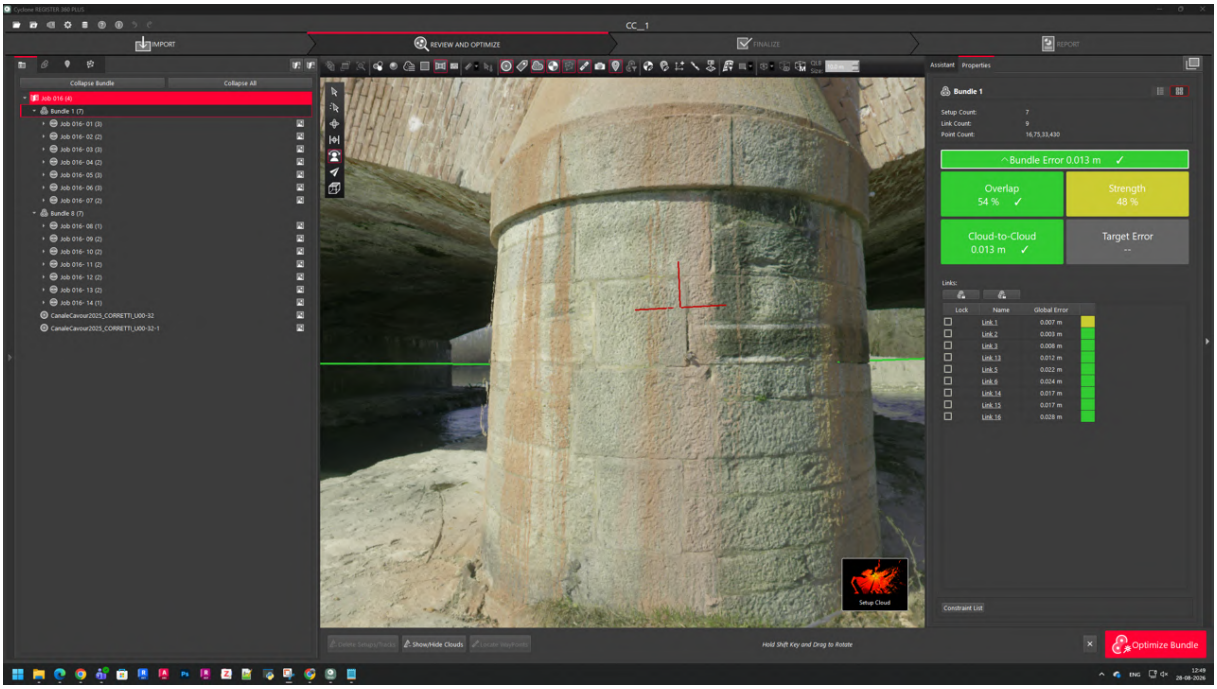


Figure 2.17: *Manual virtual marking*

Before, continue reading the Unmanned aerial vehicle Photogrammetric workflow. jump to the 2.2.2 to follow the workflow of georeferencing and Indexing data in cyclone register 360 plus.

Unmanned Aerial Vechile Photogrammetric Workflow (Agisoft Metashape)

To capture the upper deck, parapets, and hard to reach exterior masonry faces, aerial photogrammetry was integrated.

Following data acquisition, the collected scans were processed using Agisoft Metashape professional. The following workflow steps were then followed:

1. Image Import & Alignment:

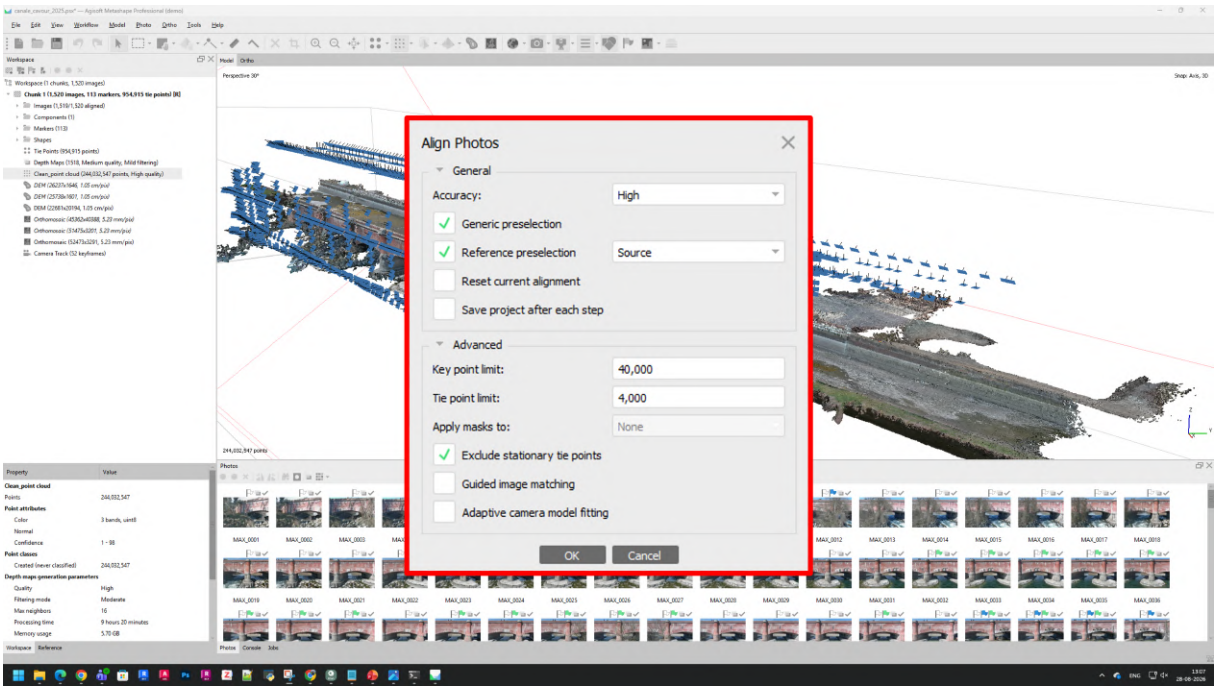


Figure 2.18: *Setting for align photos*

High resolution Unmanned aerial vehicle (UAV) photos were loaded into Agisoft Metashape. The Align Photos tool was executed using highest accuracy settings, applying Structure from Motion (SfM) algorithms to estimate internal camera calibration parameters and calculate preliminary camera poses, generating a sparse point cloud. (Figure: 2.18) During this process, Metashape identified common features between overlapping images and generated 954,915 tie points, out of 1520 photos 1519 photos are aligned, which formed the basis for the initial sparse 3D reconstruction.

Three important indicators from the alignment were considered:

Number of tie points – 954,915: A large number of tie points indicates that many common features were successfully identified between overlapping images. This was important for the bridge because the masonry and structural details provided sufficient visual features for reliable image matching. (Figure: 2.19)

RMS reprojection error – 0.14983 px: This represents the average difference between the observed image points and their calculated positions. The very small value indicates a

good initial image alignment, meaning that the detected features were generally positioned consistently across the images. (Figure: 2.19)

Average tie point multiplicity – 5.58: This means that, on average, each tie point was visible and matched in approximately 5–6 images. This is important because points observed in several photographs provide stronger connections between images and help produce a more stable photogrammetric model. (Figure: 2.19)

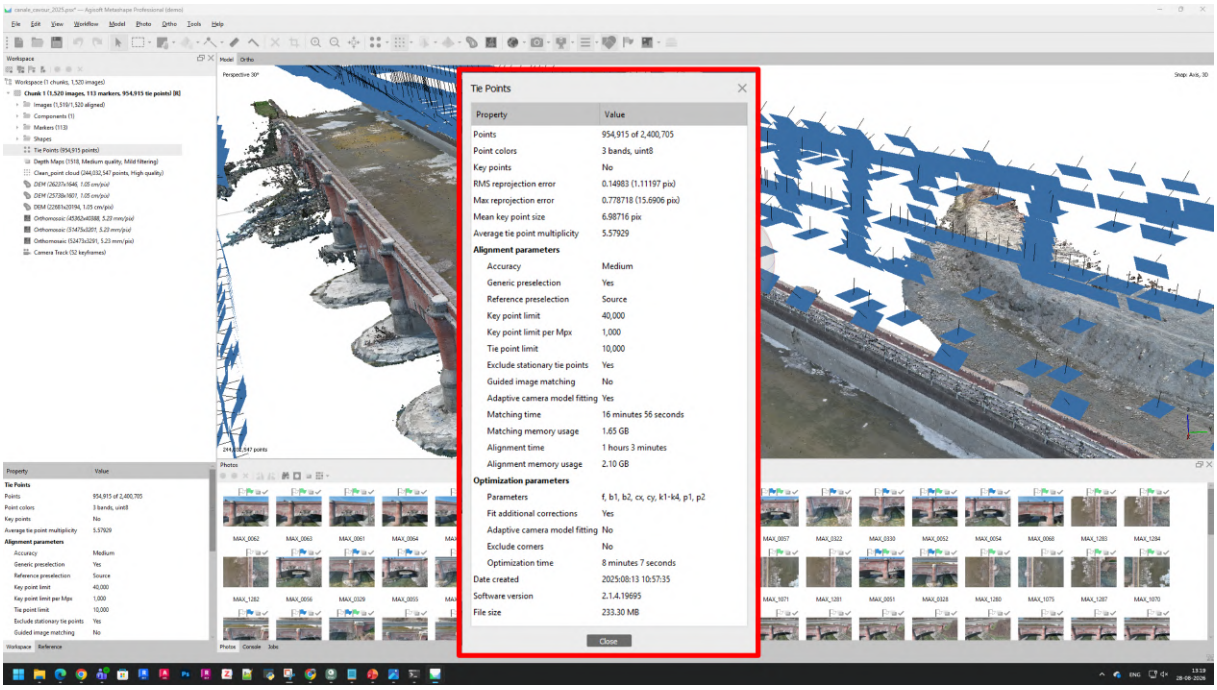


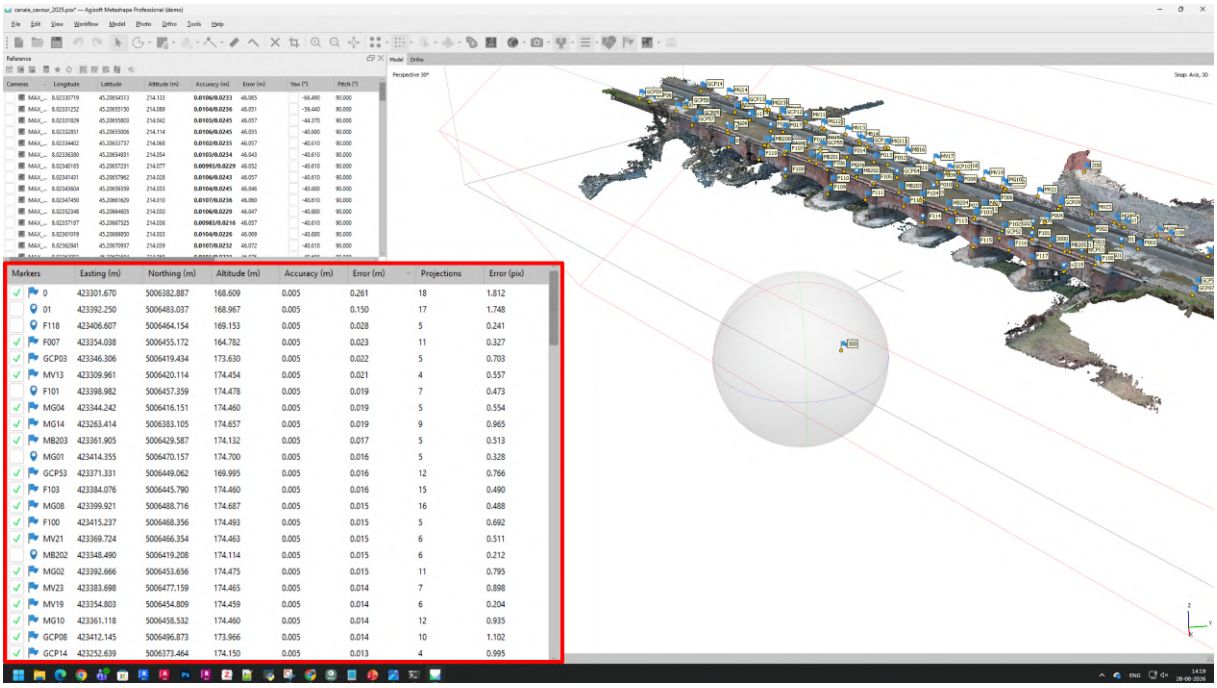
Figure 2.19: *Tiepoints info after aligning photos*

2. Reference marker import: The surveyed reference marker coordinates were imported into Agisoft Metashape to provide accurate control points for the Unmanned aerial vehicle photogrammetric model.

The markers were identified and manually positioned in the corresponding Unmanned aerial vehicle images, allowing the software to relate the image based model to the surveyed ground coordinates. These control points helped constrain the model, improve its spatial accuracy, and reduce possible geometric distortions during the optimization process.(Figure: 2.20)

The coordinate systems in Agisoft Metashape were set according to the reference data used in the survey. The camera positions were referenced using WGS 84 (EPSG:4326), while the surveyed markers were assigned to WGS 84 / UTM Zone 32N (EPSG:32632).

The coordinate systems in Agisoft Metashape were defined according to the location and reference data of the survey. The bridge is located in Chivasso, near Turin, Italy, which falls within UTM Zone 32N.



Markers	Easting (m)	Northing (m)	Altitude (m)	Accuracy (m)	Error (m)	Projections	Error (pix)
0	423301.670	5006382.887	168.609	0.005	0.261	18	1.812
01	423392.250	5006483.037	168.967	0.005	0.150	17	1.748
F118	423406.607	5006464.154	169.153	0.005	0.028	5	0.241
F007	423354.038	5006455.172	164.782	0.005	0.023	11	0.327
GCP03	423346.306	5006419.434	173.630	0.005	0.022	5	0.703
MV13	423309.961	5006420.114	174.454	0.005	0.021	4	0.557
F101	423398.982	5006457.359	174.478	0.005	0.019	7	0.473
MG04	423344.242	5006416.151	174.460	0.005	0.019	5	0.554
MG14	423263.414	5006383.105	174.657	0.005	0.019	9	0.965
MB203	423361.905	5006429.587	174.132	0.005	0.017	5	0.513
MG01	423414.355	5006470.157	174.700	0.005	0.016	5	0.328
GCP53	423371.331	5006449.062	169.995	0.005	0.016	12	0.766
F103	423384.076	5006445.790	174.460	0.005	0.016	15	0.490
MG08	423399.921	5006488.716	174.687	0.005	0.015	16	0.488
F100	423415.237	5006468.356	174.493	0.005	0.015	5	0.692
MV21	423369.724	5006466.354	174.463	0.005	0.015	6	0.511
MB202	423348.490	5006419.208	174.114	0.005	0.015	6	0.212
MG02	423392.666	5006453.656	174.475	0.005	0.015	11	0.795
MV23	423383.698	5006477.159	174.465	0.005	0.014	7	0.898
MV19	423354.803	5006454.809	174.459	0.005	0.014	6	0.204
MG10	423361.118	5006458.532	174.460	0.005	0.014	12	0.935
GCP08	423412.145	5006496.873	173.966	0.005	0.014	10	1.102
GCP14	423252.639	5006373.464	174.150	0.005	0.013	4	0.995

Figure 2.20: *Imported reference markers from .txt file*

Following Coordinate system settings are used:

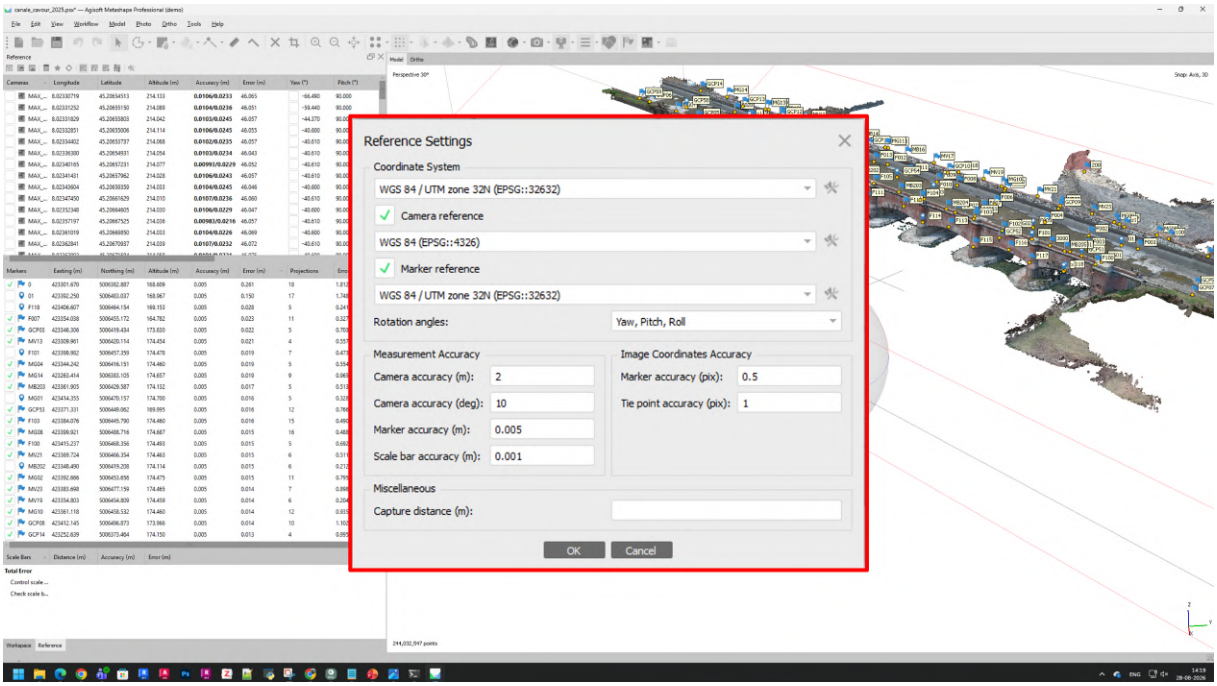


Figure 2.21: *Reference coordinae system settings used in Metashape*

2.2.2 Georeferencing and indexing

The final point cloud was georeferenced within the established survey control framework, aligning all datasets to a common spatial reference system. Control points derived from the total station network were used to validate positional accuracy and ensure consistency between Unmanned aerial vehicle (UAV), Terrestrial Lasser Scanner (TLS).

Terrestrial Laser Scanning Workflow (Cyclone Register 360 plus)

4. Georeferencing via Control Points: To tie the registered scan bundle to absolute coordinates, the surveyed target file containing X, Y, Z positions from the Total Station and GNSS network is imported. (Figure: 2.22)

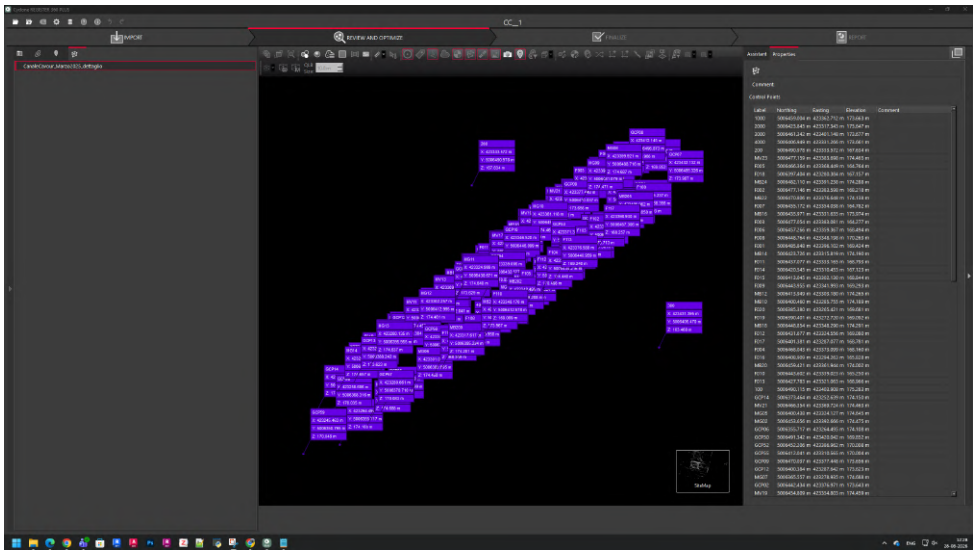


Figure 2.22: *Total Station reference coordinates imported*

5. Target matching: Black and white planar targets and spherical markers identified in the high resolution intensity photos of the scanner were manually paired with their corresponding total station reference coordinates.

As shown in (Figure: 2.22), the identified targets are displayed from total station images and associated with the corresponding target labels, such as F003 and F002. This step was important because it established a direct correspondence between the targets observed by the laser scanner and the points whose coordinates had been independently measured using the total station. In this way, the targets provided common reference points that could be used to constrain and evaluate the registration of the individual scans.

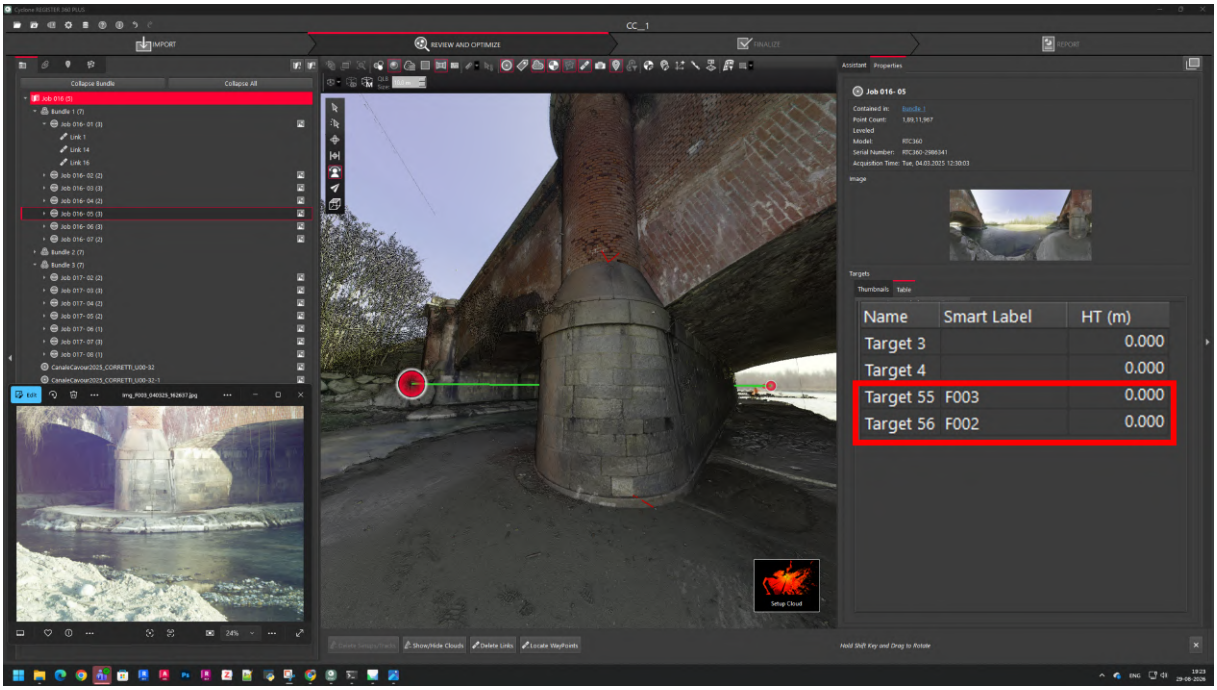
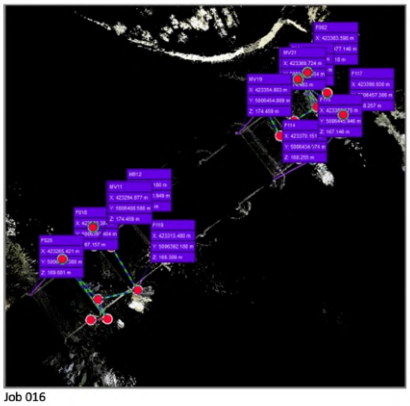


Figure 2.23: *Target matching with help of total station photos*

6. Error assessment:



Job 016

Qualità globale

Risultati Errori per Gruppo 1

Conteggio dei Setup:	8
Conteggio dei collegamenti:	9
Robustezza:	51 %
Sovrapposizione:	50 %
Errore globale del pacchetto:	0.007 m

Errore gruppo	0.006 m ✓
Sovrapposizione	50 % ✓
Robustezza	51 % ✓
Cloud-to-Cloud	0.006 m ✓
Errore target	--

Figure 2.24: *Error assessment of group 1* **??**

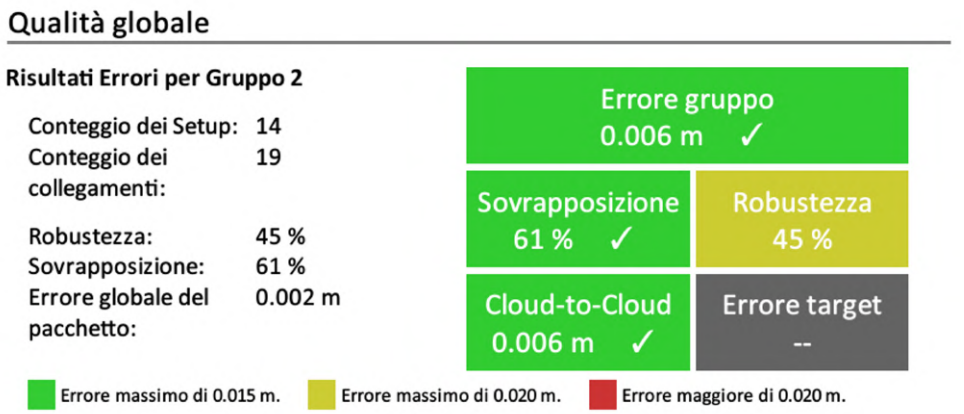
The quality of the registered point cloud was evaluated using both target based and Cloud to Cloud registration errors. A maximum registration error of 20 mm (2 cm) was adopted as the project accuracy target. This threshold was selected to limit the geometric discrepancy introduced during the alignment of individual scanner setups and to ensure that the registered point cloud remains suitable for subsequent geometric analysis.

Errore medio assoluto per il controllo di 'Gruppo 1': **0.016 m**

Nome del gruppo	Setup	Etichetta	Errore
Gruppo 1	Job 017- 02	MB12	0.015 m
	Job 017- 02	F016	0.014 m
	Job 017- 05	F020	0.012 m
	Job 017- 02	F018	0.015 m
	Job 017- 02	MV11	0.018 m
	Job 017- 04	F119	0.018 m

Figure 2.25: *Error assessment of group 1* **??**

Following target optimization, the final bundle adjustment yielded a overall registration error of 0.006m (6 mm) of group 1 in Figure 2.24 & each target error of group 1 is mentioned in Figure 2.25



Errore medio assoluto per il controllo di 'Gruppo 2': **0.012 m**

Nome del gruppo	Setup	Etichetta	Errore
Gruppo 2	Job 016- 05	F002	0.008 m
	Job 016- 08	F115	0.007 m
	Job 016- 08	F117	0.007 m
	Job 016- 08	F114	0.015 m
	Job 016- 12	F005	0.011 m
	Job 016- 12	F004	0.014 m
	Job 016- 05	F003	0.013 m
	Job 016- 05	MV21	0.014 m
	Job 016- 12	MV19	0.016 m

Figure 2.26: *Error assessment of group 2* **??**

These results indicate that the scan network achieved the required registration accuracy and that the resulting point cloud provides a consistent geometric basis for subsequent processing and measurement.

Unmanned Aerial Vechile Photogrammetry Workflow (Agisoft Metshape)

3. Manual marker pinning: Control targets were identified across overlapping source images with help of total station images. Each target was manually pinned in at least 3 4 photos to maximize ray interception accuracy and constrain bundle block adjustment. (Figure:2.27)

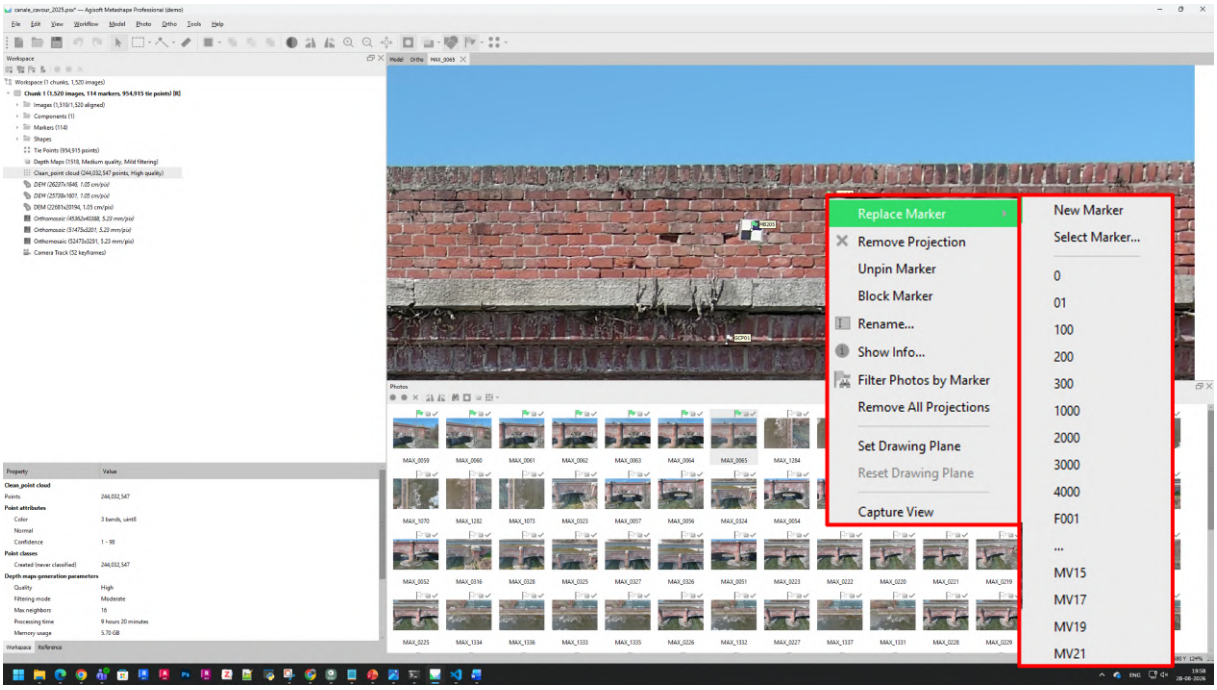


Figure 2.27: *Manual marker pinning*

For this project, the registration and optimization were carried out with the aim of maintaining the positional error within approximately 1–2 cm.

Markers	Easting (m)	Northing (m)	Altitude (m)	Accuracy (m)	Error (m)	Projections	Error (pix)
F118	423406.607	5006464.154	169.153	0.005	0.028	5	0.241
F007	423354.038	5006455.172	164.782	0.005	0.023	11	0.327
GCP03	423346.306	5006419.434	173.630	0.005	0.022	5	0.703
MV13	423309.961	5006420.114	174.454	0.005	0.021	4	0.557
F101	423398.982	5006457.359	174.478	0.005	0.019	7	0.473
MG04	423344.242	5006416.151	174.460	0.005	0.019	5	0.554
MG14	423263.414	5006383.105	174.657	0.005	0.019	9	0.965
MB203	423361.905	5006429.587	174.132	0.005	0.017	5	0.513
MG01	423414.355	5006470.157	174.700	0.005	0.016	5	0.328
GCP53	423371.331	5006449.062	169.995	0.005	0.016	12	0.766
F103	423384.076	5006445.790	174.460	0.005	0.016	15	0.490
MG08	423399.921	5006488.716	174.687	0.005	0.015	16	0.488
F100	423415.237	5006468.356	174.493	0.005	0.015	5	0.692
MV21	423369.724	5006466.354	174.463	0.005	0.015	6	0.511
MB202	423348.490	5006419.208	174.114	0.005	0.015	6	0.212
MG02	423392.666	5006453.656	174.475	0.005	0.015	11	0.795

Figure 2.28: *check markers in reference settings* **??**

The reference markers were manually identified and pinned in the UAV images to establish a reliable connection between the photogrammetric model and the surveyed coordinates. During

this process, some points were selected as control markers, while other points were kept as check points (green check mark) and distributed across the bridge to independently assess the accuracy of the final model. (Figure:2.28)

Markers	Easting (m)	Northing (m)	Altitude (m)	Accuracy (m)	Error (m)	Projections	Error (pix)
F118	423406.607	5006464.154	169.153	0.005	0.028	5	0.241
F007	423354.038	5006455.172	164.782	0.005	0.023	11	0.327
GCP03	423346.306	5006419.434	173.630	0.005	0.022	5	0.703
MV13	423309.961	5006420.114	174.454	0.005	0.021	4	0.557
F101	423398.982	5006457.359	174.478	0.005	0.019	7	0.473
MG04	423344.242	5006416.151	174.460	0.005	0.019	5	0.554
MG14	423263.414	5006383.105	174.657	0.005	0.019	9	0.965
MB203	423361.905	5006429.587	174.132	0.005	0.017	5	0.513
MG01	423414.355	5006470.157	174.700	0.005	0.016	5	0.328
GCP53	423371.331	5006449.062	169.995	0.005	0.016	12	0.766
F103	423384.076	5006445.790	174.460	0.005	0.016	15	0.490
MG08	423399.921	5006488.716	174.687	0.005	0.015	16	0.488
F100	423415.237	5006468.356	174.493	0.005	0.015	5	0.692
MV21	423369.724	5006466.354	174.463	0.005	0.015	6	0.511
MB202	423348.490	5006419.208	174.114	0.005	0.015	6	0.212
MG02	423392.666	5006453.656	174.475	0.005	0.015	11	0.795

Figure 2.29: *Error of check markers in reference panel* **??**

The control markers were used during the optimization process, and preference was given to points showing lower positional errors in the reference table. As shown in the results, several markers have errors in the range of approximately 0.013–0.023 m (13–23 mm), indicating a relatively good agreement between the surveyed coordinates and the photogrammetric model. The check points, identified separately, were not intended to control the adjustment but were used to verify how accurately the optimized model reproduced their known positions. (Figure: 2.28) This separation between control markers and check points provides a more reliable assessment of the final accuracy because the check points are not directly used to improve the model

4. Optimization & dense cloud generation: After the reference markers were identified and controlled, the camera alignment was optimized in Agisoft Metashape using the surveyed marker coordinates. This step refined the camera positions and orientations and adjusted the photogrammetric model to better fit the measured control points. The optimization was carried out with the aim of keeping the positional errors within the 1–2 cm accuracy range required for the project.

Following the optimization, a dense point cloud workflow was executed from the aligned UAV images using depth map generated by the build point cloud workflow tool. Generated depth map contains 1518 points with mid quality. As shown in the results, the final dense cloud contains approximately 244 million points (244,032,547 points), providing a detailed representation of the bridge. The dense cloud was generated using mid quality depth map processing with Moderate filtering and a maximum of 16 neighbouring images considered during processing. This high density point cloud provided the detailed 3D data required for the subsequent analysis and reconstruction of the bridge geometry.

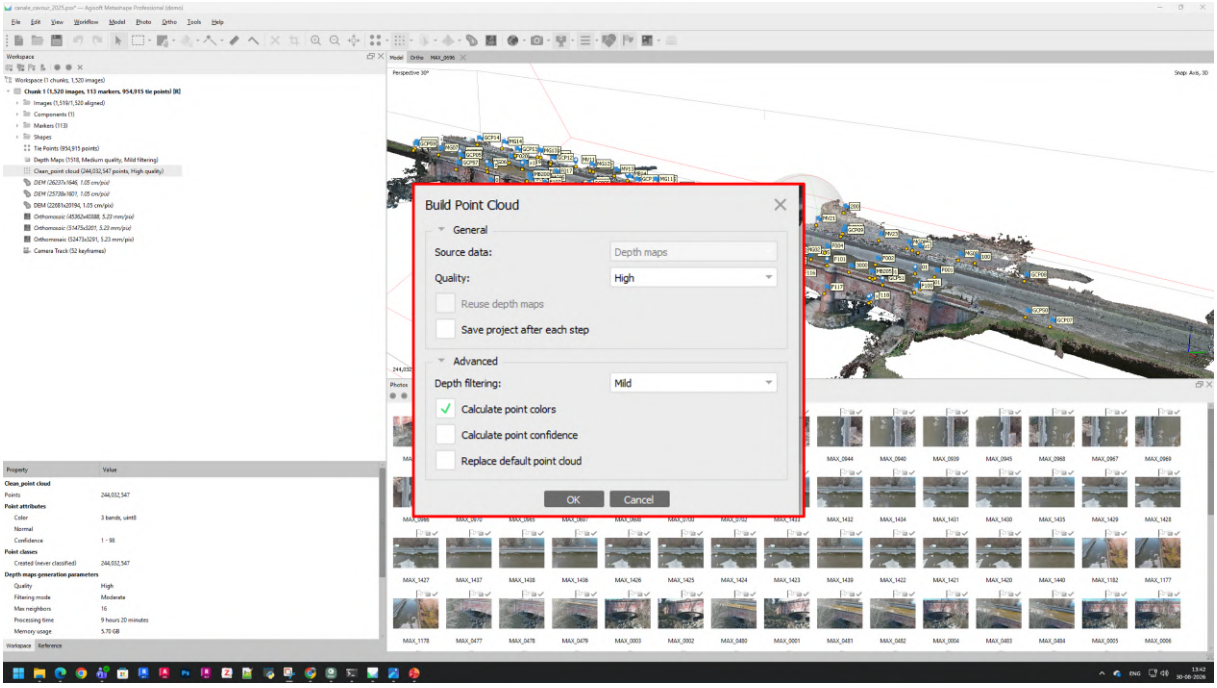


Figure 2.30: *Point cloud generating settings*

2.2.3 Noise cleaning and filtering

Following registration, the integrated point cloud underwent a cleaning and filtering phase to improve data quality and usability for BIM modelling. Noise caused by vegetation, large trees around the scan, and atmospheric artefacts was removed using statistical outlier removal and manual segmentation techniques in Agisoft metashape professional and CloudCompare.

UAV Photogrammetry in Agisoft Metashape

Following the dense point cloud generation in a similar noise cleaning procedure was carried out natively within the software to isolate the bridge geometry from the photogrammetric dataset.

While UAV photogrammetry provided excellent coverage of the upper deck, parapets, and exterior façades, the resulting dense cloud also captured substantial surrounding environmental noise. This included surrounding trees, ground terrain along the riverbanks, floating vegetation.

Using Metashape's manual selection tools—such as the Free form Selection and Polygonal Selection tools unwanted points and floating artifacts surrounding the structure were highlighted in 3D space and deleted directly from the workspace. Particular attention was given to the margins along the bridge deck and top edges of the parapets to ensure that surrounding foliage was cleanly excised without trimming the structural edge details of the masonry. (Figure: 2.31)

This step yielded a refined, high density UAV dense cloud consisting solely of the bridge's physical structure, ensuring a clean dataset for subsequent integration with the TLS point cloud and final 3D modeling. (Figure: 2.32)

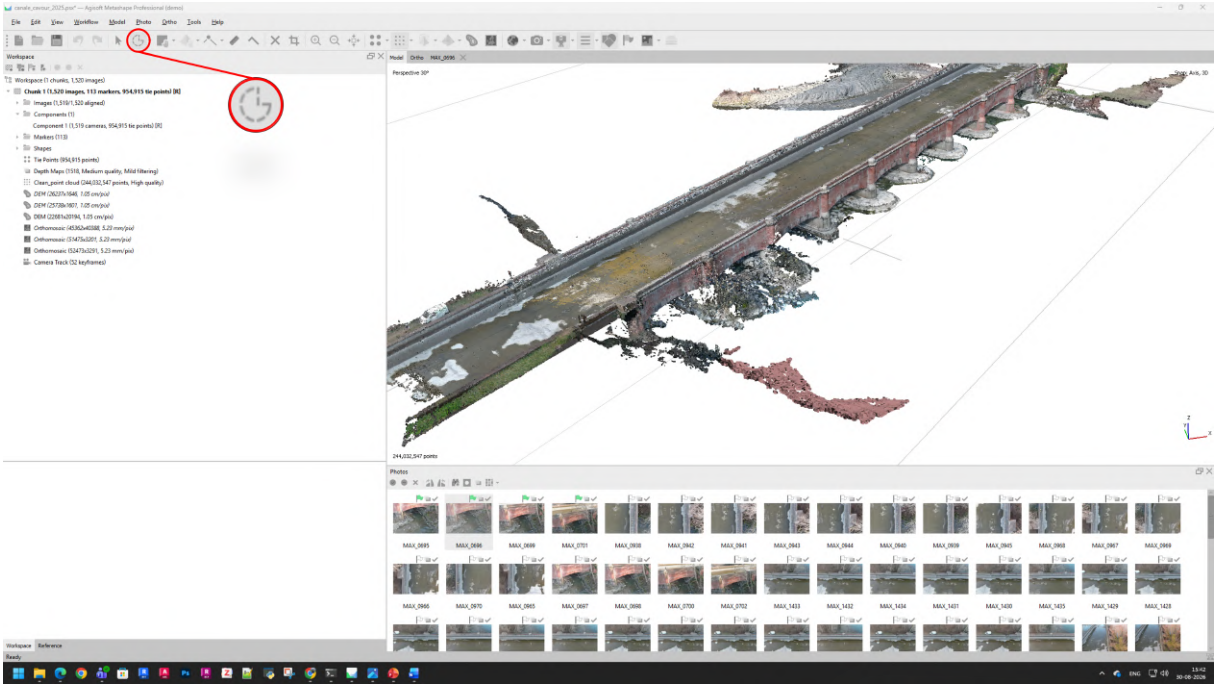


Figure 2.31: *Point cloud selection tool*

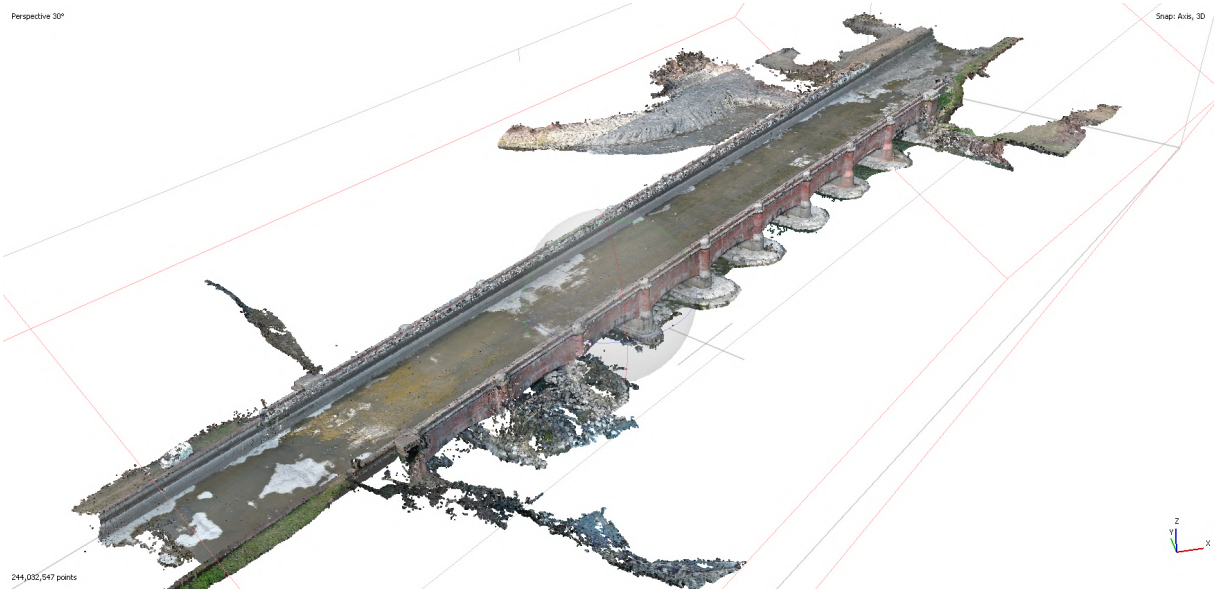


Figure 2.32: *Point cloud after cleaning the noise*

Terrestrial Laser Scanner in Cloud Compare

Following registration and georeferencing in Leica Cyclone REGISTER 360, the registered TLS scan stations were exported as individual .e57 files and imported into CloudCompare for detailed segmentation and manual noise filtering.

Raw laser scan data inherently contains extraneous spatial information, including surrounding vegetation, terrain, distant riverbanks.

Removing these non relevant elements was essential to isolate the primary masonry structure,

reduce file size, and optimize performance for downstream photogrammetric integration and 3D modeling

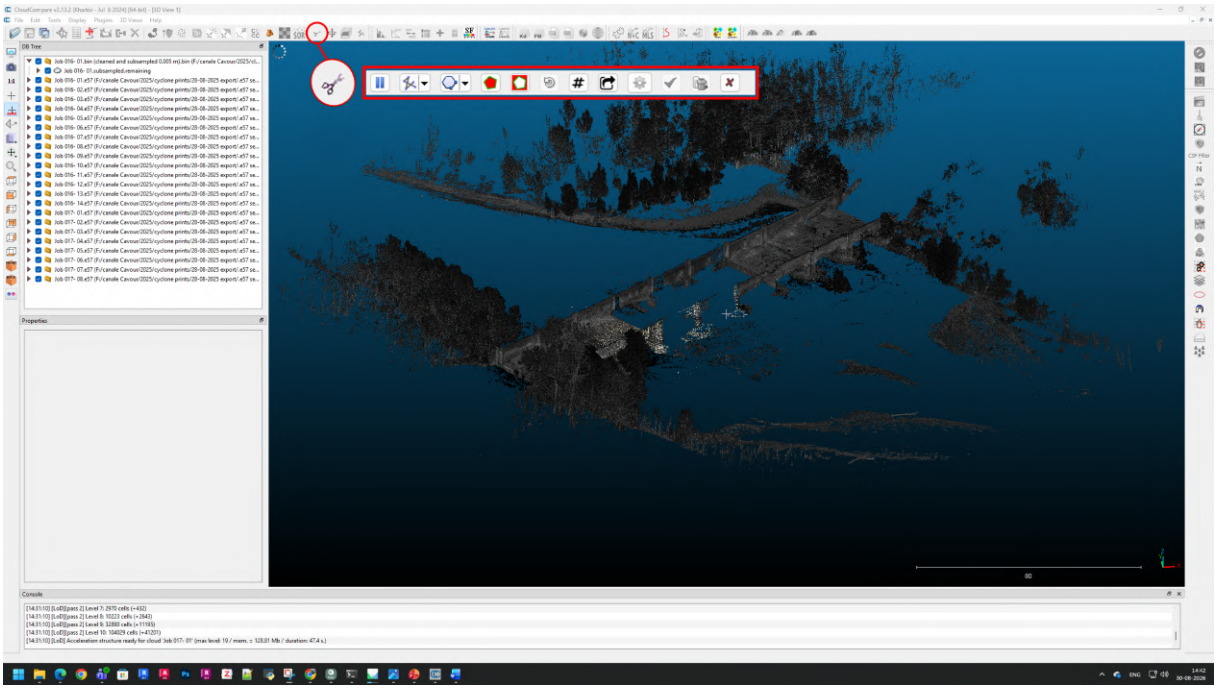


Figure 2.33: *Point cloud segmentation tool*

Manual cleaning was performed using CloudCompare's Segment (Scissors) tool. (Figure: 2.33) By drawing custom 3D polygonal boundaries around non structural features, points were split into internal and external subsets, allowing for the systematic removal of dense embankment vegetation and stray laser returns caused by the river surface. Special care was taken around the intrados and lower pier surfaces to clear away water noise without eroding the structural masonry geometry.

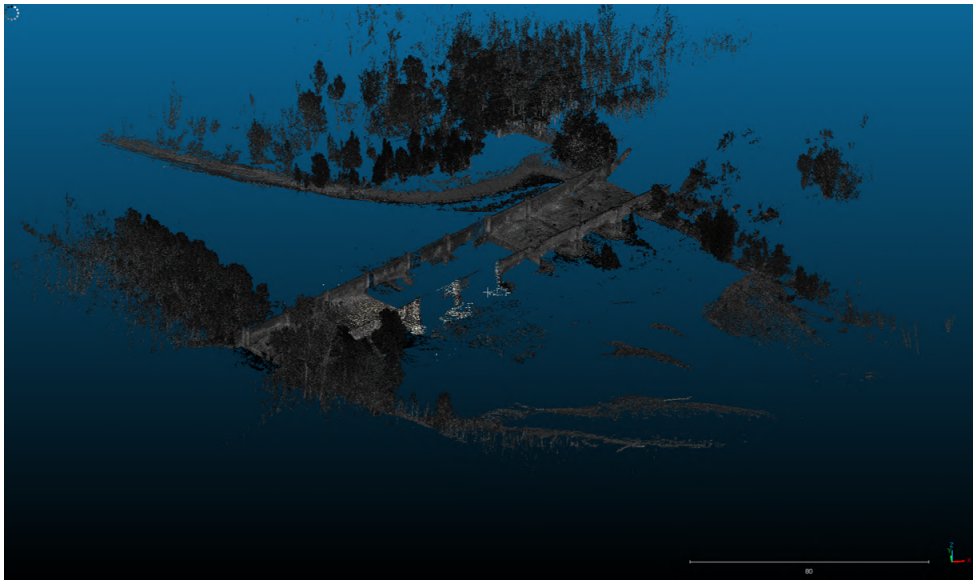


Figure 2.34: *Point cloud with noise*

The resulting cleaned point cloud isolates only the essential structural components of the bridge including the piers, arches, intrados, and deck surfaces producing an uncluttered, high density dataset ready for photogrammetric comparison and surface mesh generation. (Figure: 2.35)

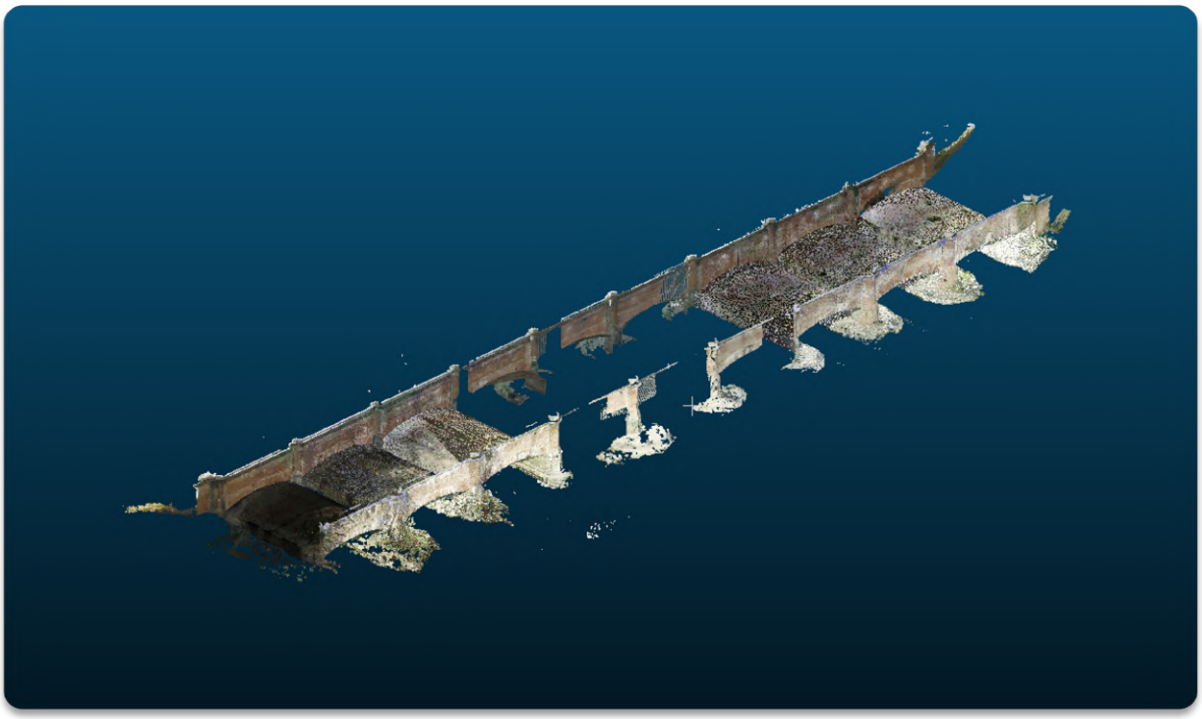


Figure 2.35: *Point cloud after cleaning the Noise*

2.3 Results

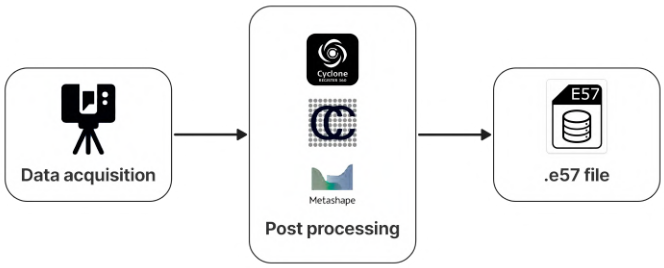
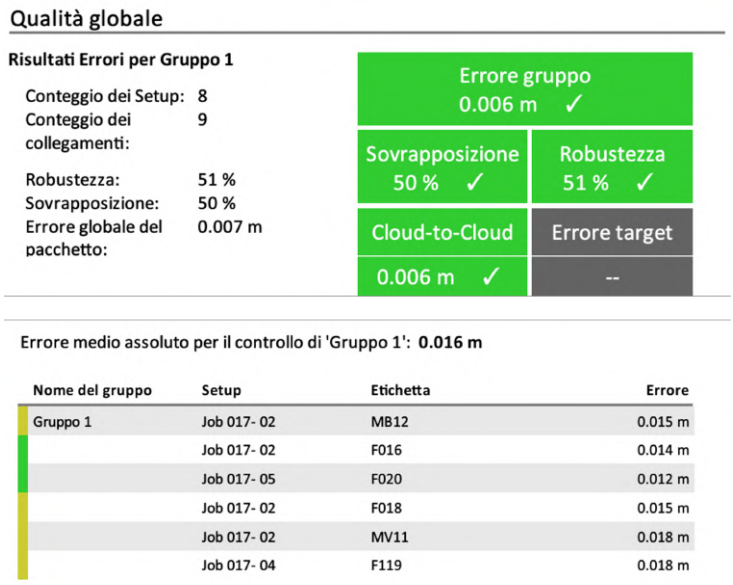


Figure 2.36: *Point cloud after cleaning the Noise*

2.3.1 Terrestrial Laser Scanner-Leica Cyclone Register 360 Plus

The Terrestrial Laser Scanning (TLS) dataset was carried out using Leica Cyclone REGISTER 360 PLUS. To manage the alignment process effectively, the scanning stations were divided into two primary processing groups.



Qualità globale

Risultati Errori per Gruppo 1

Conteggio dei Setup:	8
Conteggio dei collegamenti:	9
Robustezza:	51 %
Sovrapposizione:	50 %
Errore globale del pacchetto:	0.007 m

Errore gruppo 0.006 m ✓	
Sovrapposizione 50 % ✓	Robustezza 51 % ✓
Cloud-to-Cloud	Errore target
0.006 m ✓	--

Errore medio assoluto per il controllo di 'Gruppo 1': **0.016 m**

Nome del gruppo	Setup	Etichetta	Errore
Gruppo 1	Job 017- 02	MB12	0.015 m
	Job 017- 02	F016	0.014 m
	Job 017- 05	F020	0.012 m
	Job 017- 02	F018	0.015 m
	Job 017- 02	MV11	0.018 m
	Job 017- 04	F119	0.018 m

Figure 2.37: *Group 2 Overall bundle report and target error*

The first group comprised 8 individual scan setups connected by 9 alignment links. For this network, the optimization achieved a global bundle error of 0.007 m and a cloud to cloud error of 0.006 m. The results showed a surface overlap of 50% and a structural network strength of 51%, with a mean absolute error for the control targets of 0.016 m. (Figure: 2.37)

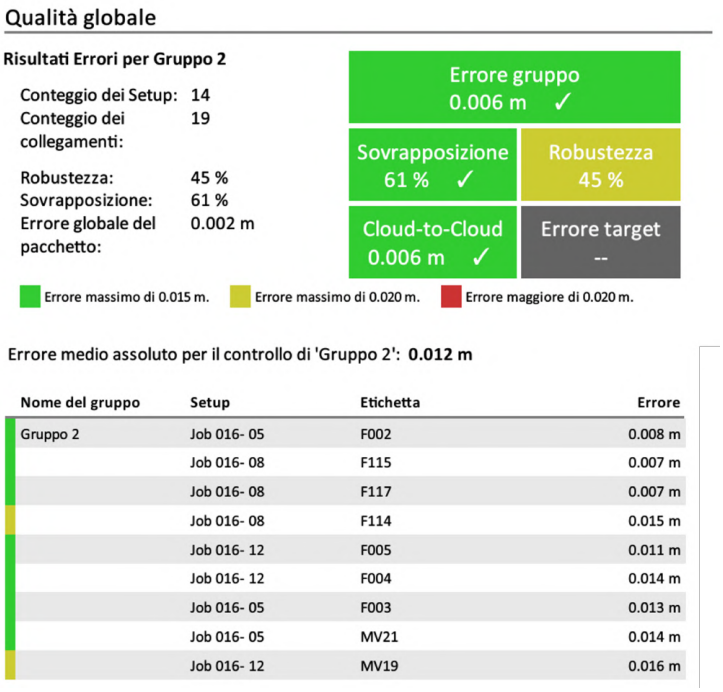


Figure 2.38: *Group 1 Overall bundle report and target error* **??**

The second group involved a larger network of 14 scan setups and 19 alignment links. The results for this group demonstrated a refined global bundle error of 0.002 m, while maintaining a consistent cloud to cloud error of 0.006 m. This network achieved a surface overlap of 61% and a structural strength of 45%, with a mean absolute error of 0.012 m for its control targets. (Figure: 2.38)

1 Panoramica

Nome collegamento	Setup 1	Setup 2	Sovrapposizione	Errore medio assoluto
Collegamento 1	Job 017- 01	Job 017- 02	85 %	0.012 m
Collegamento 2	Job 017- 06	Job 017- 07	43 %	0.002 m
Collegamento 3	Job 017- 01	Job 017- 04	64 %	0.007 m
Collegamento 4	Job 017- 08	Job 017- 06	13 %	0.007 m
Collegamento 5	Job 017- 07	Job 017- 04	21 %	0.002 m
Collegamento 6	Job 017- 02	Job 017- 05	53 %	0.005 m
Collegamento 7	Job 017- 05	Job 017- 07	73 %	0.004 m
Collegamento 8	Job 017- 05	Job 017- 06	39 %	0.003 m
Collegamento 9	Job 017- 03	Job 017- 02	56 %	0.014 m

Nome collegamento	Setup 1	Setup 2	Sovrapposizione	Errore medio assoluto
Collegamento 10	Job 016- 01	Job 016- 02	47 %	0.006 m
Collegamento 11	Job 016- 03	Job 016- 02	78 %	0.004 m
Collegamento 12	Job 016- 04	Job 016- 03	32 %	0.008 m
Collegamento 13	Job 016- 04	Job 016- 05	42 %	0.012 m
Collegamento 14	Job 016- 03	Job 016- 05	27 %	0.004 m
Collegamento 15	Job 016- 02	Job 016- 05	24 %	0.004 m
Collegamento 16	Job 016- 06	Job 016- 05	84 %	0.004 m
Collegamento 17	Job 016- 06	Job 016- 07	79 %	0.003 m
Collegamento 18	Job 016- 07	Job 016- 09	72 %	0.007 m
Collegamento 19	Job 016- 10	Job 016- 09	76 %	0.005 m
Collegamento 20	Job 016- 09	Job 016- 08	52 %	0.007 m
Collegamento 21	Job 016- 14	Job 016- 08	61 %	0.009 m
Collegamento 22	Job 016- 13	Job 016- 14	80 %	0.002 m
Collegamento 23	Job 016- 12	Job 016- 13	81 %	0.003 m
Collegamento 24	Job 016- 12	Job 016- 11	54 %	0.008 m
Collegamento 25	Job 016- 11	Job 016- 10	68 %	0.004 m
Collegamento 26	Job 016- 10	Job 016- 06	83 %	0.002 m
Collegamento 27	Job 016- 05	Job 016- 11	68 %	0.005 m
Collegamento 28	Job 016- 07	Job 016- 01	45 %	0.009 m

Figure 2.39: *Group 1 & group 2 link errors* **??**

across the entire dataset, the spatial alignment remained well within the project's established accuracy target of 1 to 2 cm. Individual target residuals recorded across the network ranged from a minimum of 0.007 m, to a maximum of 0.018 m. (Figure: 2.39)

The dataset was not exported as a single unified file. Instead, every registered scan setup was exported separately as an individual .e57 file. This step was carried out to ensure better file management and to minimize system memory overloads during data transfer. More importantly, exporting the scans individually allowed for highly precise, isolated noise cleaning of each station in CloudCompare prior to merging the dataset and integrating it with the UAV photogrammetric point cloud.

2.3.2 Unmanned Aerial Vehicle-Agisoft Metashape professional

Following the cleaning of the UAV dense point cloud, Digital Elevation Models (DEMs) and high resolution orthomosaics were generated within Agisoft Metashape. To accurately represent the bridge elevations without perspective distortion, custom planar projection planes were defined using custom markers on the facade of the bridge for both the north façade, south façade. The DEMs were built directly from the refined dense point cloud, providing a precise topographic baseline required to project the true orthophotos. (Figure: 2.40)

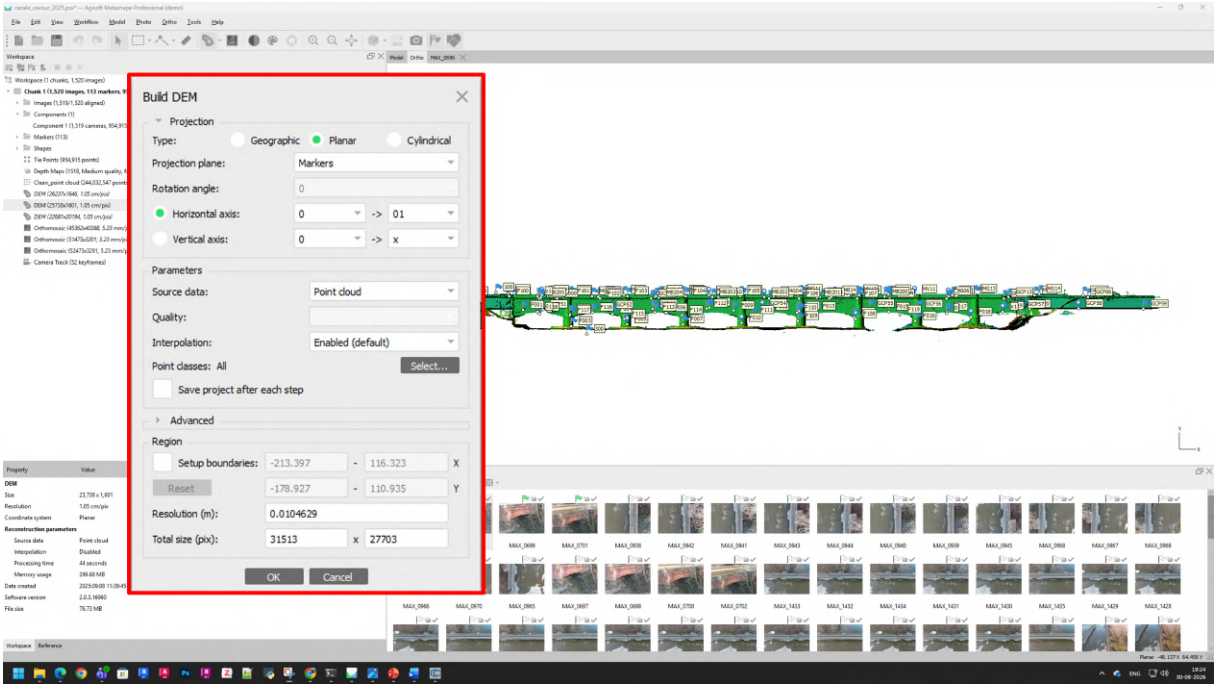
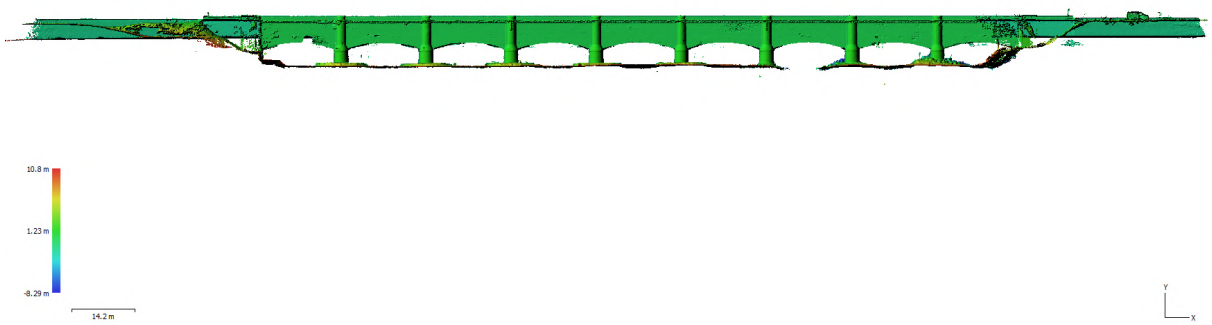


Figure 2.40: *DEM settings in build DEM workflow*

The resulting planar DEMs achieved spatial resolutions ranging between 1.00 cm/pixel and 1.05 cm/pixel. (Figure: 2.41)

Using these planar DEMs as the surface source, high detail orthomosaics were subsequently generated for each principal elevation. This step was carried out using the 'Mosaic' blending mode, with hole filling enabled to seamlessly close minor surface gaps in the point data. The results showed a final spatial resolution of 5.23 mm/pixel across the orthomosaics. Due to this high level of detail, the final raster files were substantial; the output images reached dimensions of 32,473 by 3,231 pixels and 31,475 by 8,201 pixels. (Figure: 2.42)



(a)



(b)

Figure 2.41: *(a) DEM of north facade, (b) DEM of south facade*

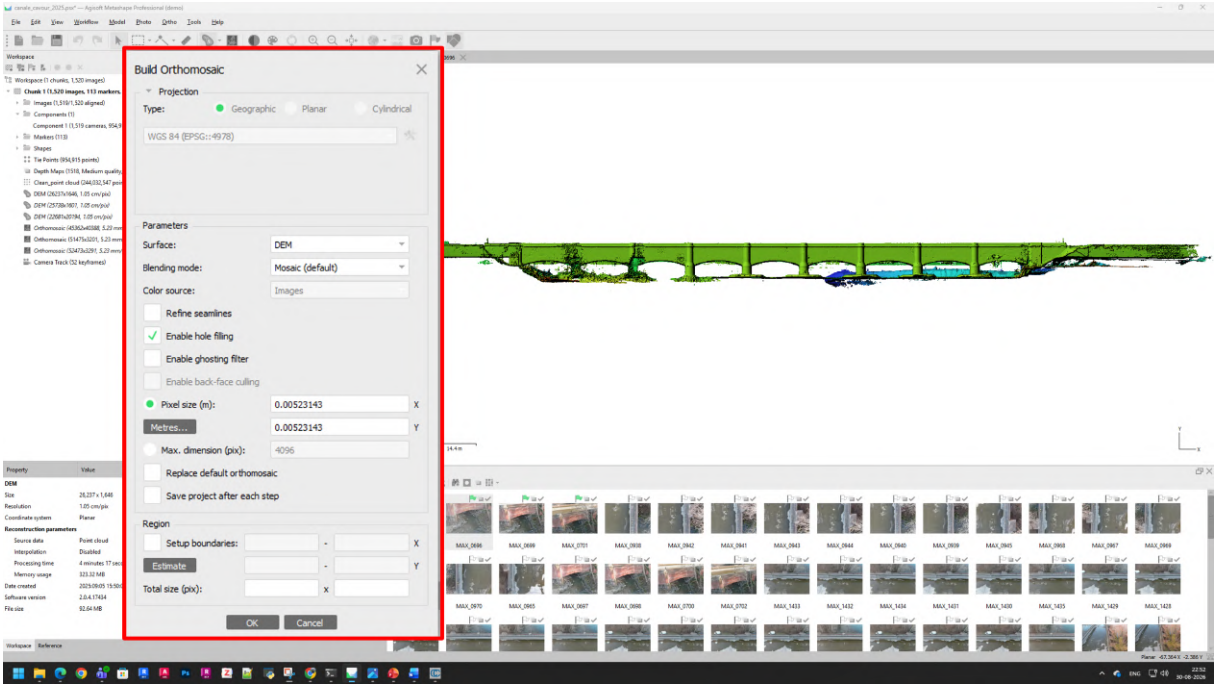


Figure 2.42: *Orthomosaic settings in build orthomosaic workflow*

The main purpose of producing these sub centimeter resolution orthomosaics and planar DEMs was to establish a foundation for non destructive structural diagnostics and heritage documentation. By eliminating scale variance and perspective displacement, the 5.23 mm/pixel orthomosaics allow for direct, millimeter scale visual tracing of structural cracks, shear fractures, and material pathologies such as brick spalling or moisture staining. (Figure: 2.43)

Furthermore, the planar DEMs effectively convert surface distances into depth maps relative to a vertical plane, allowing for the future identification of out of plane deformations or lateral wall displacement along the masonry façades. These orthophotos also serve as a highly accurate metric background for stone by stone architectural vectorization in CAD software.

While the DEMs and orthomosaics serve as the primary 2D orthographic deliverables for façade analysis, the broader scope of the project required the construction of a complete 3D digital shadow of the bridge.



(a)



(b)

Figure 2.43: *(a) Orthomosaic of north facade, (b) Orthomosaic of south facade*

For this 3D spatial reconstruction, only the cleaned UAV dense point cloud was exported from Metashape in .e57 format. Because aerial photogrammetry is inherently limited by top down and angled camera trajectories, the UAV survey could not adequately capture the shaded undersides of the arch intrados and the lower pier bases situated over the running water of the Dora River.

Exporting the photogrammetric data as an .e57 file allowed it to be imported into CloudCompare. This enabled the subsequent co registration with the Terrestrial Laser Scanning (TLS) dataset, explicitly using the ground based TLS data to fill the missing intrados geometry and form a unified, fully realized 3D spatial model.

2.3.3 Post processing-Cloud Compare

To integrate the multi sensor survey data within CloudCompare, the individual .e57 files from the Leica Cyclone REGISTER 360 TLS project were merged into a single point cloud, and the cleaned UAV photogrammetric point cloud exported from Agisoft Metashape was imported into the workspace.

Although both datasets were independently georeferenced to WGS 84 / UTM Zone 32N (EPSG:32632) using surveyed ground control points, a slight spatial discrepancy consisting of a small rotational shift and translation offset was observed when displaying them together. Such residual alignment differences are typical when combining terrestrial laser scanning and aerial photogrammetry:

Complementary vantage points: The TLS survey primarily captures ground level elements, pier bases, and the arch intrados, while the UAV survey provides extensive coverage of the top bridge deck, parapets, and exterior façades.

Survey framework & processing differences: Slight variations in target identification precision between target spheres/checkerboards in Cyclone and manual marker selection in Metashape can introduce slight relative offsets when projected into local working coordinates.

To resolve this shift and achieve precise spatial dataset for building BIM model, the UAV photogrammetric dataset was designated as the fixed Reference entity, while the TLS point cloud was assigned as the "To Align" entity to be transformed.

Automated ICP registration assessment An automated fine alignment was initially executed using CloudCompare's Iterative Closest Point algorithm ("Finely registers already (roughly) aligned entities").

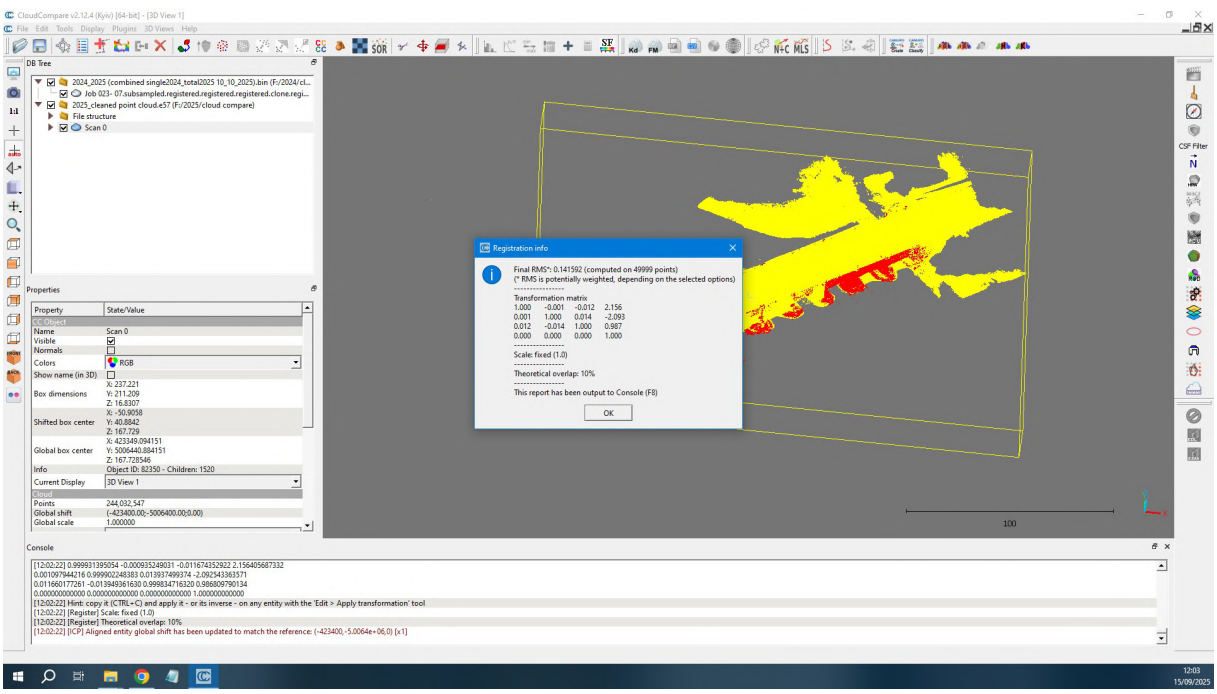


Figure 2.44: *Automated ICP registration assessment*

As shown in the initial alignment dialog, the algorithm calculated a Final RMS error of 0.141592 m (approx. 14.16 cm) based on an estimated theoretical overlap of only 10%. (Figure: 2.44)

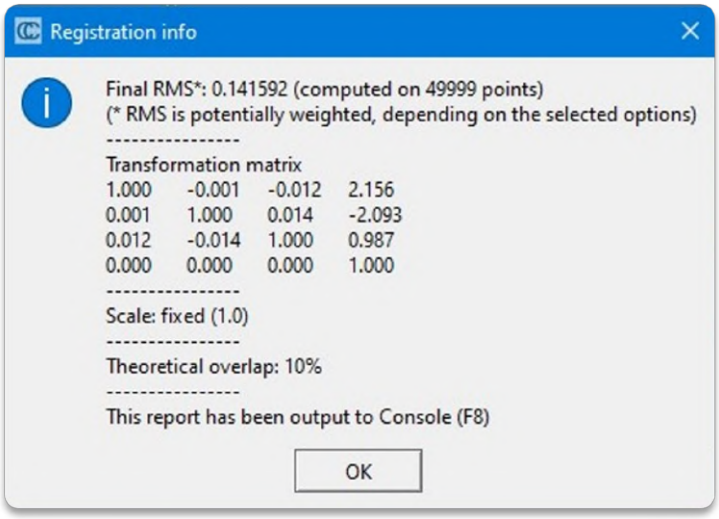


Figure 2.45: *Automated ICP registration assessment*

Because automated ICP relies heavily on matching adjacent surface points across high overlap areas, the partial geometric overlap between the ground focused TLS scan and the deck focused UAV cloud caused the optimization to converge onto a sub optimal local minimum. An error of 14.2 cm was well outside the project's target accuracy threshold of 1–2 cm, making the automated ICP result unsatisfactory for final structural analysis. (Figure: 2.45)

Point pair registration assessment To achieve the required accuracy, a manual target pair registration was performed using the "Align two point clouds by picking equivalent point pairs" tool. Corresponding structural control points common to both datasets—such as distinct masonry corners and capstone edges were identified across the Reference (UAV) and "To Align" (TLS) clouds. (Figure: 2.46)

As shown in the point pair alignment results, matching these equivalent physical points yielded a significant improvement, producing a Final RMS error of 0.00935003 m (approx. 0.94 cm) (Figure: 2.47). The individual point residual errors recorded across the selected pairs were:

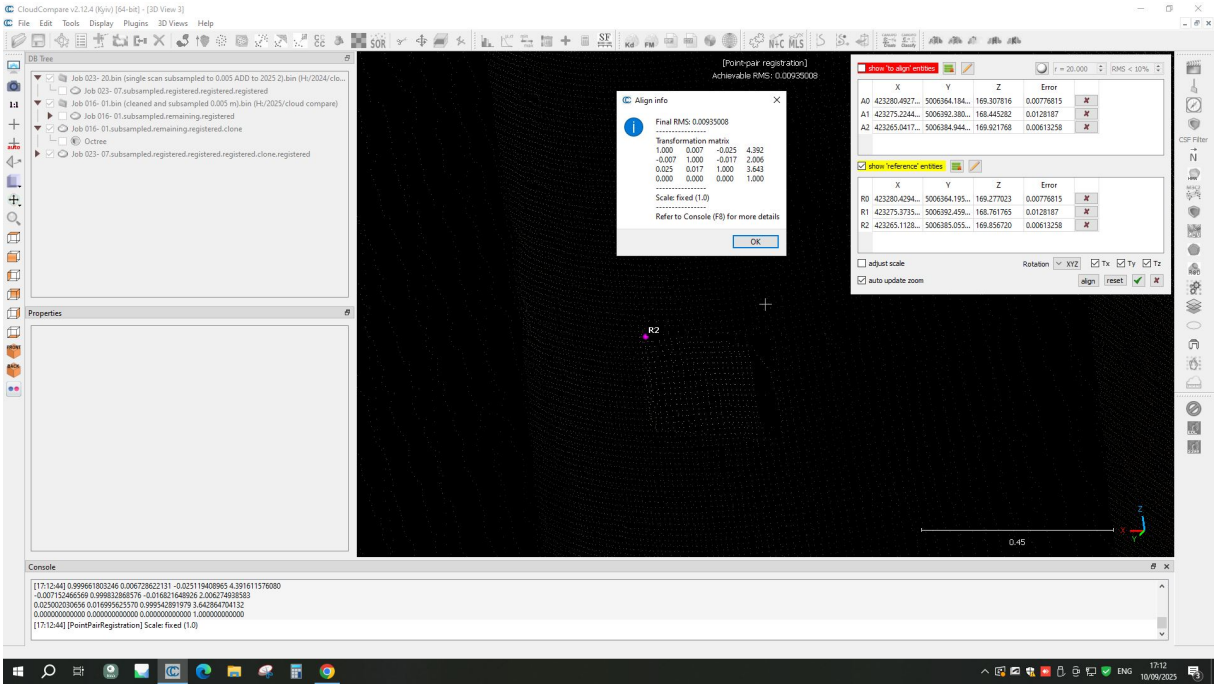


Figure 2.46: *Point pair method for roation and translation*

the point pair method successfully eliminated the rotational and translational offsets. The resulting sub centimeter RMS error satisfied the project target of under 1–2 cm, ensuring a seamless, highly accurate fusion of the TLS and UAV datasets.

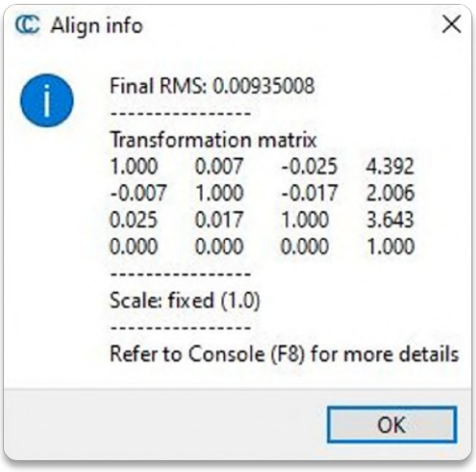


Figure 2.47: *Align info final rms*

Sub sampling point cloud CloudCompare's Subsample tool reduces point cloud file size by decreasing the number of points while retaining the overall geometry. Depending on the selected method, points are either randomly selected, spatially filtered, or selected using an octree structure. Importantly, the remaining points are original points, not newly created ones; therefore, their coordinates, colors, normals, and scalar field information are preserved.

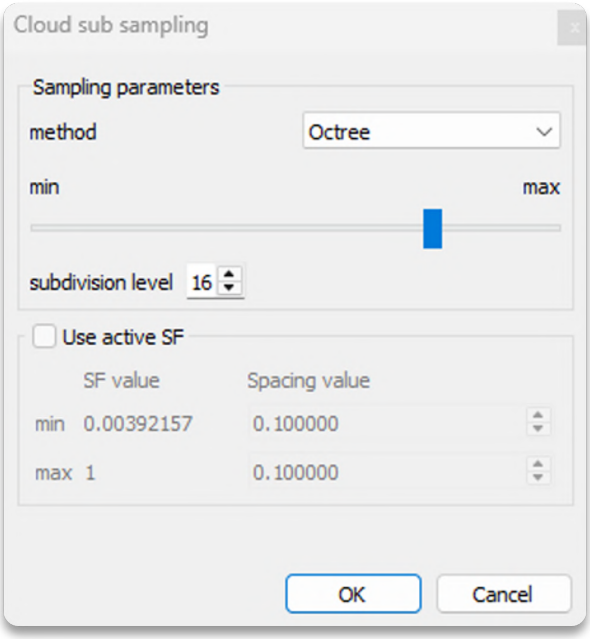


Figure 2.48: *Subsampling using spatial method settings*

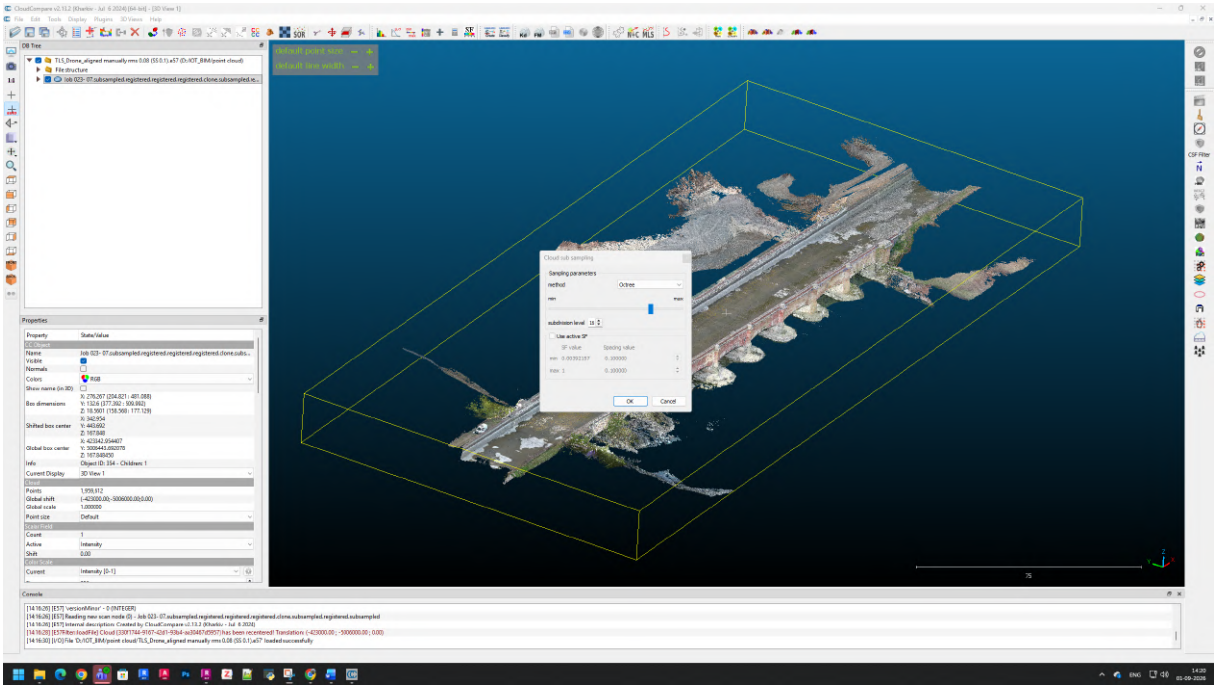


Figure 2.49: *Subsampling using spatial method*

In this study, CloudCompare's Subsample tool was applied using two approaches. The Spatial method was set to a minimum spacing of 0.10 m, (Figure: 2.49) ensuring that points closer than 10 cm were reduced. The Octree method was also tested using a subdivision level of 16, (Figure: 2.50) which reduces point density by selecting representative points within spatial cells. These settings reduced the dataset size while maintaining the overall geometry and important structural features of the point cloud.

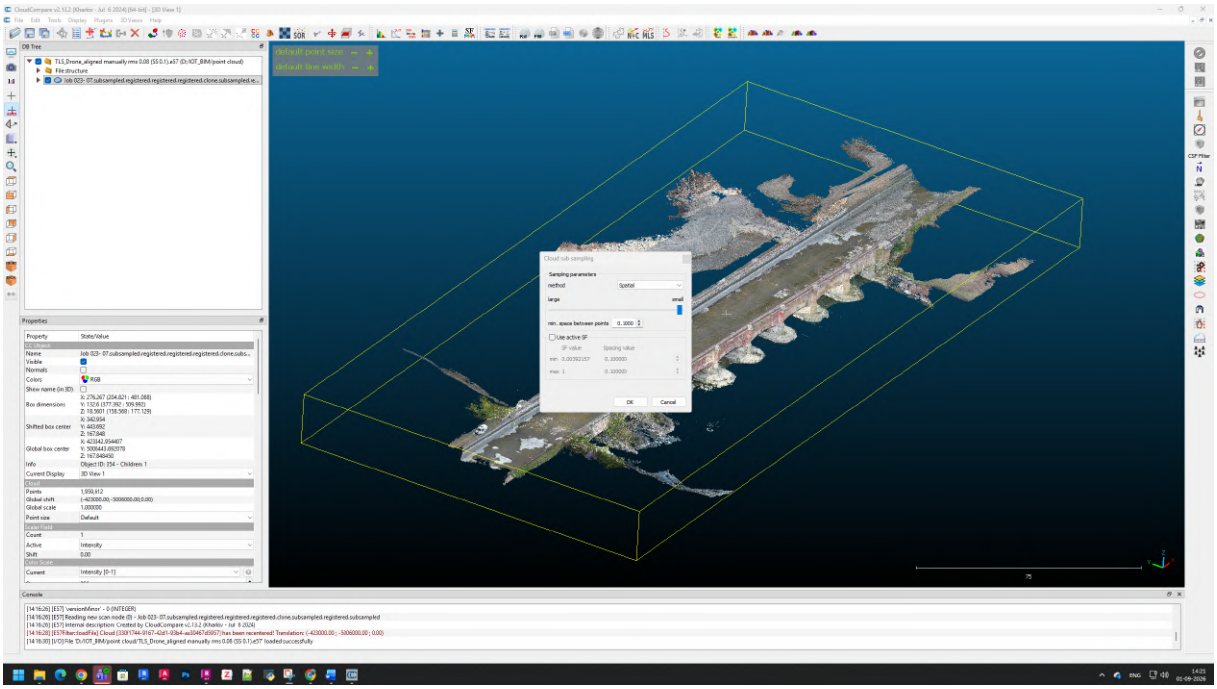


Figure 2.50: *Subsampling using spatial method*

Terrain data pre processing and alignment The raw LiDAR point cloud was first imported into CloudCompare to perform manual data cleaning. The Segment (Scissors) tool was used to crop and remove non ground noise and artificial structures, specifically removing the bridge deck to prevent artificial elevation spikes in the final terrain model. Following segmentation, the point cloud was spatially downsampled using a minimum point to point distance threshold of 0.05 m to reduce dataset density, eliminate redundancy, and standardize spatial resolution. Finally, a rigid body transformation (translation and rotation) was applied to orient and position the point cloud within the intended project reference coordinate system.

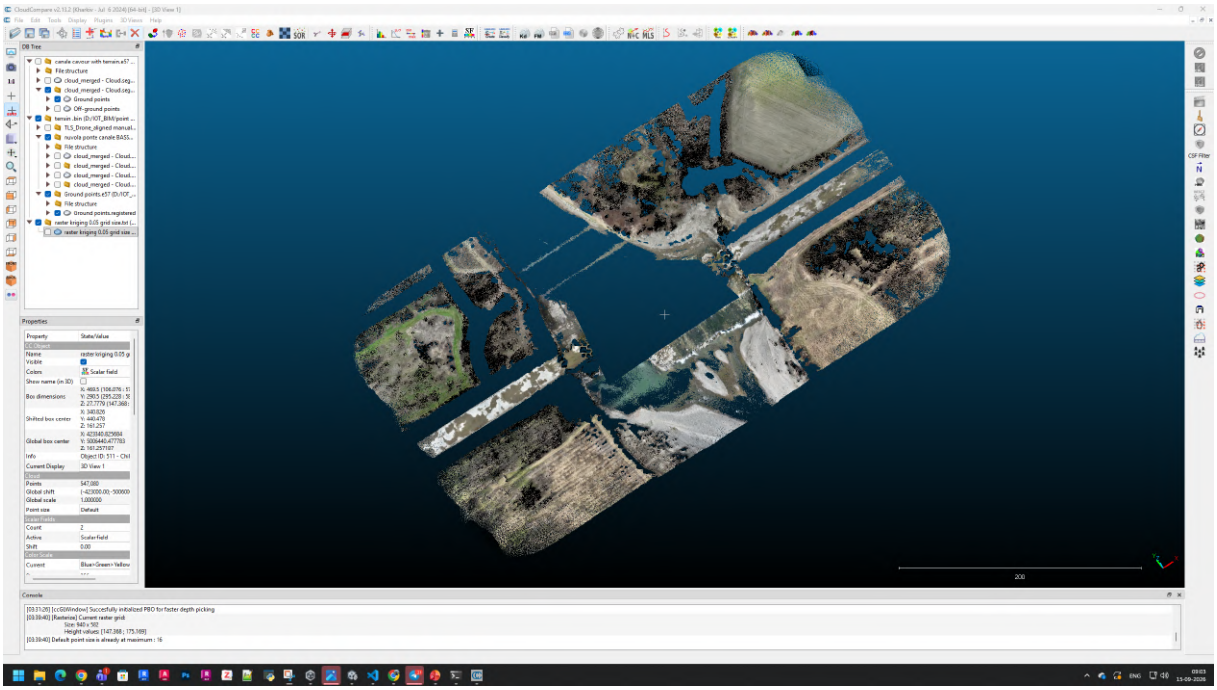


Figure 2.51: *Cleaning of LiDAR point cloud*

Ground surface extraction with Cloth Simulation Filter (CSF) To isolate ground points from residual vegetation and surrounding structures, the Cloth Simulation Filter (CSF) plugin was applied. (Figure: 2.52) The point cloud was inverted upside down, and a rigid elastic cloth surface was simulated draping over the flipped model to detect the true terrain boundary. As shown in Figure 1, the scene configuration was set to Relief, optimizing the algorithm's flexibility for moderate slope variations and micro topography while preserving natural ground contours around the removed bridge area.

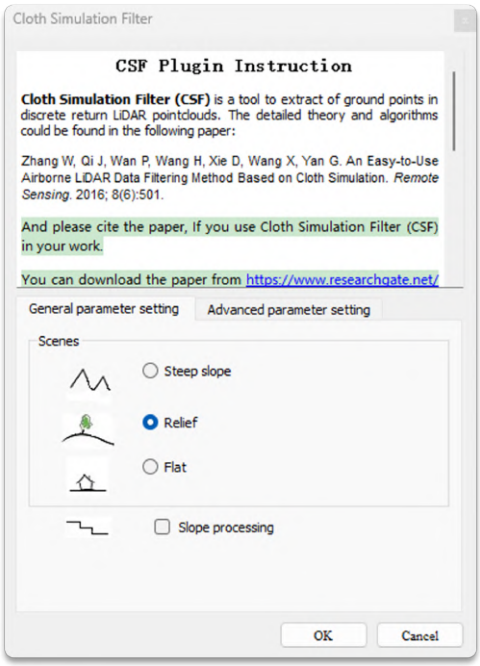


Figure 2.52: *Cloth Simulation Filter settings*

Rasterization and spatial interpolation (Kriging) The classified ground point cloud was then converted into a regularized Digital Elevation Model (DEM) using the Rasterize tool with settings detailed in Figure 2. The dataset was projected along the Z axis onto a grid with a spatial cell size (step) of 0.50 m, creating a 940×582 cell grid ($547{,}080$ total cells), with populated cell heights assigned using the average elevation value of internal points. To fill the empty grid cells resulting from the bridge removal and sparse point coverage, Kriging interpolation was selected. Kriging utilized spatial autocorrelation to predict realistic, continuous elevation values across the void left by the bridge. (Figure: 2.53)

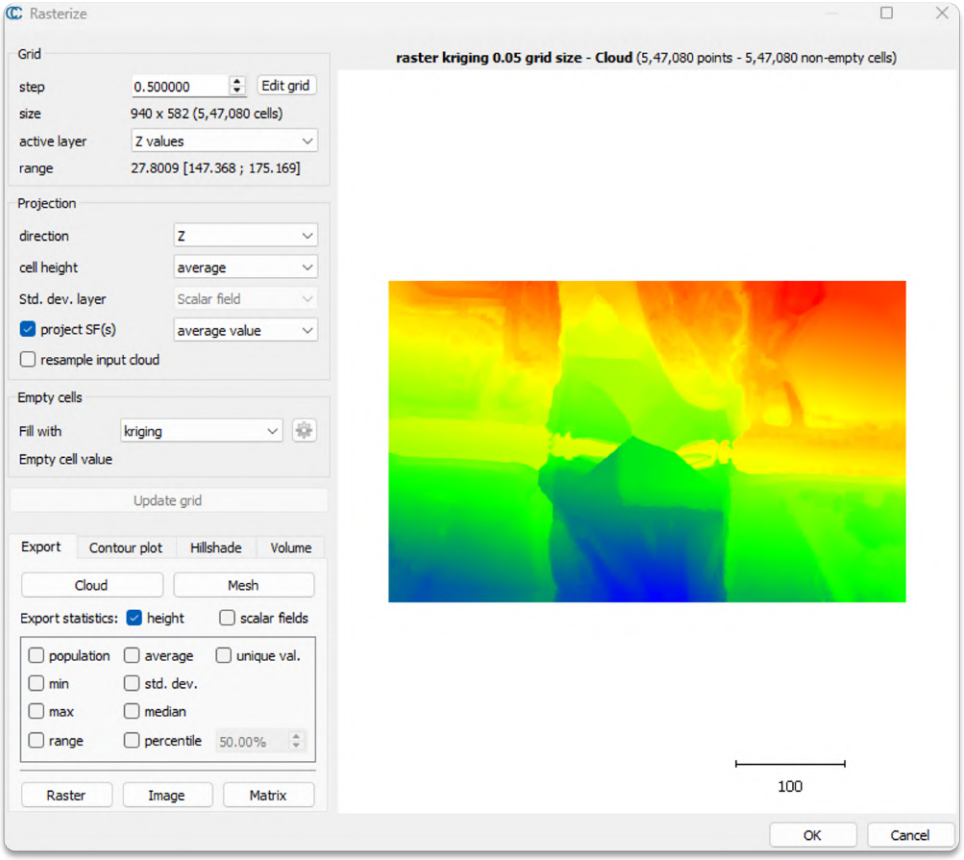


Figure 2.53: *Rasterize Kriging settings*

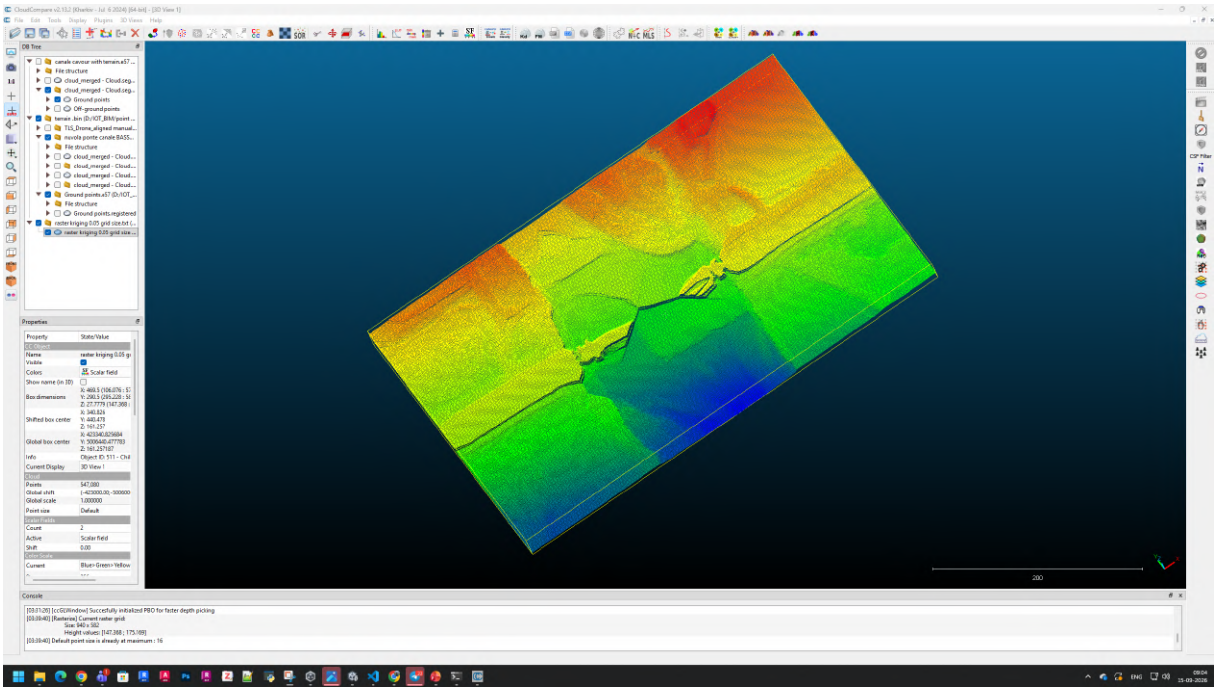


Figure 2.54: *Rasterize Kriging output*

Once the grid was fully interpolated and updated, the resulting rasterized terrain model was exported as a structured matrix text file (.txt / ASCII raster). This file format allows seamless integration into downstream engineering, hydraulic, and spatial analysis applications.

The final .e57 file has been exported from cloud compare.

Chapter 3

Scan to BIM

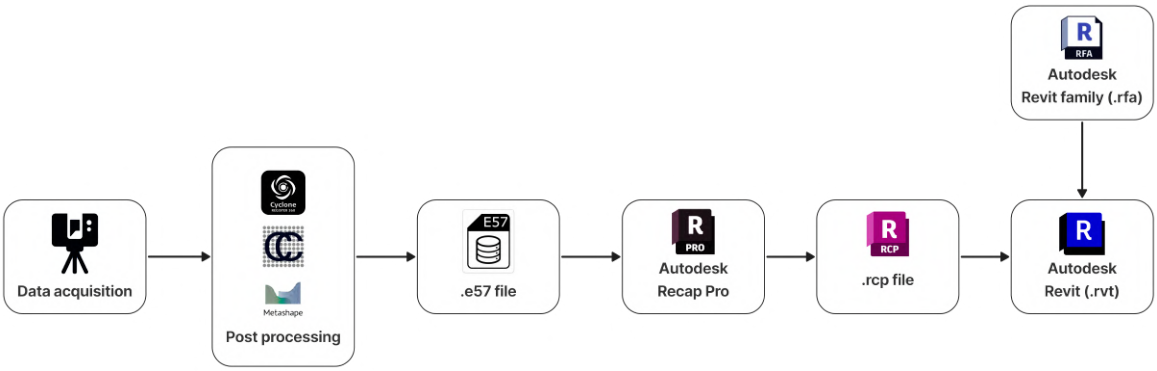


Figure 3.1: *Illustration of Scan to BIM workflow*

The transition from raw geomatics survey data to a structured 3D information model requires a systematic Scan to BIM methodology, a foundational process in the domain of Heritage Building Information Modeling (HBIM) [14]. The established workflow acts as a sequential pipeline to convert the physical, as built conditions of the masonry bridge into a parametric digital representation. While subsequent sections detail the specific modeling techniques and geometric challenges encountered, the overall process is structured into four primary stages:

1. Data acquisition and post processing: The initial phase comprised the field acquisition of spatial data through the integration of Terrestrial Laser Scanning (TLS) and Unmanned Aerial Vehicle (UAV) photogrammetry. The raw data was processed using Leica Cyclone REGISTER 360 PLUS for the active laser scans and Agisoft Metashape for the passive aerial imagery and other datasets of MMS and leveling data. These datasets were subsequently merged and filtered within CloudCompare to produce a unified, high density point cloud, which was exported as a standardized .e57 file. This type of efficient data pipeline is critical for seamlessly translating point clouds into accurate models suitable for structural analysis and HBIM [15]

2. Point cloud indexing (Autodesk ReCap Pro): Because architectural Building Information Modeling (BIM) environments cannot efficiently handle raw .e57 point clouds directly, the combined dataset was imported into Autodesk ReCap Pro. This intermediate step

indexes and compresses the spatial data into an .rcp (ReCap Project) file, a format natively supported and optimized for integration within the Autodesk software ecosystem.

3. BIM (Autodesk Revit): The indexed .rcp point cloud was directly linked into Autodesk Revit to serve as the highly accurate 3D metric reference for the structural reconstruction. Operating within the main project environment (.rvt), the point cloud geometry guided the precise spatial placement and volumetric dimensioning of the bridge's architectural elements, a methodology highly effective for documenting historic stone bridges [16].

4. Parametric component creation (.rfa): Due to the unique morphological characteristics of historical masonry bridges, standard software libraries are typically insufficient for accurate geometric representation [14]. Consequently, custom parametric componentsAutodesk Revit Families (.rfa)were developed to model specific, repeating architectural and structural features. These custom families were then instantiated within the main Revit project to assemble the detailed digital twin.

3.1 Transition from BIM to Historical BIM (HBIM)

3.1.1 Building Information Modelling (BIM)

Building Information Modelling (BIM) has fundamentally changed how the construction industry plans, builds, and manages physical structures. Moving away from traditional 2D drawings, standard BIM creates a shared 3D digital model loaded with geometric and project data. However, standard BIM methodologies were originally designed with new construction in mind. The system relies heavily on standardized, orthogonal, and mass produced components such as uniform concrete walls, standard structural columns, and prefabricated mechanical systems. Because these elements are predictable and modular, BIM makes it easy for architects, structural engineers, and contractors to collaborate, detect spatial clashes early, and manage costs across a building's lifecycle.

While standard BIM assumes clean lines, right angles, and uniform geometries, real historical structures rarely follow these ideal rules. Heritage buildings are shaped by age, hand crafted architectural styles, past repairs, structural settling, and natural material decay.

To bridge this gap, researchers introduced Historic Building Information Modelling (HBIM) [14]. HBIM adapts traditional BIM processes specifically for built heritage, conservation, and structural maintenance. Rather than forcing standard rectangular shapes onto unique structures, HBIM shifts the focus toward recording reality capturing material loss, wall tilts, surface deformations, and non standard architectural features directly inside the model's data structure.

Autodesk Revit is built primarily as a tool for modern construction, which means it initially presents real challenges when applied to complex historical architecture. Despite these limitations, Revit has become a widely used software for HBIM & BIM based projects by pairing its database structure with two specialized workflows:

Direct point cloud Referencing (Scan to BIM): High precision laser scanning (TLS) and photogrammetry produce millions of spatial data points (point clouds) that are imported directly into Revit [17]. Instead of modeling ideal shapes, designers use this point cloud as a precise template to trace existing deformations, out of plumb walls, and real world spatial geometries [18].

Custom parametric families: To represent unique historical features such as gothic vaults, classical columns, arches, or ornate stone masonry practitioners build custom parametric families [19]. These custom components can be adjusted to fit irregular real world shapes while still holding critical non geometric data, such as material composition, structural health reports, and historical conservation records.

By combining reality capture data with custom parametric modeling, researchers and conservationists can effectively overcome Revit's modern design biases, making it a viable and practical tool for managing heritage assets over the long term.

3.2 Semantic breakdown of the bridge

3.2.1 Structural understanding and historical features

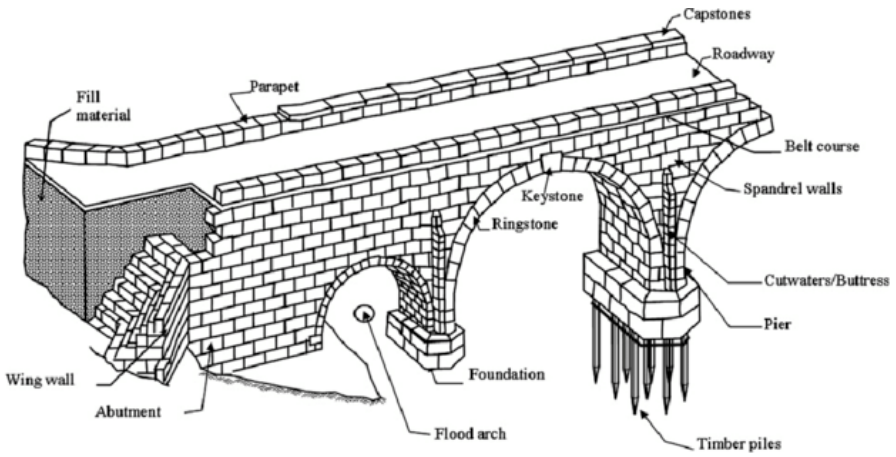


Figure 3.2: *Historical acquaduct bridge parts (Source: Scotland's Oldest Bridge)*

The Ponte Canale Cavour, located between Saluggia and Verolengo, carries the Cavour Canal across the Dora Baltea and represents an important example of nineteenth century hydraulic engineering. The bridge was designed by engineer Carlo Noè in 1854 and constructed between 1863 and 1866 as part of the Cavour Canal system. The structure is approximately 193.6 m long and consists of eight masonry piers, two abutments and nine lowered masonry arches. The arches were constructed entirely in brick, while the piers and abutments were protected with stone/granite facing.

Functionally, the structure operates as a water bridge (ponte canale), allowing the artificial irrigation canal to cross the Dora Baltea while maintaining the hydraulic level required for the downstream irrigation network. The choice of an elevated crossing rather than a siphon was related to the difference in elevation between the canal and the riverbed. Historical documentation indicates that the Cavour Canal at this location was situated substantially above the Dora Baltea riverbed, making the bridge canale solution appropriate.

From a structural perspective, the bridge can be understood as a system of interconnected masonry components comprising foundations, piers, arch barrels, spandrel walls/backfill and the hydraulic channel. This arrangement is consistent with the structural typology of historical masonry arch bridges, in which the arch barrel, backfill, spandrel walls, piers and foundations interact to transfer vertical and horizontal actions to the ground. [20]

Architecturally and structurally, historic masonry aqueduct bridges of this scale consist of interconnected structural sub assemblies designed to resist combined hydrostatic, dead, and environmental loads. Based on classical masonry bridge typologies, the key architectural elements comprise:

Substructure & Foundations: The masonry piers support the arches and transfer their vertical and horizontal reactions to the foundations. The piers and abutments are faced with dressed stone, providing additional protection within the river environment [20, 21]

Piers and cutwaters: The masonry piers support the arch superstructure and transmit the arch reactions and other loads to the foundations. The historical bridge documentation records eight brick masonry piers, with the piers and abutments faced with dressed stone [20, 22]

Superstructure arches and spandrel walls: The superstructure consists of nine lowered masonry arches constructed primarily from brick, with the arch barrels formed using temporary timber centering during construction. In traditional masonry arch bridges, the arch barrel, spandrel walls, and backfill work together to distribute loads toward the piers and foundations. [20, 23]

Belt course and aqueduct deck: The upper section carries the Cavour Canal and forms the hydraulic deck of the bridge. Historical documentation indicates that the canal bed was constructed in brickwork and finished with a waterproof plaster/render layer. [20]

3.2.2 Semantic segmentation driven by Structural Health Monitoring

In standard BIM, modeling is driven by standardized, parametric object libraries (e.g., uniform walls, slabs, columns). However, when applying Historic Building Information Modelling (HBIM) to heritage infrastructure, a masonry bridge cannot be captured or evaluated as a single, monolithic 3D solid geometry.

Treating the structure as a unified mass presents critical limitations: it obscures material variations, hides internal fill mechanics, and prevents targeted structural performance tracking. Therefore, a semantic breakdown is required. By decomposing the physical entity into discrete components, each element can carry unique geometric parameters, material decay attributes, and chronological conservation records.

Furthermore, this segmentation strategy is directly informed by the placement of structural health monitoring (SHM) sensor networks. Sensor deployment targets localized structural behaviors rather than global displacements. To pair real time monitoring data effectively with the HBIM model, the digital geometry must be divided into discrete semantic elements that correspond precisely to the spatial domain of each sensor.

3.2.3 Division strategy and sensor driven modelling logic

To establish a semantically rich and sensor integrated HBIM environment for the Ponte Canale Cavour, the bridge was systematically partitioned into logical sub assemblies.

Foundation substructure: Modeled as dedicated subterranean structural volumes representing the submerged pier footings and deep foundation beds below the impetuous bed of the Dora Baltea.

Piers and cutwaters(Inclinometers & Crack Meters): The piers were isolated from the arches and deck, modeled as independent vertical elements incorporating their specific hydrodynamic cutwaters. This segment hosts inclinometers to measure tilt and rotational movement caused by foundation settlement or river action, along with crack meters attached to monitor active masonry joints and structural fractures.

Spandrel Walls and arch systems (Crack Meters): The vertical masonry spandrel walls retaining internal fill were segregated from the main arch vaults and voussoirs. Crack meters mounted across the spandrel walls track out of plane movements and stress induced cracking

caused by internal fill thrust or thermal cycles.

Belt course: Modeled as a distinct horizontal profile family separating the lower structural masonry walls from the upper hydraulic conduit. Dividing the belt course isolates decorative architectural trim elements from primary load bearing entities.

Aqueduct deck and parapets: The topmost hydraulic channel carrying the canal water was modeled separately from the supporting substructure. This element holds key semantic properties related to hydraulic capacity, waterproofing membranes, capstones, and parapet integrity. the idea is to have a water levelling sensor in the future.

3.3 From .e57 to Autodesk ReCap Pro

Raw terrestrial laser scan exports most commonly formatted as .e57 spatial datasets represent high fidelity, unstructured spatial captures of as built environments. While an .e57 file provides a vendor neutral container for spatial coordinates (x, y, z), intensity values, and RGB color channels, its raw structure poses a significant computational bottleneck for Building Information Modeling (BIM) authoring platforms.

Software environments such as Autodesk Revit are engineered to handle object oriented parametric geometries rather than billions of unindexed vector points. Direct ingestion of uncompressed, non spatial index .e57 files into BIM tools leads to rapid RAM exhaustion, severe frame rate degradation during scene navigation, and frequent application crashes

The fundamental incompatibility stems from the lack of a hierarchical spatial index in raw formats, forcing the hardware to re evaluate the full spatial volume continuously rather than displaying localized regions dynamically.

To transition from spatial capture to parametric reconstruction, Autodesk ReCap Pro serves as the critical translation and optimization engine between geomatics engineering and BIM authoring. ReCap decodes the raw point records within the .e57 container, applies internal indexing, and converts the unstructured data into a proprietary, spatial octree structure represented by .rcp project headers linked to indexed .rcs scan files. (Figure: 3.3)



Figure 3.3: *Illustration of .e57 file to .rcp file workflow*

During this conversion, spatial data undergoes compression and spatial partition indexing, allowing downstream BIM software to perform view frustum culling and Level of Detail (LOD) streaming. Consequently, the hardware only loads point clusters within the user's active viewport and zoom proximity, maintaining computational efficiency across multi gigabyte point clouds.

Beyond format conversion, ReCap Pro functions as a spatial curation environment to prepare the point cloud for accurate parametric modeling. Key prep steps include:

Noise culling & artifact elimination: Removing atmospheric distortion, moving objects, and transient reflections that generate ghost points, preventing geometric misalignments during wall and structural tracing. (Figure: 3.4)

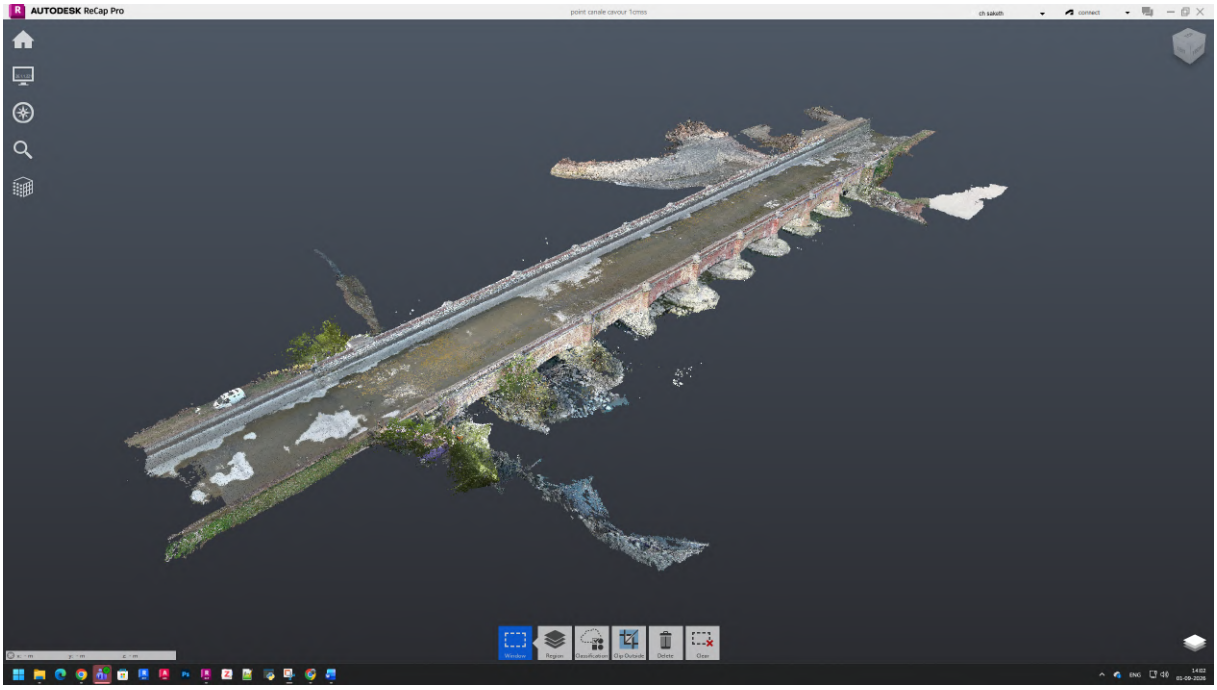


Figure 3.4: *Noise cleaning*

Volumetric Clipping: Creating predefined limit boxes and elevation clips to isolate relevant structural regions, allowing BIM modelers to focus on specific building sections while conserving active display memory. (Figure: 3.5)

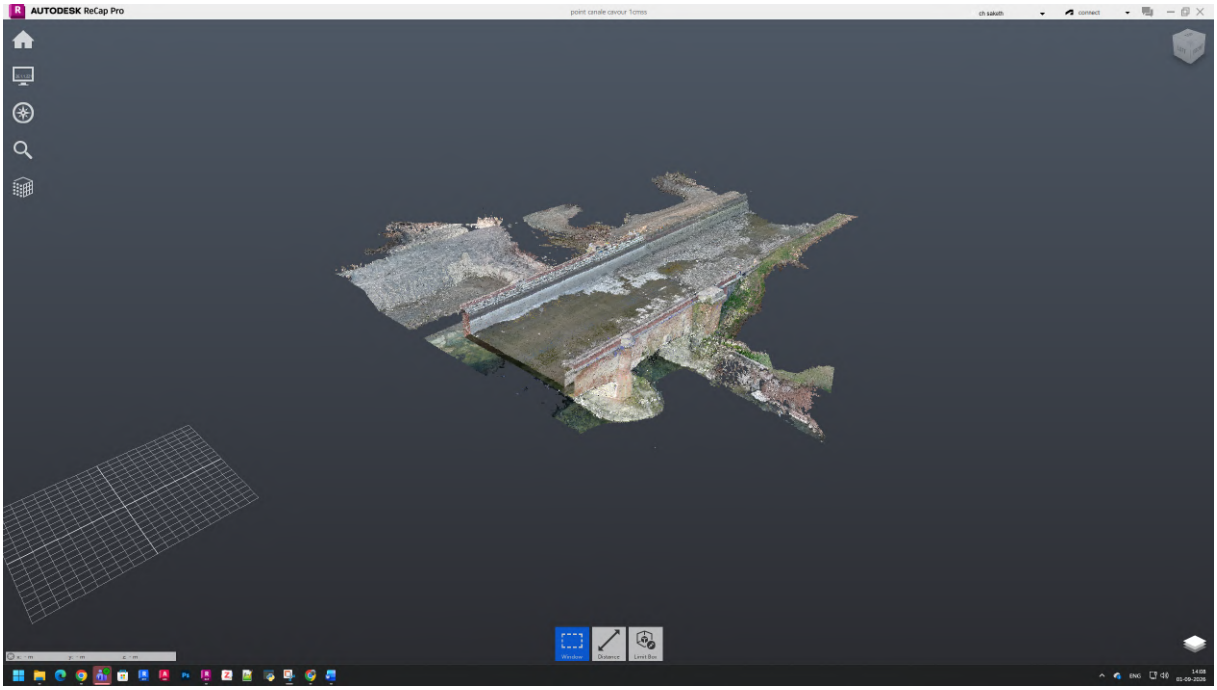


Figure 3.5: *volumetric clipping*

Exporting a unfied scan file: Consolidating individual setup locations into a single, unified .rcs scan file. This process merges individual spatial nodes into a single, continuous coordinate framework while stripping out redundant overlap. While unifying the point cloud converts structured scan setups into an unstructured format there by dropping individual scanner origin nodes and panoramic RealView setups it significantly streamlines file management, improves loading times, and optimizes memory handling when linked into Autodesk Revit. (Figure: 3.6)

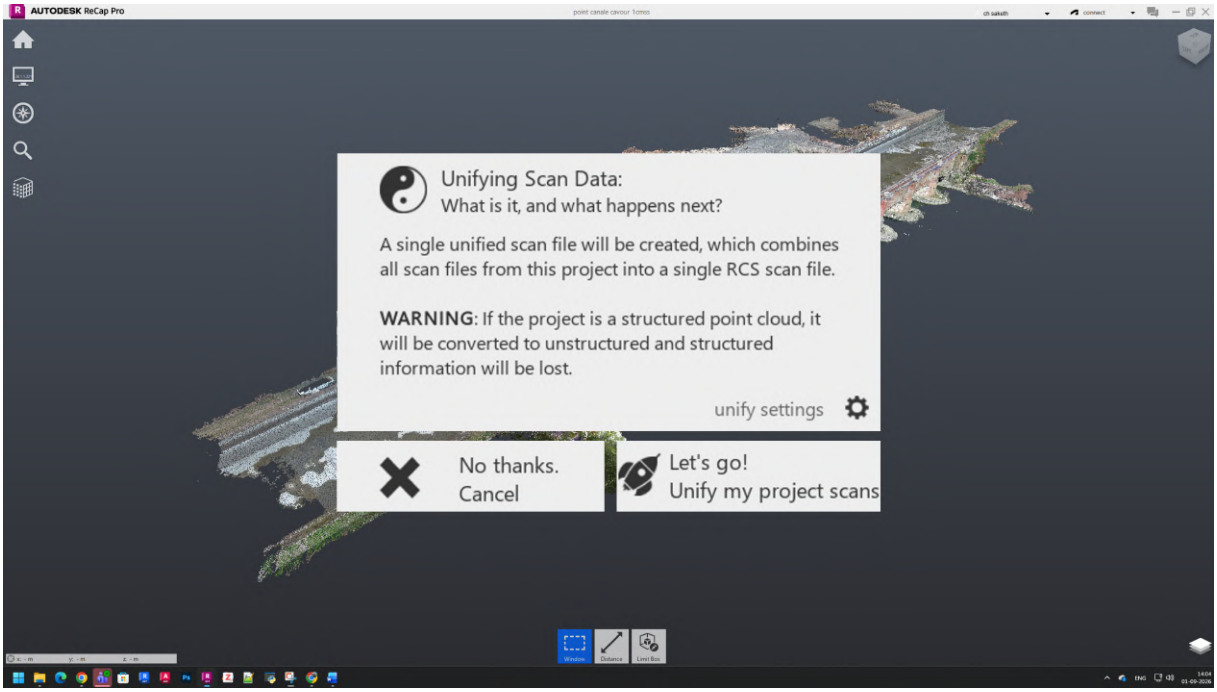


Figure 3.6: *Unifying scan data into single scan*

Volumetric size reduction & spatial decimation: Downsampling hyper dense point clusters, stripping unnecessary color channels or unused setup scans, and purging redundant overlapping scan data to substantially reduce the overall file footprint without sacrificing structural precision. (Figure: 3.7)

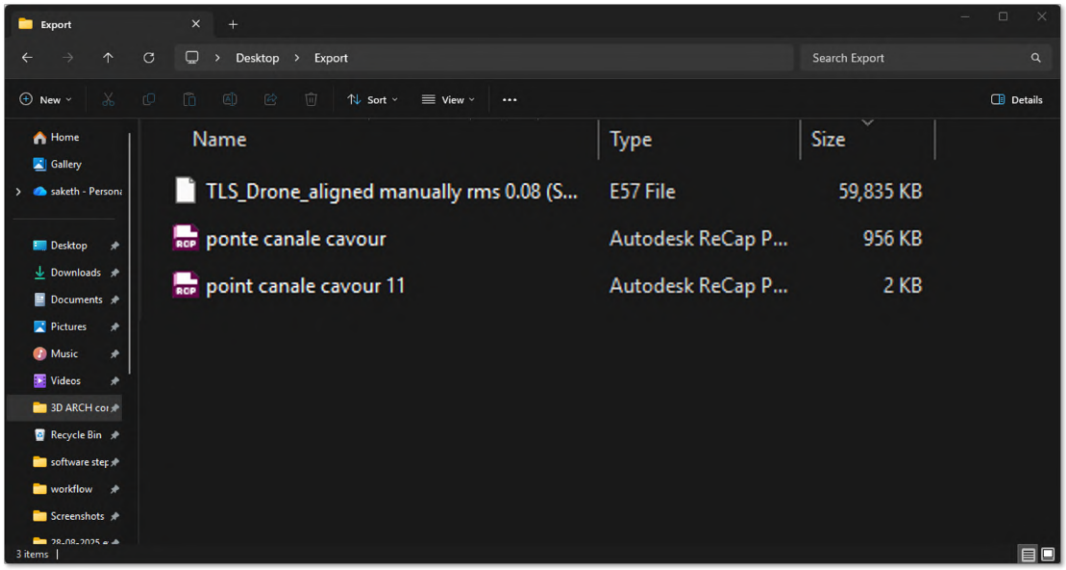


Figure 3.7: *Recap volumetric clipping*

3.4 Scan to BIM execution using Autodesk Revit

3.4.1 Setting the environment



Figure 3.8: *Autodesk Revit 2024*

In this study, It is more of point cloud based BIM and VPL workflows, Autodesk Revit 2024 can remain a practical production choice because stability and workflow reliability are often more important than adopting the newest release. Although Autodesk Revit 2025 and 2026 provide ongoing improvements, Autodesk continues to identify performance issues associated with large point cloud datasets, indicating that version upgrades do not automatically guarantee better stability. FARO As Built can provide a more specialised and controlled approach to scan to BIM, with dedicated tools for extracting and modelling BIM elements directly from point clouds. Therefore, for VPL applications, the most suitable version should be selected based on tested stability, point cloud performance, software compatibility, and repeatability rather than simply choosing the latest version.

To transition from processed spatial data to parametric reconstruction, the BIM authoring environment must be systematically configured for this.

The setup procedure establishes metric modeling precision, integrates the point cloud dataset, aligns spatial coordinate benchmarks, and sets the structural levels and reference grids that govern object placement.

Metric Unit System ConfigurationInitial project initiation within Autodesk Revit utilizes the standard Architectural Template. (Figure: 3.9)

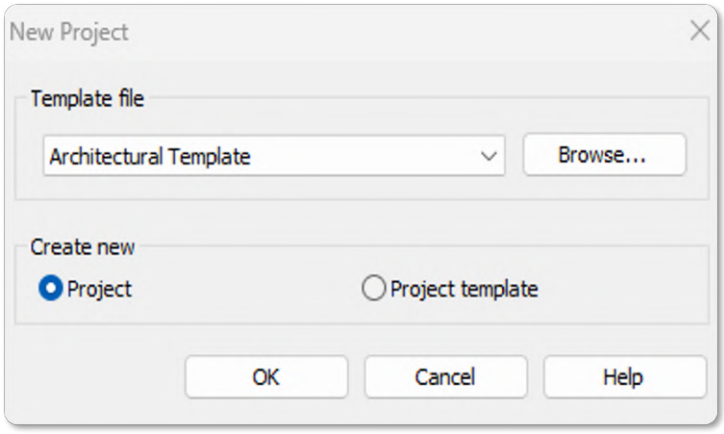


Figure 3.9: *New project in metric architecture model template*

To eliminate rounding tolerances during feature extraction, project units are explicitly set to millimeters (mm). Millimeter level precision is essential in Scan to BIM workflows to ensure accurate alignment between extracted parametric geometrysuch as masonry arches, piers, and abutments and high density point cloud cross sections. (Figure: 3.10)

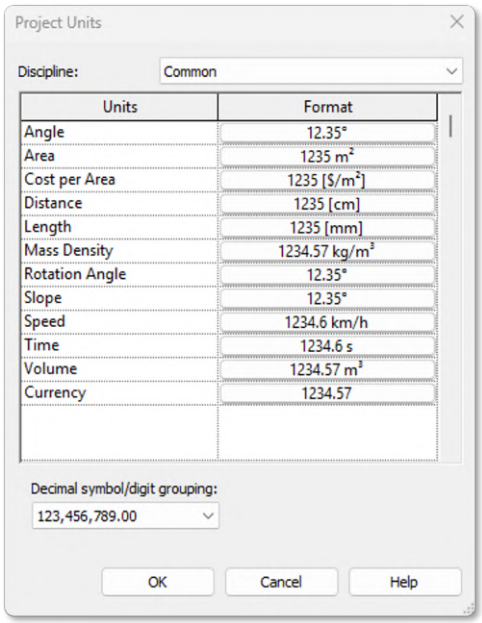


Figure 3.10: *Metric project unit settings*

Point Cloud linking and ingestion

The indexed .rcp project file is linked into Revit using the Point Cloud insertion tool located on the Insert ribbon tab, selecting the Auto Center to Center positioning method. Ingesting

the point cloud via external linking rather than direct embedding ensures lightweight memory usage and prevents file bloating. Upon import, the overall dataset representing the bridge structure and surrounding topography is placed centered within the Revit workspace (Figure: 3.11), establishing the spatial boundary for subsequent modeling operations.

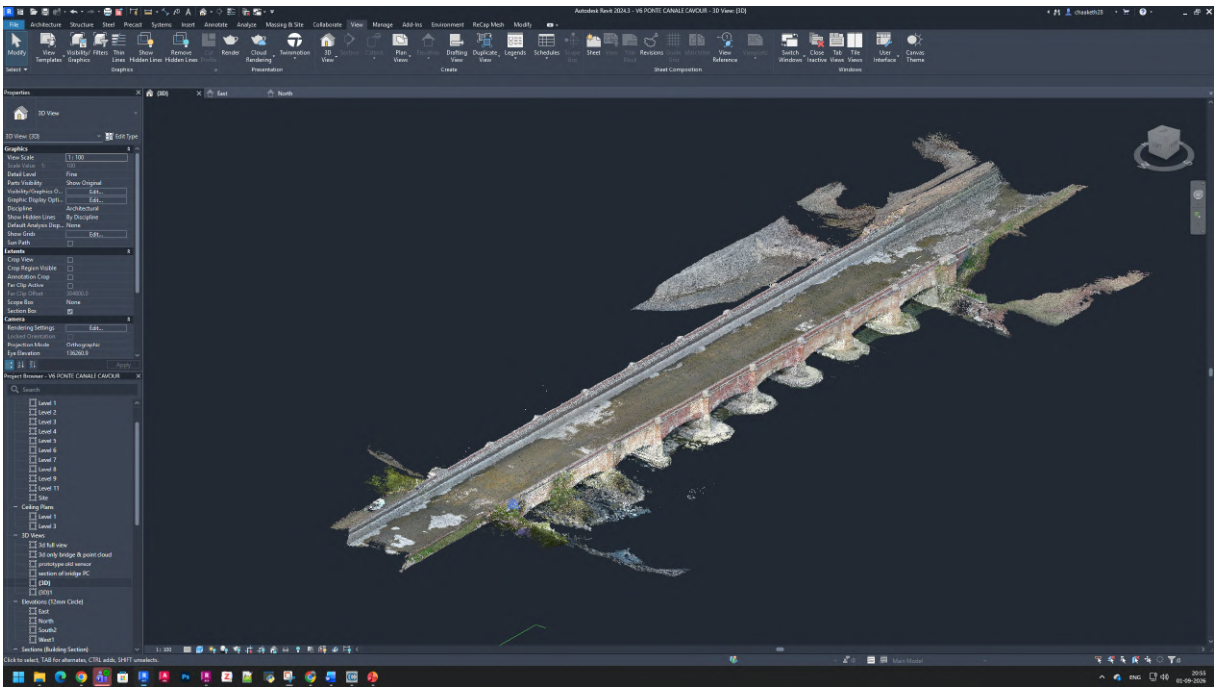


Figure 3.11: *Point cloud import*

Coordinate System Alignment and Project Base Point Setup To align Revit's internal modeling environment with absolute spatial coordinates, external point cloud inspection software (CloudCompare) is utilized alongside Revit (Figure: 3.12). Precise coordinate values (X, Y, Z and orientation angle) extracted from a specific, identifiable structural feature point within the point cloud are referenced directly.

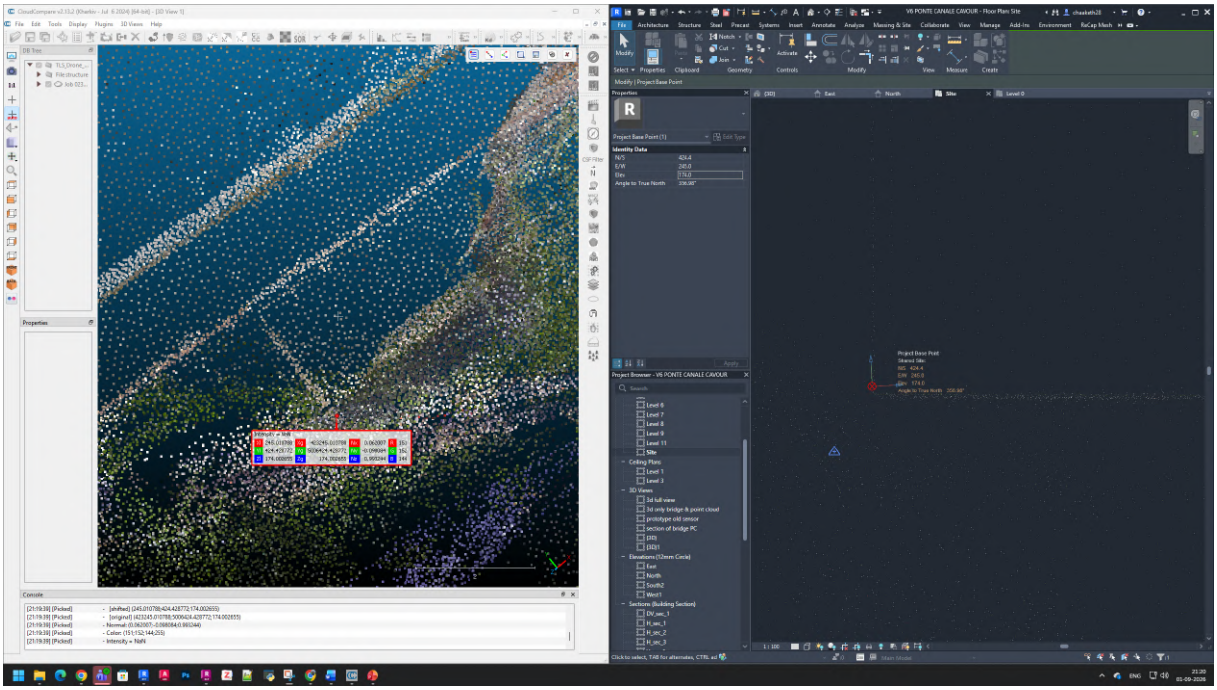


Figure 3.12: *extracting base point from cloud compare to Autodesk Revit*

The Project Base Point in Revit is subsequently modified to match these exact spatial coordinates (setting North/South, East/West, Elevation, and Angle to True North) (Figure: 3.13). This ensures that the BIM model maintains accurate georeferenced spatial alignment without risking graphical clipping issues associated with large coordinate origins.

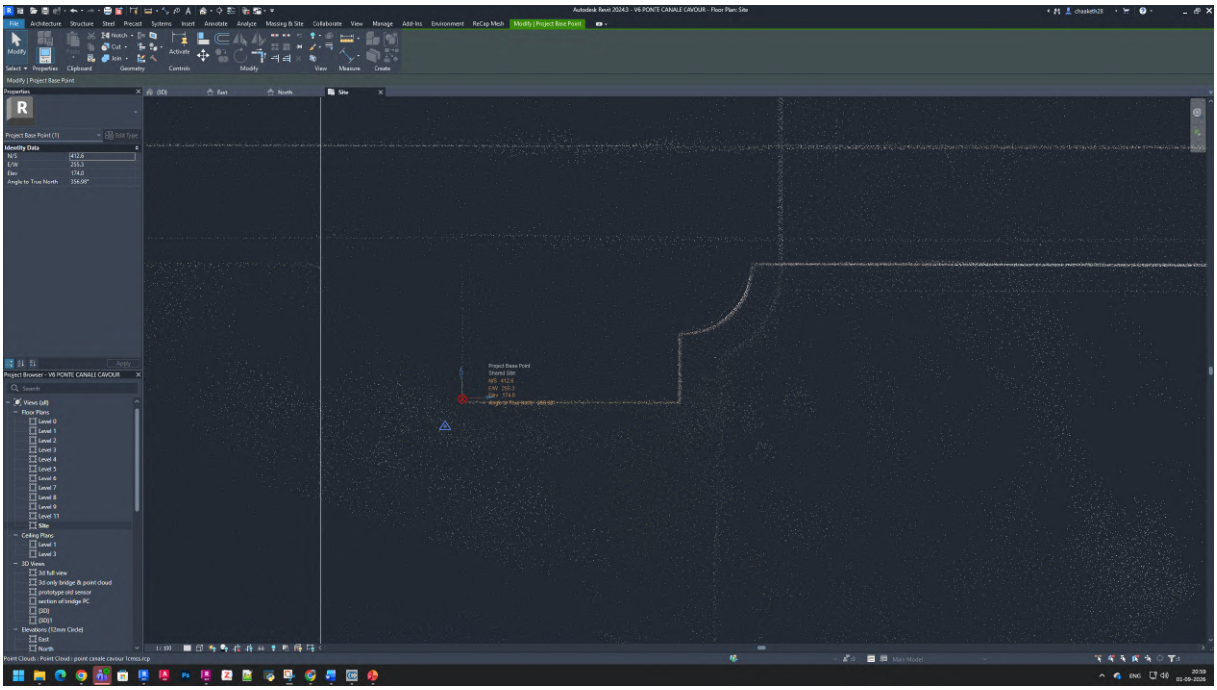


Figure 3.13: *Setting up base point coordinates*

Establishing reference levels

Vertical stratification of the bridge geometry is achieved in elevation views (Figure: 3.14). The initial idea of Reference level is to start from the bottom of the Pier calling it as Datum levels

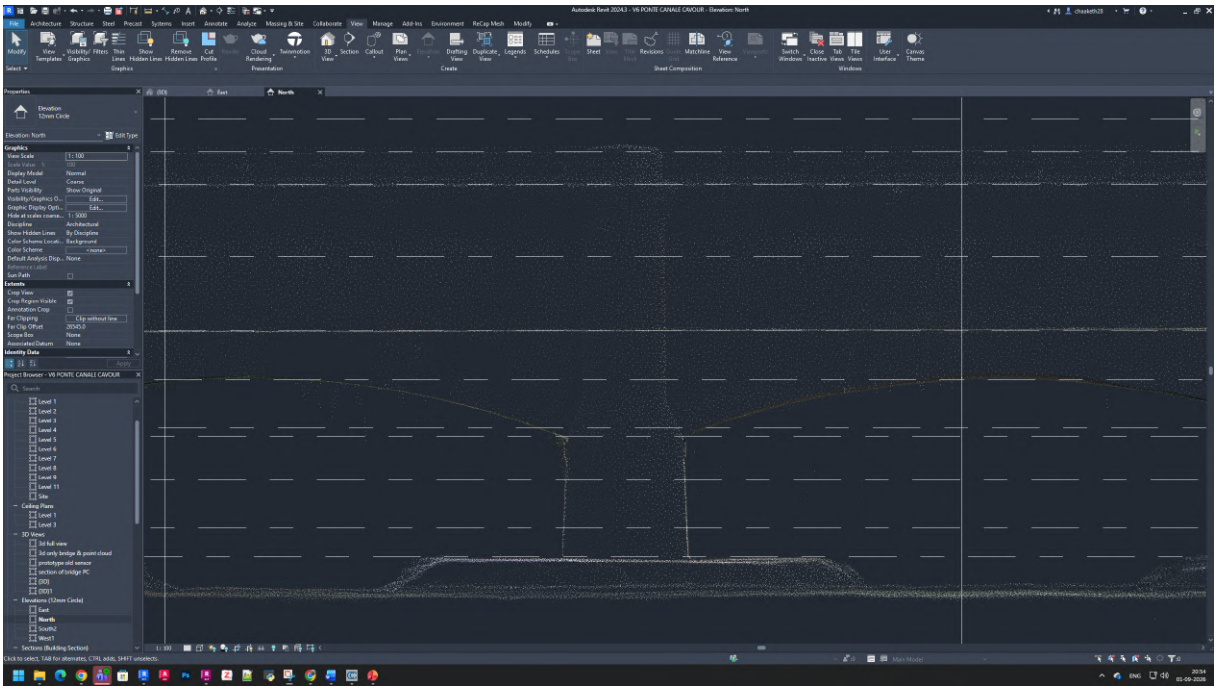


Figure 3.14: *Elevation reference levels from north elevation of point cloud*

Datum levels are established across key structural elevationsincluding the foundations, arch springings, pier caps, deck, and parapet lines (Figure: 3.15). A total of 12 vertical levels (Level 0 through Level 11) are mapped directly to physical features visible in the point cloud cross section. These levels serve as constrains for wall, column, and structural framing parametric family creation.

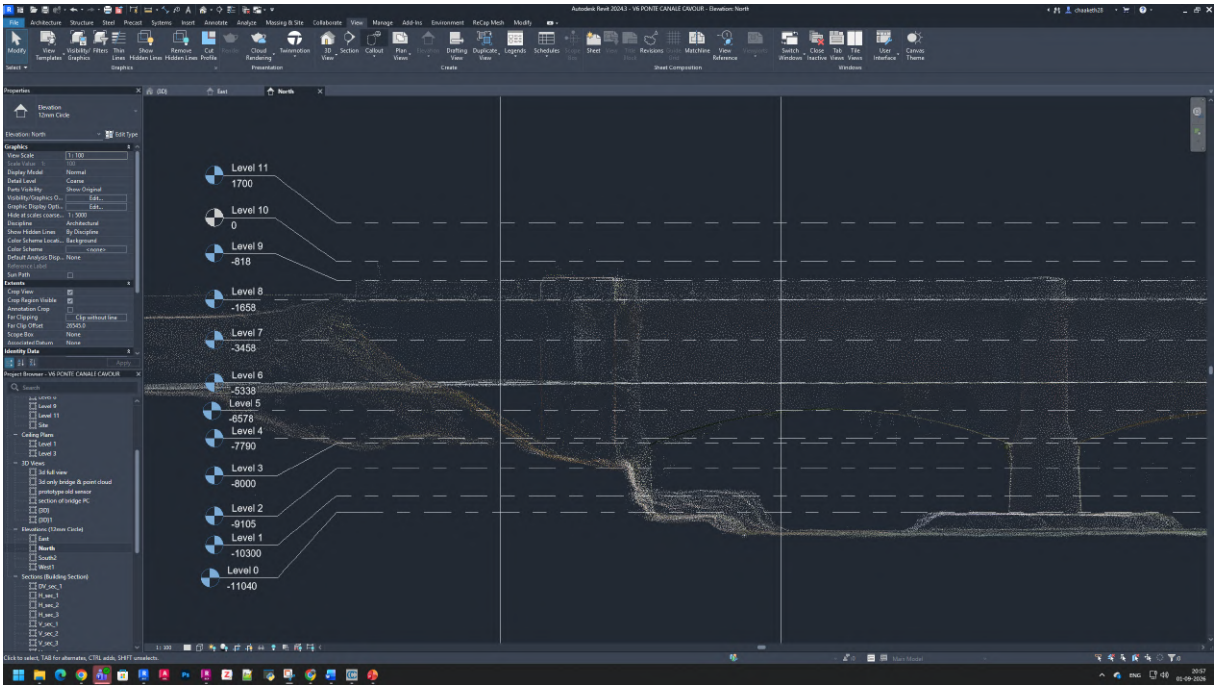


Figure 3.15: *Elevation reference levels from north elevation of point cloud*

Section placement

To facilitate high fidelity modeling of longitudinal and transverse elements, building cross sections are systematically positioned in the intervals between each pair of bridge piers (Figure: 3.16).

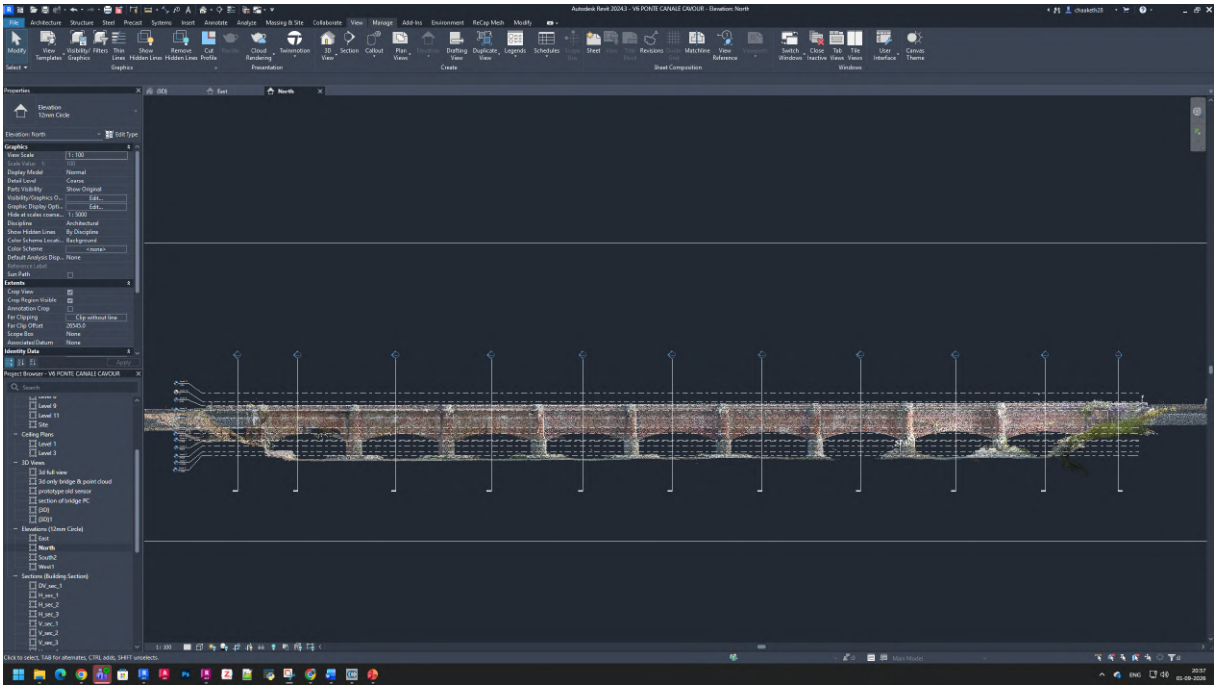


Figure 3.16: *Elevation reference levels from north elevation of point cloud*

This deliberate placement strategy isolates individual arch spans, providing localized 2D

profile views without point cloud occlusion from adjacent thick masonry piers. Establishing cross sectional cut planes midway through each span proved critical for modeling the bridge deck profile, road crown camber, and parapet walls. Because historical bridge structures exhibit non uniform sag, weathering deformation, and varying deck inclinations across spans, these localized sections allowed for precise tracing of slope variations, wall thicknesses, and edge profiles along the entire length of the structure.

3.4.2 Executing the divisions

Translating the conceptual division strategy into a functional HBIM model required selecting appropriate Autodesk Revit tools tailored to the specific geometry, complexity, and repeating nature of each structural assembly.

Aqueduct deck & channel modeling

To capture accurately and precisely the unique longitudinal profile and specialized cross section of the hydraulicc canal deck the modeling process executed directly with the aquaduuct deck using Revit's In place Mass tool rather than standard system families. As the  In place model has more flexibility in creation of accurate shape following both the point cloud and the longitudinal natural construction slope seen from the horizontal section of the bridge (Figure: 3.17)

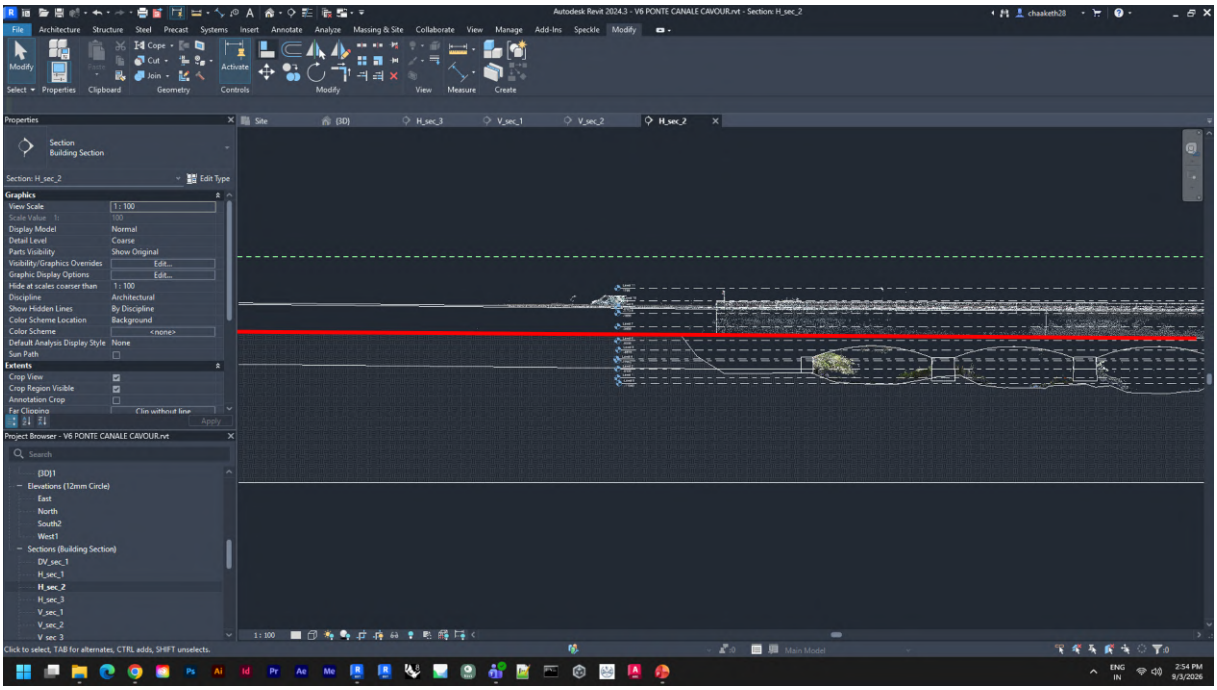


Figure 3.17: *Green dashed line perfect horizontal line following red line is with slope of the deck from longitudinal section*

Work plane setup As of historical masonry structures rarely align to pure orthogonal axes, a temporary reference cubes are created to the every longitudinal section to estabish a precise reference plan in the workspace. (Figure: 3.18)

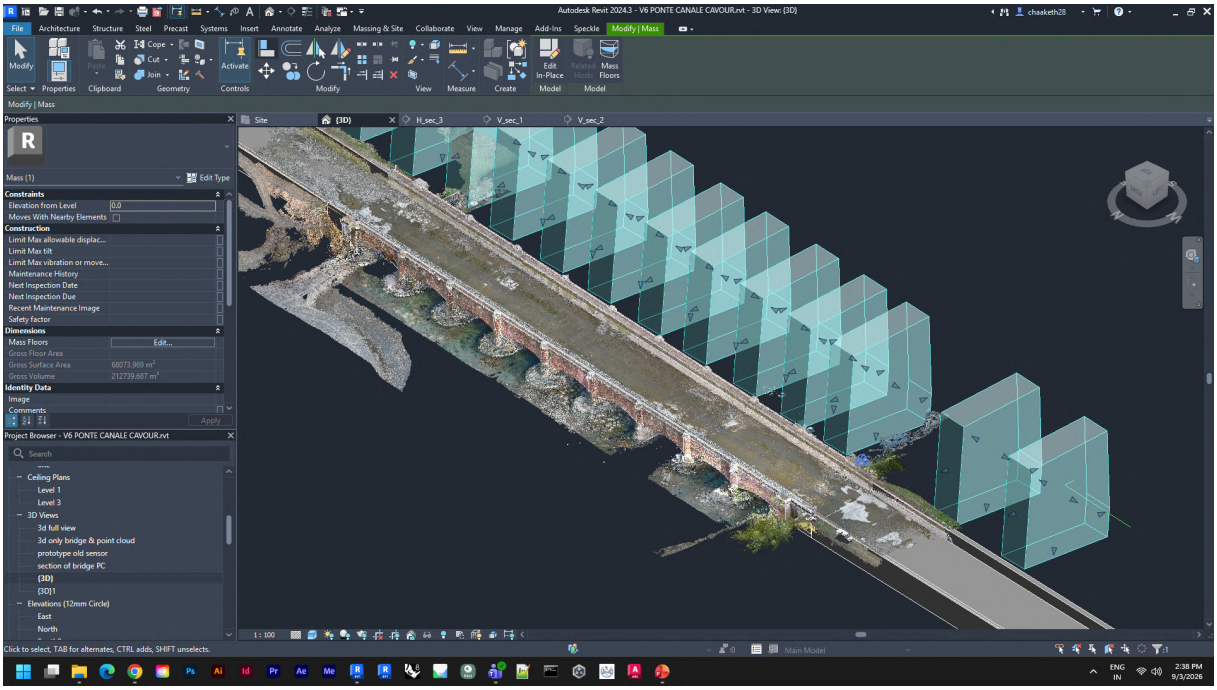


Figure 3.18: *Temporary cubes are created to setup working plane*

Sectional extrusion Using a designated each individual cross section view aligned to this reference plane, by In place mass tool the exact hydraulic cross section of the canal bed was sketched and extruded along the bridge alignment. (Figure: 3.19) for this set a plane tool is used under  set work plane tool. Created a  solid form of the deck This approach ensures that semantic properties such as profile of deck, slope of canal bed and future sensor integration zones (e.g, water levelling sensors) can be assigned to a distinct geometric volume.

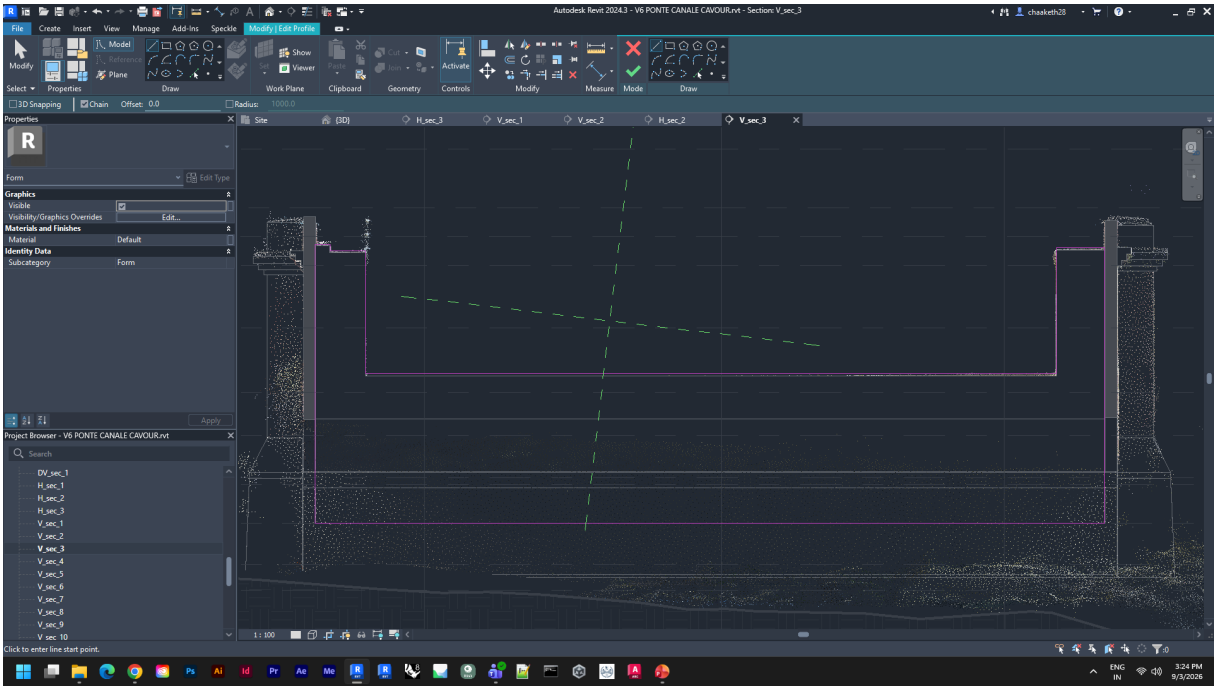


Figure 3.19: *Draw tool in cross section following point cloud*

Here, this deck profile is created for each cross section view as there is slope of the deck, selecting all the profiles created following the point cloud then creating solid form gives as the most accurate model of the canale bed.

Spandrel wall construction:

Following the generation of the primary deck mass, the vertical masonry spandrel walls retaining the internal fill were generated using system wall elements:

System wall integration The  architecture wall tool was utilized to model the spandrel walls with a uniform thickness of 350mm (35cm).

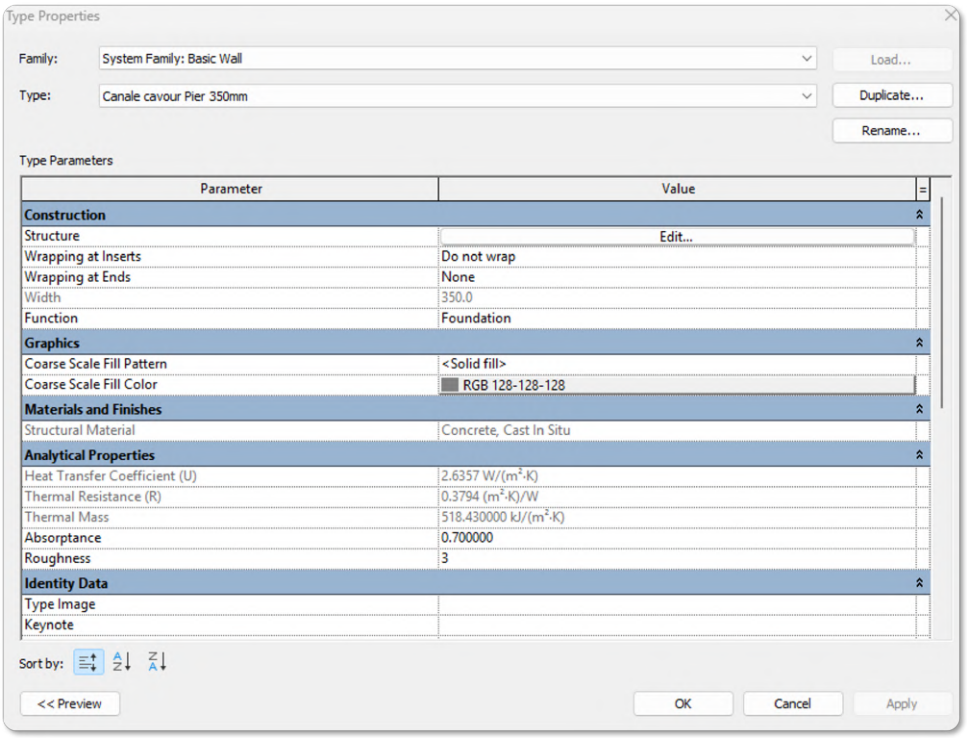


Figure 3.20: *Architecture wall 350mm wall type properties*

Structural segragation by modeling the spandrel walls as distinct system wall components attached to the deck and arch geometries, the walls are isolated from both primary load bearing arches and the main aqueduuct channel.(Figuure: 3.21)

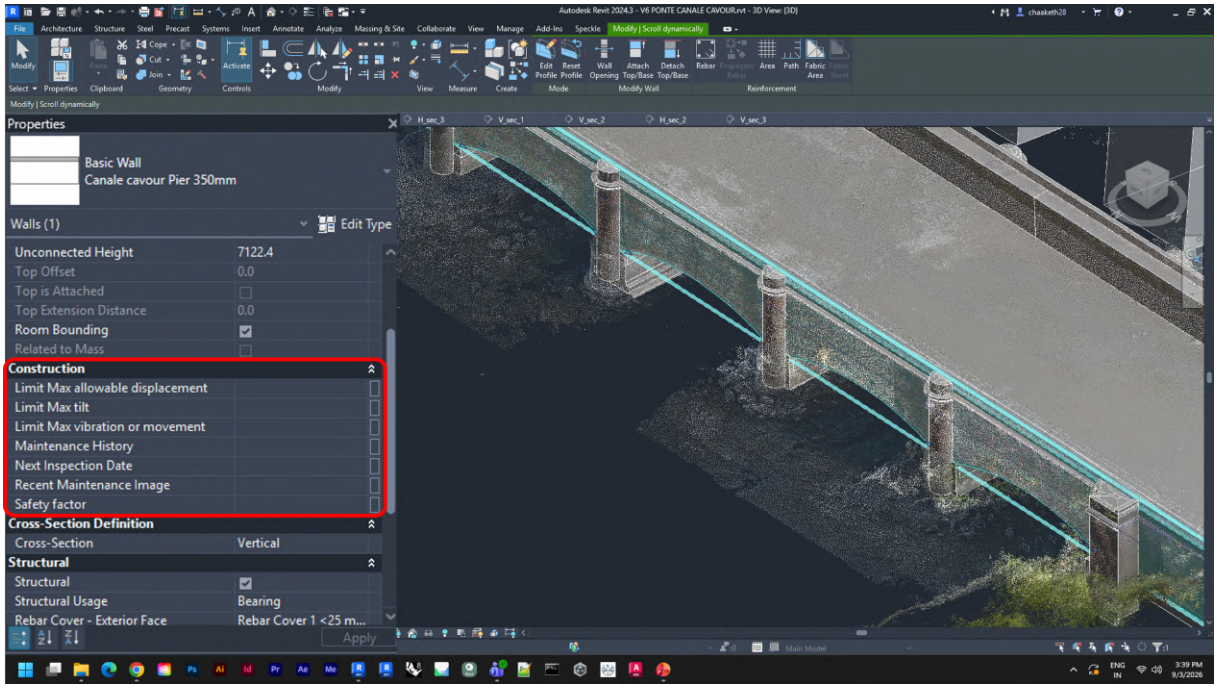


Figure 3.21: *Temporary cubes are created from section view*

This structural segration enables the model to host dedicated monitoring parameters, specifically surface mounted crack meters and temperature tracking out of plane movements and thermal expansion stress.

Piers, cutwaters, and caps

Initially, each pier assembly was conceptualized as a single monolithic element using Revit's Loadable Family (.rfa) tool. Using Generic model .rft revit family template.

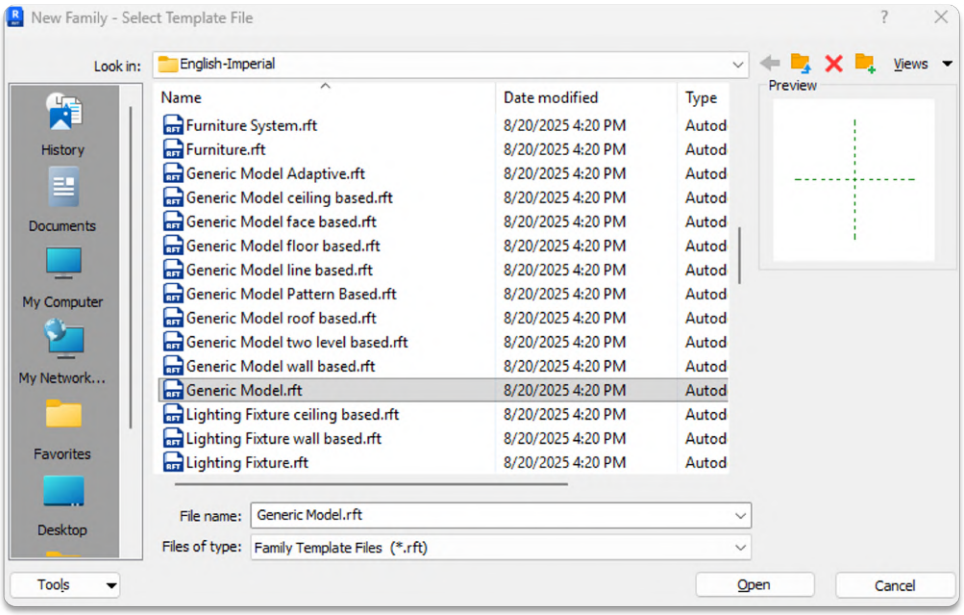


Figure 3.22: *Right Generic family template*

However, to support structural health monitoring (SHM) specifically tracking differential settlement via inclinometers and joint displacement using crack metersthe substructure was re evaluated and segregated into distinct geometric sub assemblies using Revit's Loadable Family (.rfa)

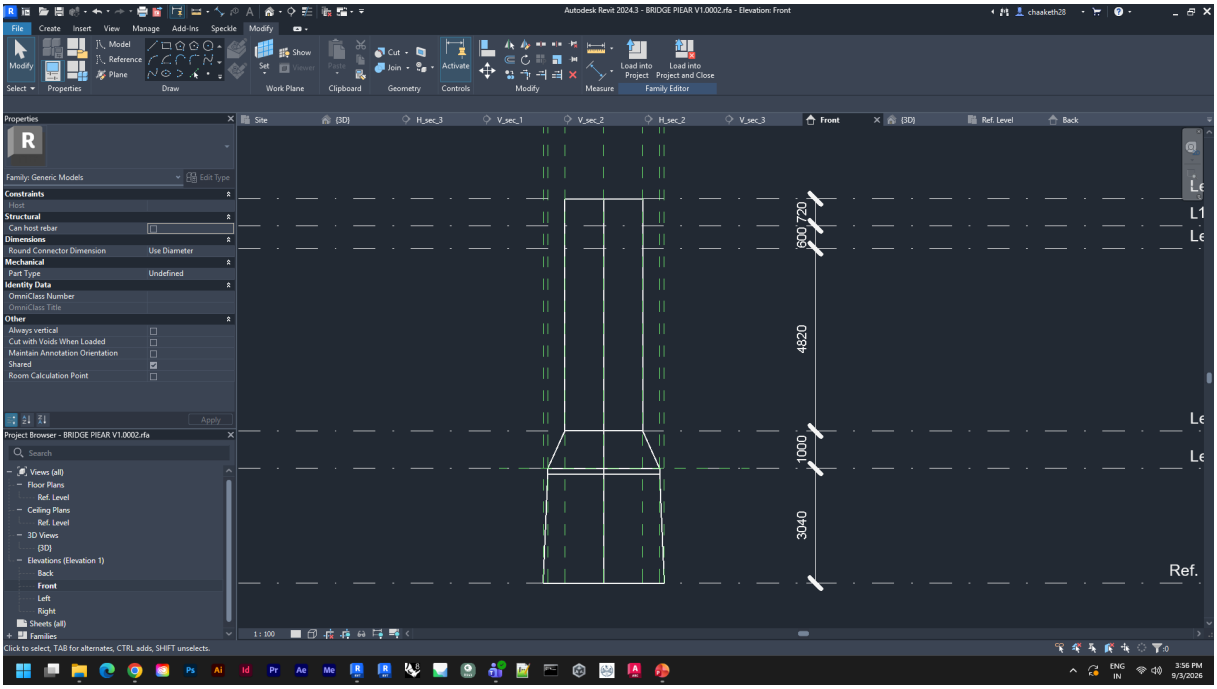


Figure 3.23: *Generic family template*

Cutwaters and submerged base The hydrodynamic cutwater base segment was isolated into a dedicated loadable family (.rfa).

Because these suubmerged elements share standardized hydrodynamic profiles across the bridge length, creating a parametric family allowed for efficient replication throughout the project while preserving specific host properties for foundation monitoring sensors. (Figure: 3.24)

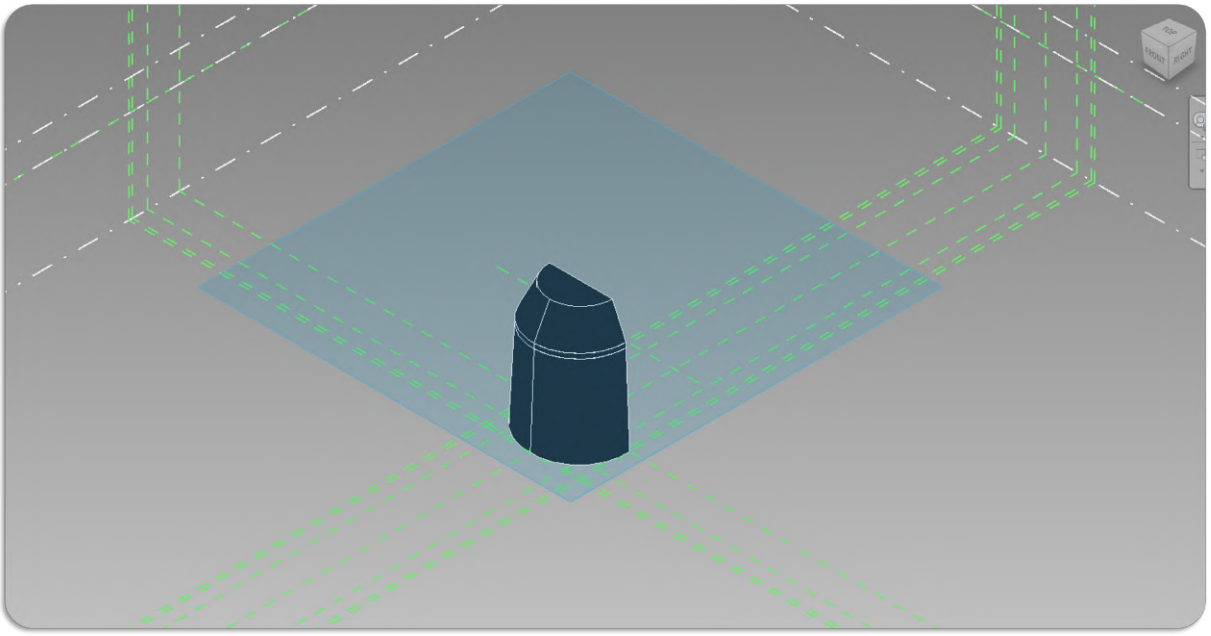


Figure 3.24: *Isolated cutwater element as loadable .rfa*

Main pier body To account for subtle geometric deviations and irregular masonry alignments present in the point cloud data, the main vertical pier body was modeled using the Architecture wall tool with the curve draw function.

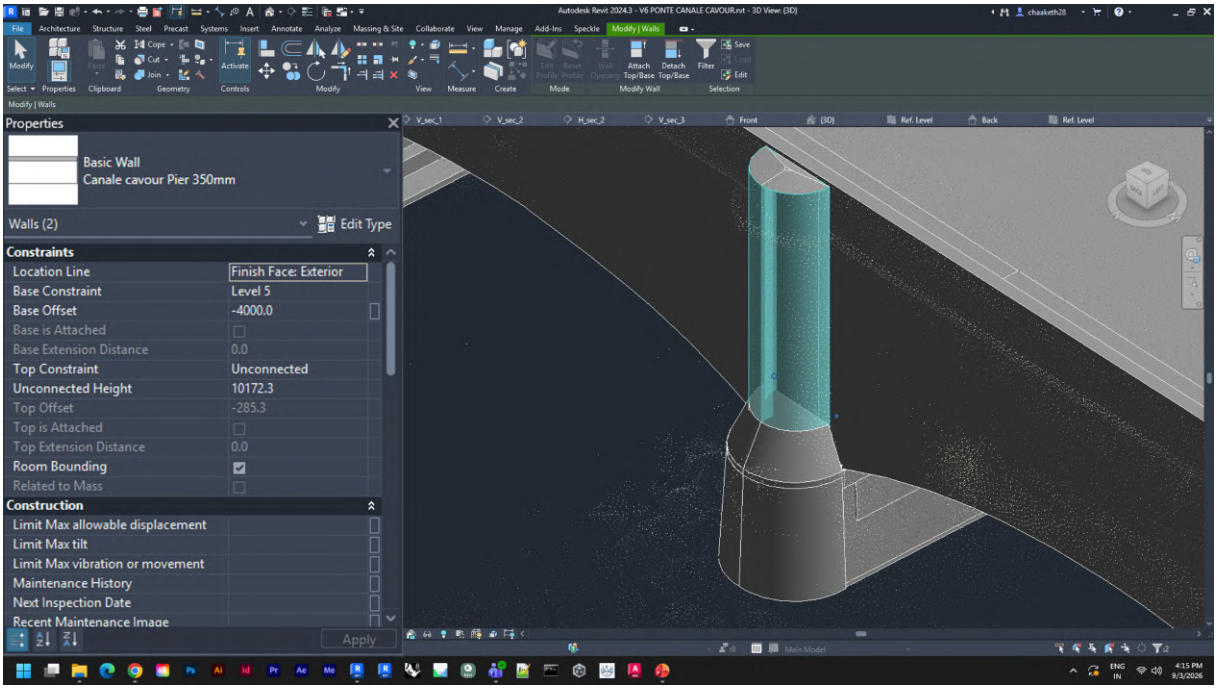


Figure 3.25: *Main Pier Body by architecture wall element*

Tracing directly over the point cloud slice ensuured high spatial accuracy. The setup of muultiple custome reerence levels provided the vertical control needed to align the tapering pier geometry accurately between the base cutwater and the upper arch springings. (Figure: 3.25)

Pier caps The transitional pier caps, which support the arch vaults and feature distinct profile geometries, were modeled using Model In place massing tools, This technique offered the flexibility needed ot captuure complex transitional curves and decorative trim details that could not be easily standardized within a generic wall or family profile.

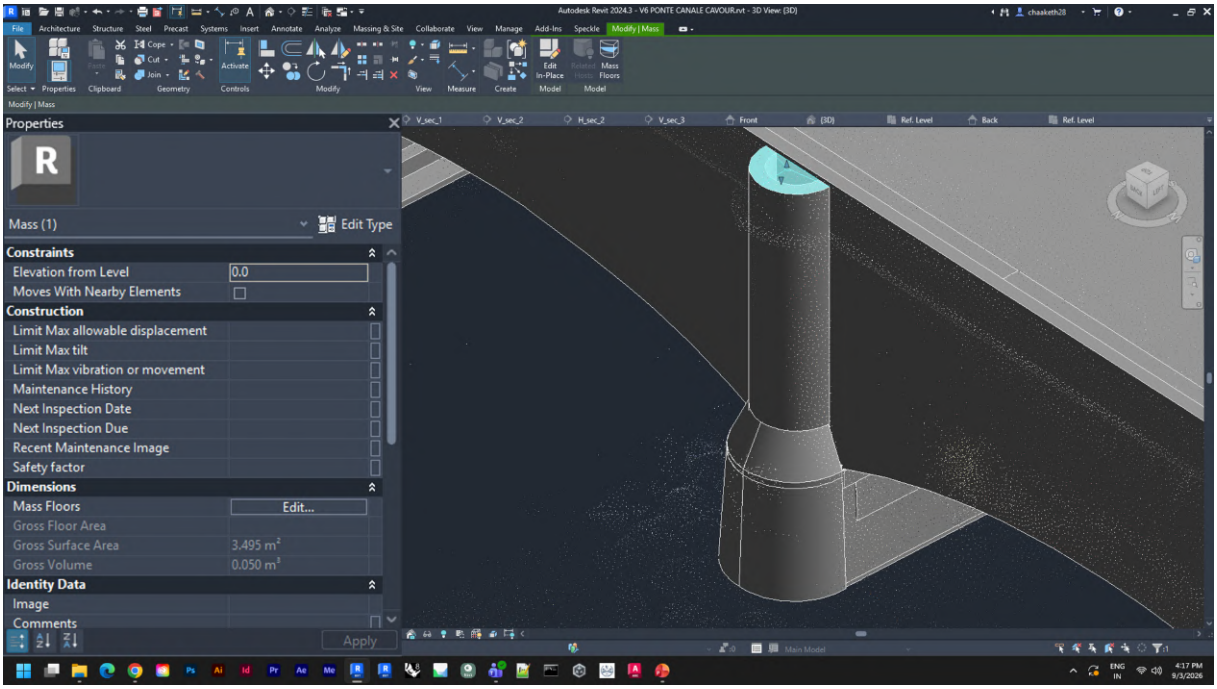


Figure 3.26: *Pier caps modelled as In-place mass*

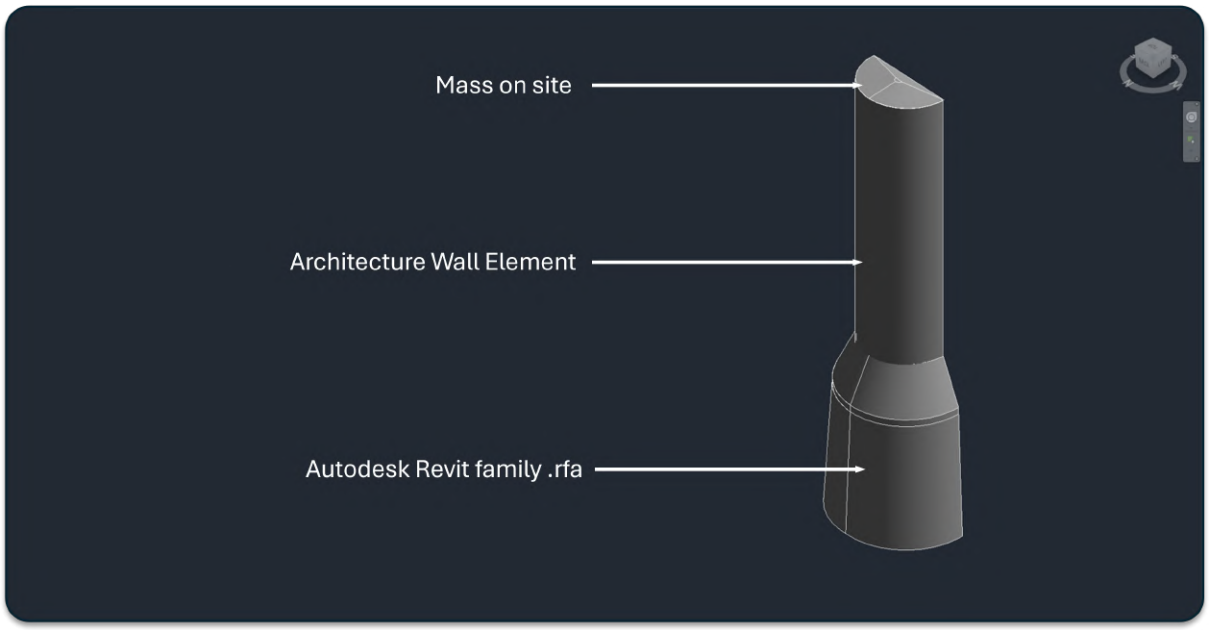


Figure 3.27: *Pier modelling divisions*

Superstructure arches and spandrel walls

Achieving the accurate curvature of the historical masonry arches following the point cloud. reuires custom modelling techniques in Autodesk Revit to reconcile non ideal field conditions such as longitudinal slope and organic deformations.

Spandrel wall arch profile The arch profile cutting into the vertical spandrel wall was generated using the  Edit Profile tool within the Architecture Wall editor. Two primary geometric constraints were considered during sketching:

Longitudinal Slope Allowance: The upper boundaries of the spandrel walls were not strictly horizontal relative to global reference levels, but sloped slightly from right to left to follow the natural incline of the canal deck.(Figure: **??**)



Figure 3.28: *Spandrel wall slope*

Point cloud curve matching: Using a dedicated section view, the lower boundary sketch of the wall profile was traced directly against the point cloud slice, capturing the true, slightly irregular shape of the supporting masonry arch vaults.

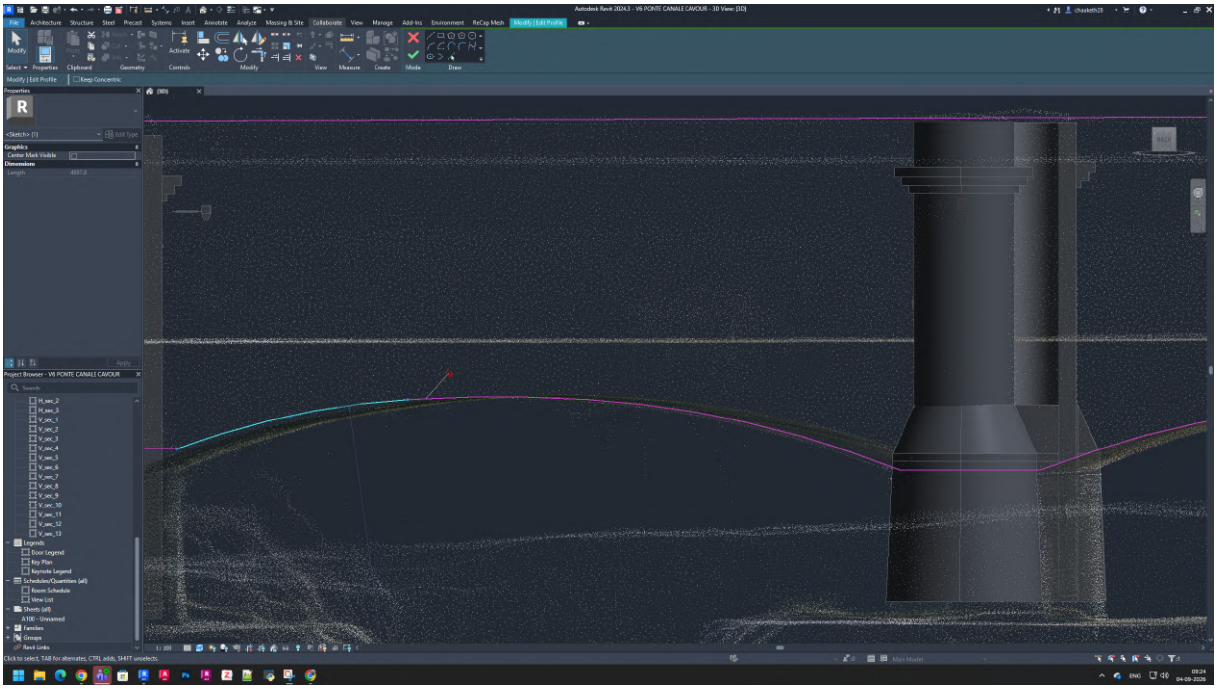


Figure 3.29: *Spandrel wall & arch curve following point cloud*

Aqueduct deck bottom arch: To carve out the vaulted undersides beneath the canal deck, In Place Mass Voids were introduced:

A continuous 3D void form was sketched using the section profile traced from the point cloud data.

The void mass was then extruded horizontally along the bridge's longitudinal axis and executed using the Cut Geometry tool to subtract material from the primary deck solid.

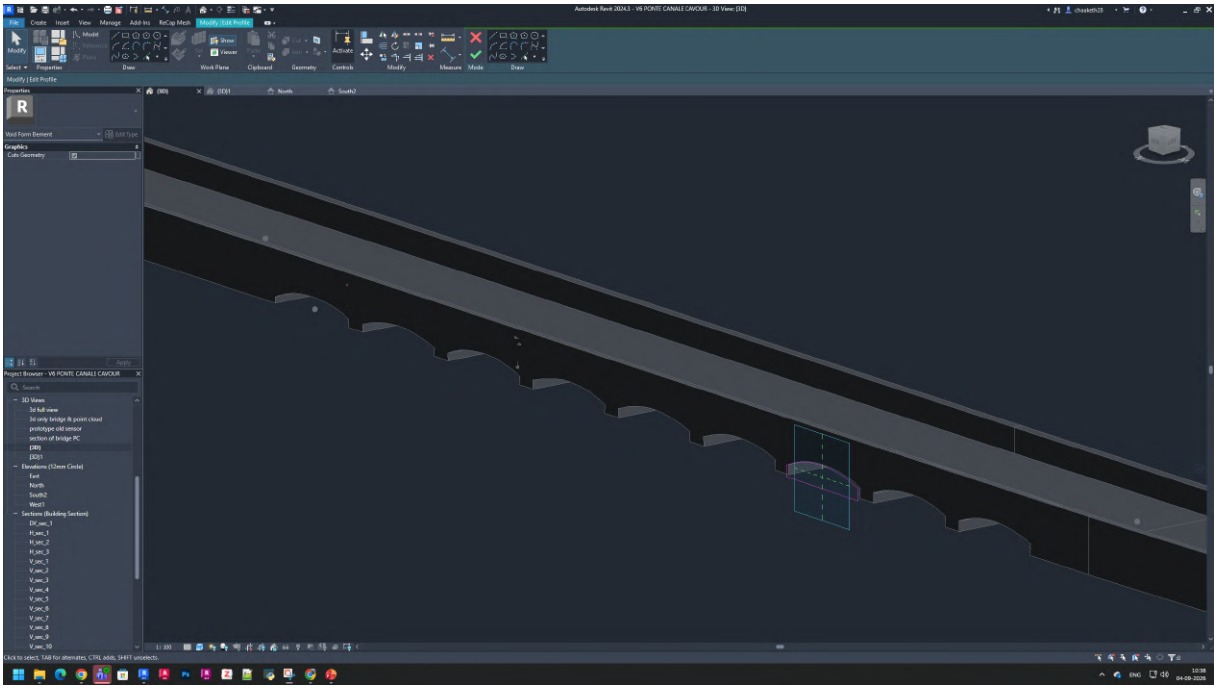


Figure 3.30: *Spandrel wall & arch curve following point cloud*

This subtractive modeling technique ensured a seamless interface between the upper hydraulic deck and the arch vaults below.

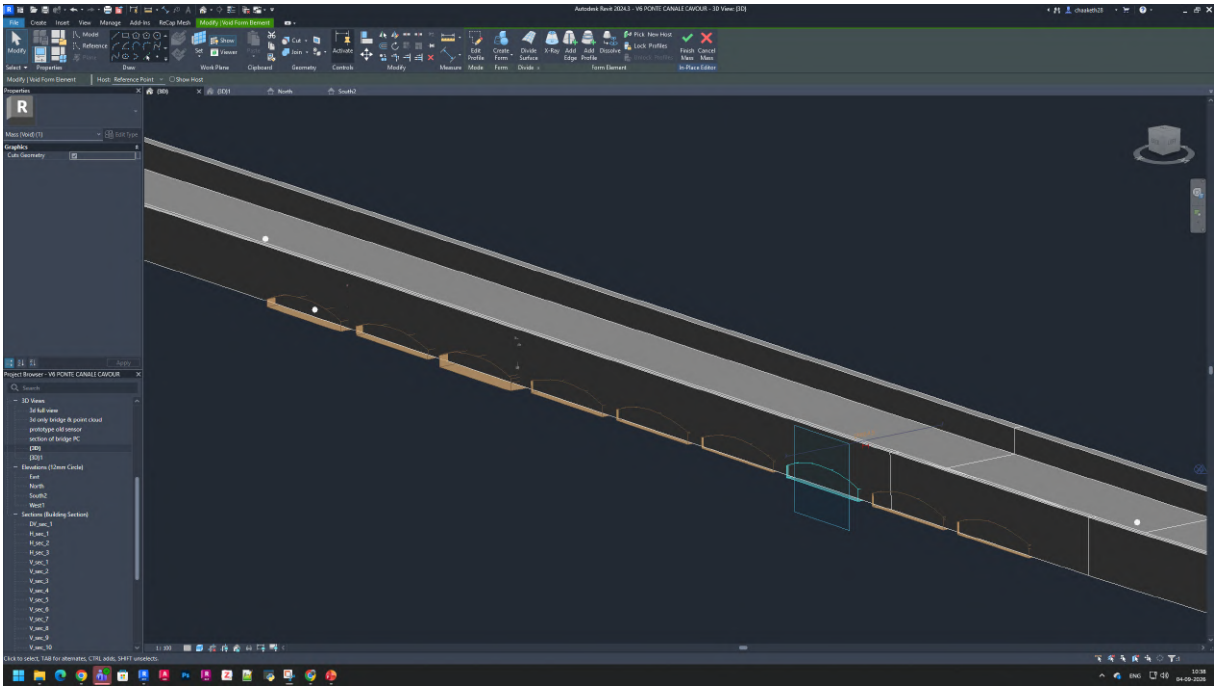


Figure 3.31: *Spandrel wall & arch curve following point cloud*

3.4.3 Modeling non geometric elements: Terrain

The exported elevation dataset was transformed into a native three dimensional building information model (BIM) terrain component using Autodesk Revit's Massing & Site toolset. The text file generated from the CloudCompare rasterization process containing organized three dimensional Cartesian coordinates (X, Y, Z) served as the direct spatial input for site modeling.

Within the Massing & Site environment, Create toposolid command is used, the point dataset was integrated using the Create from Import workflow by executing the Create from CSV command. The point cloud text file was parsed by specifying the corresponding spatial units to maintain geometric consistency with the source data and project reference system. Revit interpolated the imported point distribution to reconstruct the continuous top dynamic surface of the site.

To establish the physical geometry of the terrain, the surface was assigned the Generic 1000mm Toposolid family type via the Properties Palette. Upon completing the generation phase, Revit extruded the interpolated surface downward to a uniform thickness of 1000 mm (Figure: 3.32), converting the discrete point data into a parametric, volumetric terrain solid. This 3D Toposolid provides a structural base suitable for subsequent architectural integration, structural foundation placement, and site cut and fill volumetric analyses.

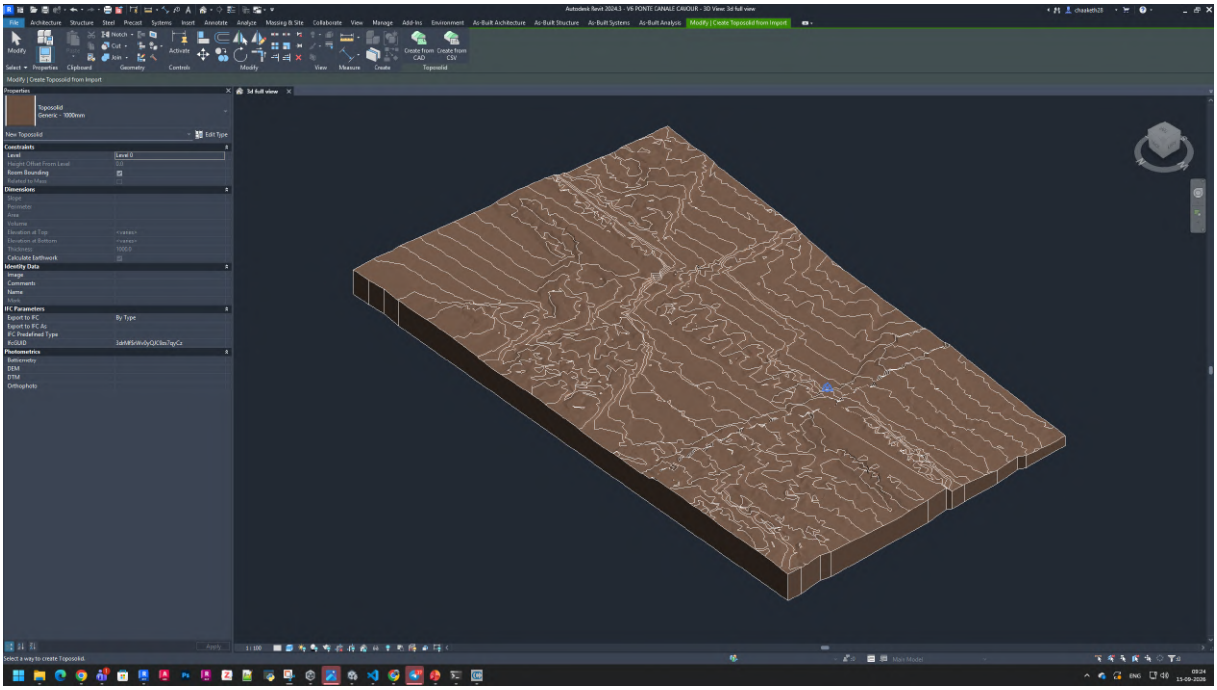


Figure 3.32: *Create toposolid using import .csv*

Chapter 4

Revit and IoT Integration

4.1 IoT sensors

The integration of Internet of Things (IoT) technology into Structural Health Monitoring (SHM) marks a paradigm shift from periodic manual inspections toward continuous, automated data collection [24]. Traditional structural evaluations often rely on visual surveys or scheduled site visits, which offer only snapshot measurements and can miss time sensitive structural dynamics or transient micro deformations [25]. In contrast, IoT based monitoring systems integrate low power sensor interfaces, microcontrollers, and wireless communication protocols such as LoRaWAN or cellular transmission to stream diagnostic telemetry continuously and remotely [26, 27].

Because these sensing devices are compact, operate on minimal battery power, and transmit data wirelessly, they bypass the requirement for extensive cable routing and invasive wired infrastructure [27, 28]. This non disruptive footprint renders IoT networks particularly suited for historic masonry structures, where strict conservation requirements mandate minimal physical interventions [28, 29]. By automating localized signal digitization and data transfer from physical sensors to cloud based management platforms, IoT networks facilitate continuous remote tracking, real time anomaly detection, and predictive maintenance strategies [24, 26].

4.1.1 IoT sensor placement on bridge

Historic masonry arch bridges are particularly vulnerable to long term degradation driven by environmental exposure, river flow dynamics, material aging, and ground movement. Over time, these actions manifest as structural fracturing across arch barrels and spandrels, differential settlement, or tilting of supporting piers, often accelerated by daily and seasonal thermal cycles.

To systematically evaluate these structural behaviors, a wireless IoT monitoring system utilizing four primary sensor types was deployed across critical zones of the bridge masonry:

Crack Meters (Fessurimetri): Measure dynamic fracture opening and closing displacements (in millimeters). Inclinometers (Clinometri): Track rotational movement and angular tilt along two orthogonal axes (x and y axes, measured in degrees). Temperature Sensors (Sensori di Temperatura): Capture ambient thermal variations (°C) to evaluate thermal expansion effects on crack widths and pier inclinations. Dataloggers / Wireless Nodes: Provide localized signal conditioning and digitization, power attached sensors, and transmit telemetry wirelessly to a centralized cloud platform.

Sensor placement and spatial configuration The sensor network was deployed strategically across both the upstream facade (Lato Monte) and downstream facade (Lato Valle) to capture the structure's response to hydraulic loads and directional sun exposure.

Upstream facade (Lato Monte):

Spatial Role: Positioned to face incoming water flow and direct solar exposure, making it ideal for monitoring thermal responses and longitudinal fracture propagation across the main arches.

Crack Meters (Fessurimetro 1, 2, and 3): Fessurimetro 1 is positioned near the bank abutment joint (sponda) to monitor vertical detachment along the structural boundary. (Figure: 4.2)

Fessurimetro 2 and Fessurimetro 3 are mounted directly across the brick masonry of the primary arch vault to measure crack behavior within the main structural span. (Figure: 4.1)

Temperature Sensor (Sensore Temperatura): Installed within a weather shielded enclosure on the upper spandrel wall adjacent to Pier 8 (Pila 8) to record ambient air temperatures. (Figure: 4.3)

Datalogger (Datalogger Monte C_8204_1): Surface mounted on the masonry wall below the temperature sensor near Pier 8 (Pila 8), aggregating wired signals from the upstream crack meters and temperature probe. (Table:4.1)



Figure 4.1: *IoT sensors installation*



Figure 4.2: *IoT sensors installation*



Figure 4.3: *IoT sensors installation*

Downstream facade (Lato Valle):

Spatial Role: Positioned along the downstream pier profile to monitor structural stability and tilt away from direct river impact.

Inclinometer (Inclinometro): Mounted near the base of Pier 6 (Pila 6) on its cylindrical masonry surface to record angular movement along orthogonal axes (x and y). (Figure: 4.4)

Crack Meter (Fessurimetro): Fixed near the upper arch to pier interface at Pier 6 (Pila 6) to track localized masonry displacement.

Datalogger (Datalogger Valle CO_8204_2): Secured to the upper portion of Pier 6 (Pila 6) beneath the stone cornice, receiving sensor inputs and transmitting data wirelessly to the cloud base station. (Table:4.2)

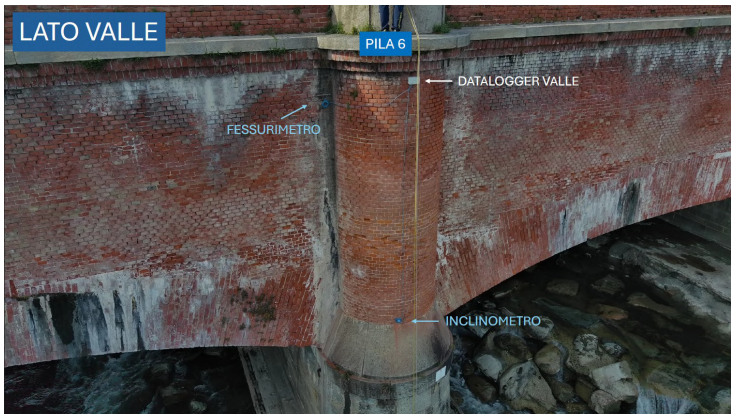


Figure 4.4: *IoT sensors installation*

Analysis of datalogger telemetry Data from both nodes is transmitted to the DIMOS (Displacements Monitoring Sensors) cloud interface managed by MicroGeo. Telemetry captured during a sample monitoring cycle demonstrates system performance and structural readings:

Upstream datalogger node (CO_8204_1 Lato Monte):

Table 4.1 outlines the telemetry configuration for the upstream datalogger node (CO_8204_1) deployed on the Cavour Canal water bridge.

It catalogs five core parameters essential for monitoring the localized structural response: the node's operational battery voltage (measured in Volts), ambient air temperature (measured in °C), and linear displacement measurements from three distinct crack meters labeled Fessurimetro

1, 2, and 3 (measured in millimeters) to systematically capture thermal effects and crack dynamics across the upstream masonry arch.

Table 4.1: *Real time IoT sensor telemetry data from the Cavour Canal water bridge lato monte CO_8204_1.*

Position	Device ID	Identifier	Parameter Name	Unit
1	CO_8204_1	BATT	Battery Voltage	V
2	CO_8204_1	VAR4	Ambient Temperature	°C
3	CO_8204_1	VAR5	Fessurimetro 1	mm
4	CO_8204_1	VAR6	Fessurimetro 2	mm
5	CO_8204_1	VAR7	Fessurimetro 3	mm

Downstream datalogger node (CO_8204_2 Lato Valle):

Table 4.2 presents the corresponding telemetry setup for the downstream datalogger node (C_8204_2).

This dataset records four key metrics: the unit's supply battery voltage (in Volts), bi axial rotational movement recorded by an inclinometer along both the x axis and y axis (measured in degrees) to track structural tilt, and dynamic crack opening displacement captured by a downstream crack meter (measured in millimeters) to evaluate pier stability.

Table 4.2: *Real time IoT sensor telemetry data from device lato valle CO_8204_2.*

Position	Device ID	Identifier	Parameter Name	Unit
1	CO_8204_2	BATT	Battery Voltage	V
2	CO_8204_2	VAR4	Inclinometer (x axis)	°
3	CO_8204_2	VAR5	Inclinometer (y axis)	°
4	CO_8204_2	VAR6	Downstream Crack Meter	mm

4.1.2 Sensor modelling in Autodesk Revit

Integrating IoT devices into a Building Information Model (BIM) transforms a static digital model into an active, information rich asset for Structural Health Monitoring (SHM). In this integration requires creating custom parametric families to represent physical sensor geometries and defining custom Project Parameters to capture diagnostic telemetry, operational metadata, and safety thresholds.

3D Parametric family creation (.rfa)

To transform standard geometric elements into dynamic data containers, custom metadata fields were defined as Shared Parameters rather than standard project parameters. Shared parameters are stored in an external text file format, ensuring that parameter GUIDs (Globally Unique Identifiers) remain consistent across Revit families, schedules, and external exports.

The decision to utilize Shared Parameters was specifically made to support downstream interoperability. By using shared parameters, all custom attributes such as real time sensor measurements, operational limits, and hardware metadata are preserved when exporting the Revit model to IFC (Industry Foundation Classes) or FBX geometry formats, alongside

corresponding JSON metadata files. This schema establishes the foundational data pipeline required for integrating the BIM model into external real time visualization engines, specifically for Unity IoT integration.

Choosing the Electrical Equipment category ensures standardized scheduling capabilities and maintains systematic element organizaiton across the BIM environment.

Crack Meter Family: Modeled with an elongated cylindrical body and mounting brackets to reflect physical displacement transducers installed across masonry fractures. (Figure: 4.5)

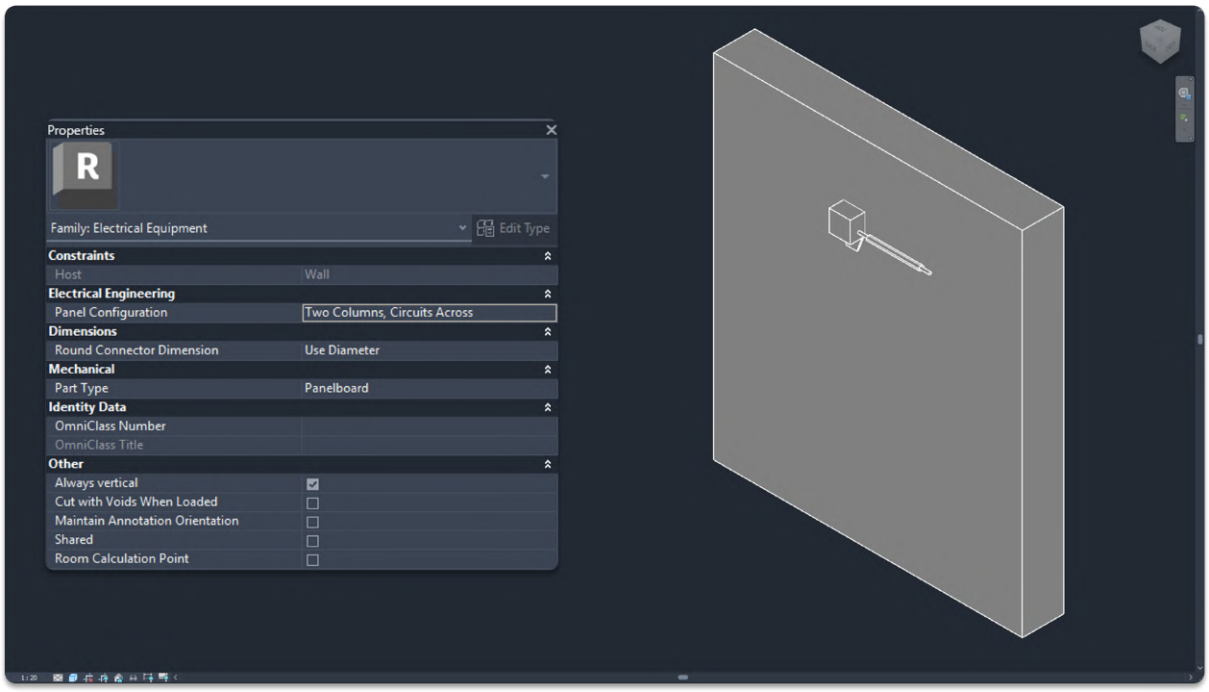


Figure 4.5: *Crack meter & its properties*

Inclinometer Family: Configured as a compact rectangular housing designed to align flush against vertical pier surfaces. (Figure: 4.6)

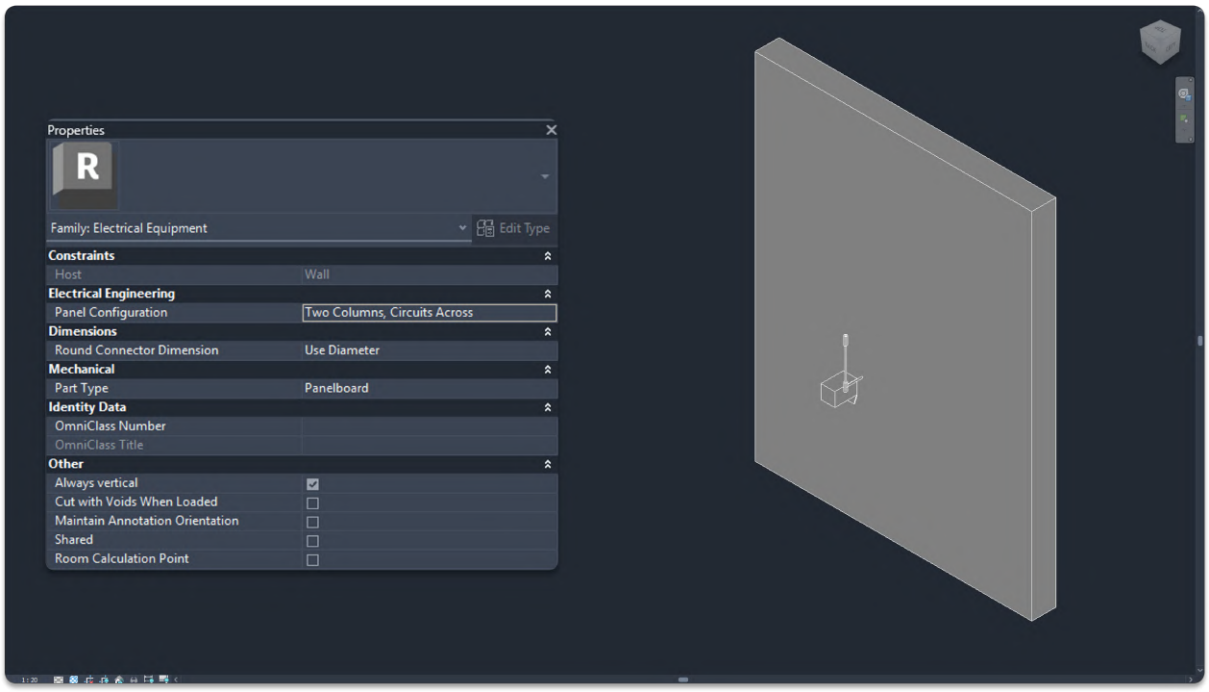


Figure 4.6: *Inclinometer & its properties*

Temperature Sensor Family: Modeled with a protective weather shield and probe geometry to mirror field installations on upper spandrel walls. (Figure: 4.7)

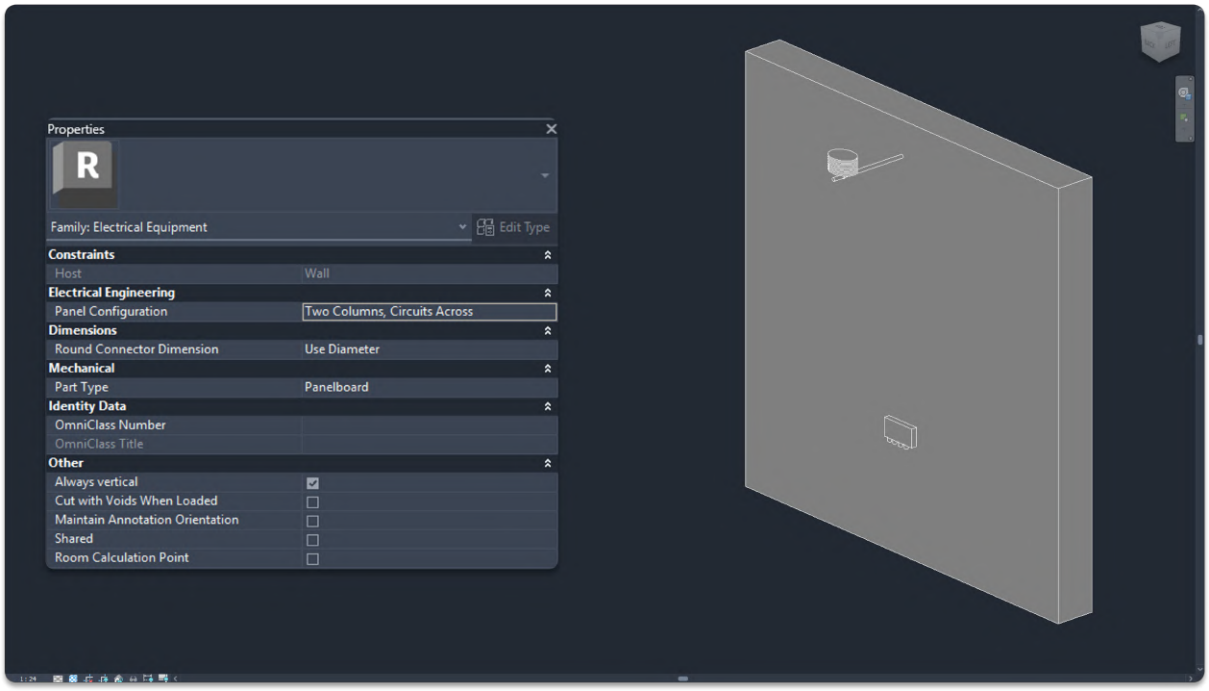


Figure 4.7: *Temperature sensor & Data logger & its properties*

Datalogger Node Family: Created as a junction enclosure that serves as the central physical and data hub for wired sensor connections. (Figure: 4.8)

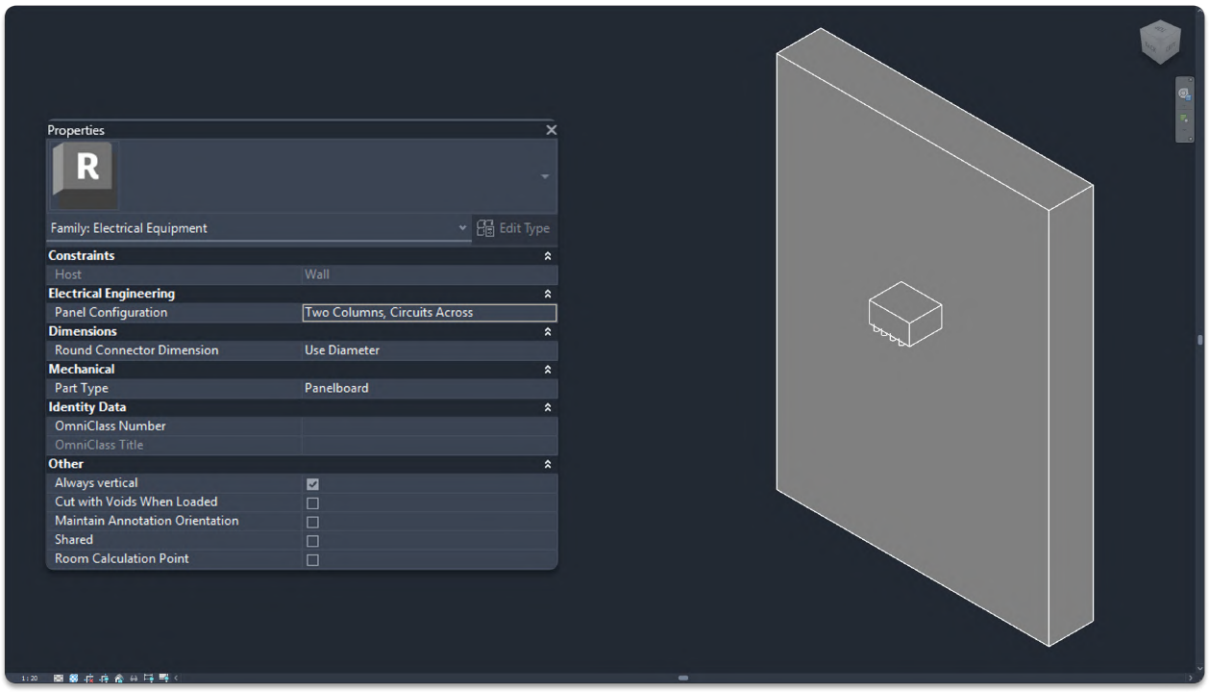


Figure 4.8: *Datalogger & its properties*

Custom shared project parameter definition

To embed sensor metadata and real time data into the Revit project environment, custom Project Parameters are defined and linked directly to the Electrical Equipment and Electrical Fixtures categories.

This setup transforms standard geometric elements into dynamic data containers.

The parameters are organized into functional groups with in the Revit family propertie, Following Data (Real time), Electrical & Hardware properties, Construction & Structural safety Thresholds. (Figure: 4.9)

Data (Real time)

- Crack meter 1, Crack meter 2, Crack meter 3, Crack meter 4: Numeric fields storing crack displacement readings (mm).
- Inclinometer x, Inclinometer y: Angular movement values along orthogonal axes (°).
- Temperature: Ambient thermal measurements (°C).
- Datalogger: Diagnostic status and telemetry records for communication nodes

Electrical properties

- Battery power: Monitors power supply levels (V) for wireless battery operated nodes.
- Sensor manufacturer, Sensor model, sensor serial number: Equipment identification attributes.
- sensor accuracy, Sensor type, Installation date: Field installation specifications.

Construction & Structural safety thresholds

- Limit Max allowable displacement: Safety threshold for fracture opening.
- Limit Max tilt: Operational limit for pier angular tilt.
- Limit Max vibration or movement: Maximum permissible dynamic displacement.
- Safety factor, Maintenance History, Next Inspection Date, Next Inspection Due, Recent Maintenance Image: Lifecycle tracking data for predictive maintenance.

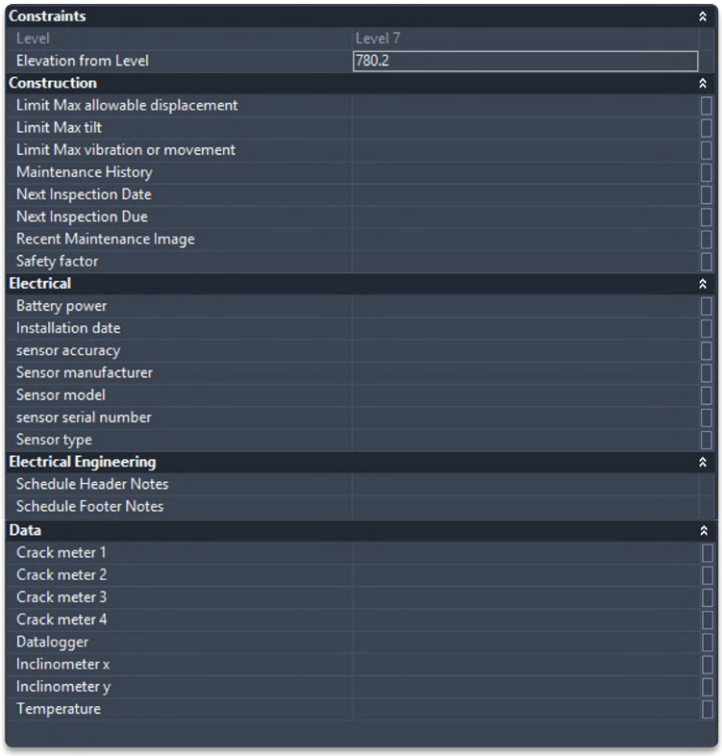


Figure 4.9: *Autodesk Revit Family, Sensor Parameter properties*

Spatial placement and automated data syncing with dynamo

Once the .rfa sensor families were loaded into the main Revit model, they were placed onto structural masonry elements at positions matching their physical field coordinates (e.g., Pier 6, Pier 8, and the arch vaults).

To avoid manual data entry and maintain continuous updates, the data pipeline is automated using custom Revit Dynamo visual programming node scripts. The Dynamo script operates as an intermediary data bridge:

Data Ingestion: The script fetches live sensor readings and operational telemetry directly from the cloud database (JSON endpoint).

Element Identification: It maps incoming telemetry values to specific sensor instances placed in the BIM model using unique device identifiers (Device ID).

Parameter Population: The script dynamically updates the corresponding Shared Instance Parameters (Crack meter, Inclinometer, Temperature, Battery power) for each sensor family in real time.

4.2 Autodesk Revit-IoT architecture

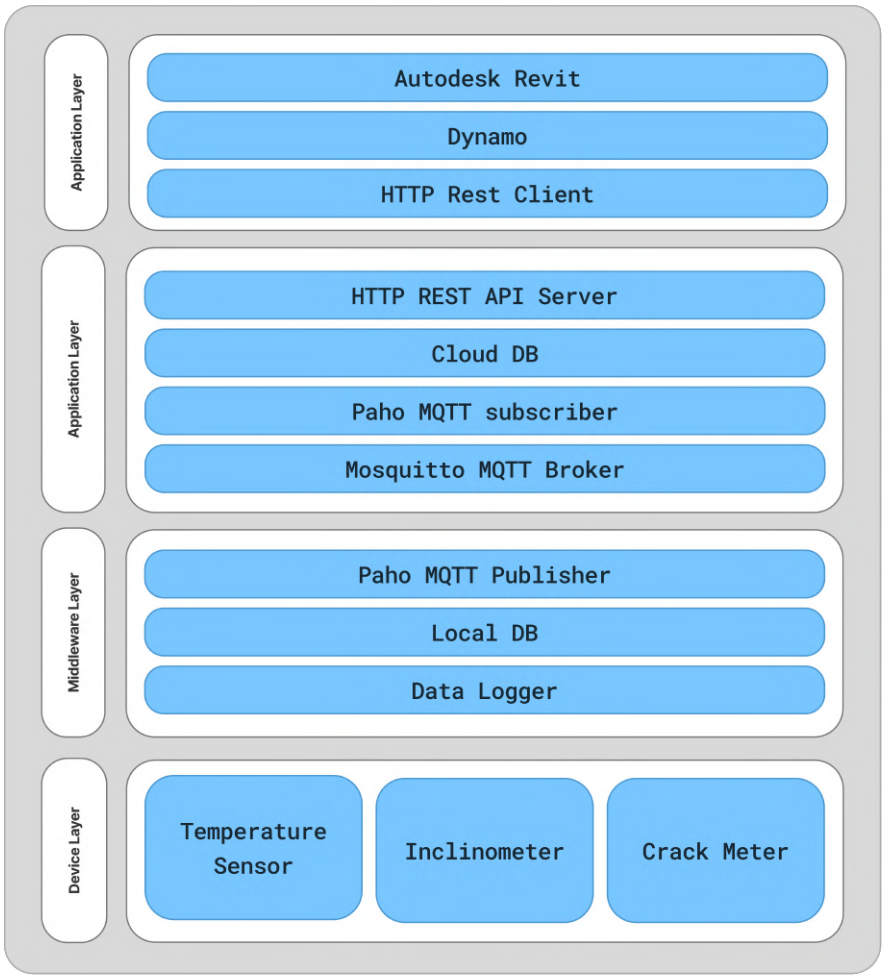


Figure 4.10: *IoT-Revit integration architecture*

This research proposes a four layer IoT BIM integration framework designed to support real time environmental and structural monitoring within a Building Information Model, enabling the implementation of a digital shadow of the monitored asset. The architecture is conceived as a modular, layered system engineering approach, ensuring robust data acquisition, resilient communication, scalable data management, and seamless integration with BIM based visualisation environments. As illustrated in Figure 4.10, the framework is structured into four functional layers: i) Device Layer, ii) Concentrator Layer, iii) Middleware Layer, and iv) Visualisation Layer. Each layer encapsulates a well defined responsibility and exposes explicit interfaces to adjacent layers, thereby promoting scalability, fault tolerance, and technology independence.

4.2.1 Device layer

The Device Layer constitutes the physical sensing infrastructure deployed directly on the bridge structure. Its primary function is to acquire in situ measurements describing the asset's structural and environmental state. Sensors are installed at strategically selected locations identified

as critical for monitoring deformation, displacement, and thermal effects. In the proposed implementation, three sensor typologies are employed: i) crack meters, ii) inclinometers, and iii) temperature sensors. All devices are connected to the Concentrator Layer through wired communication links, a design choice driven by requirements for high measurement reliability, deterministic latency, and immunity to electromagnetic interference, which are essential in long term structural monitoring applications.

Temperature sensor

Temperature sensors are used to monitor thermal conditions affecting both material behaviour and sensor measurements. These devices exploit temperature dependent physical properties to convert thermal variations into electrical signals. Resistance Temperature Detectors (RTDs), for example, rely on the predictable variation of electrical resistance in metals such as platinum, while thermocouples generate voltage differences through the Seebeck effect. Due to their accuracy, stability, and robustness, both technologies are widely adopted in structural and environmental monitoring applications [30].

Inclinometer

Inclinometers measure angular variations relative to the gravity vector, providing information on rotations, settlements, and global deformation patterns. Contemporary static inclinometers typically rely on Micro Electro Mechanical Systems (MEMS) accelerometers, which detect variations in the projection of gravitational acceleration along multiple sensing axes. The resulting signals are processed to derive high resolution tilt angles, supporting the detection and quantification of structural rotations and geotechnical movements in engineered systems [31].

Crack meter

Crack meters are displacement transducers widely used in structural and geotechnical monitoring to measure relative movements across cracks or joints in materials such as concrete, masonry, and rock. They convert mechanical displacement into electrical signals using potentiometric or vibrating wire technologies, enabling precise, stable, long term measurements of crack opening and closure. Such measurements are fundamental for assessing damage progression, evaluating serviceability conditions, and detecting early signs of structural degradation in critical infrastructure, including bridges, dams, and tunnels [32].

4.2.2 Concentrator layer

The Concentrator Layer operates as an edge gateway between the distributed sensor network and the cloud based Middleware Layer. Its primary role is to aggregate sensor data, perform local preprocessing, and ensure reliable data delivery under variable network conditions. As depicted in Figure 4.10, this layer comprises three core components: i) a Data Logger, ii) a Local Database (DB), and iii) a Paho based MQTT Publisher. Together, these elements implement an edge computing paradigm that enhances system resilience, reduces communication overhead, and preserves data integrity.

Data-logger

The Data Logger is a compact electronic unit that acquires and manages data from multiple sensors via wired interfaces. Such interfaces provide deterministic communication, low latency, and robustness in harsh environmental conditions [33]. The data logger supports heterogeneous signal types (e.g., analog, digital, or serial) corresponding to the different sensors deployed in the Device Layer. Incoming measurements are conditioned, synchronised, and time stamped before being forwarded either to the Local DB for buffering or to the MQTT publisher for transmission to the Middleware Layer.

Local database

The inclusion of a Local Database within the Concentrator Layer implements an edge computing strategy to improve system resilience and efficiency. Two main motivations underpin this design choice: fault tolerance under network disruptions and local data reduction. First, monitoring sites frequently experience intermittent or unreliable internet connectivity. In such conditions, the Local DB functions as a persistent buffer, ensuring that sensor data generated during outages is retained locally and forwarded once connectivity to the Middleware Layer is restored [34]. Second, high frequency sensors can generate large volumes of data that are impractical to transmit continuously. The Concentrator Layer can therefore perform local preprocessing, storing raw measurements while publishing only aggregated values, detected anomalies, or event based summaries. This approach significantly reduces bandwidth consumption and cloud storage requirements, which are critical considerations in scalable IoT deployments [35].

Paho MQTT publisher

Data transmission from the edge to the cloud is implemented using the Message Queuing Telemetry Transport (MQTT) protocol via the Paho open source client. MQTT follows a publish subscribe communication model and is specifically designed for constrained devices and unreliable networks [36]. In AEC monitoring scenarios, where network availability may be intermittent, and devices may operate under limited resources, MQTT offers minimal protocol overhead, with headers as small as 2 bytes. This makes it significantly more efficient than request based protocols such as HTTP [37]. Within the proposed architecture, the concentrator normalises sensor data into structured payloads (e.g. JSON) and publishes them to dedicated topics on a central broker, enabling an asynchronous, scalable, and decoupled data flow to the Middleware Layer.

4.2.3 Middleware layer

The Middleware Layer constitutes the logical integration core of the architecture, bridging the gap between telemetry oriented IoT systems and data centric BIM and enterprise applications. Its primary functions are data ingestion, persistent storage, semantic enrichment, and controlled data exposure through standardised interfaces. As illustrated in Figure 4.10, this layer includes four main components: i) a Mosquitto MQTT Broker, ii) a Paho MQTT Subscriber, iii) a Cloud Database, and iv) an HTTP REST API Server.

Mosquitto MQTT broker

The Mosquitto MQTT Broker is a widely used open source implementation of the MQTT standard and serves as the central message routing component of the system [36]. It receives sensor data published by multiple edge concentrators and distributes them to subscribed middleware services. Deploying the broker in the cloud or on a central server enables scalable handling of concurrent data streams and ensures loose coupling between data producers and consumers, thereby improving extensibility and fault tolerance.

Paho MQTT subscriber

The Paho MQTT Subscriber consumes messages from the broker and transforms transient telemetry into persistent, queryable data records. Upon message reception, the subscriber validates payload structure, performs preprocessing operations (e.g., unit conversion or timestamp normalisation), and forwards the processed data to the Cloud DB. Since MQTT does not inherently support long term data storage, the subscriber plays a critical role in bridging real time messaging and historical data management, enabling longitudinal analysis and condition assessment [38].

Cloud database

The Cloud Database provides persistent storage and acts as a data historian for the monitoring system. It stores time stamped sensor measurements, derived indicators, and contextual metadata required for structural analysis and BIM integration. At this stage, schema mapping and semantic enrichment are performed. Sensor data are associated with BIM object identifiers, spatial locations, and asset hierarchies, ensuring traceability between physical measurements and their digital counterparts within the BIM environment.

HTTP REST API server

The HTTP REST API Server provides controlled and interoperable access to monitoring data. While MQTT is optimised for machine to machine communication, most BIM tools and user facing applications rely on request response paradigms over HTTP. The API exposes structured endpoints that allow visualisation and analytics tools to retrieve specific data subsets (e.g., by sensor, time window, or BIM element) without maintaining persistent broker connections. This decoupling is essential for integration with legacy AEC software ecosystems and supports scalable and secure data access [39].

4.2.4 Visualisation layer in Autodesk Revit & IoT integration

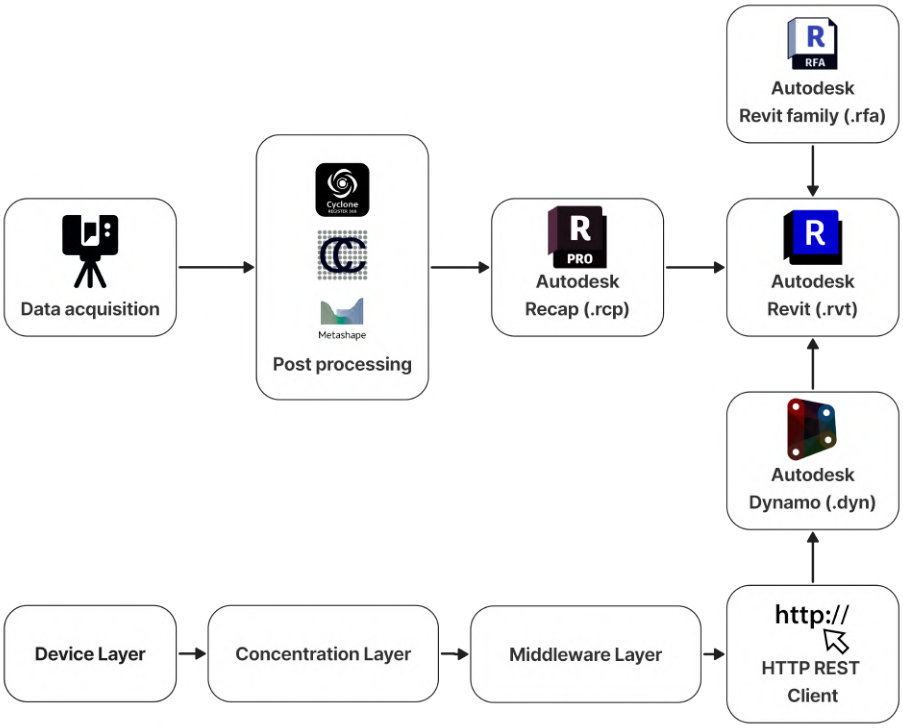


Figure 4.11: *IoT-BIM integration architecture*

The Visualisation Layer represents the final stage of the architecture, where monitoring data are contextualised and interpreted by human stakeholders. This layer operationalises the digital shadow by embedding live and historical sensor data within the BIM environment, enabling spatially informed analysis of structural and environmental conditions. As shown in Figure 4.10, this layer consists of three tightly integrated components: i) an HTTP REST Client, ii) Dynamo, and iii) Autodesk Revit.

HTTP REST client and data update workflow

The HTTP REST Client implemented within Dynamo is responsible for interfacing with the Middleware Layer. It periodically issues HTTP `GET` requests to the REST API Server and retrieves the latest sensor data stored in the Cloud DB.

Dynamo

Dynamo is a visual programming environment embedded within Revit that enables custom data processing and automation workflows. In the proposed architecture, Dynamo serves as the orchestration layer, retrieving external data, performing logical transformations, and updating BIM parameters programmatically. Its capability to access and modify Revit elements without altering the core application makes Dynamo the key enabler for implementing a live digital shadow within a traditionally static BIM environment.

Autodesk Revit

Autodesk Revit serves as the primary BIM authoring and visualisation platform. It contains the authoritative geometric and semantic model of the asset, including structural elements, parameters, and relationships. However, Revit is inherently a static, design oriented system and does not natively support continuous data streams or direct IoT integration. To overcome this limitation, the proposed architecture introduces an intermediary data injection mechanism that enables dynamic parameter updates while preserving Revit's internal data structures and workflows.

The operational workflow is defined as follows:

1. Dynamo executes at a predefined temporal interval.
2. An HTTP request is sent to the Middleware Layer's API endpoint.
3. The API server returns structured sensor data (e.g., JSON).
4. Dynamo parses the response and extracts sensor identifiers, timestamps, and values.
5. Sensor IDs are mapped to corresponding Revit element IDs.
6. Revit parameters are updated with the latest measurements.
7. View filters and graphical rules automatically reflect updated values through colour coded visualisation.

Through this mechanism, Revit evolves from a static design model into an interactive monitoring interface, enabling intuitive assessment of structural conditions directly within the BIM environment and supporting informed decision making across the asset lifecycle.

Results of the following section is continued in

Chapter 5

Unity and IoT integration

5.1 Digital Twin vs Digital Shadow

The evolution of digital representation in Structural Health Monitoring (SHM) and Industry 4.0 ranges from static digital models to fully autonomous digital twins. The classification established by [40] differentiates these systems based on the level of automated data flow and control loops between physical assets and their virtual counterparts. (Figure: 5.1)

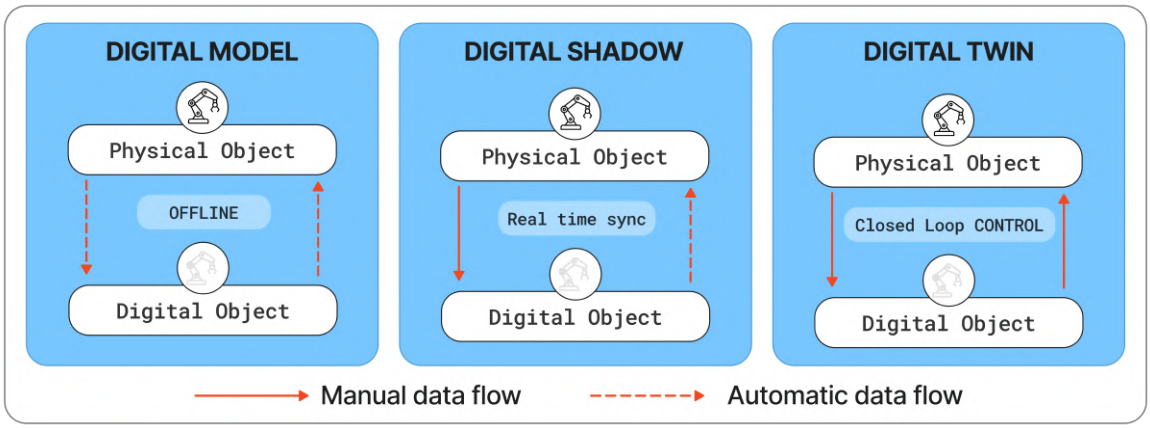


Figure 5.1: *Digital Twin vs Digital Shadow*

Digital Model

A Digital Model is a virtual representation of a physical object without any automated data exchange [40]. The flow of information between the physical object and its digital counterpart is entirely manual [40, 41]. Any changes made to the state of the physical structure do not automatically update the virtual model, and structural modifications made in the digital model must be manually implemented on the physical structure [40]. Typical examples include standard Building Information Models (BIM) or CAD representations used purely for static geometric visualization, archival documentation, and offline engineering analysis [41, 42].

Digital Shadow

A Digital Shadow introduces automated, real time data synchronization flowing in one direction from the physical asset to its virtual representation [40, 43]. Sensors deployed on the physical object (such as crack meters, inclinometers, and temperature sensors) transmit real time telemetry to update the digital model continuously [43, 27]. While the digital object accurately mirrors state changes of the physical structure in real time, it cannot automatically send control signals back to modify or actuate the physical object [40, 27]. The system described in this thesis where IoT sensors stream data from the bridge to a Autodesk Revit, web based platform dashboard functions fundamentally as a Digital Shadow [40, 43, 29].

Digital Twin

A Digital Twin represents the highest level of digital integration, featuring automated bidirectional data flow that forms a closed loop control system [40, 41]. In this setup, live sensor data updates the virtual model in real time, while analytical algorithms, AI, or decision support rules within the digital object generate feedback or command actions sent directly back to the physical asset [40, 41, 44]. This closed loop allows the virtual system to automatically adjust active controls, actuate physical components, or trigger preventive physical actions without requiring manual human intervention [40, 44].

5.1.1 Digital Twin in AEC industry

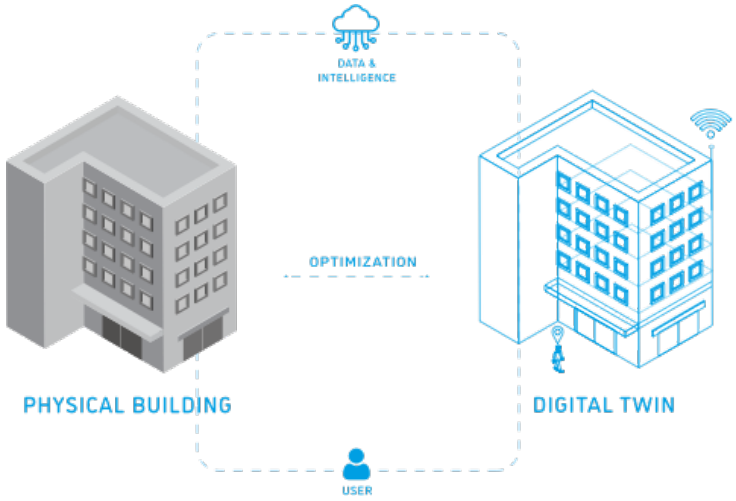


Figure 5.2: *Digital twin in AEC industry (Source: e verse learn)*

A digital twin is a dynamic virtual representation of a physical asset that integrates geometric information, operational data, and real time sensor inputs. Unlike traditional 3D models, digital twins continuously reflect the condition and behaviour of their physical counterparts, enabling monitoring, analysis, and decision making throughout the asset lifecycle.

Within the AEC industry, digital twins are increasingly being adopted for the management of buildings, bridges, transportation infrastructure, and smart cities. By combining BIM data with real time operational information, digital twins provide stakeholders with a comprehensive understanding of asset performance. This enables improved maintenance planning, condition assessment, energy management, and operational efficiency.

5.2 Introduction to unity game engine

Unity is a real time 3D development platform originally introduced by Unity Technologies in 2005 for the development of interactive video games. Over the years, the platform has evolved beyond the gaming industry and has become widely adopted in fields such as architecture, engineering, construction (AEC), manufacturing, automotive engineering, healthcare, and education. The increasing demand for immersive visualisation and real time data integration has positioned Unity as a powerful platform for digital twin development.

The concept of digital twins gained significant attention during the late 2010s as industries sought to create virtual representations of physical assets that could be continuously updated using real world data. Recognising this opportunity, Unity expanded its capabilities through the development of tools and services that support real time visualisation, Internet of Things (IoT) connectivity, cloud integration, and simulation based workflows. Today, Unity is widely used for creating interactive digital twins of buildings, infrastructure, industrial facilities, and urban environments.

5.2.1 Unity game engine and autodesk revit interoporability

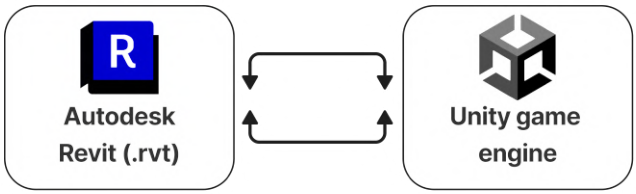


Figure 5.3: *Illustration of interoperability*

The integration of Building Information Modelling (BIM) platforms with real time visualisation environments has become increasingly important for the development of digital twins in the Architecture, Engineering, and Construction (AEC) industry. Among the available BIM authoring tools, Autodesk Revit is widely used for creating detailed building models, while Unity provides a powerful platform for real time visualisation, simulation, and digital twin applications. Establishing interoperability between these two environments is therefore essential for transferring both geometric and semantic information from BIM models into interactive digital twin systems.

A BIM model contains significantly more information than its geometric representation. In addition to three dimensional geometry, Autodesk Revit stores valuable metadata such as element identifiers, categories, material properties, dimensions, construction information, and asset related parameters. Preserving this information during the transfer process is critical because it enables the digital twin to maintain a meaningful connection with the original BIM model.

5.2.2 Unity Reflect for BIM integration



Figure 5.4: *Unity Reflect 360 overview (Source: Autodesk)*

To address the challenge of BIM interoperability, Unity introduced Unity Reflect, a dedicated solution designed to facilitate the transfer of BIM data from Autodesk Revit into the Unity engine [45]. Unity Reflect enabled users to synchronize Autodesk Revit models directly with Unity while preserving much of the associated metadata and project structure [45, 46]. This workflow allowed BIM models to be visualized and explored interactively without requiring extensive manual conversion procedures [46].

One of the main advantages of Unity Reflect was its ability to automatically translate Autodesk Revit elements into Unity objects while maintaining relationships between model components and their associated properties [45]. This allowed users to access BIM information directly within Unity applications, making it suitable for visualization, design collaboration, and early digital twin development [45, 46].

However, Unity Reflect was primarily developed as a commercial plug and play interoperability solution and offered limited flexibility when custom data structures, specialized workflows, or project specific requirements were needed [45, 43]. Furthermore, long term digital twin applications often require greater control over the transfer, storage, and management of BIM metadata than standard synchronization tools can provide [43].

Importantly, Unity Reflect has since been officially discontinued and phased out by Unity Technologies. Its deprecation highlights the risks associated with relying on proprietary commercial middle layer software for structural health monitoring platforms. The lack of ongoing support and customizability in discontinued commercial tools further justifies the adoption of custom, open standard data pipelines utilizing exported IFC, FBX, and JSON metadata streams linked via Dynamo to ensure full data ownership, long term stability, and complete flexibility for Unity IoT integration.

5.2.3 Custom autodesk revit plugin for metadata export

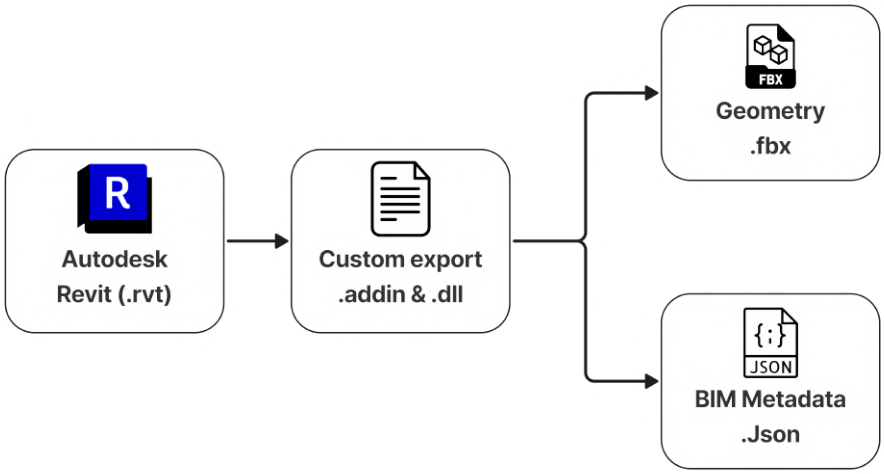


Figure 5.5: *Autodesk Revit plugin visulisation*

To overcome these limitations and support the requirements of this research, a custom plugin was developed within the Autodesk Revit environment. Instead of relying on proprietary data exchange workflows, the plugin extracts both geometric and semantic information directly from the BIM model and exports the metadata into a structured JSON (JavaScript Object Notation) format. (Figure: 5.5)

The developed plugin accesses Autodesk Revit elements and retrieves relevant information such as element identifiers, categories, family names, types, and user defined parameters. This information is then organised into a machine readable JSON structure that can be processed efficiently by external applications, including Unity.

The use of JSON offers several advantages. It provides a lightweight and widely supported data format, facilitates interoperability between software platforms, and enables flexible storage and retrieval of BIM information. Additionally, the exported data can be extended to include project specific parameters, sensor identifiers, maintenance records, or other information required for digital twin applications.

Architecture & code structure

Architecture & Code StructureThe core functionality is split across four distinct primary C# source files:

- App.cs: Implements IExternalApplication. Handles add-in lifecycle events (startup/shutdown) and registers the ribbon panel and button in Autodesk Revit's UI.
- ExportCommand.cs: Implements IExternalCommand. Serves as the main entry point when the user clicks the button, managing document references, file dialogs, and triggering export logic.
- FbxExporter.cs: Uses Autodesk Revit's native CustomExporter or export API to extract scene geometry and save the output as an .fbx file.

- MetadataExporter.cs: Iterates through model elements, reads built-in and custom parameters, and serializes the structured BIM metadata into a .json file.

Metadata mapping structure (JSON) To maintain alignment between geometry and parameters, the JSON file follows a clear structural hierarchy:

Table 5.1: *Autodesk Revit element metadata mapping structure for BIM-to-web platform integration.*

Key	Description	Example Data
ElementId	Unique Autodesk Revit integer ID matching the FBX mesh node	452319
UniqueId	Persistent Autodesk Revit GUID	c2a13f28-8d4e-4e89...
Category	Autodesk Revit element category	"Walls"
FamilyAndType	Family name and Type designation	"Basic Wall : Generic - 200mm"
Parameters	Key-value dictionary of instance/type properties	{"Length": "5000 mm", "Fire Rating": "2 hr"}

Execution Workflow

- User Action: Click the  external tool button on the Autodesk Revit Ribbon.
- Command Handler: ExportCommand.cs prompts the user for a destination save path.
- Geometry Export: FbxExporter.cs writes the 3D representation to disk.
- Data Extraction: MetadataExporter.cs parses visible elements and exports parameters to JSON. (Figure: 5.6)
- Completion: A confirmation dialog displays the output location and export status

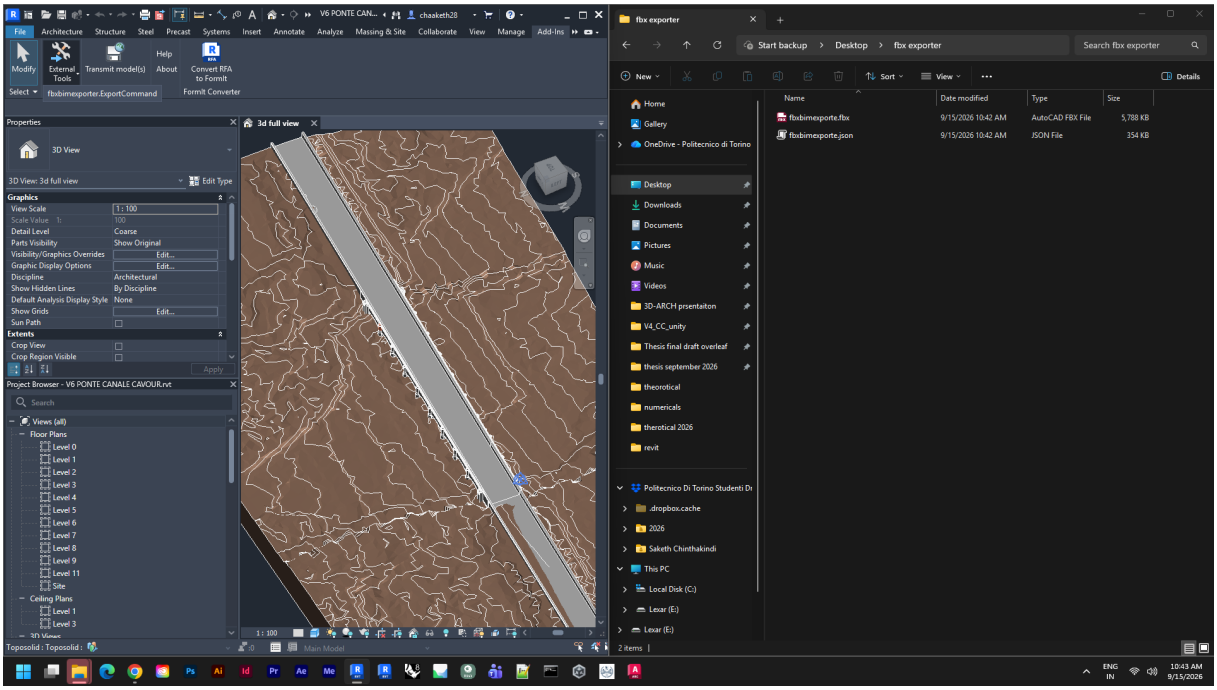


Figure 5.6: *Left Autodesk Revit, Custom plugin exicuted & Right destination folder exported both .fbx and .json files*

5.2.4 Integration with in unity game engine

In this project, by developing a custom Autodesk Revit exporter plugin, the workflow bypasses intermediate proprietary conversion tools, allowing BIM semantic metadata to be extracted into a lightweight bim_data.json structure alongside the corresponding geometric mesh (v6 Ponte canale cavour v1.fbx) at the click of a button. This approach guarantees that even large scale project models retain their associated semantic data during transfer into the gaming engine.

Once both assets are imported into Unity, C# scripting (BIMElementManager.cs and related utility handlers) dynamically maps the visual 3D elements to their corresponding JSON attribute records. This linking is established by matching the unique Autodesk Revit Element ID encoded within the FBX node naming hierarchy (e.g., [Element ID]) against the keys in bim_data.json. While linking via string parsed Element IDs served as a initial proof of concept for rapid prototyping, future iterations of this framework will transition toward open BIM standards (such as Industry Foundation Classes, or IFC GUIDs) to enhance schema standardization, multi platform interoperability, and long term asset management.

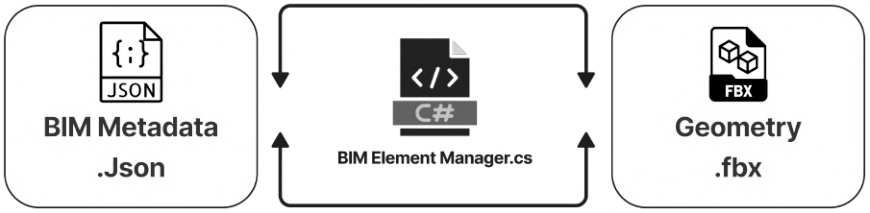


Figure 5.7: *Illustration of workflow using BIMElementManager.cs*

Within the runtime Unity WebGL environment, this mapping allows semantic properties to be queried interactively in real time. Selecting a 3D component in the viewport prompts the C# runtime to retrieve its associated BIM attributes and stream them directly to the web interface. Furthermore, establishing this unified geometric and semantic data bridge forms the baseline required for integrating real time telemetry streams, threshold alerts, and structural health inspection logs directly onto the corresponding 3D asset components.

5.3 Unity game engine-IoT architecture

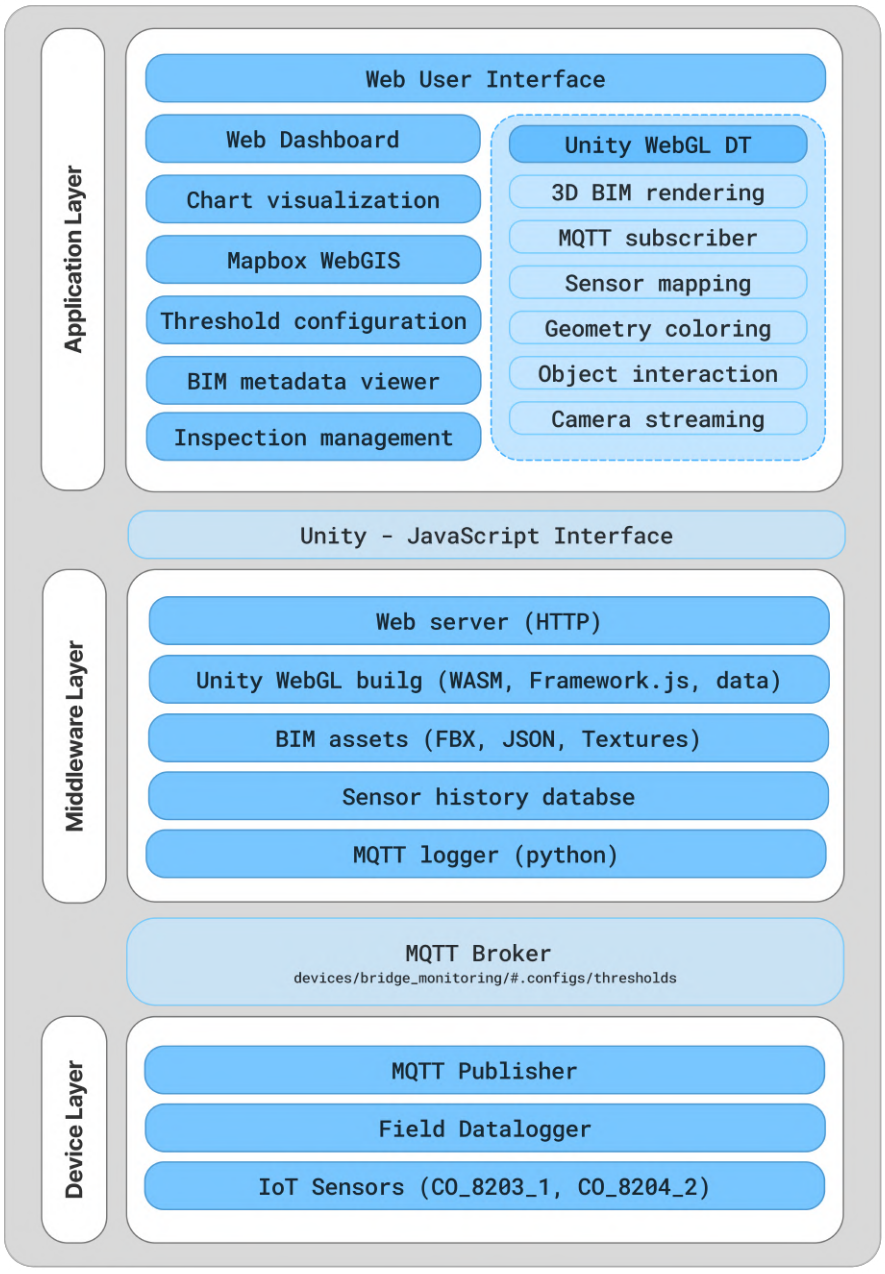


Figure 5.8: *IoTUnity game engine architecture*

The digital twin framework employs a multi tiered architecture designed to bridge physical structural monitoring hardware with an interactive, web based digital model. The data pipeline transitions from physical field measurements up to user facing visual and analytical environments, ensuring high data reliability, decoupled processing, and accessible presentation for AEC domain experts.

5.3.1 Device layer

The Device Layer represents the physical and edge infrastructure deployed directly on the monitored asset. It is responsible for physical parameter acquisition, local signal handling, and secure data transmission.

Field sensors

Physical instrumentation installed on structural components to capture mechanical and environmental parameters. In structural monitoring contexts, these typically comprise crack displacement sensors to track joint movement, thermal sensors to monitor ambient and material temperatures, and biaxial inclinometers to track angular tilt across orthogonal axes.

Field datalogger

Dedicated acquisition hardware positioned on site that interfaces with physical sensors. Dataloggers continuously query physical instruments, convert raw instrumentation signals into structured physical engineering units, apply temporal timestamps, and buffer readings to ensure data preservation during intermittent network connectivity.

MQTT publisher

An edge computing routine executing on the local gateway. It normalizes logged physical data into a standardized JSON payload structure attributing standardized variable identifiers, physical units, and timestamps and securely publishes telemetry updates to the central communication service.

MQTT message broker

The MQTT Message Broker operates as an independent central communication hub separating physical data collection from digital application surfaces. Configured with dedicated network channels, it manages message traffic across two primary operational domains:

Telemetry channels: Dedicated channels handling continuous streams of live sensor data pushed from the site. This enables multi cast distribution, ensuring that all connected monitoring surfaces receive real time structural updates simultaneously without overloading edge hardware.

5.3.2 Middleware layer

The Middleware Layer hosts server side data persistence, web asset management, and historical data logging, providing a stable backbone that remains independent of active user browser sessions.

Web server (HTTP)

Serves static web assets, visual dependencies, client side scripts, and compiled 3D runtime binaries to user devices. Additionally, it exposes secondary REST API endpoints to accept sensor ingestion payloads via standard web protocols as an operational fallback.

Unity WebGL build

Unity WebGL Digital Twin serves as the primary interactive spatial environment for the system. Rather than running as a standalone desktop application, the digital twin is compiled directly into standard browser native web files. This compilation strategy enables the 3D model to execute smoothly within modern web browsers via WebAssembly without requiring external software plugins or manual client side installations.

The underlying WebGL build structure relies on six core web artifacts to instantiate and render the digital twin context. The HTML shell (index.html) establishes the overall web layout and canvas container, while the stylesheet (style.css) manages responsive styling and viewport positioning across both the main web interface and the 3D canvas. Upon initialization, the engine bootstrap loader script (mydigitaltwinv1.0.loader.js) verifies browser capability, allocates memory, and manages the downloading sequence for the remaining core engine assets. The compiled runtime executable (raw_built_dt.wasm) houses the application logic, structural calculation routines, and internal engine systems encoded in high performance binary code. This binary is linked directly to the browser's native execution context by the framework bridge (raw_built_dt.framework.js), which manages low level graphics rendering contexts and memory buffers. Finally, the asset data package (raw_built_dt.data) acts as a compressed container storing all visual building elements, 3D geometries, textures, material definitions, and spatial scene setups required to construct the digital bridge asset.

BIM assets & parametric datasets

Offline building information modeling assets containing 3D geometric components, physical element identification tags, material textures, and associated non geometric metadata files. Unique identification keys embedded within the 3D geometry link visual elements directly to external parametric datasets.

Sensor history database

Time series database infrastructure dedicated to long term telemetry storage. It maintains durable records of structural readings and threshold violation events, enabling longitudinal trend analysis, historical data retrieval, and structural behavior playback beyond real time memory limits.

MQTT data logger

A server side service that continuously subscribes to telemetry channels on the message hub and commits incoming data streams to the database. This guarantees continuous data retention regardless of whether end user visual interfaces are actively open.

Unity javascript interface

The interface acts as a runtime operational bridge facilitating bidirectional communication between the web browser context and the embedded 3D WebGL engine environment. sits inside the unity webGL Build files.

Browser to engine communication

Translates standard web events such as UI navigation commands, threshold configuration updates, or metadata query requests into executable methods within the 3D environment to alter camera orientation, object highlighting, or internal state.

Engine to browser communication

Emits internal 3D engine events such as object click selection, camera movement coordinates, or dynamic frame captures back to the web interface to refresh surrounding dashboards, open inspection forms, or synchronize external spatial maps.

5.3.3 Application layer

The Application Layer encompasses the client side user interfaces, structured into two parallel presentation environments that process live incoming message data independently for fault tolerant operation.

Web user interface

Web dashboard

Main operational view providing global project information, visual documentation of model development maturity, and high level health status tables with visual indicators driven by live structural telemetry. The Web Dashboard serves as the central user interface layer, combining operational status overview panels, model maturity documentation, and real time asset health indicators. Page layouts and visual positioning are dynamically rendered through standard native DOM and CSS Object Model interfaces, while asset metadata querying relies on asynchronous HTTP Fetch requests to ingest static BIM attribute datasets. Continuous client server communication is maintained via the native browser WebSocket interface, which establishes a full duplex TCP connection to receive real time telemetry updates and update status indicators dynamically without requiring page reloads [47].

Chart visualisation

Graphical analytics platform plotting continuous time series trends for each monitored variable. It incorporates threshold reference lines, view navigation controls, and historical range filtering.

For structural health analytics, the monitoring interface leverages the Chart.js canvas API to instantiate and maintain dynamic time series graph instances, updating datasets efficiently via frame by frame canvas redrawing as incoming telemetry populates client side memory ring buffers [48]. Specialized extension interfaces complement the core graphing engine: the chartjs plugin annotation interface programmatically renders horizontal reference baselines over time series axes to mark critical structural thresholds [49], while the chartjs plugin zoom interface provides pan and zoom event listeners that allow operators to switch between locked real time tracking and historical data inspection [50]

WebGIS context

Geospatial map interface displaying broader asset location data, regional infrastructure networks, and contextual geographic features paired with dynamic information panels. Geospatial context

is delivered through the Mapbox WebGIS module, which utilizes the client side Mapbox GL JS engine to parse vector tiles and render geographic layers directly onto a WebGL accelerated canvas context [51]. Spatial dataset integration relies on the map engine's source and layer management APIs (addSource and addLayer) to programmatically register and style regional vector datasets, while feature interaction APIs process pointer click events on spatial geometries to extract attributes and populate contextual information panels [51]. Additionally, external environmental REST APIs, including the Open Meteo API for real time weather metrics [52] and the Sunrise Sunset API for exact solar coordinate calculations, are queried asynchronously to synchronize atmospheric conditions and solar lighting within the digital twin context.

Threshold configuration

Management console allowing operators to establish critical structural performance boundaries. Modifying threshold limits both minimum and maximum by the user which automatically updates graphical analytical baselines and transmits modified parameters across the communication service to the 3D digital model.

BIM metadata viewer

Interactive attribute panel displaying extracted Building Information Modeling properties. Selecting structural elements via search utilities, spatial hierarchy lists, or direct 3D clicks resolves parameter lookup files to present detailed physical and functional properties.

Inspection management

Site condition reporting interface linked to physical coordinates. Operators can record, view, and store structural observations, notes, and severity ratings associated with specific spatial markers placed directly on structural elements.

Unity WebGL Digital Twin

3D BIM rendering

High fidelity spatial environment rendering structural geometry, synchronized lighting cycles matching real world site solar positions, local atmospheric conditions derived from weather services, and procedural material effects. thanks to C# programming done in the Unity game engine custome files have been coded for weather, climate controls helping the 3d render to act accordingly.

MQTT engine bridge

Internal telemetry subscriber that parses incoming data from the web messaging client and routes parsed payloads to appropriate mapping, visual, or dynamic simulation scripts. to setup this MQTT engine Bridge for Unity game engine and unity M2MQTTUnity plugin [53] from github helped following the readme steps

Sensor mapping

Spatial assignment layer linking physical sensor identifiers to their corresponding 3D GameObjects and performance limits within the visual scene.

Geometry coloration

Dynamic visual feedback system that evaluates live physical readings against active threshold boundaries, automatically recoloring structural components (e.g., green for normal state, orange for caution, red for threshold breach) or applying rotational transforms to simulate measured angular tilt.

Object interaction

Direct spatial engagement routines supporting element selection, automated camera focus on specific structural assemblies, and marker placement for site condition tracking.

Camera streaming

In engine streaming component that processes virtual camera views positioned at key structural locations, converting rendered frames into visual streams pushed directly to the external web dashboard for real time monitoring.

5.3.4 Visualisation layer in Unity game engine (WebGL Build) and IoT integration

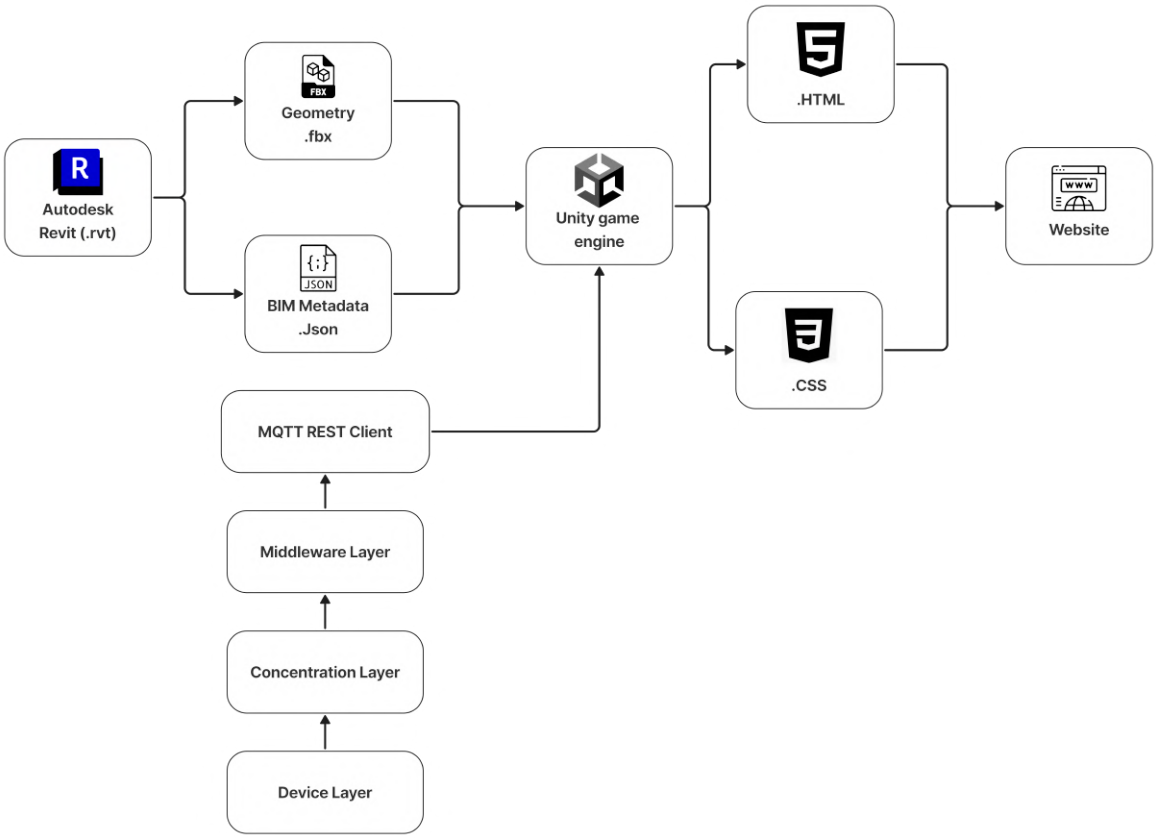


Figure 5.9: *IoT-Unity game engine workflow*

The Visualization Layer serves as the primary operational environment of the proposed digital shadow framework. While raw sensor telemetry, spatial point clouds, and structural data are gathered in the background, this layer translates complex datasets into an intuitive, real time visual interface accessible directly through a standard web browser. Built on the Unity game engine using its WebGL build profile, the platform achieves a balance between high fidelity 3D rendering and lightweight, client side web accessibility without requiring dedicated desktop software installations.

This section details the end to end implementation of the web based visual interface, highlighting the bridge between the 3D WebGL viewport and the web application dashboard wrapper.

5.3.5 Build deployment and dashboard architecture

To transition the digital twin from a standalone desktop graphics environment to an accessible, zero install web application, the project utilized Unity's WebGL Build Profile. (Figure: 5.10)Following the initial import and assembly of the 3D bridge geometry (.fbx) and structural metadata (.json), the project was compiled into a optimized WebGL distribution package.

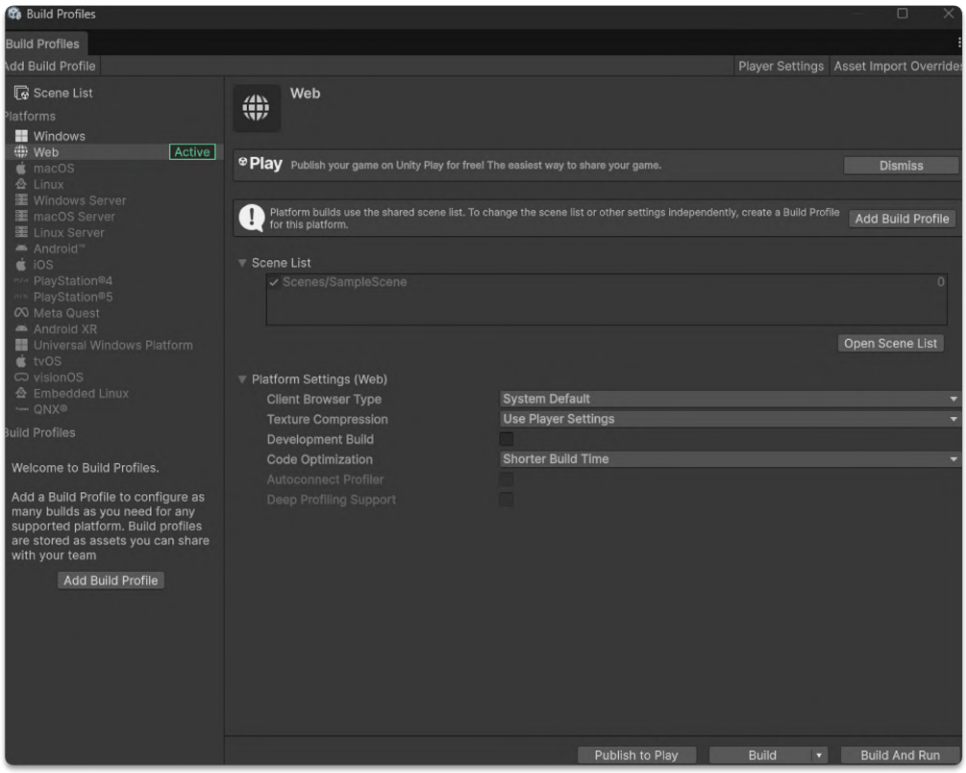


Figure 5.10: *Unity game engine, Build profile settings page*

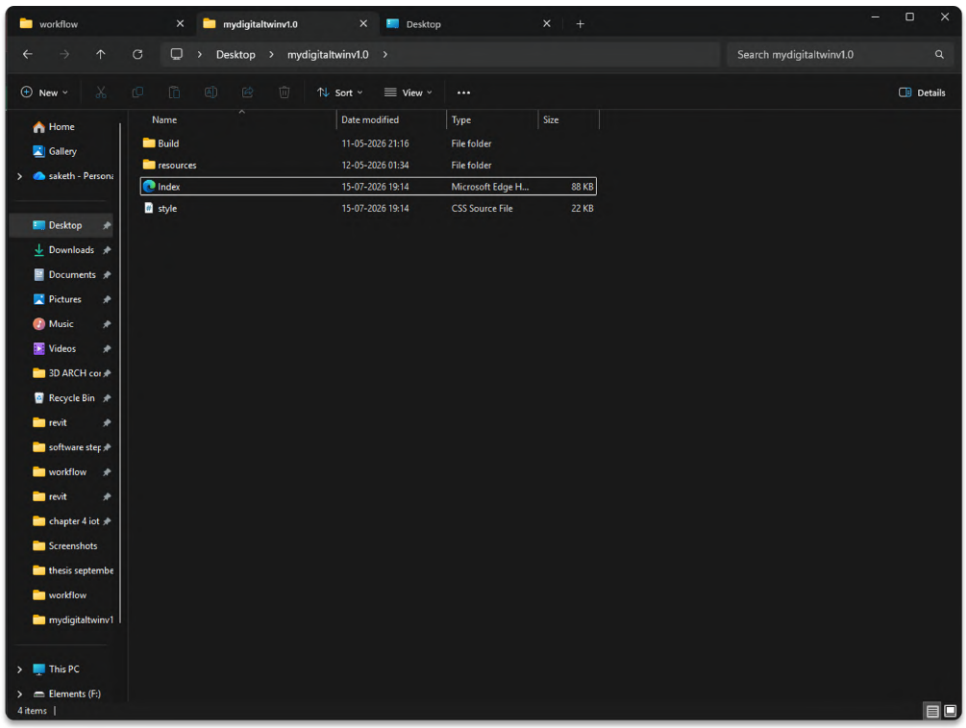


Figure 5.11: *Unity game engine, Build profile exported files*

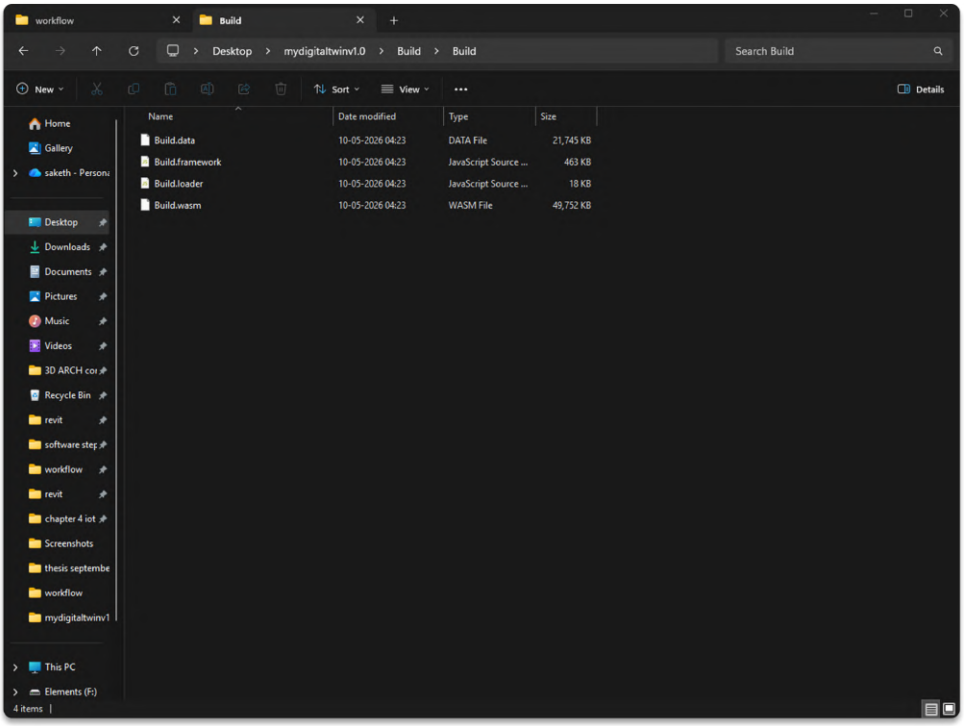


Figure 5.12: *Build WebAssembly Binary & Javascript loader files*

The compilation process automatically generates three primary web assets:

WebAssembly binary (raw_built_dt.wasm): The compiled C# logic and engine code, enabling high performance, near native execution directly within the browser's graphics engine. (Figure: 5.12)

JavaScript loader (mydigitaltwinv1.0.loader.js): A wrapper script responsible for instantiating the Unity engine canvas, managing memory allocation, and monitoring build startup progress. (Figure: 5.12)

Core HTML & CSS web wrapper (index.html & style.css): The structural canvas and layout stylesheet that host the application. (Figure: 5.11)

Rather than viewing the WebGL canvas in isolation, a custom web dashboard architecture was developed around the central HTML wrapper. Using a responsive CSS grid and flexbox layout, the single page application wraps the 3D WebGL viewport inside a broader BIMmodel interface. This architecture separates the user interface into distinct functional layers: a fixed navigation sidebar, an interactive top widget bar (displaying time, weather, and system status), collapsible metadata panels, and full width chart display grids. (Figure: 5.13) the Unity engine canvas is maintained at native resolution in the background while its visual visibility, opacity, and mouse interaction states are managed dynamically as the user navigates across different dashboard pages.

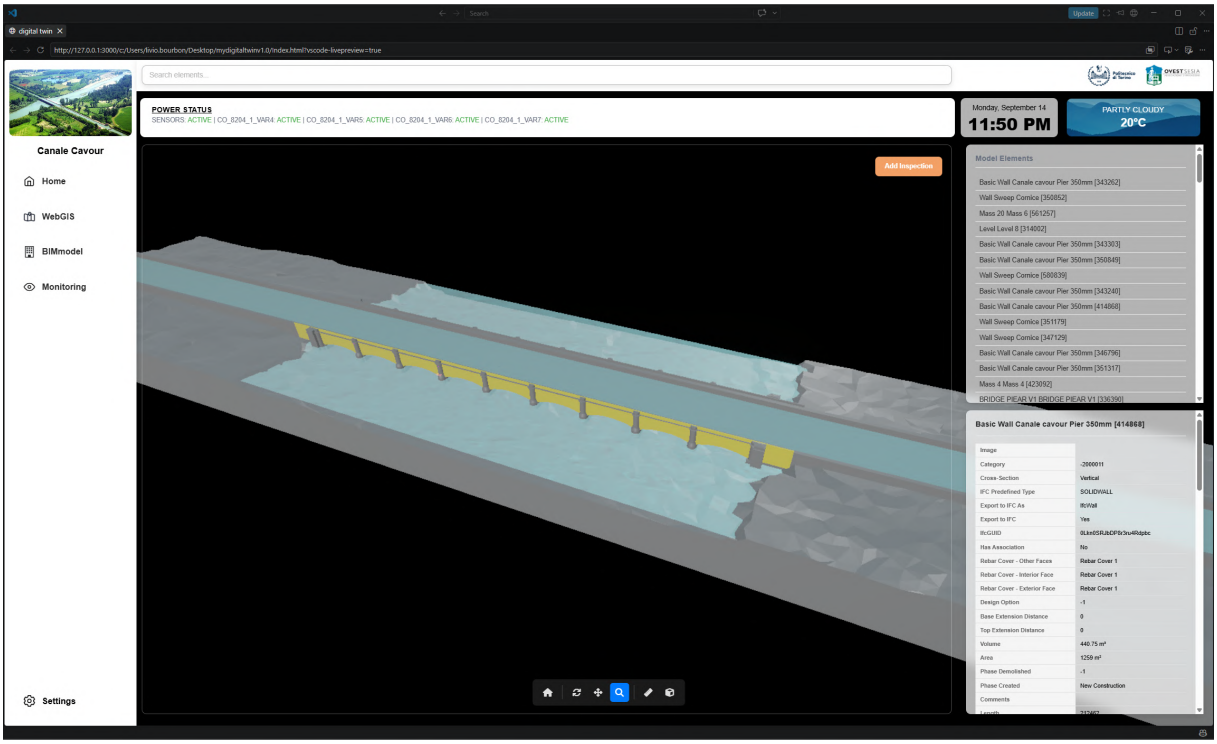


Figure 5.13: *Distinct user interface*

5.3.6 Engine to browser communication architecture

Because the Unity WebGL engine runs inside an isolated WebAssembly sandbox within the browser, a dedicated, bi directional communication bridge was implemented to pass data seamlessly between the C# engine environment and the external JavaScript web dashboard. (Figure: 5.14)



Figure 5.14: *Unity game engine, C# scripts used in the project*

Browser to engine communication (JavaScript to C#)

To send commands from web dashboard elements (such as UI buttons, range sliders, or search inputs) into the active 3D model, the web application utilizes Unity's built in messaging interface:

$$unityInstance.SendMessage(TargetGameObject, MethodName, Parameter)$$

When a user interacts with a web control such as clicking a camera control button or searching for a BIM element a JavaScript event listener fires this function. The message targets a specific C# script attached to an active GameObject in the 3D scene, invoking the named function and passing parameters such as element IDs, numerical camera speeds, or JSON strings containing user configured alarm limits.

Engine to browser communication (C# to JavaScript)

To send updates from the 3D scene back out to the web dashboard (such as spatial clicks, model element lists, or live virtual camera imagery), C sharp scripts utilize external JavaScript function declarations. By referencing native JavaScript bridge functions using [DllImport("_Internal")], C# methods can directly trigger JavaScript functions operating in the host web page:

Structural hierarchy synchronization: Upon loading the scene, BIMElementManager.cs compiles all 3D element tags into a string array and calls SendHierarchyToWeb(), passing the JSON data to populate the web dashboard's sidebar element list.

3D Mouse click interseption: Clicking a physical component directly inside the 3D viewport triggers OnMouseDown() in BIMElementManager.cs, which invokes SendClickToWeb(), informing JavaScript to query the associated BIM database and display object parameters in the web sidebar.

Streaming virtual visual feeds: SensorCameraBridge.cs extracts frame pixel data from cameras placed inside the 3D scene, encodes the images into Base64 strings, and calls PushCameraFrame(), instantly updating <img> tags on the monitoring dashboard to function as live virtual CCTV channels.

5.3.7 System visualisation & interactive workflows

To provide an intuitive interface for structural monitoring, the digital shadow web platform structures its operations across five dedicated pages. Each page addresses a specific operational requirement ranging from high level portfolio overviews to detailed 3D inspections, spatial mapping, live analytical graphing, and threshold configurations. Underlying these interfaces is a coordinated network of web components, external databases, and specialized Unity C# scripts (Figure: 5.10). The following breakdown outlines the functional role, underlying files, and data workflows associated with each page.

Home dashboard Serves as the central operational hub, summarizing bridge statistics, environmental conditions, and live sensor health.

- Files Used: index.html, style.css, HTTPReceiver.cs, MQTTReceiver.cs, MqttBridge.cs, UnityMainThreadDispatcher.cs.
- Operational Focus: Processes incoming sensor streams (tilt, displacement, temperature) to maintain continuous system awareness.

- Web to 3D Interaction: Clicking shortcut cards triggers web navigation. Selecting the 3D BIM Model shortcut reveals the 3D viewport and enables mouse interactions.
- 3D to Web Updates:On load, the 3D scene transmits a structural element inventory to the browser, updating the total component count. Incoming sensor data updates the home monitoring table and toggles green, orange, or red status indicators.

WebGIS spatial view Provides regional and territorial context by isolating spatial mapping from the 3D graphics engine.

- Files Used: index.html, style.css, Mapbox GL JS API.
- Operational Focus: Uses a 2D mapping framework to display regional canal networks, waterways, and terrain.
- Web to 3D Interaction: Navigating to WebGIS completely hides the 3D viewport and disables its mouse input to ensure smooth map interactions.
- 3D to Web Updates: Operating independently of the 3D pipeline, clicking map layers queries geographical attributes (e.g., canal names or zones) in the right hand panel.

BIM model page Acts as the visual inspection hub, pairing 3D model navigation with BIM engineering data and defect tracking.

- Files Used: index.html, style.css, bim_data.json, BIMElementManager.cs, WebController.cs, InspectionMarker.cs, InspectionMarkerManager.cs.
- Operational Focus: Manages component highlights, automatic camera focusing, virtual defect placement, and viewport controls.
- Web to 3D Interaction:
- View Controls: Toolbar buttons send direct camera commands (Orbit, Pan, Zoom, Reset View) to the 3D engine.
- Search & Selection: Choosing a component from the sidebar list or search bar highlights the geometry in yellow and zooms the camera onto it.
- Reset: Pressing Escape or clicking off the model clears highlights and resets the view.
- 3D to Web Updates:Loads the 3D structural hierarchy directly into the web sidebar.
- Clicking a 3D component opens its physical engineering parameters in an editable web panel.
- Clicking a 3D inspection marker displays recorded defect notes, severity tags, and timestamps.

Monitoring page Functions as the live structural health center, combining real time mathematical charts with virtual camera streams.

- Files Used: index.html, style.css, Chart.js, SensorCameraBridge.cs, MQTTReceiver.cs, MqttBridge.cs, SensorRotationController.cs.

- Operational Focus: Streams compressed visual feeds from virtual cameras inside the 3D model while processing alert thresholds.
- Web to 3D Interaction:Viewport Alignment: Calculates screen card dimensions and aligns virtual 3D camera angles to match the web layout.
- Threshold Limits: Submitting new safety limits sends updated alarm ranges into the 3D scene.
- 3D to Web Updates:Virtual cameras inside the 3D model stream live visual frames directly to the dashboard viewports like virtual CCTV feeds. Web charts plot incoming data, draw min/max limit lines, and dynamically color graph borders (green, orange, red).

Settings page Provides a control panel for tuning 3D viewer behavior and configuring safety alert thresholds.

- Files Used: index.html, style.css, WebController.cs, BIMElementManager.cs, MQTTReceiver.cs.
- Operational Focus: Receives user adjustments for camera sensitivity, zoom framing buffers, and safety alert bounds.
- Web to 3D Interaction:3D Tuning: Sliders adjust camera distance, pan/orbit speed, scroll zoom sensitivity, and BIM focus framing buffers in real time.
- Threshold Limits: Entering min/max values and clicking "Update Unity" pushes JSON configuration packages to the 3D engine.
- Input Isolation: Focusing on numeric input fields temporarily disables 3D keyboard shortcuts to prevent accidental camera movement.
- 3D to Web Updates: Submitting parameters instantly updates live chart threshold lines; the update button briefly flashes green with a "SUCCESSFUL" confirmation.

Chapter 6

Experimental Results

6.1 Visualisation in Autodesk Revit using Dynamo

At the Visualization Layer, the HBIM model served as the central interface for data representation. A Dynamo script was created to retrieve live sensor data via HTTP requests using a web request node (Figure: 6.1). The returned JSON data were parsed and mapped to corresponding BIM sensor elements using instance parameters embedded within the sensor families. VAR4, VAR5, VAR6, VAR7 of sensor information from the real world are mapped to the sensor Autodesk Revit family in the virtual world. As previously mentioned, for visualisation reasons, only VAR4 (the temperature sensor) is chosen, to show the variation and change of colour in the model element in Autodesk Revit.

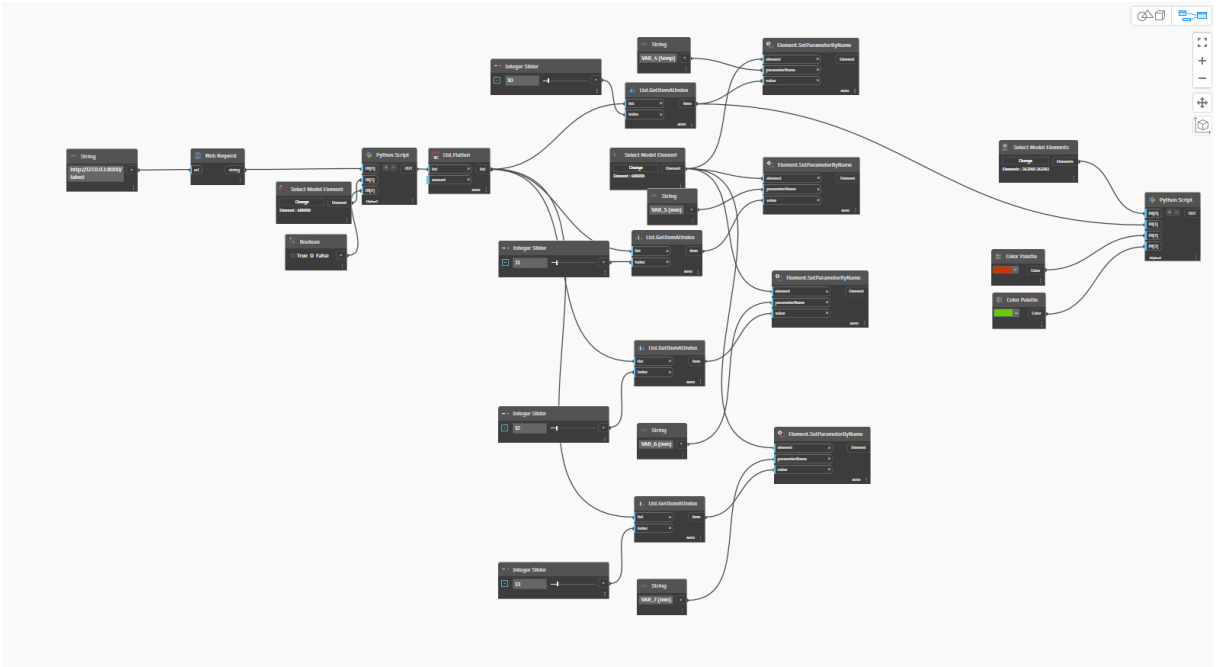


Figure 6.1: *Dynamo script to retrieve data from a web request and set the data as parameters.*

In particular, the parameter VAR4 was used to store temperature related values. A Python node within Dynamo was employed to evaluate the incoming data and apply rule based visualisation logic. Sensor elements were colour coded based on temperature thresholds, with

values below 6 °C displayed in green and values exceeding 6 °C displayed in red. This visual feedback enabled immediate identification of potential thermal anomalies directly within the BIM environment (Figure: 6.2).

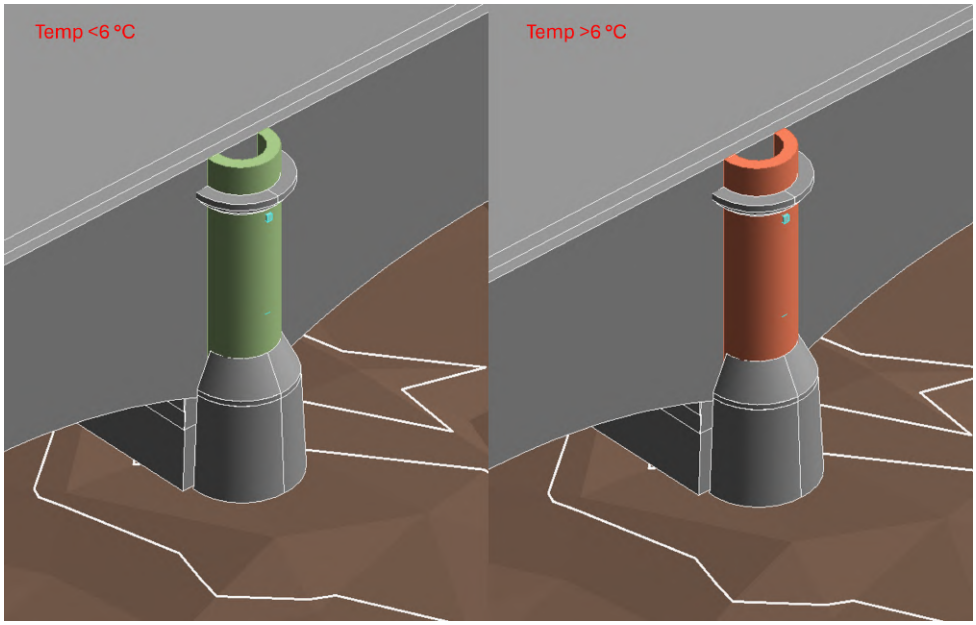


Figure 6.2: *Automatic change of element colour based on the temperature parameter in the Autodesk Revit family.*

The Dynamo workflow was configured to execute periodically with 1000ms, allowing the BIM model to update automatically as new data became available. This approach transforms the static As is BIM into a dynamic, data driven representation, supporting the conceptual implementation of a digital shadow for the Cavour Canal water bridge.

Dynamo nodes. With respect to (Figure: 6.1), this section provides a detailed description of the main nodes. (Figure: 6.3) shows the nodes from 1 to 4.

- 1. Node 1. String: text based node input, the HTTP link used to publish the JSON live data.
- 2. Node 2. Web Request: retrieve data from the JSON link.
- 3. Node 3. Python script: A custom Python script was developed to read the JSON file from the web request and parse the selected element information from the select model element node. Three inputs are necessary for the Python script: the JSON data from the web request, the selected model element, and a Boolean value (true or false). The final output consists of two lists: the first is the name of the data container, and the second is the data to be set to the element.
- 4. Node 4. List Flatten: it is used to flatten two lists into a single list to set the parameters for the following nodes.

  Figure 6.4 shows nodes 5 and 6.
- 5. Node 5. List get item at index: this node allows for getting specific information from the given list (Figure: **??**).

- 6. Node 6. Element set parameter by name: this node sets the given value to the selected element parameter on the bases of the parameter name. It has three inputs: the "element" input is given with the select model element node, the "parameter name" is given with the string node, which is case sensitive and should match the parameter name created in the Autodesk Revit model, and the "value" is the data requested to set to the parameter. In this case, it is from the web request temperature sensor value.

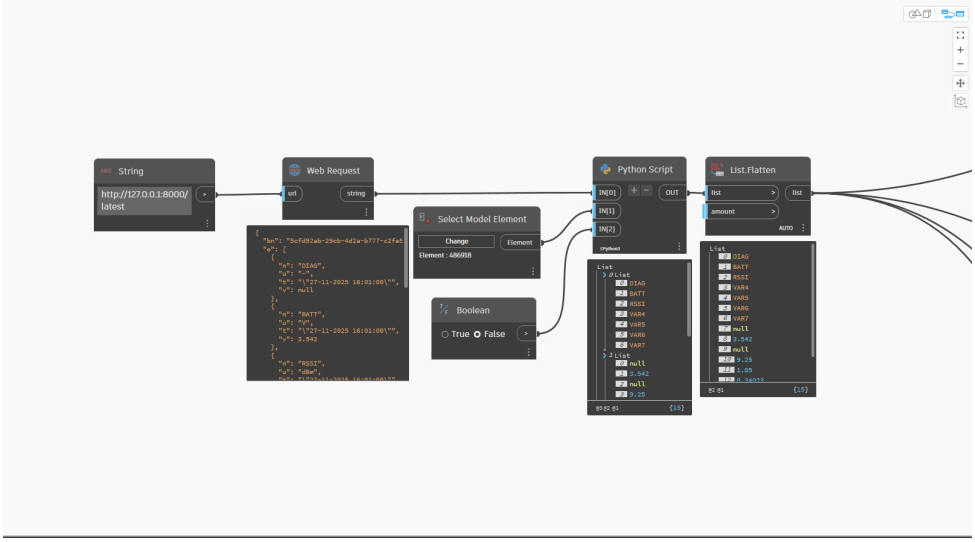


Figure 6.3: *Dynamo script showing the connection of String, Web request, Python script and List flatten nodes.*

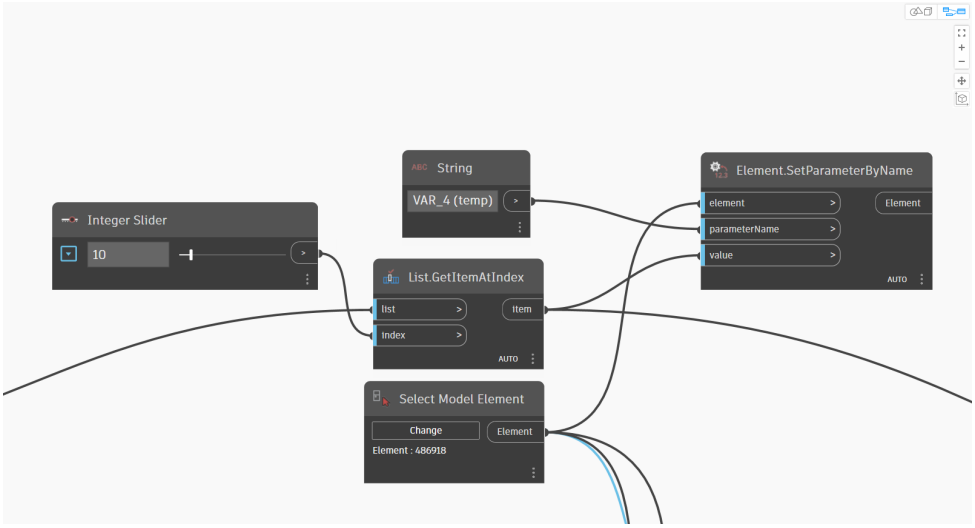


Figure 6.4: *Dynamo script showing the connection of List get item at index and Set parameter by name nodes.*

7. Node 7. Python script node (Figure: 6.5): this custom Python script node triggers the temperature value, which is retrieved from the flattened list, and if this value is less than a certain threshold (6 in this specific case), the colour code assigned to the selected element is green; if higher than 6, the colour code will be red. This node has four inputs: "input 0" to

select the model element pier, “input 1” to get the temperature value that triggers the colour change, “input 2” for the red colour palette, and “input 3” for the green colour palette.

The colours and thresholds could be arbitrarily set and constitute a future work of this research. In fact, in collaboration with structural monitoring experts, the most appropriate value ranges for the crack meters and inclinometers will be defined, and the visualisation in the HBIM model will be updated accordingly.

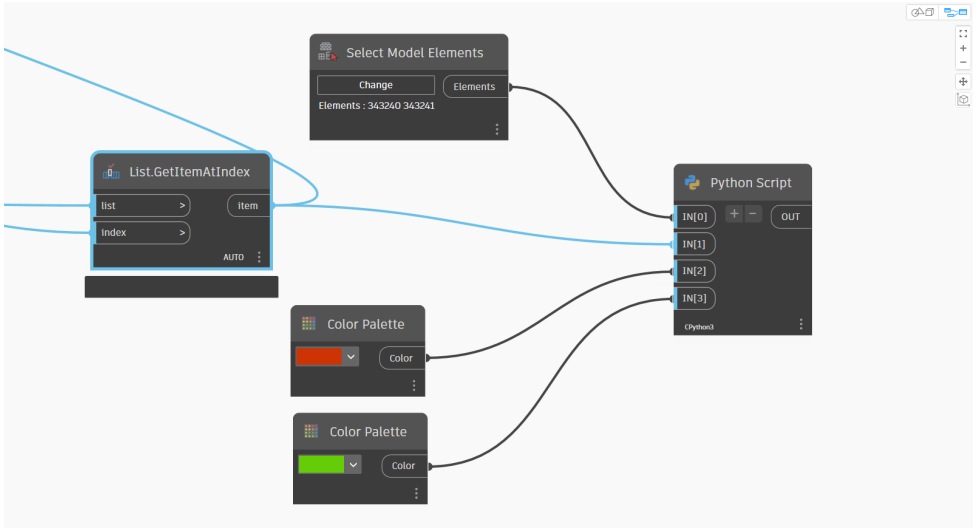


Figure 6.5: *Dynamo script showing the connection of the Python script and its inputs to change the colours of the selected element in the model.*

6.2 Web platform visualisation using Unity game engine

To provide domain experts, structural engineers, and site managers with an intuitive point of access, the digital shadow framework wraps its real time 3D WebGL engine within a comprehensive single page web application dashboard. Rather than treating the 3D model as an isolated graphical asset, the web platform embeds the Unity rendering canvas alongside dynamic spatial mapping, real time sensor charts, structural hierarchy lists, and operational control panels.

This architecture eliminates the need for specialized desktop software, enabling web browser based monitoring and inspection across standard devices. The web platform is organized into five specialized workspace modules, accessible via a fixed navigation sidebar.

This section presents a complete walkthrough of the web dashboard interface, detailing the visual layout, user workflow, and operational capabilities of each page as captured in the deployed system:

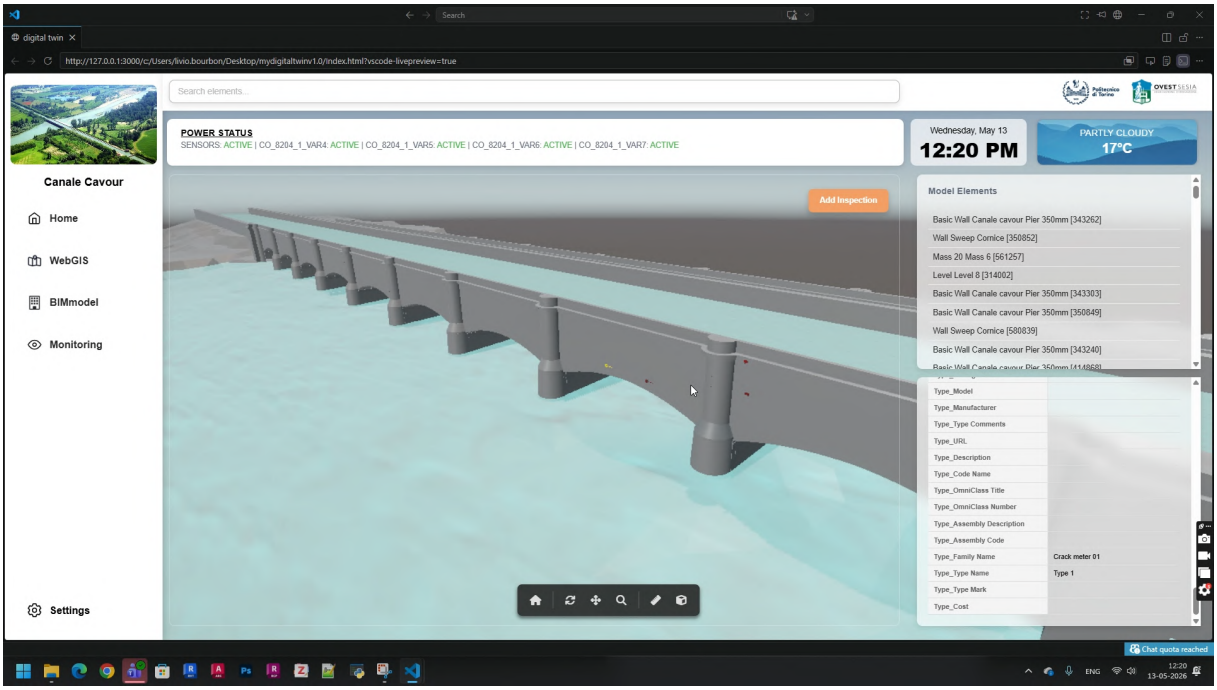


Figure 6.6: *digital shadow web platform*

This thesis presented a methodology for integrating IoT based monitoring data into an HBIM model to create a digital shadow for the preventive conservation of historical hydraulic infrastructure.

6.2.1 Home page

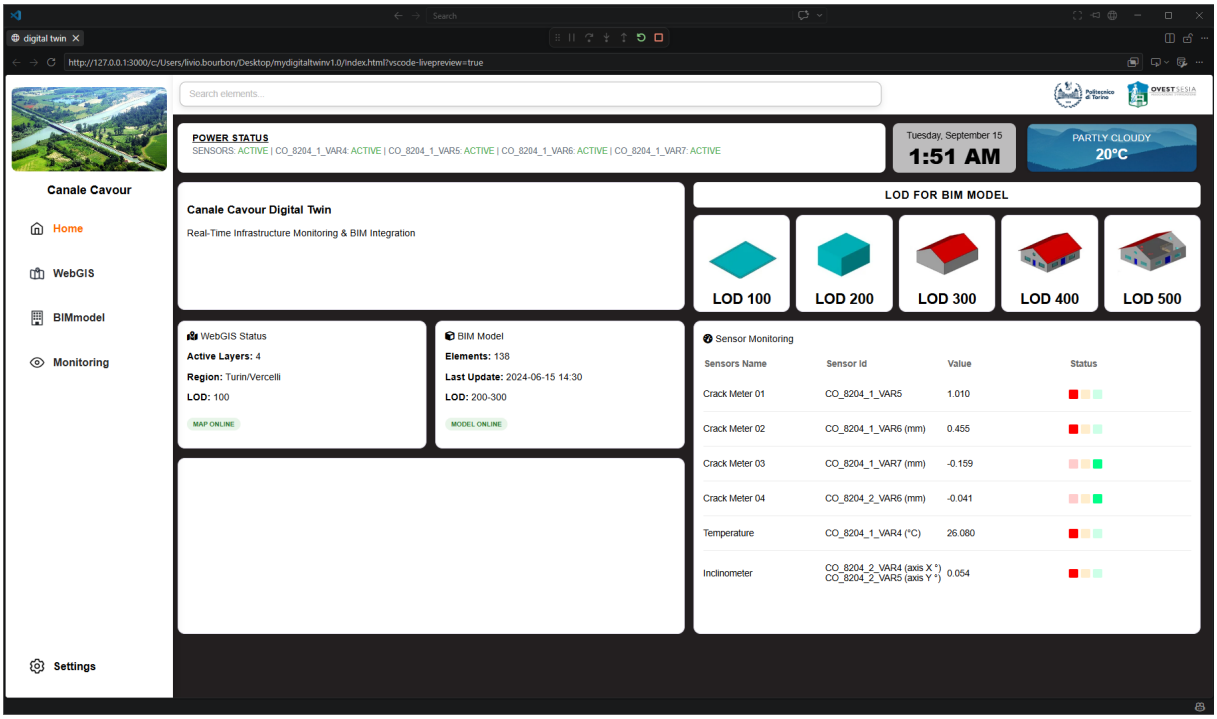


Figure 6.7: *digital shadow web platform, Home page*

Serves as the main operational overview dashboard, providing a quick summary of the bridge's health, system status, and model parameters:

Top status widgets: Displays power status across active sensors, current date/time, local weather conditions, and a global element search bar.

System overview cards: Quick summary boxes displaying active WebGIS layers and basic BIM model information.

Level of Detail (LOD) indicators: Displays visual cards representing the model's geometric detail levels (LOD 100 to 500).

Live sensor monitoring table: A summarized list of physical sensors (crack meters, temperature gauge, and inclinometers) showing current readings and simple color coded status dots (green, orange, red) to indicate safety levels.

6.2.2 WebGIS page

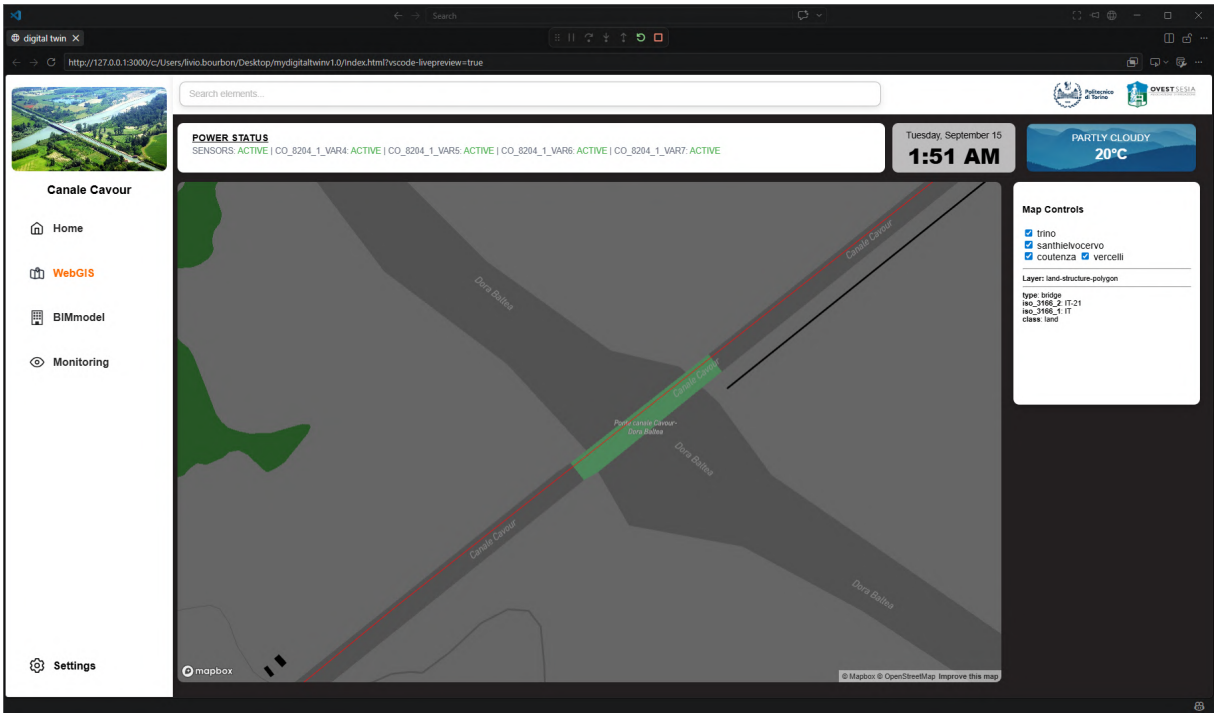


Figure 6.8: *digital shadow web platform, WebGIS page*

Provides dynamic geographical context by embedding a Mapbox satellite and topographic map:

Geographical mapping: Displays the canal infrastructure, regional bridge location, surrounding terrain, and waterways.

Map controls sidebar: Located on the right, allowing operators to toggle individual GIS spatial layers (such as regional canal networks) on or off and view metadata by clicking directly on map features.

6.2.3 BIMmodel page

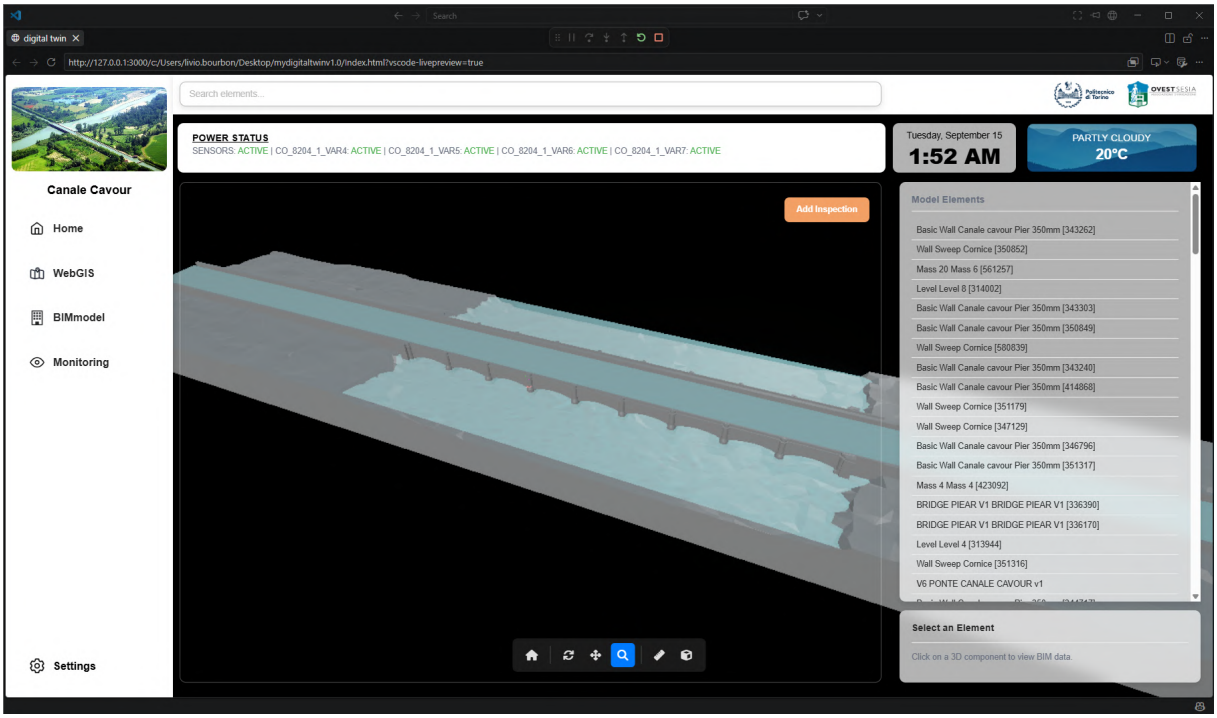


Figure 6.9: *digital shadow web platform, BIMmodel page*

Hosts the interactive 3D WebGL viewport for inspecting the physical geometry and BIM metadata of the structure:

Interactive 3D viewport: Renders the 3D bridge structure and surrounding channel environment. Users can orbit, pan, zoom, or click components directly.

3D navigation toolbar: Positioned at the bottom center of the viewport to quickly reset views or switch camera modes.

Model hierarchy & BIM panel: Located on the right, listing every structural component in the 3D model. Selecting an element from this list or clicking it in 3D automatically focuses the camera and displays its physical parameters (dimensions, materials, and structural properties) in the metadata box below.

Add inspection button (underdevelopment): Prompts an overlay panel for placing interactive 3D defect markers (e.g., recorded cracks) directly onto the model

6.2.4 Monitoring page

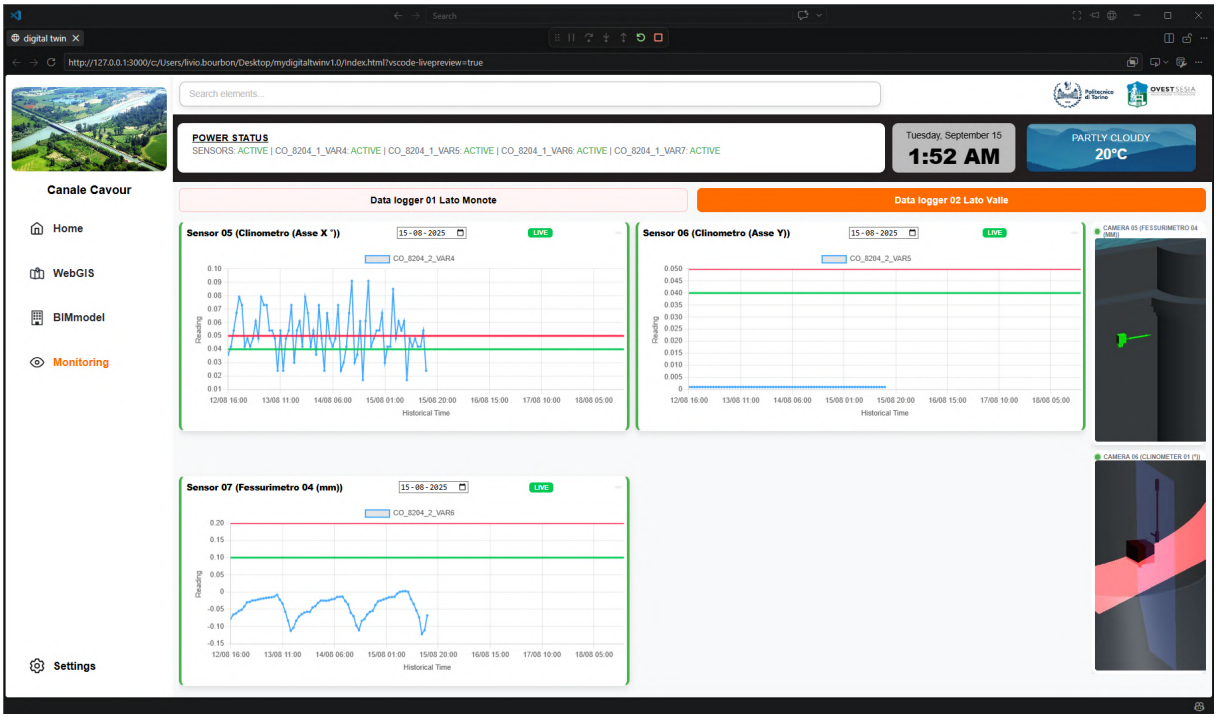


Figure 6.10: *digital shadow web platform, Monitoring page*

Acts as the main live data analytics dashboard, organized into two logger views via top tab buttons:

Data logger 01 View : Features live time series graphs for Crack Meters 01, 03, 04, and Temperature 01. Horizontal green and red threshold lines indicate safe operating bounds directly on the plots.

Data logger 02 View : Displays live telemetry graphs for Clinometer structural tilt (X and Y axes) and additional structural displacement sensors.

Virtual CCTV Sidebars: Positioned along the right side of both logger views, showing live video viewports captured directly from virtual cameras positioned around key inspection points on the 3D bridge model.

6.2.5 Settings page

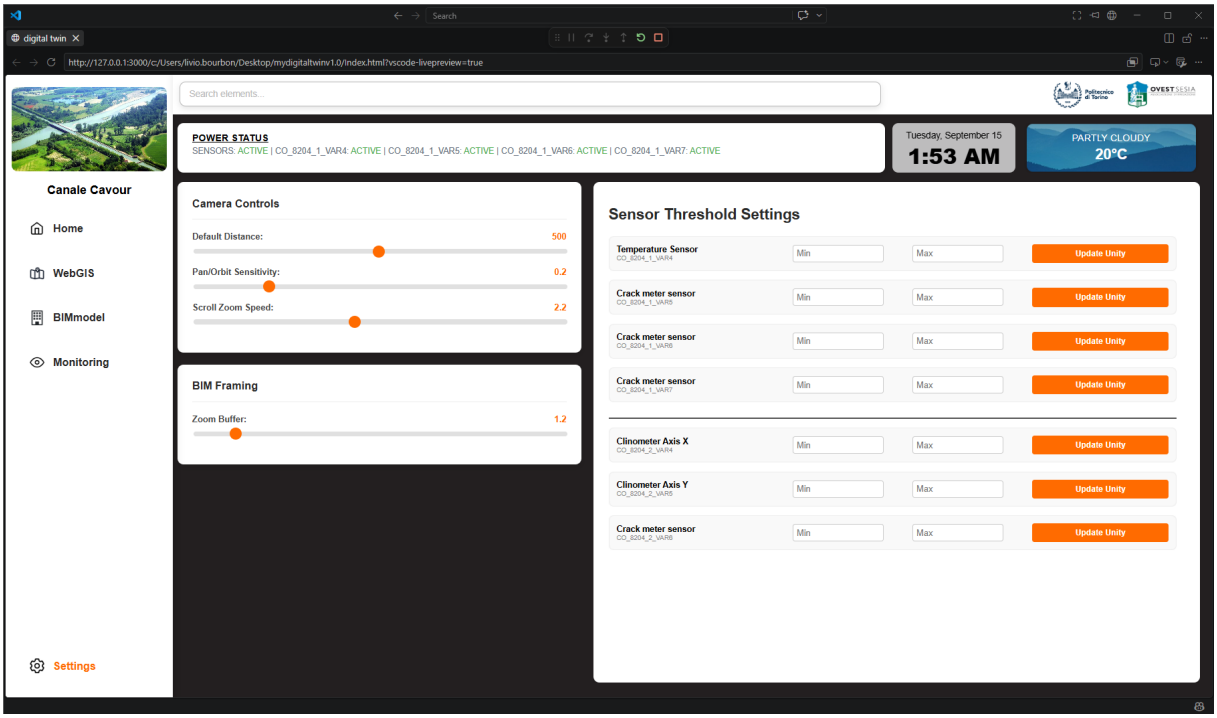


Figure 6.11: *digital shadow web platform, Settings page*

Provides a user configuration panel to fine tune 3D navigation and customize alert parameters:

Camera & BIM framing controls: Interactive sliders on the left allow operators to adjust default view distance, panning/orbiting sensitivity, scroll zoom speed, and camera focus framing buffers.

Sensor threshold settings: Form fields on the right where operators can input custom minimum and maximum safety limits for each sensor. Clicking "Update Unity" pushes these values into the live 3D engine, updating threshold warning lines and color indicators across the platform in real time.

Chapter 7

Conclusions

This thesis set out to determine whether a historic masonry structure could be placed under continuous, quantitative observation without recourse to invasive intervention. The Ponte Canale Cavour, the first and largest water bridge in Italy, served as the case study for a methodology integrating geomatic survey, Historic Building Information Modelling (HBIM), and IoT based structural health monitoring within a single, browser accessible digital shadow. An integrated 3D survey combining Terrestrial Laser Scanning, UAV photogrammetry, and mobile mapping achieved registration accuracies of 0.002–0.007 m, providing the metric foundation for a parametric HBIM model built in Autodesk Revit using custom families to capture the bridge's non-standard historical geometry. A network of crack meters, inclinometers, and temperature sensors was then integrated into this model through a layered device–middleware–visualisation architecture, evolving from an initial Dynamo based desktop prototype into a Unity driven, web based platform that makes live structural data accessible without specialist software. The resulting system fulfils the objective set out at the start of this work: a web based model able to detect and communicate critical structural conditions to the responsible authority in near real time.

This methodology is consistent with, and extends, established research on Scan to HBIM workflows for heritage structures and on the broader shift from periodic manual inspection toward continuous IoT based monitoring. this thesis explicitly identifies its platform as a digital shadow, clarifying what such a system can and cannot be relied upon to do and offering a clearer reference point for classifying comparable HBIM–IoT platforms reported elsewhere.

The principal achievement of this work is demonstrating, on a real and structurally compromised asset, that survey grade HBIM can be coupled to live sensor telemetry and delivered through an accessible interface usable by non-specialist stakeholders as well as engineers. The modular, non-proprietary system architecture has direct practical value for conservation practice, where monitoring commitments span decades and cannot depend on a single vendor's product lifecycle; the migration to a web platform demonstrates a transferable route to broadening access to monitoring data; and the semantic linkage between each sensor reading and a specific structural element establishes a traceable connection between abstract data and physical structure that is a prerequisite for any future move toward automated maintenance scheduling. The one directional nature of the system was a deliberate design choice, reflecting the principle that decisions to intervene on a protected structure must remain human led.

These achievements are nonetheless bounded by clear limitations. The methodology was validated on a single case study, leaving its transferability to other structures untested; the

sensor set captures only surface observable behaviour, with no insight into subsurface conditions such as moisture migration or internal degradation; the alert thresholds are a first approximation rather than a calibrated structural model;

the system, as a digital shadow stops at detection rather than triggering a scheduled response. Addressing this last point through integration with the 4MAIN10ANCE planned conservation platform is the most immediate priority for future work, alongside validating the methodology on additional heritage assets, incorporating thermal and multispectral diagnostics to assess subsurface conditions, refining alert thresholds with structural health monitoring specialists, and continuing to build the platform on open, interoperable components. Together, these steps would move the system from a validated proof of concept toward a mature, transferable tool for the proactive conservation of historic infrastructure.

Chapter 8

Future Devleopments

The system presented in this thesis should be understood as a working foundation rather than a finished product. Several directions naturally follow from where this project leaves off, and it's worth outlining them here both to situate this work within a longer-term research agenda, and to be transparent about what remains open.

Connecting the digital shadow to planned maintenance. Perhaps the most immediate and concrete next step is integrating this Unity-based platform with 4MAIN10ANCE, a planned-conservation and sustainable heritage management tool. This collaboration, to be pursued together with Prof. Marco Zerbinatti of the Department of Structural, Building and Geotechnical Engineering (DISEG) at Politecnico di Torino, would close a gap that the current system leaves open by design: today, the platform can detect and flag an anomaly, but it stops there someone still has to decide what to do about it and schedule the intervention manually. Linking the real-time IoT monitoring layer developed here directly to 4MAIN10ANCE's structured maintenance protocols would turn detection into action, giving conservation teams a direct line from "something changed on the bridge" to "here is the maintenance task that follows from it."

Testing the methodology beyond a single case study. This thesis focused deliberately on one structure the Cavour Canal water bridge precisely so the workflow could be developed and validated in depth. The natural next step is breadth, applying the same digital-shadow methodology to other cultural heritage assets, with different construction typologies, different sensor requirements, and different institutional contexts. Doing so would also mean testing the platform with a wider range of end users than this project could reach on its own multidisciplinary conservation teams.

Seeing beneath the surface. The current sensor set captures structural behavior displacement, tilt, temperature but says little about what's happening inside the masonry itself. A meaningful next layer of development would incorporate thermal imaging and multispectral orthophotos directly into the 3D WebGL environment, allowing the model to visualize things like subsurface moisture migration, material degradation, and internal defects that aren't visible, or aren't yet visible, from the outside.

Making the alert thresholds smarter. The green/orange/red thresholds used throughout this system were a necessary starting point, but they are, admittedly, a relatively simple first approximation. As more sensor history accumulates, there is a real opportunity to work with structural health monitoring specialists

Keeping the system open. Finally, and underlying all of the above, is a commitment to keeping this framework open rather than letting it calcify into something only its original author

can maintain. That means continuing to broaden the underlying data pipeline to support additional IoT communication standards beyond the ones used here, improving how data is cached and handled on the browser side for better performance, and ensuring the platform remains interoperable across different web environments. This is, in many ways, a continuation of the same modularity principle that shaped the system from the beginning the goal was never to build a finished tool, but to build one that could keep evolving.

Bibliography

[1] Huachun Cui and Jiawei Wu. «How Architectural Heritage Is Moving to Smart: A Systematic Review of HBIM». In: *Buildings* 15.15 (2025), p. 2664 (cit. on p. 1).

[2] Letizia Martinelli, Filippo Calcerano, Francesco Adinolfi, Dario Chianetta, and Elena Gigliarelli. «Open HBIM-IoT Monitoring Platform for the Management of Historical Sites and Museums. An Application to the Bourbon Royal Site of Carditello». In: *International Journal of Architectural Heritage* 19.2 (2025), pp. 153–170 (cit. on p. 1).

[3] C. Baratti. *I Fontanili del Novarese*. Novara: Associazione Irrigazione Est Sesia, 1997 (cit. on p. 2).

[4] Hanjin Kim, Chang-Uk Hyun, Hyeong-Dong Park, and Jongmun Cha. «Image Mapping Accuracy Evaluation Using UAV with Standalone, Differential (RTK), and PPP GNSS Positioning Techniques in an Abandoned Mine Site». In: *Sensors* 23.13 (2023), p. 5789. DOI: 10.3390/s23135789 (cit. on p. 6).

[5] Bernhard Hofmann-Wellenhof, Herbert Lichtenegger, and Elmar Wasle. *GNSS – Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and More*. Vienna: Springer-Verlag, 2008. ISBN: 978-3-211-73012-6. DOI: 10.1007/978-3-211-73017-1 (cit. on p. 6).

[6] Peter J. G. Teunissen and Oliver Montenbruck, eds. *Springer Handbook of Global Navigation Satellite Systems*. Springer International Publishing, 2017. ISBN: 978-3-319-42926-7. DOI: 10.1007/978-3-319-42928-1 (cit. on p. 6).

[7] Jianguo Zhou, Linshu He, and Haitao Luo. «Real-Time Positioning Method for UAVs in Complex Structural Health Monitoring Scenarios». In: *Drones* 7.3 (2023), p. 199. DOI: 10.3390/drones7030199 (cit. on p. 7).

[8] Charles D. Ghilani. *Elementary Surveying: An Introduction to Geomatics*. 15th. Pearson, 2017. ISBN: 978-0134604657 (cit. on p. 7).

[9] Jean M. Rüeger. *Electronic Distance Measurement: An Introduction*. 4th. Berlin, Heidelberg: Springer-Verlag, 1996. ISBN: 978-3-540-61159-2. DOI: 10.1007/978-3-642-80233-1 (cit. on p. 7).

[10] Jean Doumit. *From Drones to Geospatial Analysis*. ResearchGate, 2019. URL: https://www.researchgate.net/publication/330555127_From_drones_to_geospatial_analysis (cit. on p. 8).

[11] Treesa Sebastian. *Laser Scanning and the Different Types of Laser Scanners*. https://bimexeng.com/blog/technology/laser-scanning-and-the-different-types-of-laser-scanners. Accessed: 2026-09-15. Dec. 2021 (cit. on p. 10).

[12] George Vosselman and Hans-Gerd Maas, eds. *Airborne and Terrestrial Laser Scanning*. Dunbeath, UK: Whittles Publishing, 2010. ISBN: 978-1904445-87-6 (cit. on p. 10).

[13] Historic England. *3D Laser Scanning for Heritage: Advice and Guidance on the Use of Laser Scanning in Archaeology and Architecture*. 3rd. Swindon, UK: Historic England Publishing, 2018. ISBN: 978-1848021-12-9 (cit. on p. 10).

[14] Maurice Murphy, Eugene McGovern, and Sara Pavia. «Historic building information modelling (HBIM)». In: *Structural Survey* 27.4 (2009), pp. 311–327 (cit. on pp. 44, 45).

[15] S. Bruno, A. Musicco, L. Fatiguso, and G. R. Dell'Osso. «An Efficient Pipeline to Obtain 3D Model for HBIM and Structural Analysis from Point Clouds». In: *Applied Sciences* 10.4 (2020), p. 1235 (cit. on p. 44).

[16] Carlos León-Robles, Juan Francisco Reinoso-Gordo, José González-Tello, and Antonio Gómez-Blanco. «Heritage Building Information Modeling (H-BIM) Applied to a Stone Bridge». In: *ISPRS International Journal of Geo-Information* 8.3 (2019), p. 121 (cit. on p. 45).

[17] Fabrizio Banfi. «The integration of a Scan-to-HBIM process in BIM application: The development of an add-in to guide users in Autodesk Revit». In: *Applied Sciences* 9.7 (2019), p. 1497. DOI: 10.3390/app9071497 (cit. on p. 45).

[18] Lucy J. Lovell, Richard J. Davies, and Dexter V. L. Hunt. «The Application of Historic Building Information Modelling (HBIM) to Cultural Heritage: A Review». In: *Heritage* 6.10 (2023), pp. 6691–6717. DOI: 10.3390/heritage6100350 (cit. on p. 45).

[19] Maurice Murphy, Eugene McGovern, and Sara Pavia. «Historic Building Information Modelling – Adding intelligence to laser and image based surveys of European classical architecture». In: *ISPRS Journal of Photogrammetry and Remote Sensing* 76 (2013), pp. 89–102. DOI: 10.1016/j.isprsjprs.2012.11.006 (cit. on p. 46).

[20] Associazione Irrigazione Est Sesia. «Il Grande Canale Cavour: 150 anni». In: *Est Sesia* 62.119 (2016). URL: https://www.estsesia.it/wp-content/uploads/2019/10/EstSesia-119.pdf (cit. on pp. 47, 48).

[21] Studio Ingegneria Balossi Restelli e Associati. *Ponte del Canale Cavour sulla Dora Baltea: interventi di sottofondazione e rinforzo strutturale*. Project documentation. 1998. URL: https://www.balossirestelliassociati.it/progetti/restauro-del-ponte-del-canale-cavour-sulla-dora-baltea/ (cit. on p. 47).

[22] Comune di Saluggia. *Ponte di attraversamento del Canale Cavour sulla Dora Baltea*. Comune di Saluggia. 2026. URL: https://comune.saluggia.vc.it/vivere-il-comune/luoghi/ponte-di-attraversamento-del-canale-cavour-sulla-dora-baltea/ (cit. on p. 48).

[23] Yanyang Zhang, Enrico Tubaldi, Lorenzo Macorini, and Bassam A. Izzuddin. «Mesoscale partitioned modelling of masonry bridges allowing for arch-backfill interaction». In: *Construction and Building Materials* 173 (2018), pp. 820–842. DOI: 10.1016/j.conbuildmat.2018.03.272 (cit. on p. 48).

[24] Vincenzo Fioriti, Ivan Roselli, Angelo Tatı`, and Gerardo De Canio. «Internet of Things (IoT) for Masonry Structural Health Monitoring (SHM): Overview and Examples of Innovative Systems». In: *International Journal of Architectural Heritage* 16.8 (2022), pp. 1170–1182. DOI: 10.1080/15583058.2021.1894503 (cit. on p. 69).

[25] Jerome P. Lynch and Kenneth J. Loh. «A Summary Review of Wireless Sensors and Sensor Networks for Structural Health Monitoring». In: *Shock and Vibration Digest* 38.2 (2006), pp. 91–128. DOI: 10.1177/0583102406061499 (cit. on p. 69).

[26] C. Athanase Tokognon, Bin Gao, Gui Yun Tian, and Yina Yan. «Structural Health Monitoring Framework Based on Internet of Things: A Review». In: *IEEE Internet of Things Journal* 4.3 (2017), pp. 629–647. DOI: 10.1109/JIOT.2017.2664072 (cit. on p. 69).

[27] Carmelo Scuro, Pierluigi Sciammarella, Francesco Lamonaca, Renato S. Olivito, and Domenico L. Carnı`. «IoT-Based Architecture for Structural Health Monitoring and Management of Cultural Heritage». In: *IEEE Transactions on Instrumentation and Measurement* 67.8 (2018), pp. 1814–1823. DOI: 10.1109/TIM.2018.2809141 (cit. on pp. 69, 84).

[28] Paolo Barsocchi, Gianni Bartoli, Michele Betti, Maria Girlanda Girardi, Daniele Pellegrini, and Alice Scacco. «Wireless Sensor Networks for Continuous Structural Health Monitoring of Historic Masonry Towers». In: *International Journal of Architectural Heritage* 15.1 (2021), pp. 22–44. DOI: 10.1080/15583058.2020.1719229 (cit. on p. 69).

[29] Manojit Mishra, Anushree Bhatia, and Damodar Maity. «Structural Health Monitoring of Historical Masonry Structures: A Comprehensive Review». In: *Journal of Performance of Constructed Facilities* 35.4 (2021), p. 03121001. DOI: 10.1061/(ASCE)CF.1943-5509.0001602 (cit. on pp. 69, 84).

[30] National Instruments. *Measuring Temperature with Thermocouples, RTDs, and Thermistors*. https://www.ni.com/en/shop/data-acquisition/sensor-fundamentals/measuring-temperature.html. Accessed January 2026. 2025 (cit. on p. 78).

[31] Encardio Rite. *Inclinometer Types, Working Principle, and Applications*. https://www.encardio.com/blog/inclinometer-types-how-it-works-uses. Accessed January 2026. 2026 (cit. on p. 78).

[32] Sisgeo S.p.A. *Crackmeters and Jointmeters for Structural Monitoring*. https://sisgeo.com/product-category/crackmeters-jointmeters/. Accessed January 2026. 2025 (cit. on p. 78).

[33] National Instruments. *What is a Data Logger? Fundamentals and Use Cases*. https://www.ni.com/en/shop/data-acquisition/data-loggers.html. Accessed January 2026. 2025 (cit. on p. 79).

[34] Xiang Liu, Maxwell Fordjour Antwi-Afari, Jue Li, Yongcheng Zhang, and Patrick Manu. «BIM, IoT, and GIS integration in construction resource monitoring». In: *Automation in Construction* 174 (2025). Accessed January 14, 2026, p. 106149. DOI: 10.1016/j.autcon.2025.106149. URL: https://www.researchgate.net/publication/390442121_BIM_IoT_and_GIS_integration_in_construction_resource_monitoring (cit. on p. 79).

[35] Somit Panchal. *BIM and Big Data: Transforming the AEC Industry*. https://www.techture.global/blog/bim-and-big-data-integration. Accessed January 14, 2026. Jan. 2025 (cit. on p. 79).

[36] Hailay Gebremeskel. *Understand the protocols behind IoT messaging*. https://www.cloudamqp.com/blog/understand-the-protocols-behind-iot-messaging.html. CloudAMQP blog; Accessed January 2026. Dec. 2025 (cit. on pp. 79, 80).

[37] ThingDash. *MQTT vs AMQP Deciding on Your IoT Protocol*. https://www.thingdash.io/blog/mqtt-vs-amqp-deciding-on-your-iot-protocol. Accessed January 14, 2026. Aug. 2025 (cit. on p. 79).

[38] Tesla Outsourcing Services. *The Smart Building Revolution: How BIM and IoT Convergence is Redefining North American AEC*. https://www.teslaoutsourcingservices.com/blog/the-smart-building-revolution-how-bim-and-iot-convergence-is-redefining-north-american-aec/. Accessed January 14, 2026. Nov. 2025 (cit. on p. 80).

[39] Vikash Singh. *BIM In Construction: Integrating AI And IoT For Smarter Builds*. https://www.capstonees.com/blog/what-is-bim-in-construction-exploring-its-integration-with-ai-iot. Accessed January 14, 2026. Apr. 2025 (cit. on p. 80).

[40] Werner Kritzinger, Matthias Karner, Georg Traar, Jan Henjes, and Wilfried Sihn. «Digital Twin in manufacturing: A categorical literature review and classification». In: *IFAC-PapersOnLine* 51.11 (2018), pp. 1016–1022. DOI: 10.1016/j.ifacol.2018.08.474 (cit. on pp. 83, 84).

[41] Fei Tao, Meng Zhang, Yushan Liu, and A. Y. C. Nee. «Digital Twin Driven Smart Manufacturing: Methodology and Applications». In: *IEEE Transactions on Industrial Informatics* 14.4 (2018), pp. 1771–1780. DOI: 10.1109/TII.2017.2737880 (cit. on pp. 83, 84).

[42] Qiuchen Lu, Xiaojun Xie, Ruodan He, Ajith Kumar Parlikad, and Jennifer M. Schooling. «Digital twin-enabled anomaly detection for built asset management in operation and maintenance phase». In: *Journal of Management in Engineering* 36.6 (2020), p. 04020072. DOI: 10.1061/(ASCE)ME.1943-5479.0000850 (cit. on p. 83).

[43] Chen Boje, Annie Guerriero, Sylvain Kubicki, and Yacine Rezgui. «Towards a Semantic Construction Digital Twin: Directions for Future Research». In: *Automation in Construction* 114 (2020), p. 103179. DOI: 10.1016/j.autcon.2020.103179 (cit. on pp. 84, 86).

[44] Mengnan Liu, Shuai Fang, Hui Dong, and Chao Zhang. «Review of digital twin technology for structural health monitoring of bridges». In: *Structure and Infrastructure Engineering* 17.9 (2021), pp. 1285–1301. DOI: 10.1080/15732479.2020.1799070 (cit. on p. 84).

[45] Carlo Rainieri and Giovanni Fabbrocino. «Integration of Building Information Modeling and Real-Time Interactive Engines for Structural Health Monitoring: A Review». In: *Automation in Construction* 125 (2021), p. 103608. DOI: 10.1016/j.autcon.2021.103608 (cit. on p. 86).

[46] Michele Morganti, Matteo Del Giudice, and Anna Osello. «BIM-to-Game Engine Workflows for Cultural Heritage VR/AR Applications: Potentialities and Open Challenges». In: *Applied Sciences* 12.14 (2022), p. 7012. DOI: 10.3390/app12147012 (cit. on p. 86).

[47] World Wide Web Consortium (W3C). *The WebSocket API*. https://www.w3.org/TR/websockets/. W3C Candidate Recommendation. 2011 (cit. on p. 94).

[48] Chart.js Plugin Contributors. *chartjs-plugin-annotation: Dynamic annotations for Chart.js*. https://www.chartjs.org/chartjs-plugin-annotation/latest/. Version 3.0.1. 2024 (cit. on p. 94).

[49] Chart.js Development Team. *Chart.js: Simple yet flexible JavaScript charting for designers & developers*. https://www.chartjs.org/docs/latest/. Version 4.4.0. 2024 (cit. on p. 94).

[50] Chart.js Zoom Contributors. *chartjs-plugin-zoom: Pan and zoom plugin for Chart.js*. https://www.chartjs.org/chartjs-plugin-zoom/latest/. Version 2.0.1. 2024 (cit. on p. 94).

[51] Mapbox. *Mapbox GL JS API Reference*. https://docs.mapbox.com/mapbox-gl-js/api/. 2024 (cit. on p. 95).

[52] Open-Meteo. *Free Weather API: Open-Meteo Data Service*. https://open-meteo.com/en/docs. 2024 (cit. on p. 95).

[53] Giovanni Paolo Vigano. *M2MqttUnity: MQTT Client for Unity*. 2026. URL: https://github.com/gpvigano/M2MqttUnity (visited on 09/14/2026) (cit. on p. 95).