

# NEW ALGORITHM FOR THE MERGING OF GEOMETRIC ENTITIES TOWARDS THE CORRECT GENERATION OF SEMANTIC gbXML MODELS

R. Otero<sup>1\*</sup>, S. Lagüela<sup>2</sup>, P. Arias<sup>1</sup>

<sup>1</sup> Applied Geotechnologies Research Group, University of Vigo, 36310 Vigo, Spain - (rotero, parias)@uvigo.es

<sup>2</sup> Polytechnic School of Ávila, University of Salamanca, 05003 Ávila, Spain – sulaguela@usal.es

Commission I, WG I/7

**KEY WORDS:** BIM, gbXML, 3D Model, 3D Mapping, Indoor Mapping, Geometrical Analysis.

## ABSTRACT:

This paper presents an algorithm developed to solve a duplicity problem found during the development of a software system for BIM generation. The chosen schema for the BIM is gbXML. The Manhattan World Assumption is adopted because the objective of the system is to obtain simplified and regularized 3D models of the indoor environment to perform a thermal analysis. At the moment of writing the gbXML, some duplicity and errors were found in the polygons of both sides of walls that divide adjacent rooms. The algorithm presented in this paper was developed to detect and solve these errors. Also, this paper includes the testing and validation of the algorithm. The test is based on the application of the algorithm to six different scenarios of study: four of them are artificial scenarios developed to test the different issues detected and two of them are real cases.

## 1. INTRODUCTION

Indoor mapping and BIM (Building Information Models) generation have been popular topics in recent years. Their applications seem uncountable and increase every year. A BIM is a methodology that allows integrating all the information of a building into the same model in all stages of its lifespan. Therefore, its utility includes the design of the building and allows knowing the state of the building at all moments, improving its management. There are different standards for the BIM methodology such as IFC (“Industry Foundation Classes (IFC) - buildingSMART International,” n.d.) and gbXML (“gbXML - An industry supported standard for storing and sharing building properties between 3D Architectural and Engineering Analysis Software,” n.d.), both open. BIM can also integrate useful thermal data of the building like the building heat load, materials of the building, which are interesting for the performance of thermal analysis of the building. This topic has received increasing attention since global government policies focus on energy efficiency of the buildings. An example is the Energy Performance of Buildings Directive (“EUR-Lex - 02010L0031-20181224 - EN - EUR-Lex,” n.d.) which, with the NZEB ( Nearly zero-energy buildings), promotes that every new building has nearly zero energy consumption for 2020 or with or the energy rulebook from EU called “ Clean energy for all Europeans package” (“Clean energy for all Europeans package | Energy,” n.d.) with a subsection about energy efficiency in buildings and how to renew existing buildings.

However, energy inspections are especially complicated in patrimonial buildings, where planes can be outdated and even not exist. First, the geometric model of the building needs to be generated, which is typically a time-consuming process both in the data acquisition and the data processing procedures (Tang et al., 2010) (Hong et al., 2015). Then, the geometric model has to be adapted to the requirements of the BIM used with energy software, usually through a simplification process from the ornamentally rich geometries of patrimonial buildings to the polygonal geometries introduced in energy analysis (Díaz-

Vilariño et al., 2013). That is why an automatic process to generate the models of the buildings, combined with indoor mapping systems is so important. These methodologies use the point cloud obtained with the indoor mapping systems to estimate the 3D models. Several methodologies of cloud-to-BIM have been developed, like (Jung et al., 2014), (Adán et al., 2020) or (Ochmann et al., 2019).

The 3D models of the BIM consist of the vertexes of the surfaces which compound the building. Thus, the first step of the “as-built” systems is to estimate the vertexes of the building using the surveyed point cloud. The biggest complexity that these systems have to deal with is how to work with multiple rooms. This presents some questions like considering the thickness of the walls, which walls belong to each room, which rooms are adjacent, and how the rooms are connected.

This article presents the algorithm to solve a problem detected regarding the duplicity of polygons which correspond with the two sides of a wall in a software for the estimation of BIM without considering the thickness of the walls. The algorithm is tested among six different scenarios. Four of them are synthetic and represent the basic problems that the algorithm must be able to detect and correct. The other two are real scenarios to test the performance of the algorithm in real conditions.

This article is organized as follows: Section 2 describes the proposed algorithm and the scenarios to test it. Section 3 presents de results of the algorithm. Section 4 discusses the proposed method. Finally, Section 5 includes the conclusions reached after this study.

## 2. BACKGROUND

With the aim at generating BIM for existing buildings, it has been developed a software system to obtain the BIM from indoor point clouds, for thermal analysis purposes. The chosen schema for BIM representation is gbXML, with proven efficiency for these thermal purposes (Díaz-Vilariño et al., 2013; Lagüela et al., 2014; Wang and Cho, 2015). This

software obtains a regularized 3D model of the previously scanned point cloud, where adjacent walls are merged, avoiding the thickness of the walls. Because of this regularization and the requirements of the system developed, some explanations about real scenarios are required:

First, Manhattan World is assumed, and the scenarios consists of horizontal floors and ceilings, vertical walls vertical and oriented according to the cardinal axis.

Second, each point cloud inputted in the software corresponds only to one floor. Therefore, the algorithm is developed to work with this same characteristic.

Also, walls are considered without thickness for the gbXML generation. However, if thickness was considered, walls should be parallel but with a distance between them.

Another assumption is that to regularize the 3D models, the software system considers as in the same plane those walls that are parallel but present a distance between their planes less than 0.5 m.

In addition, all rooms in each scenario should present floors and ceilings in the same plane, respectively.

However, the performance of the software presented some problems. The main problem is the duplicity of adjacent surfaces, due to the thickness of real walls (side surfaces of each wall are identified as two different surfaces). However, gbXML requires the correction of this duplicity. Each wall must be represented by only one surface, defined by its vertexes, and the presence of two walls or the portion of two walls in the same location is considered as an error and must be corrected (“Green Building XML (gbXML) Geometry Test Cases - Test Case 6,” n.d.). As long the objective of the software developed is to automatize the processing of indoor point clouds to obtain a BIM, the developed algorithm for correction of the duplicity of surfaces must work automatically, robustly and in a wide variety of scenarios. The algorithm matches with the same assumptions than the software where it is integrated: Manhattan World, walls without thickness and restricted to only one floor by by analysis.

### 3. MATERIALS AND METHODS

#### 3.1 Algorithm

In this section, the performance of the algorithm is explained. Figure 1 shows the flow chart of the algorithm.

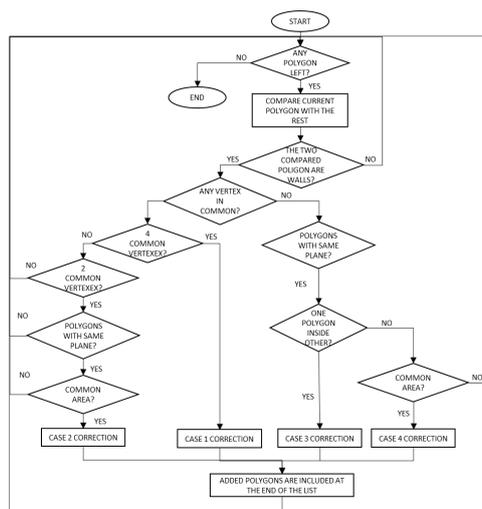


Figure 1. Merge Algorithm Diagram

In gbXML, each surface is described by the vertexes of its polygon. Assuming Manhattan World and that all walls are rectangles, each wall is described by 4 vertexes. Floors and ceilings can have different geometries, but walls area always considered as vertical rectangles. Each polygon is compared with the rest to find the following cases, which need to be corrected:

1. Two adjacent rooms share a wall. Before the correction, the wall is represented as two equal polygons, one for each room (Figure 2).

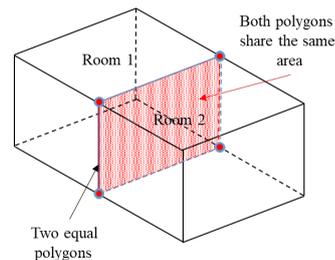


Figure 2. Case 1: Both rooms share equal polygons.

2. Two adjacent rooms share a portion of the division wall and one of its vertical edges. Both polygons have two vertexes (an edge) and a part of its area in common (Figure 3.)

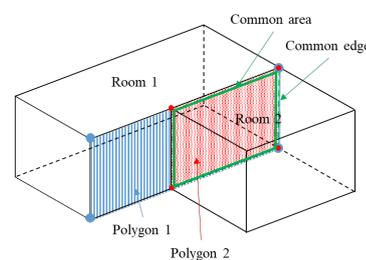


Figure 3. Case 2: Both rooms share an edge and a portion of the area of the neighbouring wall.

3. Two adjacent rooms share a portion of the division wall. This portion corresponds with the polygon of the smallest room and is totally contained into the polygon of the biggest one (Figure 4).

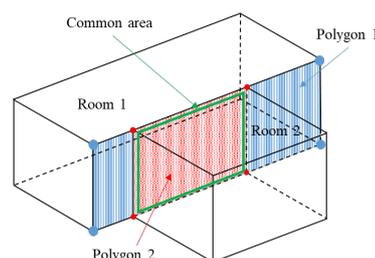


Figure 4. Case 3: Both rooms share a common surface which corresponds with the polygon of the smallest room

4. Two adjacent rooms share a portion of the division wall. Neither polygon of the rooms is contained in another (Figure 5).

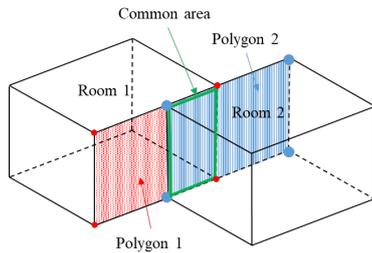


Figure 5. Case 4: Both rooms share a common surface without sharing their edges and none is contained in the other.

The input of the algorithm is a list of all polygons of the building, but only the walls are analysed because the problem of duplicity is only present for them. The algorithm compared each polygon with the rest only if both are walls. Otherwise, it jumps to the next comparison. The algorithm searches for the following conditions:

- Two compared polygons have four vertices in common. This condition denotes the case 1 in the algorithm (Figure 2). Both polygons are equal. This case is corrected by deleting one of the polygons (Figure 6).

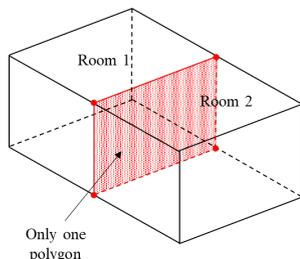


Figure 6. Case 1 correction.

- Two compared polygons have two vertices in common. If this condition is fulfilled the algorithm evaluates if the polygons are in the same plane and have a common area. With these conditions, the algorithm reaches the case 2 (Figure 3). This case is corrected by dividing the polygon of the biggest adjacent wall into two polygons: one for the common part and another for the uncommon part (Figure 7).

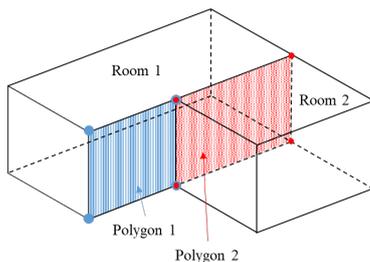
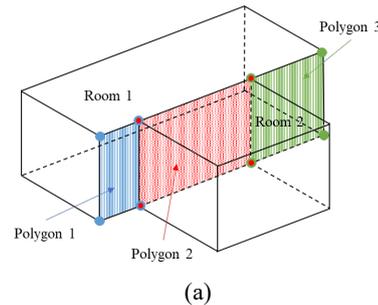


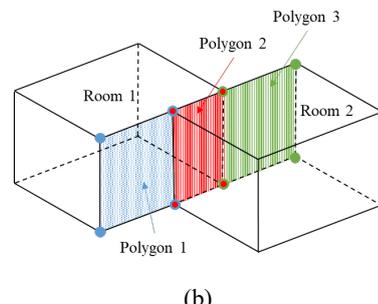
Figure 7. Case 2 correction.

- Two compared polygons have no vertices in common, but the two polygons are in the same plane. Then the algorithm evaluates if they share a portion of their area and if this area corresponds with the area of one of the polygons (case 3 and Figure 4) or it corresponds with a segment of both polygons (case 4 and Figure 5). In case 3, the biggest polygon is

divided into three different polygons: two for the uncommon parts and another for the common part which is in between (Figure 8.a.). The polygon of the smallest adjacent room is deleted. In case 4, one of the adjacent polygons is divided into two polygons, one for its common part and the other for the uncommon part. The polygon of the other adjacent space is limited to the uncommon part (Figure 8.b.).



(a)



(b)

Figure 8. Cases 3 (a) and 4 (b) correction.

If the algorithm detects any of the previously explained cases, it corrects the surfaces and, in the case that the correction implies the generation of new surfaces (cases 3 and 4), these are added at the end of the list. Then, the algorithm compares again the current surface with the rest of the list to test if there are new incidences because of the change on the current surface. Only if no new corrections are performed the algorithm continues to the next comparison.

### 3.2 Scenarios for analysis

Six different scenarios are used for testing the performance of the algorithm. Four of them are synthetic scenarios generated with the purpose of testing the four cases described in the previous section. All scenarios present the adjacent walls represented by two different parallel surfaces with a gap between them, either synthetically or from a scanning system. The synthetic scenarios are the following:

- Scenario 1: Two equal rooms, adjacent by one wall (Figure 9.a.). This scenario was made to test case 1 (Figure 2).
- Scenario 2: Two rooms, one bigger than the other, with one adjacent wall and an edge of this wall in common. (Figure 9.b.). This scenario was designed to test case 2 (Figure 3).
- Scenario 3: two rooms, one bigger than the other, with one adjacent wall between them. They form a shape similar to a "T". The surface which corresponds with the smallest room is the same as the centre area of the surface of the biggest room

(Figure 9.c.). This scenario was designed to test case 3 (Figure 4).

- Scenario 4: two equal rooms with one adjacent portion of their walls. It represents two rooms displaced one from the other along the plane of the adjacent wall (Figure 9.d.). This scenario was designed to test case 4 (Figure 5).

The other two scenarios are real cases, where a combination of the previously studied cases can be obtained. There are used for testing the flow of the algorithm and its performance with multiple simultaneous incidences. The first real case represents three consecutive classrooms (Figure 9.e.) whereas the second is formed by three spaces where each one is adjacent to the other two (Figure 9.f.).

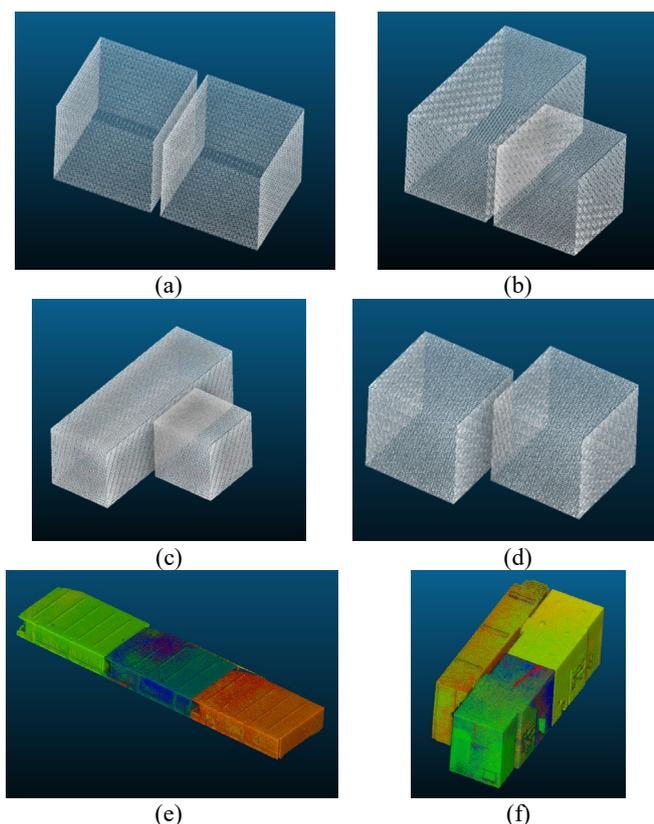


Figure 9. Scenarios: (a), (b), (c) and (d), synthetic scenarios; (e) and (f) real scenarios.

#### 4. RESULTS

In this section, the results of the algorithm after testing the proposed six scenarios are shown. As mentioned before, the algorithm is integrated into a system that analyses and processes point clouds to obtain a 3D model in gbXML. Thus, each solution is doubly shown: visualised via the software SketchUp and a schema that explains the correction performed.

##### 4.1 Scenario 1

This scenario was designed to test the case 1 of the algorithm. The two equal polygons belonging to the adjacent rooms are corrected into one polygon as seen in Figure 10.b.

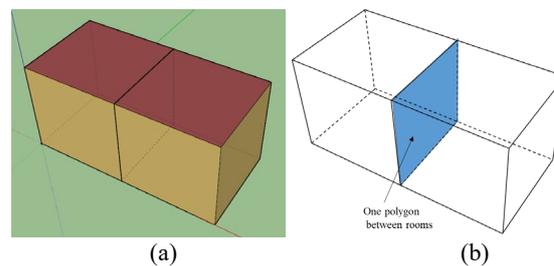


Figure 10. Scenario 1 result. (a) SketchUp visualization; (b) Schema

##### 4.2 Scenario 2

This scenario was designed to test the case 2 of the algorithm. Originally in the scenario, two surfaces were present in the same plane. The biggest one, which belonged to the room at the left, and the smallest one, which belonged to the room at the right and matched with the common part of two rooms. As can be seen in Figure 11 the correction made is to transform the surface of the biggest room into the uncommon part. As a result, there are two consecutive surfaces, one which corresponds to the common part and the other, which corresponds to the uncommon.

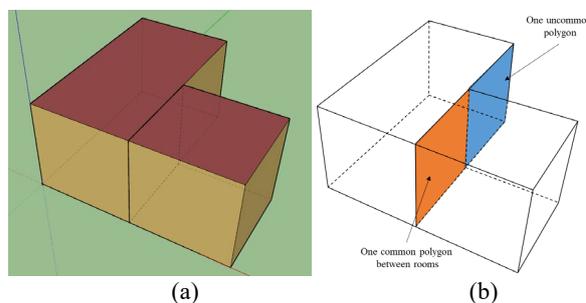


Figure 9. Scenario 2 result. (a) SketchUp visualization; (b) Schema

##### 4.3 Scenario 3

This scenario was designed to test case 3 of the algorithm. Originally there are two polygons in the same plane. One which belongs to the bigger room at the left and another which belongs to the smallest one at the right and is located in the centre of the other polygon. With the correction, three consecutive polygons are generated. The smallest polygon remains unaltered. The biggest one is divided into two different polygons which correspond with the uncommon parts as can be seen in Figure 12.b.

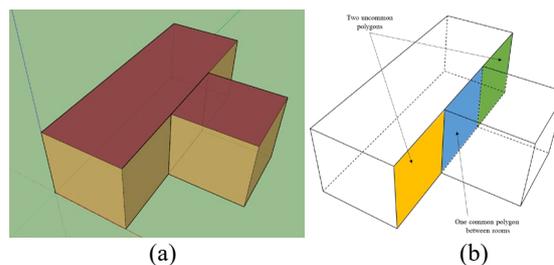


Figure 10. Scenario 3 result. (a) SketchUp visualization; (b) Schema

#### 4.4 Scenario 4

This scenario was developed to test the performance of the case 4 of the algorithm. The original model had two different polygons which belonged to the adjacent rooms respectively. The two polygons had a portion of this area in common. The algorithm divides one of the polygons into two, which correspond with the common and uncommon part. The polygon which belongs to the other room is transformed into the other uncommon part (Figure 13.b.).

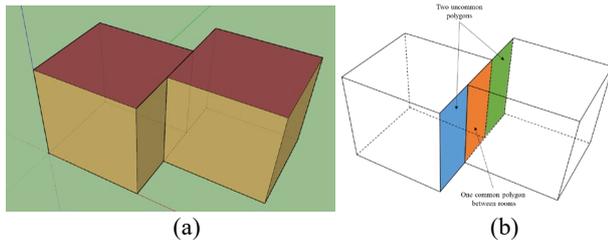


Figure 113. Scenario 4 result. (a) SketchUp visualization; (b) Schema

#### 4.5 Scenario 5

This scenario was captured with the Zeb-Revo (“ZEB Revo - GeoSLAM,” n.d.) system and belongs to three. As a real scenario, it combines different cases evaluated by the algorithm. This scenario is formed by two cases 1. Therefore, as a result, the model obtained corrects the adjacent polygons between rooms obtaining only one polygon between each pair of adjacent rooms (Figure 14.b.).

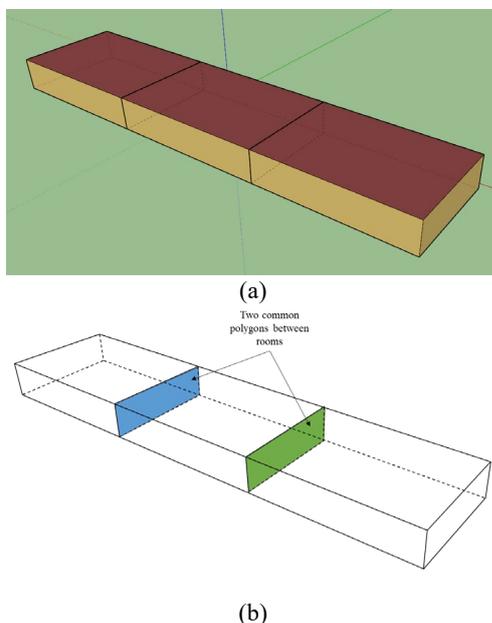


Figure 12. Scenario 5 result. (a) SketchUp visualization; (b) Schema

#### 4.6 Scenario 6

This scenario corresponds with two rooms and a corridor. It was also captured with the Zeb-Revo system. Scenario 6 is more complex than the previous scenario. It can be decomposed in the following cases: the corridor with each of the rooms correspond to case 2; the two adjacent rooms present the problem of case 1. Thus, Scenario 6 consists of one case 1

and two cases 2. The model without correction had two extra polygons. One is corrected by deleting the extra polygon between the two rooms. The other one is corrected by the modification of the polygons between the rooms and the corridor. This scenario also tested the performance of the algorithm when several problems are present. When the algorithm corrects the first problem between the corridor and one room, it operates like case 2. The polygon of the room remains equal and the polygon of the corridor is transformed in the uncommon part between them. But the modified polygon must be tested again, against the rest of the polygons because there is another incidence. With this correction, another case 1 incidence is found between the new polygon of the corridor and the polygon of the second adjacent room. As a result, the gbXML of Figure 15 is obtained.

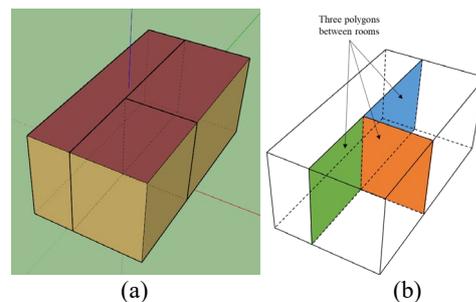


Figure 15. Scenario 6 result. (a) SketchUp visualization; (b) Schema

## 5. DISCUSSION

The algorithm developed works inside a software system that performs the modelling of the 3D indoor environments, but also their simplification and regularization for later thermal analysis.

Without this algorithm, the models generated cannot be used for energy analysis unless they represent only one space (room). Thus, its application is indispensable in all scenarios with more than one space.

The performance of the algorithm developed performance was very satisfactory. It worked as expected in all testing scenarios and without representing a working load for the system: the algorithm lasts less than 1 second in all cases, which is depreciable. It is difficult to establish the mean time of the system because it is a semi-automatic software and depends on the time of reaction of the user and the complexity of the analysed point cloud. In any case and discounting the time required for loading the point cloud in the software, the mean computation time for gbXML generation is around 1 minute and 30 seconds.

## 6. CONCLUSIONS

This paper describes an algorithm integrated into a software for the automatic processing of indoor point clouds to generate a BIM in gbXML. The objective of this algorithm is to solve the problem of the generation of duplicated and inexact polygons of walls in gbXML archives by the general software for BIM generation. The algorithm proposed analyses the list of polygons which correspond to each wall, detects the incidences and corrects them. The algorithm has been successfully tested against six different scenarios. Four of them consist of artificial scenarios developed to test the troubles found separately. The other two are real scenarios. All

scenarios were processed by the developed system and the algorithm was tested following the flow chart of the software where it is integrated. The algorithm worked fine in all cases. As future work, the functionality will be extended to multiple floor and rooms with different heights, and the performance of the algorithm without need for Manhattan World Assumption will be developed.

### ACKNOWLEDGEMENTS

Acknowledgments: This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 769255. This document reflects only the views of the author(s). Neither the Innovation and Networks Executive Agency (INEA) or the European Commission is in any way responsible for any use that may be made of the information it contains.

Authors would also like to thank Iberdrola S.L. and University of Salamanca

### REFERENCES

- Adán, A., Quintana, B., Prieto, S.A., Bosché, F., 2020. An autonomous robotic platform for automatic extraction of detailed semantic models of buildings. *Autom. Constr.* 109, 102963. <https://doi.org/10.1016/j.autcon.2019.102963>
- Clean energy for all Europeans package | Energy [WWW Document], n.d. URL [https://ec.europa.eu/energy/topics/energy-strategy/clean-energy-all-europeans\\_en](https://ec.europa.eu/energy/topics/energy-strategy/clean-energy-all-europeans_en) (accessed 4.20.20).
- Díaz-Vilariño, L., Lagüela, S., Armesto, J., Arias, P., 2013. Semantic as-built 3d models including shades for the evaluation of solar influence on buildings. *Sol. Energy* 92, 269–279. <https://doi.org/10.1016/j.solener.2013.03.017>
- EUR-Lex - 02010L0031-20181224 - EN - EUR-Lex [WWW Document], n.d. URL <https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1583922805643&uri=CELEX:02010L0031-20181224> (accessed 4.20.20).
- gbXML - An industry supported standard for storing and sharing building properties between 3D Architectural and Engineering Analysis Software [WWW Document], n.d. URL <https://www.gbxml.org/> (accessed 4.13.20).
- Green Building XML (gbXML) Geometry Test Cases - Test Case 6 [WWW Document], n.d. URL [https://www.gbxml.org/TestCase6\\_Geometry.html](https://www.gbxml.org/TestCase6_Geometry.html) (accessed 3.31.20).
- Hong, S., Jung, J., Kim, S., Cho, H., Lee, J., Heo, J., 2015. Semi-automated approach to indoor mapping for 3D as-built building information modeling. *Comput. Environ. Urban Syst.* 51, 34–46. <https://doi.org/10.1016/j.compenvurbsys.2015.01.005>
- Industry Foundation Classes (IFC) - buildingSMART International [WWW Document], n.d. URL <https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/> (accessed 4.13.20).
- Jung, J., Hong, Sungchul, Jeong, S., Kim, S., Cho, H., Hong, Seunghwan, Heo, J., 2014. Productive modeling for development of as-built BIM of existing indoor structures. *Autom. Constr.* 42, 68–77. <https://doi.org/10.1016/j.autcon.2014.02.021>
- Lagüela, S., Díaz-Vilariño, L., Armesto, J., Arias, P., 2014. Non-destructive approach for the generation and thermal characterization of an as-built BIM. *Constr. Build. Mater.* 51, 55–61. <https://doi.org/10.1016/j.conbuildmat.2013.11.021>
- Ochmann, S., Vock, R., Klein, R., 2019. Automatic reconstruction of fully volumetric 3D building models from oriented point clouds. *ISPRS J. Photogramm. Remote Sens.* 151, 251–262. <https://doi.org/10.1016/j.isprsjprs.2019.03.017>
- Tang, P., Huber, D., Akinci, B., Lipman, R., Lytle, A., 2010. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Autom. Constr.* <https://doi.org/10.1016/j.autcon.2010.06.007>
- Wang, C., Cho, Y.K., 2015. Automated gbXML-Based Building Model Creation for Thermal Building Simulation. *Proc. - 2014 Int. Conf. 3D Vis. Work. 3DV 2014* 111–117. <https://doi.org/10.1109/3DV.2014.109>
- ZEB Revo - GeoSLAM [WWW Document], n.d. URL <https://geoslam.com/solutions/zeb-revo/> (accessed 2.11.20).