

ALGORITHM FOR THE COUNTERCLOCKWISE ORDERING OF VERTEXES OF SLANTED SURFACES TOWARDS THE GENERATION OF SEMANTIC GBXML MODELS

R. Otero^{1*}, S. Lagüela², P. Arias¹

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

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

Commission IV, WG 10

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

ABSTRACT:

This paper presents an algorithm developed to solve a problem during the development of a system to estimate de 3D model of buildings roofs and its incorporation into a BIM. The chosen schema for the BIM is gbXML and one of its condition is that the vertexes that describe the model surfaces must be counterclockwise ordered. Due to the variability in the orientation of the analysed surfaces, the algorithm must be able to work with any surface regardless its orientation. Also, this paper includes the testing of the algorithm. This test is based on the application of the developed algorithm among three different real-world scenarios.

1. INTRODUCTION

One world which has gained more and more importance in construction and related technologies is BIM (Building Information Models) and not only in academic environments. One simple search in google shows 141.000.000 results in less than one second. It can be considered the technology to centralize the digital data of buildings (Zhou et al., 2012) in the whole life cycle (Jin et al., 2019). There are different standards of 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.), which integrate the 3D information of the model and semantic data like their thermal properties.

The use of BIM technology eases the integration of the 3D models of buildings into a data structure designed to be used in architectural or engineering analysis software and ease to share (“gbXML - An industry supported standard for storing and sharing building properties between 3D Architectural and Engineering Analysis Software,” n.d.). Being more specific, this paper is the result of the necessity of de development of algorithms for estimate the 3D model of roofs. The importance of these elements is unquestionable: they are one of de main isolation parts of the building, they are used as support of aeriels and solar panels...An example of this last application of solar PV installations for self-supply in buildings can be seen in (Martín-Jiménez et al., 2020). In those cases, the geometry of the roofs is necessary in order to design the position of the solar PV panels and estimate the consequent production of energy. Furthermore, for this purpose the localization data of the building (latitude, longitude, and orientation) is necessary. Traditionally, roofs are surveyed by aerial methods. However, indoor analysis of the roofs eases the process. The reason is that indoor geometric data acquisition is more affordable than

other methods such as the use of drones or aerial vehicles and requires operators with lower expertise.

The algorithm presented in this paper aims at completing the automatic 3D modelling of roofs using indoor point clouds and the gbXML schema. One of the conditions of the gbXML schema is that each surface must be compound by its vertex points, ordered counterclockwise from the inside of the space, which belongs to the surface. The main challenge of this procedure lies on the geometric complexity of the roofs: although many roofs are formed by planes with several (mostly two) vertexes in common, there are roofs formed by planes with only one vertex in common.

This work presents an evolution with respect to other works (Otero et al., 2020), solving some restrictions like The Manhattan World Assumption. The algorithm was developed to work in real-world scenarios. The performance of this algorithm was tested among three different real-world scenarios. This article is organized as follows: Section 2 describes the background and assumptions made for the correct performance of the algorithm. Section 3 describes the algorithm and the scenarios where it is tested. Section 4 discusses the proposed method. Finally, Section 5 includes the conclusions reached after this study.

2. BACKGROUND

There exist several studies that prove the utility of the gbXML schema for BIM, like (Díaz-Vilariño et al., 2013). This schema allows the inclusion of the geometric data of the roof, its localization data, and has a proven utility in energy analysis (Di Giuda et al., 2015; Jalaei and Jrade, 2014; Wang and Cho, 2015) For this reason, it was chosen for the 3D modelling of roofs using indoor point clouds. The basic structure of a gbXML document can be seen in Figure 1. The basic entities of this schema are surfaces. The 3D model is divided into planar surfaces defined by its vertexes. And these vertexes must be ordered counterclockwise (“GreenBuildingXML_Ver6.01,” n.d.).

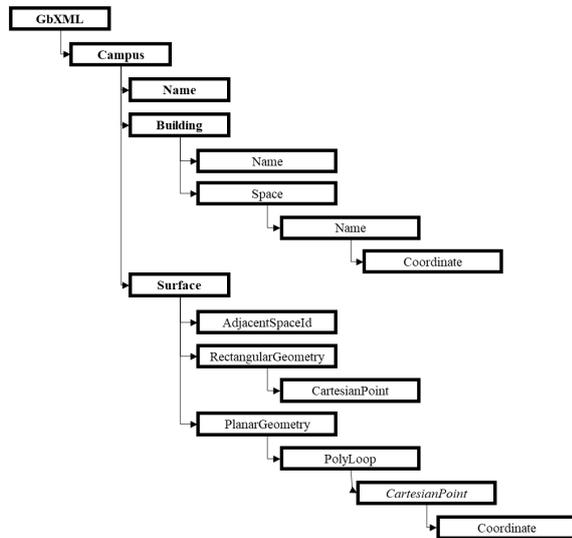


Figure 1. GbXML basic schema

The counterclockwise ordering of the vertices has already been addressed by works like (Otero et al., 2020), where the application is limited to scenarios with Manhattan World Assumption, that is, orthonormal planes, and parallel to one of the cardinal planes. However, the methodology proposed in this work does not limit the orientation and relative positions of the planes, although some hypothesis are established:

- Surfaces are considered with thickness = 0 for the gbXML generation.
- Walls are considered as vertical.
- The floor of the building is considered as horizontal.

3. MATERIALS AND METHODS

3.1 Algorithm

The purpose of this section is to explain the performance of the algorithm. Figure 2 shows its general flow chart.

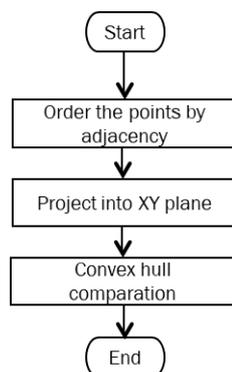


Figure 2. Counterclockwise ordering algorithm flow chart.

The algorithm presented in this paper is based on the hypothesis that each point is generated by the intersection of at least three non-parallel surfaces. Therefore, after estimating the vertices as the intersection of surfaces, the algorithm associates each point to the planes intersected for its generation. The input of the algorithm consists of three C++ map structures (“map - C++ Reference,” n.d.): the first map structure contains the relation between the indexes of the

analysed surface and the indexes of the unorganised vertices of the surface; the second map structure contains the relation between the indexes of each point and the corresponding value of the 3D point; the third map contains the index of each point and the planes which intersect in this point. The output of the algorithm is another map structure, similar to the second input structure but with the indexes of the points counterclockwise ordered.

The algorithm organizes the points of each surface individually using the input data as follows:

The first step is to obtain a closed loop in the points of a surface (Figure 3): each point is associated with the surfaces intersected for its generation. Adjacent points have in common two intersected surfaces: the surface which is being described and the surface that connects the two points (Figure 4). The points are ordered the common surfaces. The algorithm converges when a point that is already ordered is found, for the surface under study. Thus, the repeated points and the points between them conform the close loop and a surface. If there are any remaining points, the process is repeated, considering these points belonging to an additional surface, until no point remains.

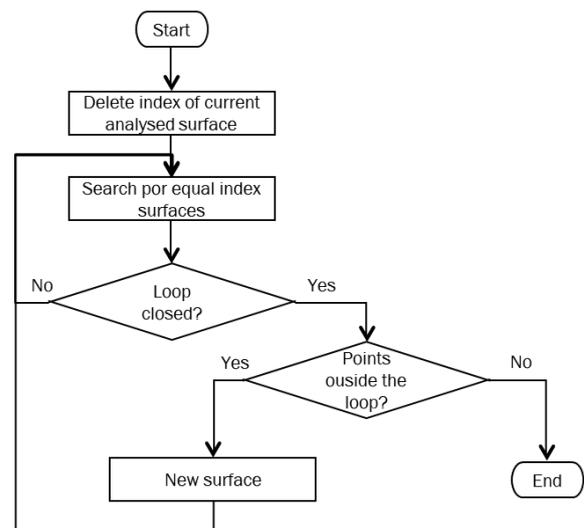


Figure 3. Flow chart of adjacency ordering of vertices.

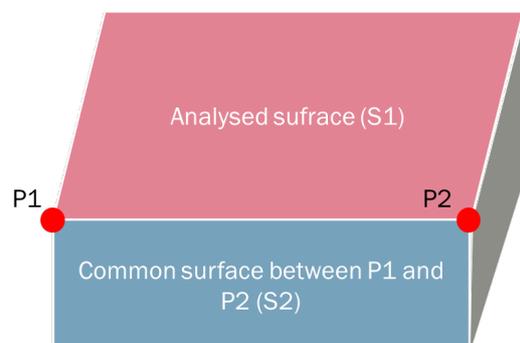


Figure 4. Schema of adjacency of two points using two common surfaces.

At this point, the vertices are ordered by their adjacency. Then they are counterclockwise ordered using PCL features (“Introduction — Point Cloud Library 0.0 documentation,” n.d.). This process can be seen in Figure 5. First, the ordered points are projected into the XY plane and then the convex hull (“Point Cloud Library (PCL): pcl::ConvexHull< PointInT >

Class Template Reference,” n.d.) is estimated. The PCL method for the estimation of the convex hull gives the mentioned convex hull points clockwise ordered. Therefore, the sequence must be inverted. The convex hull provides a reference for the sequence of points. They are compared and, if the points in the sequence follow the same order than the inversed convex hull, they are already counterclockwise ordered. If they follow the opposite order, the sequence must be reversed.

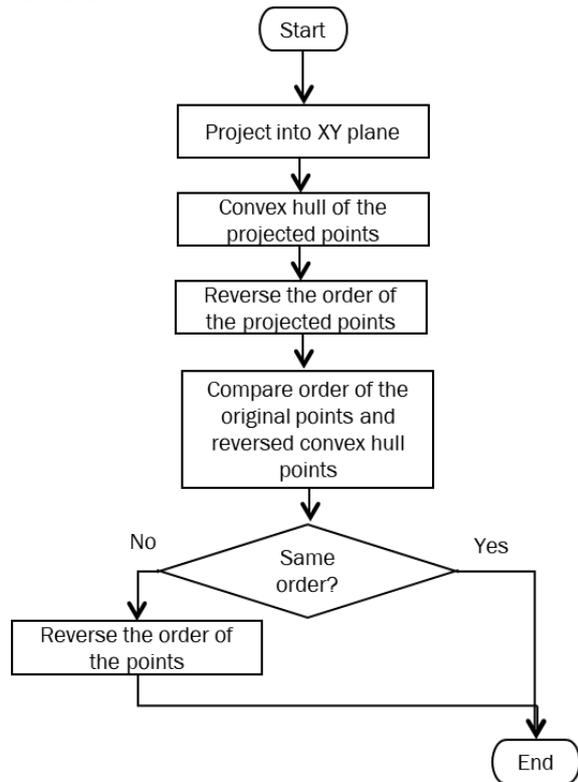


Figure 5. Use of PCL features for counterclockwise ordering.

3.2 Scenarios

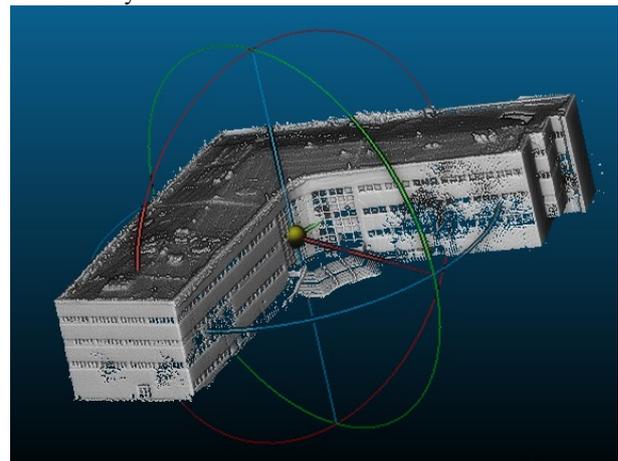
Three different real-world scenarios were used to exemplify and test the algorithm. The points and the classification of intersected surfaces used as input for the algorithm were estimated through the simplification and regularization of the point cloud, prior to the computation of their intersection for the definition of the vertexes as intersection points.

Scenario 1, which can be seen in Figure 6a is a near L-shaped building with a flat roof. The roof is defined by 6 vertexes originated by the intersection of 7 different planes. Each vertex is originated by the intersection of the roof surface with two non-parallel surfaces (walls).

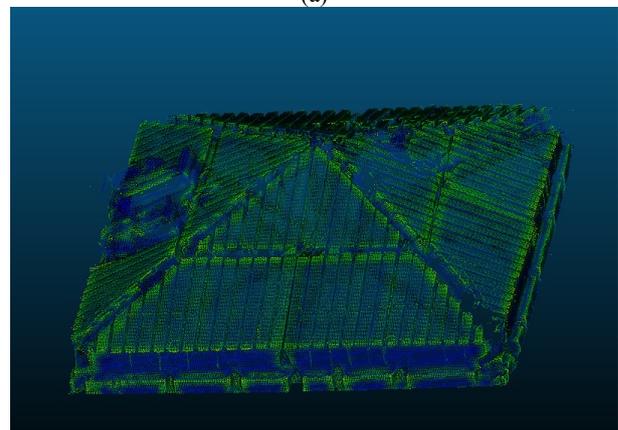
Scenario 2, shown in Figure 6b, is the roof of the tower of a patrimonial building with a rectangular projection. It is formed by four different triangular surfaces. Each surface is described by three points: one obtained by the intersection of the four roof surfaces, and the other two obtained by the intersection of each two adjacent roof surfaces with other two non-parallel surfaces (walls).

Scenario 3, which can be seen in Figure 6c, is the roof from the tower of another patrimonial building, in this case with a near-octagonal projection. Its regularized form is composed of eight triangular surfaces, one of them shorter than the rest. Each surface is defined by three vertexes, except for the two adjacent surfaces to the short one, which are defined by four

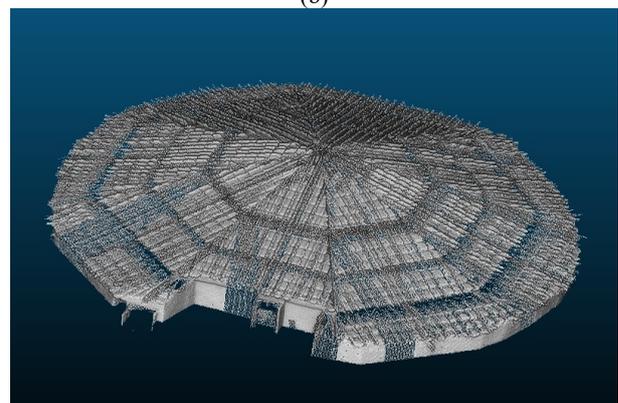
points: three vertexes and an additional point in the middle of the common segment with the short surface (which can be treated as vertexes for the algorithm). Like in the previous scenario, all surfaces share a common vertex on the top of the building, estimated by the intersection of the 8 surfaces. The rest of the vertexes of the surfaces correspond with the intersection of two adjacent roof surfaces with two non-parallel surfaces (walls) or the intersection of the roof surface with two walls directly.



(a)



(b)



(c)

Figure 6. Point clouds of the scenarios: (a) Scenario 1; (b) Scenario 2; (c) Scenario 3.

4. DISCUSSION

In this section the performance of the algorithm is tested against three different real scenarios previously described. Most of the examples have a very repetitive workflow into the

different surfaces analysed. To explain and test the performance of the algorithm, the most representative surfaces of each example are discussed.

Scenario 1 is composed of only one surface. This surface presents index 1, The rest of the surfaces are ordered giving consecutive indexes to near-parallel surfaces. The order of the vertices is given following the order followed for the intersection of the vertical surfaces. This can be seen in Figure 7.

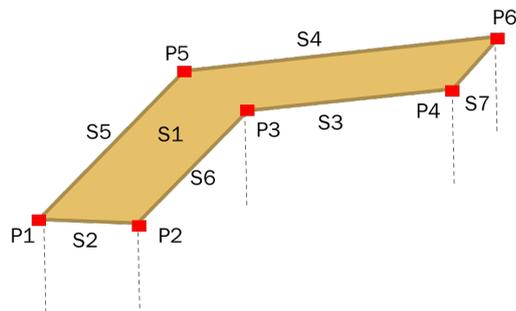


Figure 7. Schema of vertices and surfaces for Scenario 1

Thus, as input data the algorithm has three different structures:

- Map of indexes of the analysed surface and indexes of the vertices of the surface (Table 1).
- Map of the index of each vertex and its 3D value (Table 2).
- Map of the index of each vertex and the planes used for its estimation (Table 3).

| Index of analysed surface | Indexes of vertices of the surface |
|---------------------------|------------------------------------|
| 1 | 1, 2, 3, 4, 5, 6 |

Table 1. Map of indexes of the analysed surfaces and indexes of the vertices of the surface for Scenario 1.

| Index of the vertex | Value of the vertex |
|---------------------|--------------------------|
| 1 | -55.669, -41.865, 15.964 |
| 2 | -34.675, -33.480, 15.964 |
| 3 | -51.955, 10.509, 15.964 |
| 4 | -21.525, 36.967, 15.964 |
| 5 | -80.468, 19.476, 15.964 |
| 6 | -80.468, 19.476, 15.964 |

Table 2. Map of the index of each vertex and its 3D value for Scenario 1

| Index of the vertex | Planes used to estimate the vertex |
|---------------------|------------------------------------|
| 1 | 1, 2, 5 |
| 2 | 1, 2, 6 |
| 3 | 1, 3, 6 |
| 4 | 1, 3, 7 |
| 5 | 1, 4, 5 |
| 6 | 1, 4, 7 |

Table 3. Map of the index of each vertex and the planes used to estimate it for Scenario 1.

By the combination of these three structures, the vertices can be counterclockwise ordered. The first step is adjacency order. As mentioned before, the surface analysed is omitted: in this

case the surface with index 1. The ordering process can be seen in the graph of Figure 8.

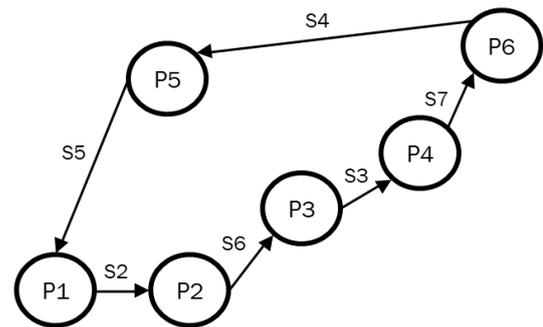


Figure 8. Graph diagram of the adjacency ordering for Scenario 1.

The vertices are ordered by adjacency but their counterclockwise order must be tested. The convex hull of the vertices is estimated using PCL libraries. This result is presented in clockwise order so it must be reversed. First, the vertices must be projected into the XY plane (in this case they are already in a parallel plane) and then the convex hull is estimated. The result of this estimation is the following vertices order: 1, 2, 4, 6, 5. Vertex 3 is outside the convex hull. Comparing this result with the vertices ordered by adjacency, it can be seen that they are in the same order, so the vertices are already counterclockwise ordered after being reverted.

The output of the algorithm is a map structure of size 1 for this case, with association of the index of the surface and the ordered indexes of the vertices of this surface (Table 4).

| Index of ordered surface | Ordered points |
|--------------------------|------------------|
| 1 | 1, 2, 3, 4, 6, 5 |

Table 4. Map structure with association of the index of the surface and the ordered indexes of the vertices of this surface for Scenario 1.

In Scenario 2, four non-parallel surfaces are modelled. All four surfaces are described by three vertices, and the process is equal for all of them. Surfaces and vertices are modelled like in Scenario 1, with one particularity. The first vertex in this case is the one obtained by the intersection of the four surfaces of the roof. This is followed by the vertices obtained by the intersection of surfaces from the roof and surfaces from the walls. The schema of the vertices and surfaces can be seen in Figure 10.

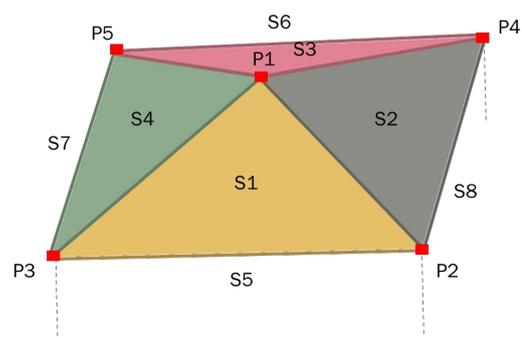


Figure 9. Schema of vertices and surfaces for Scenario 2.

As in Scenario 1, the input data is:

- Map of indexes of the analysed surfaces and indexes of the vertexes of the surface (Table 5).
- Map of the index of each vertex and its 3D value (Table 6).
- Map of the index of each vertex and the planes used to estimate it (Table 7).

| Index of analysed surface | Indexes of vertexes of the surface |
|---------------------------|------------------------------------|
| 1 | 1, 2, 3 |
| 2 | 1, 2, 4 |
| 3 | 1, 4, 5 |
| 4 | 1, 3, 5 |

Table 5. Map of indexes of the analysed surfaces and indexes of the vertexes of the surface for Scenario 2.

| Index of the vertex | Value of the vertex |
|---------------------|-------------------------|
| 1 | 0.253, -4.572, 234.462 |
| 2 | 4.639, -9.450, 232.597 |
| 3 | -4.271, -9.450, 232.597 |
| 4 | 4.639, -0.308, 232.597 |
| 5 | -4.271, -0.308, 232.597 |

Table 6. Map of the index of each vertex and its 3D value for Scenario 2.

| Index of the vertex | Planes used to estimate the vertex |
|---------------------|------------------------------------|
| 1 | 1, 2, 3, 4 |
| 2 | 1, 2, 5, 8 |
| 3 | 1, 4, 5, 7 |
| 4 | 2, 3, 6, 8 |
| 5 | 3, 4, 6, 7 |

Table 7. Map of the index of each vertex and the planes used for its estimation for Scenario 2.

In this case, the process is equal in the four surfaces. As an example, the algorithm for surface 1 can be discussed. The ordering by adjacency can be seen in Figure 10, and it is very similar to the process of scenario 1 but with fewer steps.

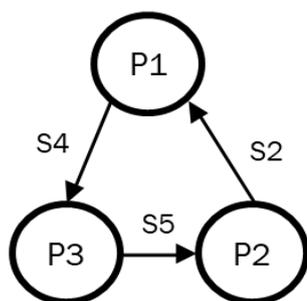


Figure 10. Graph diagram of the adjacency ordering for Scenario 2.

The vertexes for surface 1 are ordered by adjacency. If the convex hull is calculated, the result shows that vertexes are clockwise ordered and their order must be reversed. The same process is repeated for the other three surfaces.

The output of the algorithm applied to Scenario 2 is a map structure of size 4, where each element is a pair with the index of the surface and the ordered indexes of this surface (Table 8).

| Index of ordered surface | Ordered points |
|--------------------------|----------------|
| 1 | 1, 3, 2 |
| 2 | 1, 2, 4 |
| 3 | 1, 4, 5 |
| 4 | 1, 5, 3 |

Table 8. Map structure with association of the index of the surface and the ordered indexes of the vertexes of this surface for Scenario 2.

Scenario 3 is a roof from a patrimonial building, composed of eight non-parallel surfaces. In this case, there are three different types of surfaces. Surface 1 is represented by three vertexes and is a little smaller than the other seven surfaces. Surfaces 2 and 8 are composed of four points that can be considered as vertexes for the algorithm (despite points 2 and 3 which are in the same line as points 1 and 5 and 1 and 11, respectively). The rest of the surfaces are composed of three vertexes, as shown in Figure 11.

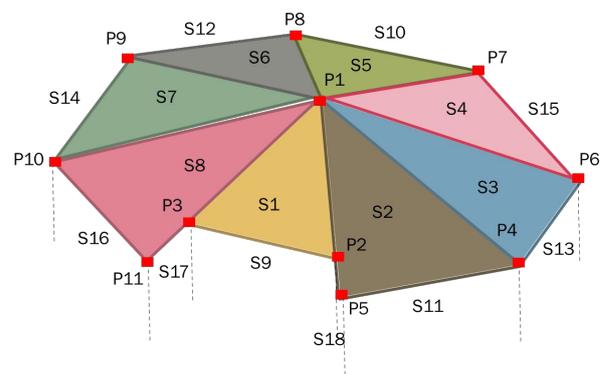


Figure 11. Schema of vertexes and surfaces for Scenario 3.

As in the two previous scenarios, the input data are:

- Map of indexes of the analysed surfaces and indexes of the vertexes of each surface (Table 9).
- Map of the index of each vertex and its 3D value (Table 10).
- Map of the index of each vertex and the planes used for its estimation (Table 11).

| Index of analysed surface | Indexes of vertexes of the surface |
|---------------------------|------------------------------------|
| 1 | 1, 2, 3 |
| 2 | 1, 2, 4, 5 |
| 3 | 1, 4, 6 |
| 4 | 1, 6, 7 |
| 5 | 1, 7, 8 |
| 6 | 1, 8, 9 |
| 7 | 1, 9, 10 |
| 8 | 1, 3, 10, 11 |

Table 9. Map of indexes of the analysed surfaces and indexes of the vertexes of the surface for Scenario 3.

| Index of the vertex | Value of the vertex |
|---------------------|-------------------------|
| 1 | -4.335, 0.882, 31.017 |
| 2 | 2.658, 1.997, 29.183 |
| 3 | 1.638, -2.699, 29.183 |
| 4 | 0.829, 7.670, 28.748 |
| 5 | 3.832, 2.273, 28.748 |
| 6 | -5.461, 8.998, 28.748 |
| 7 | -11.079, 5.756, 28.748 |
| 8 | -12.231, -0.666, 28.748 |
| 9 | -8.622, -5.934, 28.748 |
| 10 | -2.477, -7.584, 28.748 |
| 11 | 2.682, -4.103, 28.748 |

Table 10. Map of the index of each vertex and its 3D value for Scenario 3.

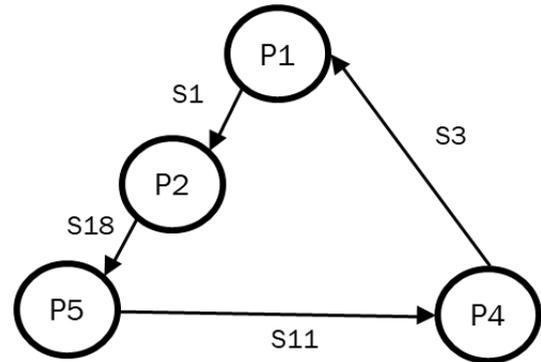


Figure 13. Graph diagram of the adjacency ordering for surface 2 of Scenario 3.

| Index of the vertex | Planes used to estimate the vertex |
|---------------------|------------------------------------|
| 1 | 1, 2, 3, 4, 5, 6, 7, 8 |
| 2 | 1, 2, 9, 18 |
| 3 | 1, 8, 9, 17 |
| 4 | 2, 3, 11, 13 |
| 5 | 2, 11, 18 |
| 6 | 3, 4, 13, 15 |
| 7 | 4, 5, 15, 10 |
| 8 | 5, 6, 10, 12 |
| 9 | 6, 7, 12, 14 |
| 10 | 7, 8, 14, 16 |
| 11 | 8, 16, 17 |

Table 11. Map of the index of each vertex and the planes used to estimate it for Scenario 3.

The algorithm performs the ordering process for the eight surfaces. The process is equal for surfaces 2 and 8 and for surfaces 3, 4, 5, 6, 7 and 1. Surfaces 1, 2 and 3 are selected for further description as examples.

The adjacency ordering for surface 1 can be seen in Figure 12. After this ordering process, the vertexes are tested among the convex hull as done in the previous scenarios. The result of the convex hull is the same points as in the surface 1 in order 1, 3, 2. Therefore, the vertexes are clockwise ordered and must be reversed.

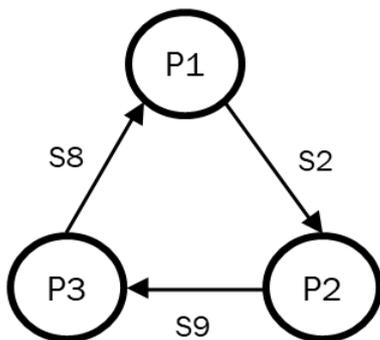


Figure 12. Graph diagram of the adjacency ordering for surface 1 of Scenario 3.

For surface 2, the adjacency ordering process is represented by Figure 13. After adjacency, the clockwise or counterclockwise order is tested using the convex hull. The result of the convex hull is points 1, 2, 5 and 4. For this case, the vertexes are already counterclockwise ordered. The same process is applied to surface 8.

For surface 3, the adjacency ordering process is represented by Figure 14. After adjacency, the clockwise or counterclockwise order is tested using the convex hull. The result of the convex hull is the same vertexes in order 1, 6 and 4. For this case, the vertexes are already counterclockwise ordered. The same process is repeated for surfaces 4, 5, 6 and 7.

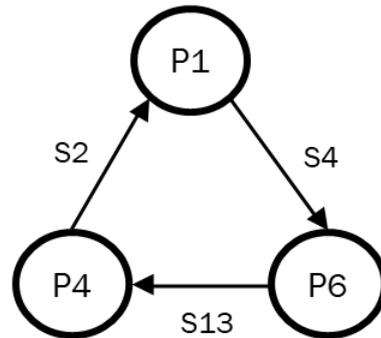


Figure 14. Graph diagram of the adjacency ordering for surface 3 of Scenario 3.

The output of the algorithm is a map structure of size 8 for this case, where each element is a pair of the index of the surface and the ordered vertexes of the surface, as shown in Table 12.

| Index of ordered surface | Ordered points |
|--------------------------|----------------|
| 1 | 1, 3, 2 |
| 2 | 1, 2, 5, 4 |
| 3 | 1, 4, 6 |
| 4 | 1, 6, 7 |
| 5 | 1, 7, 8 |
| 6 | 1, 8, 9 |
| 7 | 1, 9, 10 |
| 8 | 1, 10, 11, 3 |

Table 12. Map structure with association of the index of the surface and the ordered indexes of the vertexes of this surface for Scenario 3.

In all three scenarios, the algorithm lasts less than 1 second in the ordering of the points.

5. CONCLUSIONS

This paper describes an algorithm for the automatic modelling of roofs using indoor point clouds to generate a BIM. The chosen schema for the BIM is gbXML and one of its conditions is that the vertexes of the surfaces of the model must be counterclockwise ordered. This algorithm orders the vertexes based on the hypothesis that each vertex is generated by the intersection of at least three non-parallel surfaces and two adjacent vertexes share two surfaces: the described surface and another. In order to do this, the algorithm uses some features of PCL.

The algorithm is advancement respect previous works allowing to overcome the Manhattan World Assumption and making able to work with surfaces in any orientation, being this last point especially important to work with roofs, as can be seen in the scenarios used as tests.

Computationally is simple and robust with times of execution lower than 1 second in all the performed tests.

Its principal flaw is its dependency in the received data, requirement of the algorithm because its integration in a biggest system.

As future work, the functionality of the algorithm will be extended all type of building surfaces. This functionality falls out of the scope of the system where is integrated the algorithm but seems promising for the integration in systems that model the whole building.

REFERENCES

- Di Giuda, G.M., Villa, V., Piantanida, P., 2015. BIM and energy efficient retrofitting in school buildings. *Energy Procedia* 78, 1045–1050. <https://doi.org/10.1016/j.egypro.2015.11.066>
- 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>
- 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).
- GreenBuildingXML_Ver6.01 [WWW Document], n.d. URL https://www.gbxml.org/schema_doc/6.01/GreenBuildingXML_Ver6.01.html#Link19D (accessed 6.18.20).
- 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).
- Introduction — Point Cloud Library 0.0 documentation [WWW Document], n.d. URL <https://pcl.readthedocs.io/projects/tutorials/en/latest/#> (accessed 3.31.21).
- Jalaei, F., Jrade, A., 2014. Integrating Building Information Modeling (BIM) and Energy Analysis Tools with Green Building Certification System to Conceptually Design Sustainable Buildings. *ITcon* 19, 494–519.
- Jin, R., Zhong, B., Ma, L., Hashemi, A., Ding, L., 2019. Integrating BIM with building performance analysis in project life-cycle. *Autom. Constr.* <https://doi.org/10.1016/j.autcon.2019.102861>
- map - C++ Reference [WWW Document], n.d. URL <http://www.cplusplus.com/reference/map/map/> (accessed 3.31.21).
- Martín-Jiménez, J., Del Pozo, S., Sánchez-Aparicio, M., Lagüela, S., 2020. Multi-scale roof characterization from LiDAR data and aerial orthoimagery: Automatic computation of building photovoltaic capacity. *Autom. Constr.* 109, 102965. <https://doi.org/10.1016/j.autcon.2019.102965>
- Otero, R., Frías, E., Lagüela, S., Arias, P., 2020. Automatic gbXML modeling from LiDAR data for energy studies. *Remote Sens.* 12, 2679. <https://doi.org/10.3390/RS12172679>
- Point Cloud Library (PCL): pcl::ConvexHull< PointInT > Class Template Reference [WWW Document], n.d. URL https://pointclouds.org/documentation/classpcl_1_1_convex_hull.html (accessed 2.2.21).
- 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>
- Zhou, W., Whyte, J., Sacks, R., 2012. Construction safety and digital design: A review, in: *Automation in Construction*. Elsevier, pp. 102–111. <https://doi.org/10.1016/j.autcon.2011.07.005>