



Heritage Information Modeling: The Case of Chellah's Gate

Sharif Anouar¹(✉), Adam Anouar², and Ayoub Lharchi³(✉)

¹ International University of Rabat (UIR), Salé, Morocco
sharif.anouar@uir.ac.ma

² Vacken, Rabat, Morocco

³ Centre of Information Technology and Architecture (CITA), Copenhagen, Denmark
alha@kglakademi.dk

Abstract. This paper aims to propose an integrated workflow for the digitization of the built cultural heritage. To this end, we leverage the power of computational tools and the relevancy of Building Information Modeling (BIM) process to overcome the limitations and challenges faced by Scan-to-BIM. We describe the automatic generation of an as-built BIM model of a heritage building in a three-step procedure. Firstly, we outline the data acquisition method of the point cloud. Secondly, we describe the automatic processing and segmentation of the point cloud according to architectural elements using Machine Learning. Then, we tested and compared various meshing algorithms and utilized a combination depending on the desired level of details. Lastly, the resulting geometry is converted into a BIM object that will be subsequently semantically labeled. We used a UNESCO world heritage in Morocco—Chellah, as a case study to test the robustness of our protocol.

Keywords: Built cultural heritage · Digitization · Point cloud · Machine learning · Building information modeling

1 Introduction

Built heritage carries the legacy of humanity. It encapsulates the memory of the people, their accomplishment, and their aspirations. However, architectural heritage worldwide faces severe threats from anthropogenic hazards and natural disasters, making its preservation, documentation, and dissemination one of the challenges of our time [10]. The semantic digitization of the architectural heritage has provided an effective solution to this issue [14], while using remote sensing technologies in conjunction with BIM authoring tools (Scan-to-BIM) has become standard procedure [13].

In recent years, Building Information Modeling (BIM) has become widely accepted within the Architecture, Engineering, and Construction (AEC) industry [5]. When implemented correctly, it facilitates integrated workflows covering design, construction, and management processes [7]. Despite this integration in design practices, the use of BIM for remodeling and analysis of historic buildings remains an unexplored area [4]. The

complex nature of such buildings and the logistical constraints that usually arise when addressing structural and preservation considerations. Therefore, existing BIM workflows for digitization hit their limitations. This is particularly the case in the generation of reusable models that feature different Levels of Details (LOD) and a high number of irregular geometries and patterns [3], which is typically the case in old construction. Although Scan-to-BIM workflow is well established, it still requires additional complex manual modeling, which is often time-consuming and prone to errors [17]. Besides, the BIM platform offers neither the possibility to process massive point clouds data nor advanced modeling tools for the high-fidelity reconstruction of the architectural artifact, which in return leads to the usage of several expensive specialized software.

To tackle these shortcomings, this paper will aim to describe an agile, zero data loss methodology to digitize the built heritage. Its novelty lies in the use of Machine Learning and a reduced set of software deployed under a BIM platform. The end goal is to achieve an as-built BIM model, robust and flexible enough to be used for further analysis and visualization scenarios.

2 Background

Heritage Building information modeling (HBIM) is a multidisciplinary and evolving system. It consists of a reverse engineering process based on creating parametric objects derived from remote survey data of existing constructions [11]. These objects represent architectural elements and hold comprehensive information, including geometrical, material, texture, and historical and structural data. As an application of BIM to heritage buildings, HBIM opens promising prospects for managing the ever-increasing complexity of the conservation practice [8]. Moreover, the understanding of undocumented heritage, its analysis, maintenance, restoration planning, and simulation, communication, among others, become then achievable.

Remote sensing technologies such as 3D scanning and photogrammetry are the foundation of HBIM. The procedure of transferring the acquired spatial data and its transformation into BIM models is commonly called Scan-to-BIM. This procedure is threefold: It starts with the survey and mass data capture. Then comes the processing phase, which encompasses point clouds cleaning, segmentation, and modeling. Finally, the last step involves the conversion of the generated geometry into BIM objects that will be subsequently semantically labeled [16].

While studies related to the survey side of Scan-to-BIM show established techniques and frameworks, research about the two following phases is growing, with a multitude of approaches being investigated. This interest is explained by problems inherent to the acquired point cloud, software used, and the BIM procedure.

The difficulties in the cleaning and segmentation of point cloud datasets have been addressed by employing a deep learning framework using either DGCNN (Dynamic Graph Convolutional Neural Network) for point-based approaches [12], voxel-based (Zhou and Tuzel 2018), or multiview-based methods [2]. Others worked on the automatic detection and segmentation of building components from the point cloud using the Hough transform algorithm [6]. In contrast, others investigated semantic-based methodology [9].

Recent studies have investigated the possibilities of (semi-) automating the geometric reconstruction of as-built models of built heritage. Some authors explored the parametric generation of NURBS-based 3d models from mesh [18], others focused on the automatic conversion of Terrestrial Laser Scanning (TLS) and Structure from Motion (SfM) point cloud data into textured BIM objects [1]. Despite the intense focus on automatic cleaning of noisy point clouds and the recognition and segmentation of architectural elements, there is, to the best of our knowledge, a lack of studies leveraging the power of Machine Learning (ML) in HBIM-based workflows.

3 Methodology

Building upon the standard procedure of the Scan-to-BIM process, we propose an integrated workflow for Heritage BIM (Fig. 1) enabled by ML.

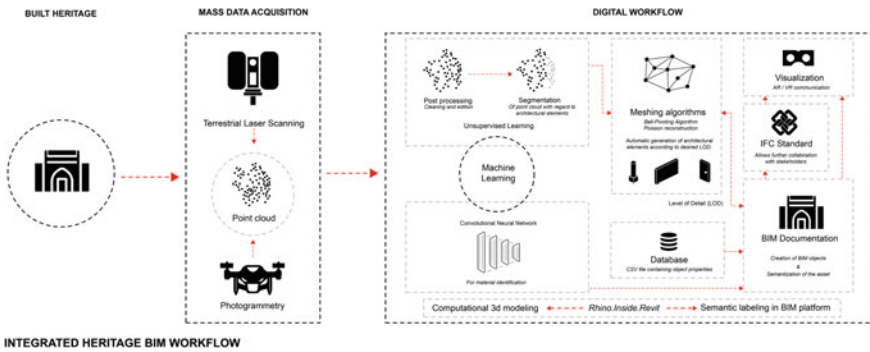


Fig. 1. Overview of the proposed workflow.

3.1 Case Study: The Main Gate of Chellah

The necropolis of Chellah (Fig. 2) is an iconic cultural site of the capital city of the Kingdom of Morocco. The oldest building in Rabat, was successively a Roman city, a Berber city, sometimes a capital, and finally a Marinid necropolis. Its rich heritage can be discovered through the multiple vestiges of the civilizations that succeeded each other over almost three millennia. Classified as a historical monument since 1913, it integrates in 2012 with a set of other remarkable buildings of the capital, the UNESCO world heritage list. The gate of Rabat’s most venerable building seemed to us to be the most legitimate to inaugurate the phase of digitization of the architectural heritage, which is undoubtedly in Morocco, only at its beginnings.

3.2 Data Acquisition

The data acquisition was achieved using two different methods depending on the intervention scale. In the field of mass data acquisition for larger-scale artifacts, Terrestrial



Fig. 2. Chellah's gate.

Laser Scanners (TLS) appear to be the preferred choice. Despite being an expensive piece of equipment necessitating a high degree of expertise, TLS can output highly accurate spatial data. The information requirements for the project determined the required scan data quality. A LOD of 500 has been chosen to facilitate the automatic reconstruction of the architectural artifact in the following stages of the workflow.

The survey was conducted with a FARO Focus M 70. After setting the resolution parameters, the scanning speed, and precision, multiple scans of Bab Chellah have been performed from different perspectives. This strategy produces overlapping point clouds data yet avoids the problems of occluded areas. Unfortunately, the cumulative point clouds that were obtained were not usable. Too much noise was picked and due to the current restoration works in Chellah, we could not perform additional scans to improve the quality of the point cloud. The resulting point clouds were not pre-processed using commercial software like FARO Scene,¹ Leica Cyclone² or RecapPro³ but directly transferred to Rhinoceros 3D. On a smaller scale, we used a Microsoft Kinect Azure (MKA) to create 1-2 m scan section of elements of interest that were further processed and segmented (Fig. 3).

¹ <https://www.faro.com/fr-FR/Products/Software/SCENE-Software>.

² <https://leica-geosystems.com/products/laser-scanners/software/leica-cyclone>.

³ <https://www.autodesk.com/products/recap/overview>.

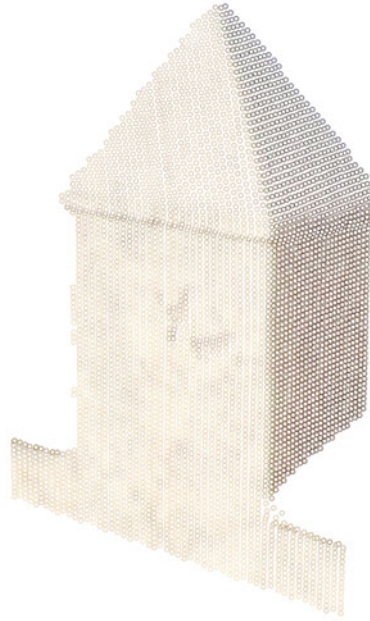


Fig. 3. Small scale point cloud obtained using a MKA and a laptop.

3.3 Processing and Segmentation Using RANSAC

There are many algorithmic methods for grouping point cloud data. In the case of Chellah, two parameters must be considered. The nature of the data to be grouped and the size of the project.

The cloud point here is an unsorted set of unlabeled data; therefore, the clustering method is preferred to the classification method, which is not feasible. The size of the Chellah model to reconstruct is relatively small, and it can be subdivided into a not very high number of primitive geometries, in our case, 2D planes. For this purpose, several methods are available to us: First, we discard kernel methods like k means, etc., they have the disadvantage of assuming somewhat rounded cluster shapes, and above all, they require giving the number of clusters as input. Geometric diversity, even in the smallest and simplest projects, prohibits us from doing this, which would significantly reduce the automation of the cloud point segmentation process. The unsupervised algorithm to group unlabeled data will also have to do without the number of clusters as input. In addition, the algorithm must not be density-based because the clusters to be produced will be adjacent (the faces being adjacent to each other). Therefore, there will be no gaps separating the different groups. The clustering algorithm must therefore be able to detect the primitive shapes. Thus, the final choice was between the clustering Region Growing approach and the Model Fitting method. We preferred the latter and especially the Random Sample Consensus (RANSAC) method because it works well with noisy data, unlike the former, which requires a preliminary cleaning step. Concretely, we

imported the Point cloud in Grasshopper⁴ platform and used the RANSAC component of its free plug-in Cockroach [15] for the detection and clustering of the plane distribution of the point cloud.

3.4 Mesh Generation Procedure

To generate a usable 3D mesh from the collected set of point clouds, we used three different algorithms:

- **Ball-Pivoting Algorithm (BPA):** It is based on the use of a “Virtual Ball” that is rolled across the point cloud with a variable diameter. Ideally, the diameter should be larger than the average distance between the points. With continuous rolling and pivoting, triangles are generated and finally combined to constitute the mesh.
- **Poisson Reconstruction (PR):** This method tries to envelop the point cloud with a “cloth” that keeps shrinking depending on the depth parameter.
- **MeshFromPoints (MFP)** feature included in the modeling software (Rhinoceros 3D).

For the first two algorithms, we used the open-source Python library “Open3D” [19]. Since our modeling software does not support natively using CPython libraries, we developed an intermediate external script, that was capable of loading the XYZ file format that contains the points coordinates and the faces normals. Once loaded, the point cloud was meshed, and then reimported within Rhinoceros 3D. It is critical to note that extensive testing was necessary to optimize the script and find the correct parameters for each algorithm to obtain a valid mesh with minimal artifacts (Fig. 4).

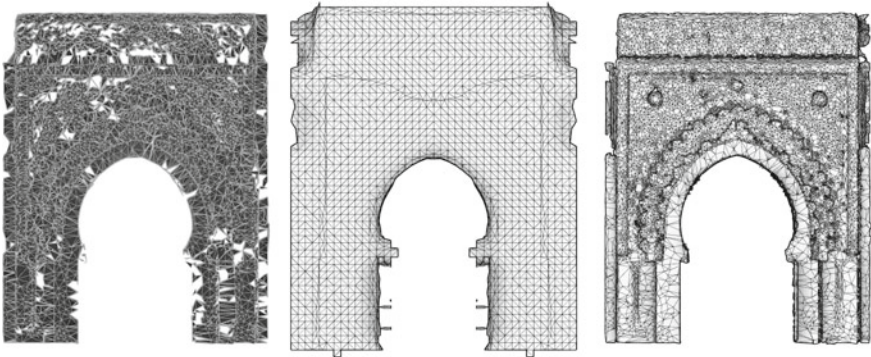


Fig. 4. Comparison of the obtained mesh with the 3 meshing methods. From Left to Right: BPA, MFP, PR.

While the MFP might seem to be the best option, tests showed that the PR method is the most effective as it produces satisfactory results, much faster, and is the least affected by noise in the input data. Although PR is generating an irregular mesh topology, it was

⁴ <https://www.grasshopper3d.com/>.

trivial to simplify it and regenerate different topologies (Fig. 5), which is a necessary step for further BIM element generation.

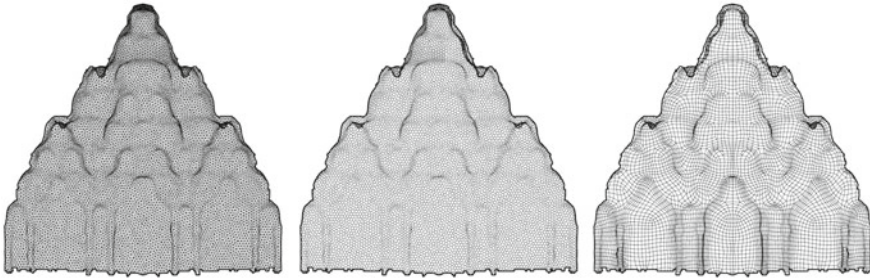


Fig. 5. Processed results. From left to right: Triangulated, Dual Ngon, Quadrated meshes.

3.5 BIM Documentation

All the meshes produced already form a faithful representation of the model. However, to transfer it to Revit and develop an informed model, it is necessary to create a family. To do this, the mesh must be converted into a NURBS surface since the families on Revit only accept closed poly-surfaces. To this end, the first code in Grasshopper deconstructs each mesh into its many points and boundaries and then reconstructs it into a single surface, or at least into a minimum of surfaces. The process is then applied in a loop to all the meshes representing all the faces of the model. Among the different mesh typologies created and tested, the unstructured triangulated mesh is preferred because triangles surfaces are planar, and Revit has a greater tolerance to planar surfaces, their insertions in families, and their conversions into walls and other native Revit elements. Moreover, Revit is sometimes quite reticent and unpredictable to accept surfaces created with Rhinoceros 3D and Grasshopper, even when they are qualitative and well defined. This matter is explained by the difference the two software have in handling tolerances; where Rhinoceros 3D can go to tolerances of 0.001 mm, Revit can only accept 1-mm tolerance.

Each project has its own physical and material characteristics that distinguish it, both in size and complexity of form, requiring a particular and adapted approach. Therefore, the following approach is not intended to be the universal method to be adopted but the best option available to us in the context of this particular project. Starting from the characteristics of the heritage artifact that is the Chellah Gate, it appeared to us that the best approach to follow would be to algorithmically create a mass via the “conceptual mass” family template using the Rhino.Inside plugin, through which the surfaces, thickened on occasion so that they become solids with a thickness, will be transferred to Revit. Inside Revit finally, a code on Dynamo; the visual programming language of Revit, will take care of generating via the “wall-by-face” tool, the walls from the faces of the conceptual mass generating the model of Bab Chellah with Revit native elements. The door itself was modeled using. Each element is finally generated according to its nature (the wall modeled with the wall tool, the door created with the door family etc.).

Regarding the LOD (levels of detail), dealing mainly with the issue of the graphic level of detail (representations of muqarnas, low relief frescoes on walls), it was decided that they will be generated via in-place family, categorized as a wall, they will be produced according to the same process as for the walls, namely, by creating a conceptual mass, and its automatic export from Grasshopper to Revit via a code using the plugin Rhino.Inside.Revit, and thereby they will be placed within the in-place family as a component.

Separating the walls from their ornamentation allows a practical and straightforward management of the LOD at the time of the export and exchange of information by the IFC, according to what is defined by the “information requirements” (Figs. 6 and 7).



Fig. 6. Three stages of Chellah's gate reconstruction.

From left to right: Raw point cloud, clustered point cloud, BIM model LOD 100.



Fig. 7. Chellah tower. From left to right: Segmented point cloud, mesh reconstruction, BIM model LOD 100, BIM model LOD 400.

Finally, given the quality of the point cloud and the complexity of the architectural elements, it was decided to reconstruct them from the existing survey as here in the case of the muqarnas (Fig. 8).

The level of detail of the created mesh is directly correlated to the resolution of the point cloud. The survey should represent the existing building; the maximum LOD corresponds to the LOD 500 “as Built”. To propose different levels of detail (and therefore necessarily lower, LOD 400, 300, 200, etc.), we can play on the resolution of the

generated surfaces by reducing it at the time of the conversion of the mesh into the surface. This produces simplified surfaces given with washed-out ornamentation, allowing a saving of computational resources when a high level of detail is not necessary.

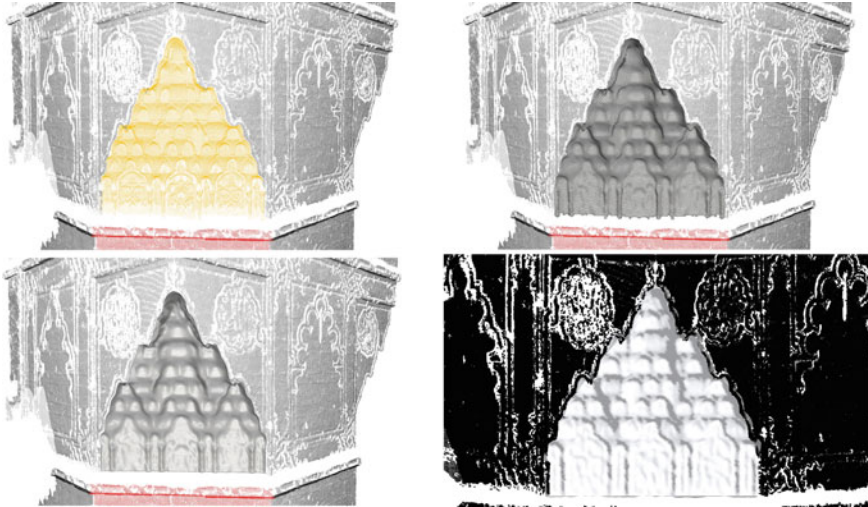


Fig. 8. Muqarnas detail reconstruction (from top left to bottom right): Point cloud segmentation, mesh generation, NURBS-based surface conversion, Revit in-place family (wall).

Afterwards, a text-based CSV file (Comma-Separated Values), containing the mechanical properties of the architectural elements is linked to the BIM model to automatically label its different parts.

4 Results

The outcome of this work is an attempt to automate the process of Scan-to-BIM while reducing both the number of software involved and the time spent on cleaning, segmenting, modeling and labeling based on an initial point cloud. Furthermore, the workflow proposed based on Rhinoceros 3D-Revit maintained robust interoperability, allowing the designer to have complete control of the overall process and giving the possibility of using Machine Learning to speed up the process while reducing human involvement. Finally, the compatibility and flexibility of the tandem Rhino-Revit linked by Rhino.Inside through Grasshopper platform enables flexible strategies for transferring more or less detailed geometry depending on the LOD required and needed for the IFC export.

5 Conclusion and Discussion

HBIM is gaining momentum within the AEC industry. Although studies focusing on the management of point cloud data or the automatic reconstruction of architectural heritage

exists, there is still a lack of an exhaustive and comprehensive framework for HBIM. This paper tries to narrow this gap by proposing an integrated workflow relying mainly on close interoperability between a BIM platform and a 3D modeler enhanced by ML methods. This combination showed to be a powerful tool for the culturally built heritage.

Future work could expand on the reconstruction of other building types. It might as well integrate different phases into the workflow, such as fabrication. Besides, 3D laser scanning being expensive, we expect to experiment with photogrammetry using DSLR cameras and smartphones mixed with free, open-source software to propose an Open-HBIM workflow that can be vendor agnostic.

References

1. Andriasyan M, Moyano J, Nieto-Julián JE, Antón D (2020) From point cloud data to building information modelling: an automatic parametric workflow for heritage. *Remote Sens* 12:1094. <https://doi.org/10.3390/rs12071094>
2. Boulch A, Saux BL, Audebert N (2017) Unstructured point cloud semantic labeling using deep segmentation networks. In: *Eurographics workshop on 3D object retrieval*, p 8. <https://doi.org/10.2312/3DOR.20171047>
3. Brusaporci S, Maiezza P, Tata A (2018) A framework for architectural heritage Hbim Semantization and development. In: *The international archives of the photogrammetry, remote sensing and spatial information sciences*. Presented at the ISPRS TC II Mid-term symposium towards photogrammetry 2020 (vol XLII-2) - 4;7 June 2018, Riva del Garda, Italy, Copernicus GmbH, pp 179–184. <https://doi.org/10.5194/isprs-archives-XLII-2-179-2018>
4. Capone M, Lanzara E (2019) Scan-to-Bim Vs 3d ideal model Hbim: parametric tools to study domes geometry. In: *The international archives of the photogrammetry, remote sensing and spatial information sciences*. Presented at the 8th international workshop 3D-ARCH 3D virtual reconstruction and visualization of complex architectures - 6;8 February 2019, Bergamo, Italy, Copernicus GmbH, pp 219–226. <https://doi.org/10.5194/isprs-archives-XLII-2-W9-219-2019>
5. Chen, Chao & Tang, Llewellyn. (2019). Development of BIM-Based Innovative Workflow for Architecture, Engineering and Construction Projects in China. *International Journal of Engineering and Technology*. 11. 119-126. 10.7763/IJET.2019.V11.1133.
6. Díaz-Vilariño L, Conde B, Lagüela S, Lorenzo H (2015) Automatic detection and segmentation of columns in as-built buildings from point clouds. *Remote Sens* 7:15651–15667. <https://doi.org/10.3390/rs71115651>
7. Eastman, C., Eastman, C. M., Teicholz, P. et al. (2011). *BIM Handbook: A Guide to Building Information Modeling for Owner, Manager, Designer, Engineers and Contractors* (p. 243). Hoboken, NJ: John Wiley & Sons.
8. Gigliarelli E, Calcerano F, Cessari L (2017) Heritage Bim, numerical simulation and decision support systems: an integrated approach for historical buildings retrofit. *Energy Proc* 133:135–144. <https://doi.org/10.1016/j.egypro.2017.09.379>
9. Hmida HB, Cruz C, Boochs F, Nicolle C (2013) From 3D point clouds to semantic objects an ontology-based detection approach. [arXiv:1301.4783](https://arxiv.org/abs/1301.4783) [cs]
10. Holman N, Ahlfeldt G (2015) No escape? The coordination problem in heritage preservation. *Environ Plan A* 47:172–187. <https://doi.org/10.1068/a130229p>
11. Murphy M, McGovern E, Pavia S (2009) Historic building information modelling (HBIM). *Struct Surv* 27:311–327. <https://doi.org/10.1108/02630800910985108>

12. Pierdicca R, Paolanti M, Matrone F, Martini M, Morbidoni C, Malinverni E, Frontoni E, Lingua A (2020) Point cloud semantic segmentation using a deep learning framework for cultural heritage. *Remote Sens* 12:1005. <https://doi.org/10.3390/rs12061005>
13. Rocha M, Fernández F (2020) A scan-to-BIM methodology applied to heritage buildings. *Heritage* 3:47–67. <https://doi.org/10.3390/heritage3010004>
14. Salvador García E, García-Valldecabres J, Viñals Blasco MJ (2018) The use of Hbim models as a tool for dissemination and public use management of historical architecture: a review. *Int J SDP* 13:96–107. <https://doi.org/10.2495/SDP-V13-N1-96-107>
15. Vestartas P, Settimi A (2020) Cockroach: a plug-in for point cloud post-processing and meshing in Rhino environment. EPFL ENAC ICC IBOIS.
16. Volk R, Stengel J, Schultmann F (2014) Building information modeling (BIM) for existing buildings — literature review and future needs. *Autom Constr* 38:109–127. <https://doi.org/10.1016/j.autcon.2013.10.023>
17. Wang Q, Guo J, Kim M-K (2019) An application oriented scan-to-BIM framework. *Remote Sens* 11:365. <https://doi.org/10.3390/rs11030365>
18. Zheliazkova M, Naboni R, Paoletti I (2015) A parametric-assisted method for 3d generation of as-built Bim models for the built heritage. Presented at the STREMAH 2015, A Coruña, Spain, pp 693–704. <https://doi.org/10.2495/STR150581>
19. Zhou Q-Y, Park J, Koltun V (2018) Open3D: a modern library for 3D data processing. [arXiv:1801.09847](https://arxiv.org/abs/1801.09847) [cs]

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

