

Automating Built Heritage Modelling for the Integration into 3D City Models

*Original*

Automating Built Heritage Modelling for the Integration into 3D City Models / Aboohamzeh, A.; Avena, M.; Spano', ANTONIA TERESA. - In: INTERNATIONAL ARCHIVES OF THE PHOTOGRAMMETRY, REMOTE SENSING AND SPATIAL INFORMATION SCIENCES. - ISSN 1682-1750. - ELETTRONICO. - 48:4/W10-2024(2024), pp. 5-12. [10.5194/isprs-archives-XLVIII-4-W10-2024-5-2024]

*Availability:*

This version is available at: 11583/2990453 since: 2024-07-07T09:08:42Z

*Publisher:*

Copernicus - ISPRS

*Published*

DOI:10.5194/isprs-archives-XLVIII-4-W10-2024-5-2024

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

## Automating Built Heritage Modelling for the Integration into 3D City Models

Ali Aboohamzeh <sup>1\*</sup>, Marco Avena <sup>1</sup>, Antonia Spanò <sup>1</sup>

<sup>1</sup> Department of Architecture and Design (DAD), Politecnico di Torino, Torino, Italy - (ali.aboomhamzeh, marco.avena, antonia.spano@polito.it)

**KEYWORDS:** 3D modelling, 3D Python libraries, LoDs, automatic/semiautomatic 3D meshing, point cloud filtering, 3D geodatabases.

### ABSTRACT:

The interest in studies on 3D city models has renewed over time thanks to technological innovations that concern the methods of data acquisition and automatic classification of unstructured point clouds, to model increasingly specialized geometric objects for the growing needs of urban transformation management. Managing geometric and semantic information in cognitive systems, based on standards, increasingly open to the inquiries, analyses, and predictions that today are necessary to implement for future assessments, can digitally twin reality. The problem focused on in this work is the modelling phase, which requires efficient automatisms to make the production phase of geometric objects more sustainable, whether they are then destined to become parametric features in a BIM/CIM environment or 3D geometric objects of a geodatabase.

A comprehensive methodology for automating the generation of 3D meshes from point cloud data using Python code, with a focus on practical implementation is presented parallelly to evaluations regarding the generation of the geometric 3D contents of the urban environment representation, depending on whether or not it includes examples of architectural heritage (of all ages). Leveraging the Open3D library and NumPy, the methodology provides different experiences, tests, and related discussions in the field of architectural and urban 3D surface reconstruction and visualization. The paper outlines essential steps including data loading and preparation, meshing strategy selection, data processing using Poisson reconstruction, and mesh exportation and visualization. Additionally, a method for generating multiple Levels of Detail (LoD) meshes is introduced, enhancing the applicability of the approach for city modelling framework applications, including architectural elements of buildings. The presented Python code serves as a practical resource for implementing 3D mesh generation workflows seamlessly.

### 1. INTRODUCTION

3D city models are a vast area of research that has aroused intense interest for many years, because they involve different fields of innovation, from data acquisition to processes intended to create a replica of the city for the most varied uses (Biljecki et al. 2015), ranging from urban monitoring, for urban planning and administrative uses, energy efficiency, for climate change, for the flow mobility control of resources and people, for autonomous mobility etc.

The attention around 3D city models in the field of geoinformation is favored by the continuous innovation of sensors essentially airborne LiDAR (Vosselman & Maas 2010), but also oblique images (Haala et al. 2015) and MMS (Elhashash et al. 2022), as well as by processes that increasingly make use of automatic strategies based on ML and DL to facilitate the generation of geometric objects to integrate into 3D urban geodatabases (Ballouch et al. 2022, Farella et al. 2021, Treccani et al. 2022, ).

Among the major interests today we must also consider the scenario of *digital twinning* paradigms, which increasingly aim to transform 3D city models into cognitive systems, which in light of the large quantities of semantic contents they incorporate, are estimated with the prospect of constituting powerful tools of monitoring the present and of future predicting, with the aim of representing effective decision supports tools. (Singh et al. 2021)

Another aspect not to be underestimated is the increasingly close connection between geoinformation management strategies with the approaches of communities that develop and enhance the AEC sector and the related paradigms of modelling, representation, and visualization based on IFC Standard (Noardo et al. 2020), which embraces the building context, urban legacies through HBIM (Heritage building information Modelling), and also City Information Models (CIM).

For this reason, important developments derive from the evolution of standards, as certainly demonstrated by the comprehensive revision made to the CityGML model (Kutzner et al. 2020). This revision was undertaken to meet the growing demand for enhanced interoperability with other pertinent standards in the field, such as the Industry Foundation Classes (IFC). Additionally, this is illustrated by the extensive and in-depth studies that have been proposed in the field of indoor modelling. (Tang et al. 2018).

One of the relevant aspects that our work underlies is the awareness that not only the level of detail that the City GML standard has certainly enriched, the necessity of extended indoor LoD (iLoD) specification (Tang et al. 2018), the rich repository of legal and physical datasets that 3d cadastre aspects require (Atazadeh et al. 2019), but also the important issue of architectural heritage modelling, which obviously implies particular attention in the point clouds segmentation and classification, and also above all the semantic contents, which for all the

---

\* Corresponding author

functions required of the 3D city model must be declined according to the fragility and cultural values of the built heritage.

### 1.1. Aim of work

Starting from the framework described, the paper aims to present some modelling experiments of urban objects, among which built contexts belonging to the sphere of urban legacies or built heritage will be particularly privileged. Of particular interest will be the study of a semi-automatic method for generating models at different resolutions, starting from point clouds, which can satisfy the concept of modelling according to LoDs.

The process of generating 3D meshes from point clouds is crucial for various applications; above all, LoD-based urban modelling, despite that fields of study, including computer graphics, virtual reality, and geospatial analysis, also benefit from such an approach. Our aim leverages some targeted experimentation based on different sources and density point clouds to ascertain a comprehensive methodology for automating this process using a Python-based approach.

As an alternative to the use of some plugins enabled by commercial software such as Grasshopper (La Russa et al. 2023), the choice of leveraging the capabilities of the Open3D library and NumPy, enables to efficiently convert raw point cloud data into usable 3D mesh representations through an open tool.

## 2. METHODOLOGY

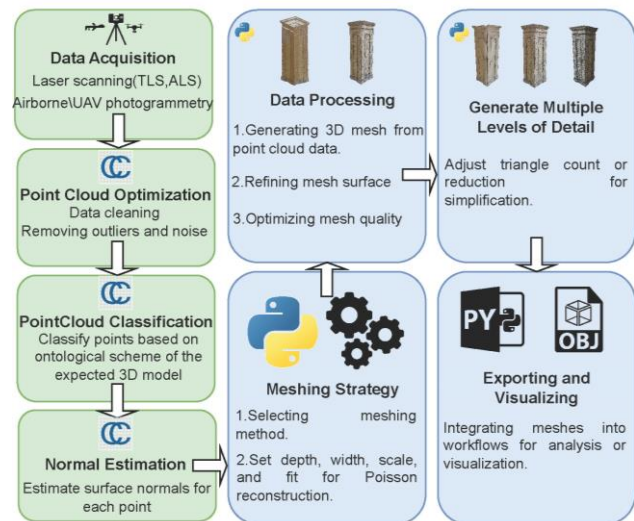
The proposed workflow (Figure 1) can be applied to any source point cloud, regardless of the acquisition technique (image or range-based) and within certain limits the density. The process comprises distinct steps that require a preliminary segmentation of the point cloud according to the aims of the work and to the ontological scheme of the subsequent model needed to achieve. This preliminary step of the process, today also and preferably obtainable using automatic strategies based on AI approach, will be described in the following paragraph 4.1, since its organization and data structuring is so related to the general aim of the project (urban or architectural scale environment modelling) and the finalization of the modelling (3D city model, digital twin of a monument/building of historic/cultural/social architectural value or also under a conservation plan). Surely, the data arrangement (optimization, segmentation) is an essential and pivotal step for the successful generation of 3D meshes from point clouds. Figure 1 represents the general organization of the method, which will be described and discussed in detail hereafter.

As known, the preliminary phase to start the process of mesh generation is to have at disposal the point cloud normal estimation and RGB data storing, and we need to consider that the point cloud has just been subjected to data cleaning, consisting in the pre-process point cloud data aimed at removing outliers, noise, and artifacts to ensure data quality.

After cleaning the data, the next step involves point cloud classification, where architectural elements such as doors, windows, and columns are identified and labeled. Similarly, in an urban scale analysis, the classification distinguishes between buildings and ground surfaces, enabling more precise analysis and visualization of the environment.

When point cloud data is loaded into a Python environment using NumPy arrays, which is a Python library for numerical computing, facilitating efficient operations on arrays and mathematical functions, and Open3D, which is an open-source Python library designed for 3D data processing and visualization, objects are created for further processing.

The process of data preparation includes handling file paths and data transformation to speed up the process, and File Parsing performs reading data files in common formats such as .xyz, .ply, or .las and extracting point coordinates, colors, and normals.



**Figure 1.** General organization of the process, foreseeing a preliminary preparation of cloud (first green column), and the multiple LoD meshing generation using Python code. *Preparing the data and process parameters*

The process offers the ability to carefully select an appropriate meshing technique based on the specific characteristics of the input data and the desired outcome for the resulting mesh. When employing Poisson reconstruction, various parameters can be adjusted to tailor the reconstruction to the user's needs. These parameters include:

- Depth: This parameter allows the user to control the level of detail present in the reconstructed mesh, enabling adjustments to match the requirements of their study.
- Width: By setting the width parameter, users can specify the desired width of the octree utilized during the reconstruction process, allowing for customization based on the complexity of the input data.
- Scale: Adjusting the scale parameter enables users to define the relationship between the diameter of the cube used for reconstruction and the bounding cube diameter of the samples, providing flexibility in adapting the reconstruction to different datasets.
- Linear Fit: Users have the option to enable or disable linear fit, providing control over the quality of the surface reconstruction achieved, which is crucial for ensuring the accuracy of their study results.

### Meshing Strategy: Poisson reconstruction

Also known, the Poisson reconstruction chosen as the meshing strategy involves fitting a watertight surface to the original point set by creating an entirely new point set representing an iso-surface linked to the normals.

During the mesh generation phase, the selected meshing strategy transforms the processed point cloud data into a 3D mesh. Additionally, it is possible to apply optional mesh optimization techniques, such as smoothing, decimation, or simplification, to enhance mesh quality and reduce complexity based on specific requirements.

After generating the mesh, it is feasible to further refine the surface to ensure smoothness and eliminate any gaps or inconsistencies present in the mesh. This refinement step is crucial to produce high-quality meshes (without irregularities) that accurately represent the underlying geometry of the original data.

#### Data Processing / Generation of multiple levels of detail

It's crucial to strike a balance between detail and computational efficiency when selecting depth and scale parameters, ensuring that the resulting mesh meets the requirements of the intended application. The resulting mesh is then subjected to further processing, such as cropping to remove unwanted artifacts or refining to improve mesh quality. Optimization techniques such as smoothing, decimation, or simplification may be applied to ensure the generated mesh is watertight, to ensure smoothness and eliminate gaps or inconsistencies, and to be suitable for subsequent analysis and visualization tasks.

During the mesh simplification phase, it is interesting for the further organization of multiscale urban/architectural objects to generate multiple levels of detail (LoD) meshes by simplifying the original mesh geometry while preserving important features. By adjusting parameters such as the target triangle count or percent reduction, it is possible to control the degree of simplification for each LoD, customizing the meshes according to their specific shape complexity requirements.

Following mesh simplification, analysts organize the LoD meshes into a hierarchical structure based on the predefined level of detail. This hierarchical arrangement facilitates efficient rendering and visualization, enabling a seamless transition between different levels of detail as needed.

#### Export and Visualization:

Following the generation of LoD meshes, the resulting meshes are exported to open file formats (.ply, .obj, .stl). This step enables seamless integration with external software tools for further analysis and visualization. Additionally, within Python, the meshes can be visualized using Open3D's built-in visualization capabilities, providing real-time feedback on the reconstructed surfaces. Finally, the exported meshes can be integrate into existing workflows or applications for further analysis, simulation, or visualization purposes. By integrating the meshes into their workflow, analysts can leverage the 3D data for a wide range of applications, including virtual simulations, computational modelling, and scientific visualization, thereby enhancing the utility and value of the generated meshes.

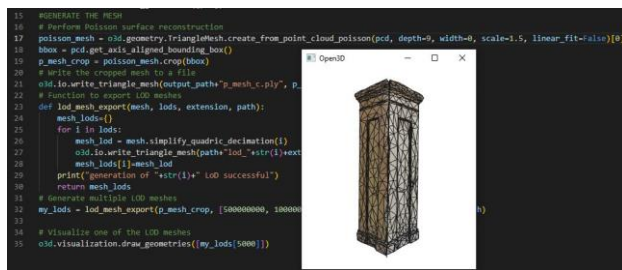


Figure 2. Python code snippet for 3D mesh generation from point cloud data using Open3D and NumPy.

### 3. APPLICATIONS (CASE STUDIES)

The case studies that the paper takes into consideration are derived from two projects led by the research group both in Turin,

whose aim was to experiment with digitisation strategies for the planned conservation of architectural heritage. Here, we aim to emphasize the advantage of evaluating an architectural-scale conservation plan at the urban scale as well. The general aim pinpointed at the base of enhancing the conservation process increasing the effectiveness of intervention technologies, and optimizing data sharing, in addition to saving resources and time, is very close and connected to the advantages of the ability to manage information about the conservation of such cultural heritage at the urban scale, possibly adopting a 3D city model used in a digital twins strategy. The first case study is an architectural complex of significant historical and cultural interest designed by L. Nervi and attributed to the modern movement (1948–1950); the second case study is the Royal Palace in Turin (seventeenth century), where a research project is underway in collaboration with restoration and conservation experts which aims to produce a prototype of a cognitive system based on BIM-GIS integration to support conservation program of the vast museum complex.

The vast and iconic building of Torino Esposizioni halls required a complex integration of range-based and image-based techniques since structural and seismic assessment analyses and investigations were planned. Even if the indoor environment was particularly focused, the external envelope of the building was surveyed using a UAV photogrammetry strategy, since a high scale and dense cloud was helpful to the documentation of the extremely thin vaulted system conceived by P.L. Nervi adopting ferrocement and a minimum use of materials.

Several GCPs measured by topographic technique related to a topographic control network enabled to obtain a general quality of the UAV point cloud close to 2 cm. (medium value of RMSE 0.017 m), (Spanò et al. 2023). (Figure 3)

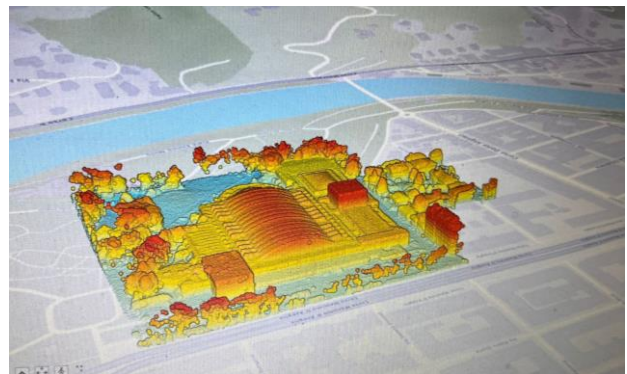
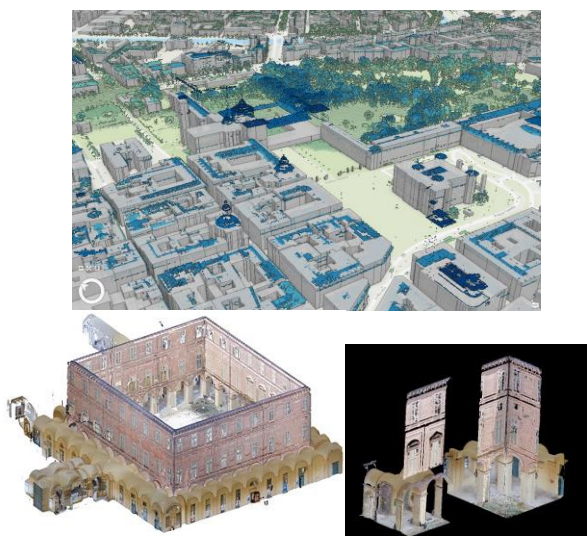


Figure 3. (above) UAV photogrammetric point cloud pertaining to the area of Torino Esposizioni in Turin (below) Lod 400 3D model achieved in manual and time-consuming way for the Torino Esposizioni architectural complex in a CIM view accomplished using Infracore sw. (Aboohamzeh 2023)

The second case study is the Royal Palace of Turin hosting a highly frequented museum is a very challenging and also proper example of a building pertaining to the cultural heritage domain that should be integrated into the urban 3D model of Turin in order to be preserved and managed by a conservation plan compliant and exchanging information with a twinned 3D representation of the urban environment, The urban scale cloud was generated from an airborne photogrammetric flight made available by the Turin municipality. (flight height 3610m with GSD equal to 15cm. Camera: UltraCam Eagle Mark 2 @ Vexcel Imaging GmbH. The reference system used for the projection of the LiDAR point cloud is WGS 84 / UTM zone 32N).

The architectural scale 3D survey and consequent 3D dense point clouds of the Turin Royal Palace have been acquired using the new Flash 3D scanning technology by Faro, exploiting a panorama camera and an upsampling technique. The medium value of the overall accuracy assessed by marker was 11 mm, while the point spacing on surface at the distance of 10m was 8 mm. (Sammartano et al. 2024) (Figure 4, below).



**Figure 4.** (above) Bird eye view of the Turin 3D geodatabase superimposed to the LiDAR airborne cloud. (below) Architectural scale 3D point cloud of the Turin Royal palace courtyard, derived from the new flash scanning technology.

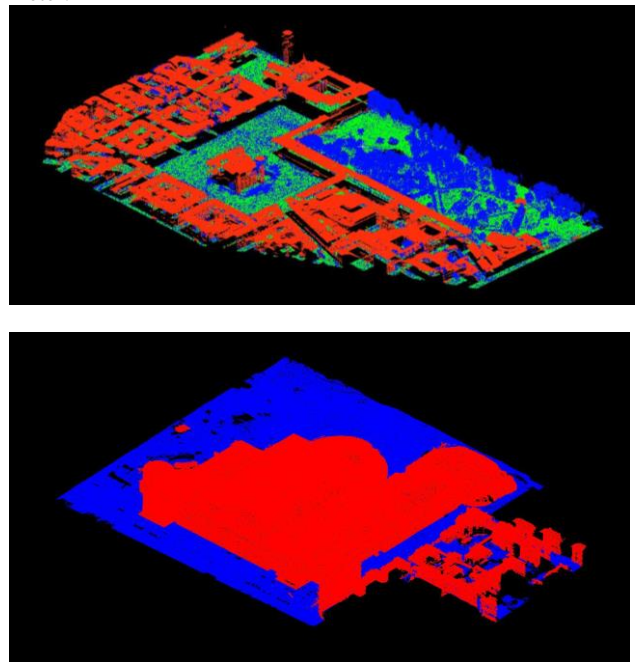
## 4. RESULTS

### 4.1. Preliminary clouds segmentation: urban and architectural scale

Although presently, the main discussion pertains to the automatic point cloud classification using a deep learning approach (Ballouch et al. 2022), we used well-known tools available on commercial software (ArcGIS pro) that allow the identification of objects in the urban environment. Starting from the LAStools which is an open source tool box for handling .las files, that Esri calls Forest-based Classification and Regression and inspired by Breiman (2001) Random forest algorithm, we obtained the classification of the ground, roofs, and vegetation (Jayaraj & Ramiya 2017).

As far as the subsampled point clouds are concerned, we can say that the royal palace sample cloud of figure 5, is about 1,8 million points, with a cloud density of 4.21 points per square meter; as said the Torino Esposizioni UAV sample cloud had to be decimated much more, and the final cloud used was about

1,1 million point, with a cloud density of 7.61 points per square meter.



**Figure 5.** Classified urban scale clouds of the two cultural heritage sites in Turin, labeling the ground, roofs, and vegetation. (above: Royal Palace site; below: Torino Esposizioni site)

As far as the architectural scale point cloud of the Royal Palace is concerned, we selected some typical architectural elements of the porch since such part of buildings are so diffuse in the historical city and crucial in consideration to the 3D city model. The cloud segmentation has been performed using a manual strategy, identifying typical elements such as walls, moldings, doors, and pilasters.

### 4.2. Multiscale automatic mesh generation

As planned, the implementation of the approach extended to both architectural element reconstruction and urban-scale building modelling, showcasing its versatility across different spatial scales. Pertaining to the architectural scale, through empirical studies focusing on elements such as pilasters, windows, doors, and architectural framing, the framework demonstrated its capacity to faithfully reproduce intricate architectural features in 3D space.

The selection of the Poisson reconstruction method was preferred over alternatives due to its proven effectiveness and versatility in handling complex point cloud data while delivering high-fidelity results. Additionally, the adjustable parameters of Poisson reconstruction provide users with flexibility and control over the level of detail and quality of the reconstructed mesh, aligning with the specific requirements and objectives of their study. (Poux 2020)

Figure 6 shows successful 3d meshing results starting from segmented clouds and the discussed method of mesh generation, enhancing also the ability to obtain a multiscale modelling adapting the automatic mesh generation parameters to different levels of detail needs.

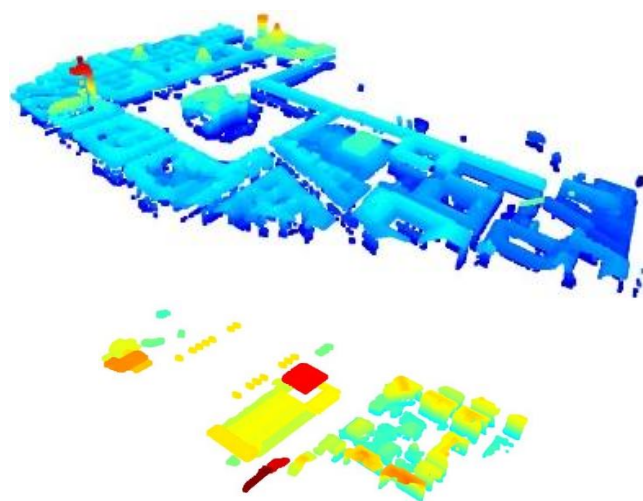
Expanding its scope to urban-scale environments (Poux 2020), the approach facilitated the creation of simplified yet accurate representations of entire buildings and urban landscapes.

Leveraging the dedicated Python script tailored for urban-scale mesh generation, the approach efficiently generated 3D meshes that captured essential architectural features and spatial configurations within their surrounding context.

The visualization presented in the figure 7 showcases the initial processing step of the point cloud data, where points classified as ground (class 6) are isolated and visualized in three-dimensional space. Utilizing the *laspy* library for data extraction and Open3D for visualization, the script effectively translates the raw point cloud data into a structured format, enabling further analysis and interpretation. This visualization serves as a foundational step in the broader workflow, setting the stage for subsequent analyses such as clustering and mesh generation.

Architectural element	PointCloud	Low Level of Detail	High Level of Detail
Wall			
Molding			
Door			
Pilaster			

**Figure 6.** the table shows the original segmented point clouds (first column) and multiple level of detail 3D mesh generation (2<sup>nd</sup> and 3<sup>rd</sup> column).



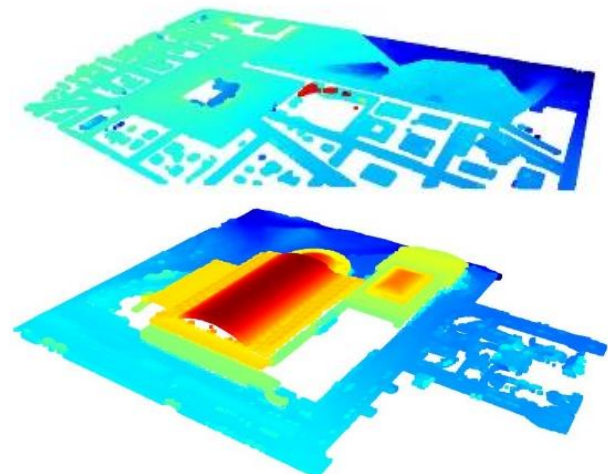
**Figure 7.** Structured point cloud format, enabling the visualization of different points classes, buildings and roofs. (above: Royal Palace site; below: Torino Esposizioni site)

Before delving into the results of the Torino Esposizioni and its neighborhood case study, it's essential to highlight some pre-processing steps and challenges encountered during the analysis.

The dense point cloud data obtained from UAV photogrammetry presented a significant computational challenge due to its size and complexity. To address this issue and facilitate smoother execution of the analysis code, the data underwent a sub-sampling process. By reducing the density of the point cloud, the computational load on the Python code was alleviated, enabling more efficient processing.

Figure 8 illustrates the extraction and visualization of points classified as ground (class 2) from the point cloud data (above: the Royal Palace site; Below: Torino Esposizioni site). By filtering the data based on classification criteria and utilizing Open3D for visualization, the script isolates ground points, providing insights into the terrain characteristics captured by the LiDAR scans or photogrammetric clouds. This visualization aids in understanding the spatial distribution of ground features and forms the basis for subsequent analysis, such as terrain modelling or object detection.

Despite these preparations, the intricate shape and architectural complexity of the buildings in Torino Esposizioni posed additional challenges. In particular, the detailed structures, such as complex roof designs, were not always accurately recognized by the analysis code. As a result, some errors were encountered in the final results, particularly in the identification and reconstruction of certain building features.

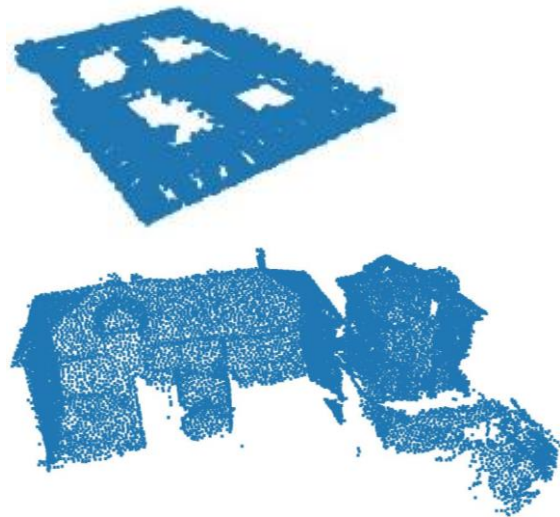


**Figure 8.** Ground points isolated from the other, allowing subsequent analysis such as terrain modelling. (above: Royal Palace site; below: Torino Esposizioni site)

The images of figure 8 depicts ground point cloud data, classified as category 2, where ground surfaces are highlighted. Notably, Torino Esposizioni's complex architectural structure, characterized by intricate features, is erroneously identified as ground due to its intricate architectural features (figure 8 below). The complexity of Torino Esposizioni's roof, differing from neighboring buildings, leads to misclassification as a ground surface. As shown in the image, the code does not recognize it as a building but rather as part of the ground surface. The explanation of this issue can be related, in our opinion, on the special roofs morphology of Torino Esposizioni: the building envelop present vaulted systems based

on pavilion or the largest hall is covered by a waved cylindrical vault.

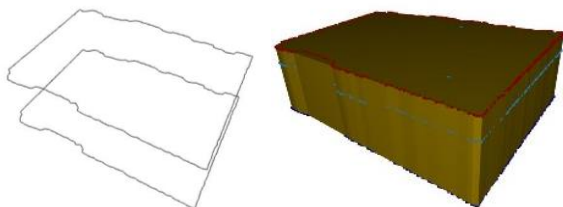
The next section of the code demonstrates the selection and visualization of a specific segment from the point cloud data, designated by the variable "sel." By isolating this segment, which may represent a distinct architectural or structural element, the script facilitates detailed analysis and processing. The subsequent steps involve extracting the outline of the selected segment using the *AlphaShape* algorithm, resulting in a 2D polygon representing the boundary of the segment. This polygon is then stored in a *GeoDataFrame*, enabling further geospatial analysis and integration with other GIS datasets.



**Figure 9.** Extraction of 2D polygons representing the boundary of cloud segments. (standard roof buildings in both sites are shown)

This part of the code involves creating an extrusion to represent the top layer of the building. The extrusion is constructed based on the vertices of the 2D polygon representing the building's footprint. Each vertex is extended vertically to the specified height, resulting in a three-dimensional representation of the building's upper boundary. The extrusion is visualized along with the 2D polygon using Open3D's visualization capabilities, providing an intuitive depiction of the building's geometry and structure. (Figure 10)

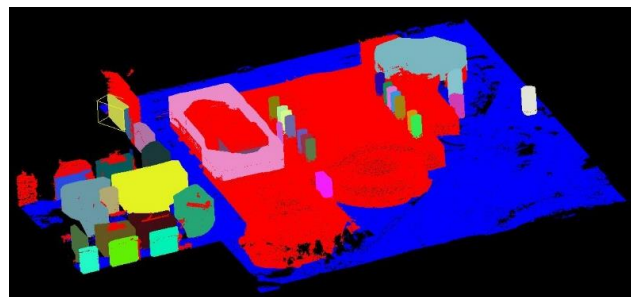
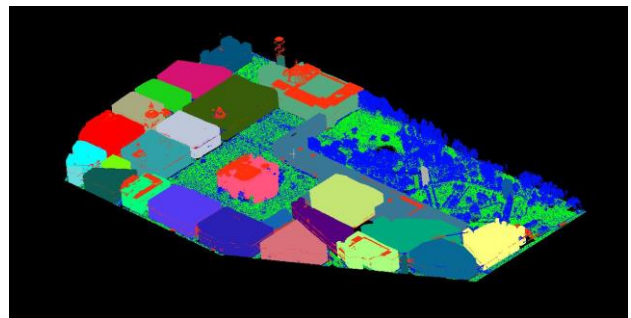
In this section of the code, the alpha shape of the combined 3D point cloud *temp\_o3d* is computed with a specified alpha value. The alpha parameter determines the tightness of the shape around the points. The resulting alpha shape is converted into a triangle mesh using Open3D's *create\_from\_point\_cloud\_alpha\_shape* function.



**Figure 10.** 2D polygon representing the building's footprint, and extrusion based on the specified height, resulting in a three-dimensional representation.

Additionally, vertex normals are computed for the mesh to assist with shading and rendering. Finally, the mesh is colored uniformly and visualized along with the original point cloud *temp\_o3d* and the selected segment. The *mesh\_show\_back\_face* parameter is set to *True* to display both the front and the back faces of the mesh during visualization.

This section of the code iterates through each segment identified in the point cloud data, extracting essential attributes and generating corresponding 3D meshes for buildings. Using techniques like the *alpha shape* algorithm, it computes the footprint and calculates the height of each building. These attributes, along with other metadata such as area, perimeter, centroid coordinates, and translation coordinates, are stored in a *GeoDataFrame*. The code then generates 3D geometry for each building by extruding its footprint to the calculated height. Random colors are assigned to each building mesh to enhance visualization (Figure 11). Subsequently, individual building meshes are saved as *.ply* files, each uniquely named based on its segment ID. Finally, all building attributes are consolidated and saved into a shapefile. The generated mesh files are then uploaded to the cloud for comparative visualization alongside the original point cloud data.



**Figure 11.** 3D geometries are generated for each building and random colors are assigned to each building mesh to enhance visualization, (above) in Royal Palace area and (below) in Torino Esposizioni surroundings.

However, the intricate architecture of Torino Esposizioni introduces complexities that challenge the code's recognition capabilities. As a result, certain parts of the structure may be misclassified or only partially recognized, leading to errors in the mesh generation process. This underscores the difficulties inherent in accurately capturing the nuanced architectural features of complex structures like Torino Esposizioni.

## 5. CONCLUSION

These case studies underscored its potential utility in modelling buildings from the perspective of urban environment city models. Since we focus on a historical town involving architectural heritage that requires a higher level of detail, we tested the Python code in crucial areas of Torino, respectively where an XVII-century building (Royal Palace) and a modern movement-reinforced concrete building (Torino Esposizioni) are located.

Obviously, we can also affirm that the architectural scale tests (on walls, doors, pilasters, and moldings) perfectly worked, so it means that such classification and modelling can also be considered valuable for architectural visualization and digital heritage preservation, providing researchers and analysts with a robust framework for capturing and analyzing architectural details.

These results highlighted the approach's versatility and scalability, positioning it as a valid alternative within tools for urban planning, virtual city modelling, and environmental simulation applications. In the research endeavors focusing on urban and architectural modelling and analysis, advanced computational tools to extract meaningful insights from complex spatial datasets have been utilized. To this end, a Python script obtained from prior studies (Poux 2020, 2023) has been adapted and integrated into the presented analytical workflow. This script, developed with the aid of prominent libraries like NumPy, Open3D, and *laspy*, provided a robust framework for processing and interpreting classified point cloud data.

Through the application of this script, the authors aimed to address key challenges inherent in urban modelling, such as accurately delineating building footprints and generating high-fidelity 3D meshes. By leveraging established algorithms for point cloud classification, clustering, and mesh reconstruction, AI strived to automate and expedite the process of transforming raw data into actionable insights.

The utilization of these Python scripts underscored a commitment to leveraging cutting-edge computational techniques for architectural analysis. Over the past few years, there has been a notable challenge for researchers in heritage documentation due to the increasing need for flexible and sustainable approaches in digitization processes. Specifically, our research group have introduced inventive methodologies that combine Visual Programming Language and 3D Python libraries into established modelling processes within the AEC sector, making these processes increasingly automated (Avena et al 2024). In this scenario, by harnessing the power of open-source software and community-developed tools, the work endeavored to enhance the efficiency and reproducibility of the research outcomes while contributing to the broader advancement of urban modelling methodologies.

In perspective, this research aims to underline two fundamental concepts: the first is that pursuant to the automation of classification and segmentation new approaches based on AI, even the automatic generation of multiscale 3D models ready to be classified into different Levels of Details, makes modelling more sustainable in terms of recourse and time, even when specific complex geometries related to heritage are required.

The use of open tools is certainly the second aspect of great importance.

Future developments can certainly concern urban objects under close observation, such as the vegetation component (Ortega 2018) and the underground component which currently still require in-depth efforts.



**Figure 12.** Examples of the urban environments and domains, such as vegetation and underground spaces (in the figures manually generated), that could benefit from the automatic mesh generation presented in this paper. (models by E. Fillia)

## References

- Aboohamzeh A., *Exploring HBIM Applications in Preserving the Architectural Legacy, The case study of Torino Esposizioni, supervisors Spanò, Avena* Politecnico di Torino, M.thesis, 2023
- Atazadeh, B.; Rajabifard, A.; Zhang, Y.; Barzegar, M. Querying 3D Cadastral Information from BIM Models. *ISPRS Int. J. Geo-Inf.* 2019, 8, 329.
- Avena, M., Patrucco, G., Remondino, F., and Spanò, A.: A SCALABLE APPROACH FOR AUTOMATING SCAN-TO-BIM PROCESSES IN THE HERITAGE FIELD, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLVIII-2/W4-2024, 25–31, <https://doi.org/10.5194/isprs-archives-XLVIII-2-W4-2024-25-2024>, 2024.
- Ballouch, Z., Hajji R., Poux F., Kharroubi A., et Billen R. 2022. A Prior Level Fusion Approach for the Semantic Segmentation of 3D Point Clouds Using Deep Learning. *Remote Sensing* 14 (14):
- Bernardini, F.; Mittleman, J.; Rushmeier, H.; Silva, C.; Taubin, G. "The ball-pivoting algorithm for surface reconstruction." *Transactions on Visualization and Computer Graphics* 1999, 5, 349–359.
- Biljecki, F.; Stoter, J.; Ledoux, H.; Zlatanova, S.; Çöltekin, A. Applications of 3D city models: State of the art review. *ISPRS Int. J. Geo-Inf.* 2015, 4, 2842–2889.
- Breiman, Leo. "Random Forests". *Machine Learning*. 45 (1): 5-32. doi:10.1023/A:1010933404324. 2001
- Elhashash, M.; Albanwan, H.; Qin, R. A Review of Mobile Mapping Systems: From Sensors to Applications. *Sensors* 2022, 22, 4262, doi:10.3390/s22114262.
- Farella, E.M.; Özdemir, E.; Remondino, F. 4D Building Reconstruction with Machine Learning and Historical Maps. *Appl. Sci.* 2021, 11, 1445. <https://doi.org/10.3390/app11041445>



Haala, N.; Rothermel, M.; Cavegn, S. Extracting 3D urban models from oblique aerial images. In Proceedings of the 2015 Joint Urban Remote Sensing Event (JURSE), Lausanne, Switzerland, 30 March–1 April 2015.

Kazhdan, M.; Bolitho, M.; Hoppe, H. "Poisson surface reconstruction." Eurographics symposium on Geometry processing 2006, 1–10.

Kutzner, T., Chaturvedi, K. & Kolbe, T.H. CityGML 3.0: New Functions Open Up New Applications. PFG 88, 43–61 (2020). <https://doi.org/10.1007/s41064-020-00095-z>

La Russa, F. M., Grilli, E., Remondino, F., Santagati, C., Intelisano, M., 2023. Advanced 3D parametric historic city block modelling combining 3D surveying, AI and VPL. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-M-2-2023, 903–91

Noardo, F., Harrie, L., Arroyo Ohoi, K., Biljecki, F., Ellul, C., Krijnen, T., Eriksson, H., Guler, D., Hintz, D., Jadidi, M. A., Pla, M., Sanchez, S., Soini, V.-P., Stouffs, R., Tekavec, J., & Stoter, J. (2020). Tools for BIM-GIS Integration (IFC Georeferencing and Conversions): Results from the GeoBIM Benchmark 2019. ISPRS International Journal of Geo-Information, 9(9), 502. <https://doi.org/10.3390/ijgi9090502>.

Poux, F., 2020. Generating 3D Meshes from Point Clouds: A Python Implementation., *Towards Data Science*.

Poux, F., 2023. 3D Python Workflows for LiDAR City Models: A Step-by-Step Guide, *Towards Data Science*.

Ortega-Córdova, Lessie. 2018. « Urban Vegetation Modelling 3D Levels of Detail ». PhD. Thesis. <https://repository.tudelft.nl/islandora/object/uuid%3A8b8967a8-0a0f-498f-9d37-71c6c3e532af>.

Singh M, Fuenmayor E, Hinchy EP, Qiao Y, Murray N, Devine D. Digital Twin: Origin to Future. *Applied System Innovation*. 2021; 4(2):36. <https://doi.org/10.3390/asi4020036>

Sammartano, G., Patrucco, G., Avena, M., Bonfanti, C., and Spanò, A.: Enhancing terrestrial point clouds using upsampling strategy: first observation and test on faro flash technology, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLVIII-2/W4-2024, 381–388, <https://doi.org/10.5194/isprs-archives-XLVIII-2-W4-2024-381-2024>, 2024.

Spanò, A., Patrucco, G., Sammartano, G., Perri S., Avena M., Fillia, E., Milan, S., Digital Twinning for 20th Century Concrete Heritage: HBIM Cognitive Model for Torino Esposizioni Halls. (2023) *Sensors*, 23 (10), art. no. 4791. DOI 10.3390/s23104791

Tang, L., Li, L., Ying, S., & Lei, Y. (2018). A Full Level-of-Detail Specification for 3D Building Models Combining Indoor and Outdoor Scenes. *ISPRS International Journal of Geo-Information*, 7(11), 419. <https://doi.org/10.3390/ijgi7110419>

Treccani, D.; Balado, J.; Fernández, A.; Adami, A.; Díaz-Vilariño, L. A Deep Learning Approach for the Recognition of Urban Ground Pavements in Historical Sites. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* 2022, 43, 321–326.

Vosselman, G. & Maas, H-G. 2010. Airborne and terrestrial laser scanning. Whittles Publishing, Scotland. p. 318. ISBN 978-1904445-87-6.