# Automatic IFC data enrichment with space geometries for Building Energy Performance Simulations

Georgios N. Lilis [1], Kyriakos Katsigarakis [1], Dimitrios V. Rovas [1]
[1]Institute for Environmental Design and Engineering, University College London, UK

## Abstract

The potential of using BIM geometry data to automate the process of setting up the geometry in whole building simulation models is very appealing. This transformation process is far from trivial and relies on good quality data. Missing or incorrect space geometries in the BIM model can create issues, especially in complex geometric configurations. In this paper, we present an algorithm to identify building space volume geometries automatically. The algorithm receives as input the geometric representations of architectural elements surrounding the building spaces and accurately reconstructs the internal building space volumes. The proposed method is tested in real-world examples and can be particularly useful for curved wall geometries. The transformation process relies on good quality geometric input data of the architectural elements avoiding clashes and surface errors; we define criteria to check for such errors and discuss the impact of such errors on the reconstructed space volumes.

## Key Innovations

- Automatic openBIM-based (IFC) building space geometry generation.
- Elimination of the error-prone manual building space design of the BEPS model generation process.
- Handling of complex building construction elements (walls, slabs, ...), that a manual space design processes cannot easily tackle.

## Practical Implications

According to the current state of the art, the generation of building energy performance simulation models from open-BIM data, is a manual and laborious process. During the design stage, certain design guidelines (rules) (Giannakis et al., 2019) should be followed to produce error-free designs. One of these rules refers to the accurate design of internal building space volumes, a difficult and time-consuming task, in cases where the surrounding to the building spaces, wall geometries are complex. Consequently, the proposed method aims to eliminate this manual internal building space design task, facilitating the overall BEPS model generation process, without introducing additional possible manual design errors.

## Introduction

Ways to automate the process of generating Building Energy Performance Simulation(BEPS) models, utilising Building Information Model (BIM) data has received significant attention lately (Andriamamonjy et al., 2018), (Cao et al., 2015). Fully automating the process can be particularly challenging: first, because complex transformations are required and not all the necessary data may be readily available on the BIM model, and second, because the form of the target BEPS model will depend on the intended use (e.g. a compliance model can be quite different from assessing, say, overheating risks). In the former case, additional information might have to be provided, e.g. from product and material libraries. In the latter case, any transformation process should encapsulate expert knowledge tailored to a particular use. While acknowledging some of these challenges, specific tasks are time-consuming and can be more readily automated. One of these is the transfer of geometric information, avoiding the need for duplicating efforts in setting up geometrical information in two separate environments (e.g. the BIM model or the BEPS model).

Even in this more limited scope, it may be hard to meet the specific end-use data requirements. For example, the building's second-level space boundaries (Bazjanac, 2010), required to set up the energy model may be missing or incorrectly defined in the BIM data files. Geometric transformation algorithms have been reported in the literature (Lilis et al., 2017), (Rose and Bazjanac, 2015), (Ladenhauf et al., 2016), (El-Diraby et al., 2017) that can automate second-level space boundary information identification. A recent review of these methods can be found in (Ying and Lee, 2021). The output of these algorithms can then enrich the BIM models. This enriched BIM model can then be readily used to generate BEPS models through a well-defined and relatively straightforward transformation process.

Like second-level space boundary information, knowledge of space volumes has many potential uses in energy modelling. Some of these include the ability to accurately compute space volumes, define space groupings that identify thermal zones, or pre-requisite for space-boundary identification algorithms. When this information is available on the BIM model, we can directly use this information. But

https://doi.org/10.26868/25222708.2021.30888

in cases where such information is missing or has errors, then we argue in this paper that this information can be automatically be generated. This paper's main focus is the Automatic Space Generation (ASG) algorithm to generates such spaces.

A prerequisite to the application of these algorithms is that the building geometry has been defined correctly. This requires BIM model checking that covers a range of data quality tests, including (i) checking the boundary representations of all building entities to ensure no missing surfaces and all surfaces are oriented correctly; (ii) clash detection, to ensure no clashes or intersections among the building architectural components and the internal building spaces, if any, should be attached to all surrounding building constructions (walls, slabs, ...). Widely available BIM model checkers, e.g. Solibri, can also be used to identify some of the errors mentioned above (Solibry, 2019).

A principle of minimisation emerges. The BIM model needs to have good quality geometric information to start. Therefore, BIM modellers can benefit from clear guidelines (Giannakis et al., 2019) and model checking tools to ensure data quality requirements are being met. Information related to space volumes and space boundaries can be algorithmically generated, reducing the need for human intervention and the opportunity for errors.

The semantic enrichment process outlined above is shown in Figure 1. Hereafter in this paper, we assume dealing with openBIM data in the Industry Foundation Class (IFC) format. This should help focus the discussion, but without loss of generality — the ASG algorithm would still work for other BIM formats. Given an IFC model of the building that has been designed following guidelines and quality checked, we are seeking to create an enriched model with geometric data for BEPS modelling.

Provided that building space volume descriptions are present in the IFC file, as indicated in (Lilis et al., 2018), error checks are required to ensure that the space definitions meet data quality requirements. These checks cover:

- **Surface Errors** are related to missing or inverted surfaces in the boundary representations (B-reps) of architectural elements and building spaces.
- **Clash Errors** refer to the intersections among the solid geometric representations of architectural elements and building spaces.
- **Space Errors** are defined as the boundary surface parts of internal space volumes which are not attached to other building architectural elements.

The term architectural elements refer to parts of the building fabric (walls, slabs, beams, columns, ...) that

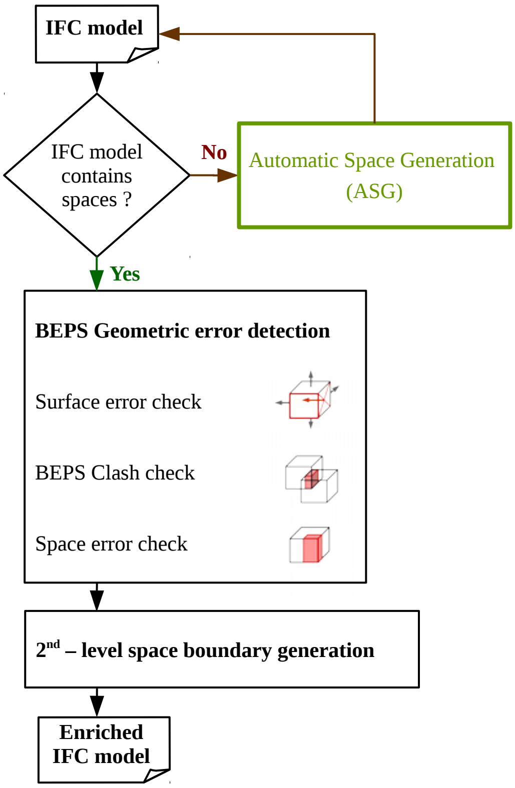
IFC model
IFC model contains spaces ?
No
Automatic Space Generation (ASG)
Yes
BEPS Geometric error detection
Surface error check
BEPS Clash check
Space error check
2nd – level space boundary generation
Enriched IFC model

*Figure 1: Semantic enrichment process that includes Automatic Space Generation, where this information is not available or has errors*

impede the building's thermal exchange with the environment. Other elements such as railings, staircases are ignored as the impact on the total building's energy balance is not significant.

In case the geometric representation of building space volumes is not present in the IFC data, or there are quality issues, the space volumes are automatically generated by the Automatic Space Generation (ASG) algorithm, which is the focus of this paper. The ASG algorithm's position in the semantic enrichment workflow is indicated, as a first-level space boundary generation operation, with a green block in Figure 1. Essentially, ASG generates the internal building space volumes (first-level space boundaries according to (Bazjanac, 2010)), from the geometrical information of the surrounding architectural elements and enriches in the output, the input IFC file with the geometrical representation of the generated space volumes.

As stated in (Pinheiro et al., 2018), although a spaces to zones mapping is required for a correct BIM to BEPS transformation process, the present work focuses on the space volume generation to facilitate the building's second-level space boundary topology generation without taking into account the building zones. Space number reduction using spaces to zones merging is investigated in second-level space boundary simplification algorithms (Lilis et al., 2019).

https://doi.org/10.26868/25222708.2021.30888

The rest of the paper is organized as follows: initially, a nomenclature table is presented, followed by the description of ASG and application examples' algorithmic process. The paper proceeds with analysing the advantages of ASG compared to manual design methods through BIM authoring design software. The paper concludes with a discussion on the limitations of the presented algorithm and its relative performance metrics.

### Nomenclature

All mathematical expressions conform to the following nomenclature.

| Symbol | Description |
|---|---|
| B-rep | Boundary representation |
| $a$ | Polygon a |
| $W_a$ | Wireframe of polygon a |
| A | Polygon set A |
| $W_\mathrm{A}$ | Wireframe of polygon set A |
| $\|A\|$ | Number of polygons in set A |
| A($i$) | Polygon i of polygon set A |
| $\partial$A | B-rep A |
| **A** | Set of polygon sets |
| LIS | Length of Isolated Segments |

## Algorithmic process

The ASG algorithmic process receives as input an IFC file that does not contain any information related to building internal space volumes (described by the IFCspace class) and enriches it with this missing information. The ASG can be split into four consecutive stages:

### Stage 1: B-rep generation

Architectural elements are described in IFC files with various geometrical representations, ranging from extruded area solids, manifolds, revolved area solids, faceted boundary representations and others. This stage aims to transform all solid representations of architectural elements in the input IFC file to a convenient form for further processing, the faceted boundary representation (B-rep).

### Stage 2: Boundary and internal surface extraction

In this stage, the common boundaries among the B-rep pairs are extracted. All possible B-rep pair combinations of all architectural elements are tested for common boundary surfaces (wall-wall, wall-slab, ...). For each B-rep entity $\partial\mathrm{B}_i$, the common boundaries shared among the B-rep and neighbour B-reps $j \in N_i$ are collected in common boundary polygon set $\mathrm{CB}_{ij}$. Here $N_i$ is the set of B-rep indices neighbour to B-rep i. The common boundary extraction is illustrated in part I of Figure 4 and implemented using the function $F_{cod}$ demonstrated in part III of Figure 2.

$$\mathrm{CB}_{ij} = F_{cod}(\partial\mathrm{B}_i, \partial\mathrm{B}_j) \qquad (1)$$

In case a clash occurs in a B-rep pair $\partial\mathrm{B}_i$ and $\partial\mathrm{B}_j$, the surfaces of B-rep i that are inside B-rep j are also collected into the internal surface set $\mathrm{IS}_{ij}$. $\mathrm{IS}_{ij}$ consists of surfaces of $\partial\mathrm{B}_i$ that are completely inside B-rep j which are returned by the function $F_{ins}$ (demonstrated in part I of Figure 2), and surfaces of $\partial\mathrm{B}_i$ which are also surfaces of $\partial\mathrm{B}_j$ which have the same normal vectors, returned by the function $F_{csd}$ (demonstrated in part II of Figure 2).

$$\mathrm{IS}_{ij} = [F_{ins}(\partial\mathrm{B}_i, \partial\mathrm{B}_j) \cup F_{csd}(\partial\mathrm{B}_i, \partial\mathrm{B}_j)] \qquad (2)$$

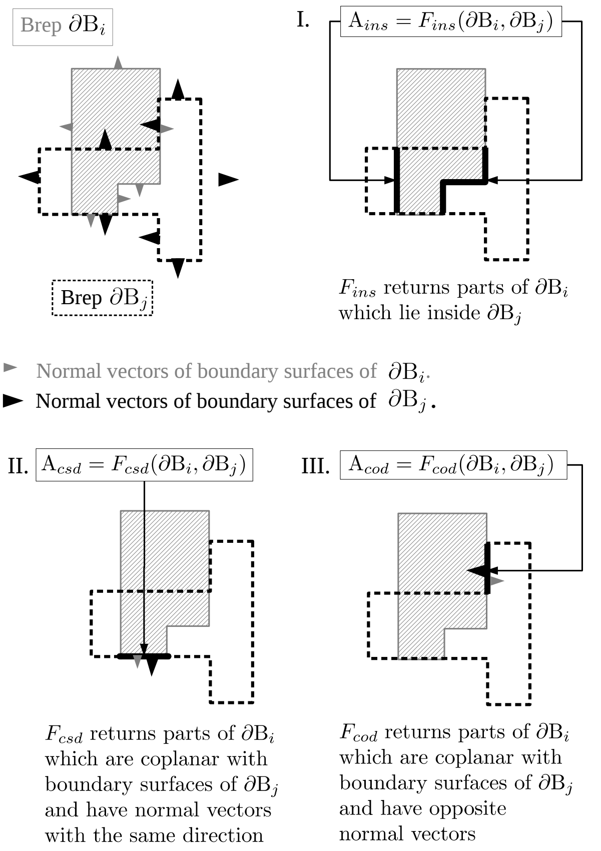
Brep $\partial\mathrm{B}_i$
I.
$\mathrm{A}_{ins} = F_{ins}(\partial\mathrm{B}_i, \partial\mathrm{B}_j)$
Brep $\partial\mathrm{B}_j$
$F_{ins}$ returns parts of $\partial\mathrm{B}_i$ which lie inside $\partial\mathrm{B}_j$
► Normal vectors of boundary surfaces of $\partial\mathrm{B}_i$.
► Normal vectors of boundary surfaces of $\partial\mathrm{B}_j$.
II.
$\mathrm{A}_{csd} = F_{csd}(\partial\mathrm{B}_i, \partial\mathrm{B}_j)$
III.
$\mathrm{A}_{cod} = F_{cod}(\partial\mathrm{B}_i, \partial\mathrm{B}_j)$
$F_{csd}$ returns parts of $\partial\mathrm{B}_i$ which are coplanar with boundary surfaces of $\partial\mathrm{B}_j$ and have normal vectors with the same direction
$F_{cod}$ returns parts of $\partial\mathrm{B}_i$ which are coplanar with boundary surfaces of $\partial\mathrm{B}_j$ and have opposite normal vectors

*Figure 2: Functions acting on B-rep pairs $\partial\mathrm{B}_i$ and $\partial\mathrm{B}_j$*

### Stage 3: Shell generation and classification

After the common boundary surfaces shared by B-rep pairs are extracted, they subtracted from the B-rep surfaces, and the remaining surfaces are obtained. This operation for B-rep $\partial\mathrm{B}_i$ can be expressed as:

$$\mathrm{R}_i = \bigcup_{\forall b\in\partial\mathrm{B}_i}\left[ b - \bigcup_{\substack{\forall c\in\mathrm{C}_i \\ c||b}} c \right], \mathrm{C}_i = \bigcup_{j\in N_i}[\mathrm{CB}_{ij} \cup \mathrm{IS}_{ij}] \qquad (3)$$

where $\mathrm{R}_i$ is the remaining polygon set of B-rep i, The subtraction (-) and union ($\cup$) are operations, on coplanar polygon pairs among the polygon sets $\partial\mathrm{B}_i$ and $\mathrm{C}_i$. If $c||b$ denotes that polygon c is co-planar with

https://doi.org/10.26868/25222708.2021.30888

polygon b, $\mathrm{C}_i$ contains the common boundaries $\mathrm{CB}_{ij}$ and internal surfaces $\mathrm{CB}_{ij}$, calculated in the previous step, of B-rep i with all neighbour B-reps $j \in N_i$.

Essentially the polygon set $\mathrm{R}_i$ contains the remaining polygons of $\partial \mathrm{B}_i$ after the common boundaries $\mathrm{CB}_{ij}$ are subtracted. The generation of $\mathrm{R}_i$ for B-rep i is illustrated in part II of Figure 4. All the polygon sets $\mathrm{R}_i$ are merged into one set R. Then the obtained set R is split again into shells which are polygon sets, forming a set of shells (polygon sets), $\mathbf{S}$. Every polygon in a shell $\mathrm{S}_k \in \mathbf{S}$, share at least one edge with another polygon in the same set every edge segment belongs to at most two polygons. This shell generation process is implemented by the shell generation function described by Algorithm 2, which is based on the common edge function $F_{ce}$ described in the Algorithm 1.

Before describing the $F_{ce}$ function, the wire-frame $W_{\mathbf{A}}$ of a polygon surface set $\mathbf{A}$ is defined as the collection of all line segments $\{W_{\mathbf{A}}(1), ...\}$ of its polygon boundaries.

$F_{ce}$ acts on two wire frames $W_1$ and $W_2$ returns a Boolean variable $c = \{\text{true, false}\}$ which indicates whether the input wire frames share a common edge. The common edge is identified by checking the whether the line segments of any line segment pair $(W_1(n), W_2(m))$ (where $n \in \{1, ..., |W_1|\}$ and $m \in \{1, ..., |W_2|\}$) intersect with each other $(W_1(n) \cap W_2(m) \neq \emptyset)$. Given any line segment pair $(A_1A_2, B_1B_2)$ in three dimensional space there are four non-empty intersection cases demonstrated in Figure 3.

**Algorithm 1**
**Common edge function:** $c = F_{ce}(W_1, W_2)$

```
$c =$ false                          ▷ Initialization of output variable
for $n = 1, ..., |W_1|$ do
    for $m = 1, ..., |W_2|$ do
        if $W_1(n) \cap W_2(m) \neq \emptyset$ then      ▷ Intersection
            $c =$ true
            break
        end if
    end for
    if $c$ then
        break
    end if
end for
```

It is important to point out that this line segment intersection condition is necessary for the wire-frames and their respective polygons to share common edges in three dimensions. Also, $F_{ce}$ reports only the common edge existence without calculating the common edge segments. For this reason, if at least one common edge is detected, appropriate break conditions are inserted to terminate the common edge function without examining all other line segment pairs. This decreases the overall execution time of ASG considerably, as $F_{ce}$ is applied to multiple polygon pairs of the sets $\mathrm{R}_i$ in Algorithm 2.

**Algorithm 2**
**Shell generation function:** $\mathbf{S} = F_{sg}(\mathrm{R}_i)$

```
$\mathbf{S} \leftarrow \{\emptyset\}$                    ▷ Initialization of output set
$\mathrm{R} \leftarrow \bigcup_i \mathrm{R}_i$                ▷ Remaining set initialization
$\mathrm{R}_{tmp} \leftarrow \{\mathrm{R}(0)\}$
$\mathrm{R} \leftarrow \mathrm{R}/\mathrm{R}(0)$
$c = 0$                              ▷ Counter initialization
while $\mathrm{R} \neq \emptyset$ do
    $r = \mathrm{R}(c)$                      ▷ Current polygon
    if $F_{ce}(W_r, W_{\mathrm{R}_{tmp}})$ then
        $\mathrm{R}_{tmp} \leftarrow \mathrm{R}_{tmp} \cup r_c$
        $\mathrm{R} \leftarrow \mathrm{R}/\mathrm{R}(0)$
        if $c = |R|$ then
            $\mathbf{S} \leftarrow \mathbf{S} \cup \{\mathrm{R}_{tmp}\}$         ▷ Output set update
        end if
        $c = 0$
    end if
    $c = c + 1$                          ▷ Counter update
end while
```

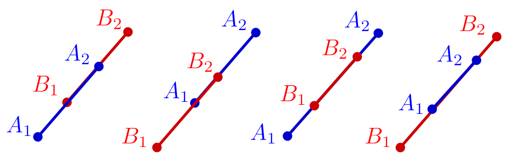

*Figure 3: Four cases of nonempty line segment pair intersection in three dimensions.*

The shell generation function's obtained shells are classified into internal and external shells using the total volume criterion. The total volume criterion is based on the fact that the volume of a shell (not necessarily closed) is negative or positive depending on the normal vector of its surfaces. If the normal vectors point outwards, the shell is external, and its total volume is positive. If the normal vectors point inwards, the shell is internal, and its volume is negative. The total volume $V_i$ of the shell $\mathrm{S}_i$ is calculated using the following formula:

$$V_i = \frac{1}{3} \sum_{k=1}^{|\mathrm{S}_i|} [< \hat{n}_k, \vec{r}_k >] \, A_k \tag{4}$$

where $\hat{n}_k$ is the normal vector, $\vec{r}_k$ is the centroid vector and $A_k$ is the area of $k_{th}$ surface of B-rep i. The notation $< \vec{a}, \vec{b} >$ is used for inner product of vectors $\vec{a}$ and $\vec{b}$. For the volume criterion to be correct all polygon subtracting operations in (3) must preserve the normal vectors of the B-rep polygons b.

### Stage 4: IFC enrichment

During the final stage of ASG, IfcSpace data classes are populated in the input IFC file with geometric representation described by the IfcFacetedBrep class and the points of the inner shells extracted by the previous stage. A custom building dedicated We used BIM library to perform the deserialization/serialization operations and bring the input IFC classes to memory, populate them and export the new enriched IFC file. If levelling information is present in the input IFC file in the IfcStorey classes, the geometry of the space cells is translated correctly to the local coordinate system of the respective floor. Otherwise, all space volumes are translated to the building's local coordinate system.

The overall algorithm is summarized in parts III and IV of Figure 4, for an imaginary cubic room example. In part III, the common boundaries among pairs of walls and slabs are identified and indicated with blue colour. These common boundaries are subtracted from their respective B-reps, leaving a remaining set of surfaces split into an inner and outer shell, indicated, with solid green and transparent green colours, in part IV.

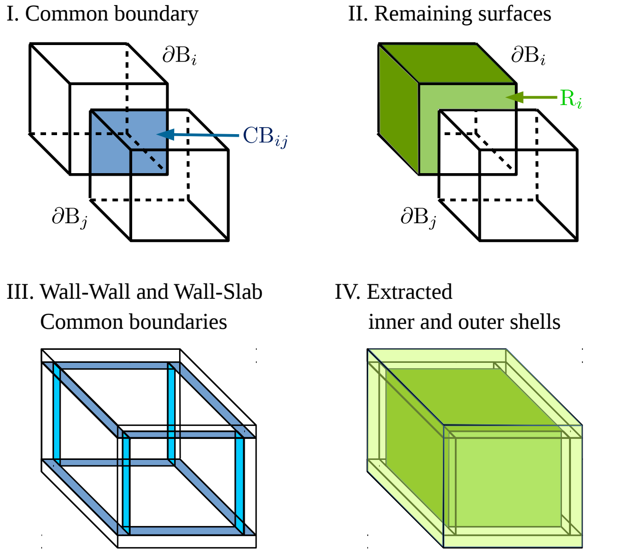

*Figure 4: Illustration of Common boundary and Remaining surfaces of B-rep i and application examples on building elements (wall, slabs) leading to inner (solid) and outer (transparent) shell extraction*

## Performance measure

Essentially ASG generates closed space shells. The quality of its output depends on whether the produced shells are closed or not. One performance criterion of ASG is the total length of isolated line segments (segments that belong to one boundary surface of the shell). If the shell is completely closed, all line segments of its boundary surfaces belong to two polygonal surfaces; there are no isolated line segments, and their total length is zero. In any other case, the bigger the total length of isolated line segments, the worse the space shell generation is.

Given any set of polygonal surfaces ($S_i$) forming a closed or open-shell, the set isolated segments can be extracted using the isolated segment extraction function ($F_{ise}$). $F_{ise}$ is applied on the wireframes of polygonal surfaces, which are the sets of line segments forming the boundary of the polygonal surfaces. In short, $F_{ise}$ checks the wire-frame of every surface in the polygonal set with the wire-frame of the rest of the set's surfaces and collects the line segments of the first wireframe that do not intersect with the second wireframe. These line segments define the shell areas, where either surfaces are missing creating holes and render the shell open (see example in part I of Figure 5, or surfaces intersect with each other (see example in part II of Figure 5). Essentially, the length of these line segments on a reconstructed building space volume shell measure how good the reconstruction is since they define areas of missing or intersecting boundary surfaces. This length defines the Length of Isolated Segments (LIS) metric.

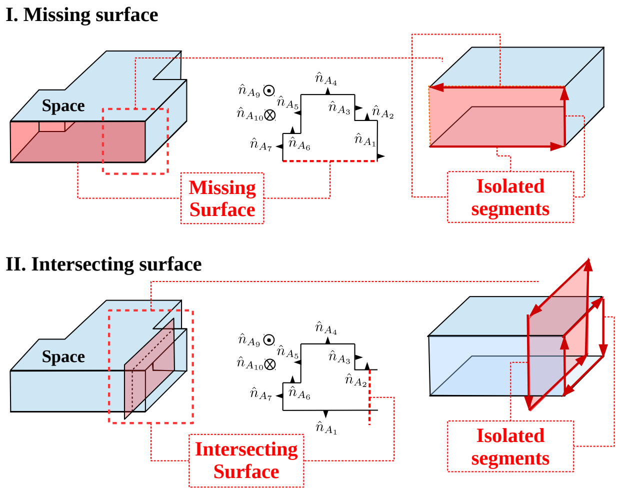

*Figure 5: Incomplete space generation and isolated segments*

## Limitations

The results of ASG are affected by geometric errors in the IFC data. For example, clashes or intersections among architecture elements cause the omission of common boundaries among these elements. Consequently, if these common boundaries are not considered in the remaining surface set extraction, the space shell generation is inaccurate. Furthermore, if the orientation of the B-rep surfaces of architectural elements is inverted, the adjacent to these elements building space volumes are not reconstructed correctly, as their related generated shells cannot be classified as inner shells (the volume determined by (4) becomes positive). Finally, if the B-reps of architectural elements have missing surfaces,

https://doi.org/10.26868/25222708.2021.30888

the algorithm cannot accurately reconstruct adjacent building space volumes because the related generated shells are open.

## Advantages over existing practices

Even with the fastest and most accurate designers available, the space volume topology generated by ASG is far superior in production time and accuracy compared to design approaches manually. Additionally, in case the building space cavities are formed by architectural elements which are represented geometrically by irregular solid shapes (which include, for example, surfaces generated by curved segments: circular, ellipsoidal, b-spline arcs), their volume is extremely tedious to be designed, even with the best BIM design software platforms. ASG algorithm can facilitate segmentation algorithms (Ying and Lee, 2019) that convert curved geometries to segmented approximations by removing their requirement to adjust the curved space volume approximations to fill in curved building cavities.

## Software dependencies

The software implementation of ASG relies on two existing frameworks: a novel cross-platform IFC library implemented in Java, which serializes and deserializes IFC files in various formats (IFC2x3, IFC4, ...) and generates appropriate Java classes and a C++ geometric library. This IFC library is used to query the building's architectural elements' geometric representations from the IFC file. The same library can also enrich the IFC file with geometric representations of the space volumes generated by the ASG algorithm. The geometric library depends on the fast and robust open-source clipper software developed by Angus Jonhson (Johnson, 2016). Clipper is based on Vatti's algorithm (Vatti, 1992). In ASG, the clipper performs basic two-dimensional polygon operations (union, intersection, difference).

## Application examples

ASG is applied on two demonstration BIMs referring to two real buildings. The first building is a student hall in the Technical University of Crete in Greece, displayed in part I of Figure 3. It was completed in 2020 and is located in the north-west part of the university campus. The building has three floors with six spaces each and is a part of a larger complex which has the same building repeated 11 times. We chose the building because of its simplicity and the fact that it had a very high quality BIM model. Consequently, ASG was applied on this building and 48 inner space shells were reconstructed successfully, illustrated in part II of Figure 7. Since the architectural content of the input IFC file for this building was geometric error free the LIS metric for the 48 reconstructed inner space shells attained close to zero values (less than $2e-4$) apart from one space shell due to a wall slab misalignment. Also some unoccupiable inner space shells were also reconstructed referring to the small volumes of the pinnacle of the two building shafts. These space volumes are highlighted with a dashed rectangle in Part II of Figure 6. This small space volume reconstruction highlights the need for post processing of the enriched out IFC file, by editing each space volume individually (deleting, labeling, ...). These post-processing operations will be reported in a future paper.

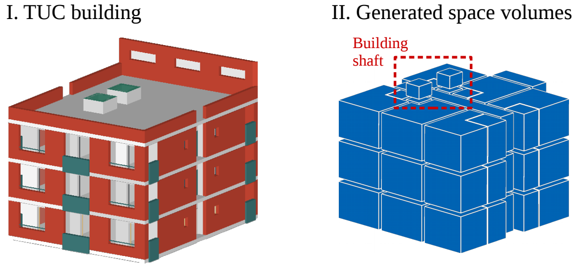

*Figure 6: Screenshot of architectural content of IFC file of TUC building (I) and its space volume generation (II)*

The second demonstration building is the Torre Turina building, in Valladolid, Spain, pictured in part I of Figure 7. It was selected because it is a large 11-story building BIM that contains many design clash errors (wall-column clashes). It was selected to highlight the importance of inner surface inclusion of ASG (equation (2)), in the avoidance of the harmful effects of these errors to the overall space volume reconstruction.

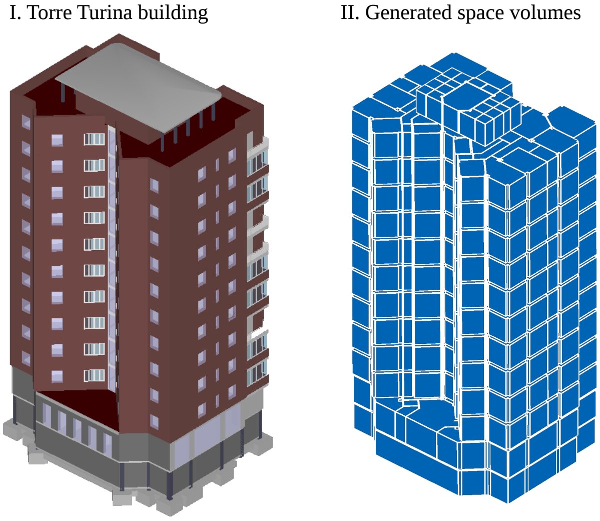

*Figure 7: Screenshot of architectural content of IFC file of Torre Turina building (I) and its space volume generation (II)*

The space volume reconstruction for this building is pictured in part II of Figure 7. Although the reconstruction seems to be accurate for most of the

https://doi.org/10.26868/25222708.2021.30888

internal space volumes, geometric errors led to imperfections in the space volume reconstruction process from a general point of view. Clashes between architectural elements generate intersections in reconstructed space surfaces of adjacent building space volumes because the clash's internal surfaces are not treated as common boundaries but as space volume surfaces, which is not correct (part I in Figure 8). If the clashes' internal surfaces, defined in (2), are included in the ASG, the reconstruction of adjacent building space volumes is correct (part II in Figure 8). On the other hand, surface errors render some shells external instead of internal. Figure 8 displays the wall-column clashes repeated in all floors of the building, which generate intersections in the reconstructed spaces' surfaces, resulting in merging two adjacent space volumes into one.

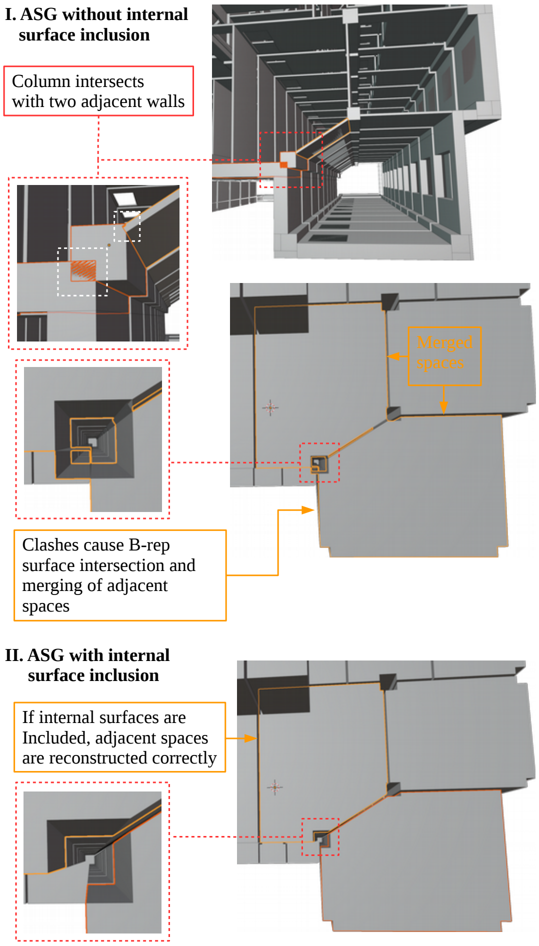
I. ASG without internal surface inclusion
Column intersects with two adjacent walls
Merged spaces
Clashes cause B-rep surface intersection and merging of adjacent spaces
II. ASG with internal surface inclusion
If internal surfaces are Included, adjacent spaces are reconstructed correctly

*Figure 8: Examples of wall-column clashes affecting ASG in Torre Turina building (I) and correct space reconstruction after internal surface inclusion (II).*

## Conclusions

An Automatic Space Generation (ASG) algorithm for enriching IFC data files with the geometric representations of building space volumes was introduced. Provided that the geometric representations of the surrounding to the space volumes building elements, are error-free (i.e. do not intersect and the surfaces of their B-reps are not inverted), ASG successfully reconstructs the building facade elements as well as all internal building spaces as outer and inner closed shells, respectively. ASG is particularly useful in cases where the manual design of building spaces for building energy performance simulation purposes is tedious, time-consuming, and even impossible in complex geometric representations of surrounding building constructions (curved geometries, parametric descriptions). The algorithm's performance can be assessed by the length of isolated segments (segments belonging to only one boundary surface) in each building space volume reconstruction.

ASG was successfully tested using IFC models of two real buildings: one model containing geometric errors, and another which was error-free. In the future, ASG will be included in a general framework containing many software tools used to offer openBIM (IFC) data quality checking and transformation web services. To improve the introduced algorithm's performance, the parts of ASG, which can run in a parallel fashion, will be implemented using the p_thread C library to enable parallel threads in a multi-core processing environment.

## Acknowledgments

The research leading to these results has been partially funded by the European Commission H2020 project "BIM-based holistic tools for Energy-driven Renovation of existing Residences" under contract #820621 (BIMERR).

## References

Andriamamonjy, A., D. Saelens, and R. Klein (2018). An automated IFC-based workflow for building energy performance simulation with Modelica. *Automation in Construction 91*, 166–181.

Bazjanac, V. (2010). Space boundary requirements for modeling of building geometry for energy and other performance simulation. In *CIB W78: 27th International Conference.*

Cao, J., R. Wimmer, M. Thorade, T. Maile, J. O'Donnel, J. Rädler, J. Frisch, and C. van Treeck (2015). A flexible model transformation to link BIM with different Modelica libraries for building energy performance simulation. In *Proceedings of the 14th IBPSA Conference.*

El-Diraby, T., T. Krijnen, and M. Papagelis (2017). BIM-based collaborative design and socio-technical analytics of green buildings. *Automation in Construction 82*, 59–74.

Giannakis, G. I., K. Katsigarakis, G. N. Lilis, and S. Álvarez-Díaz (2019). Guidelines for OptEEmAL BIM Input Files. https://www.

https://doi.org/10.26868/25222708.2021.30888

opteemal-project.eu/files/guidelines_for_opteemal_bim_input_files_v11.pdf.

Johnson, A. (2016). Clipper - an open source freeware library for clipping and offsetting lines and polygons. http://www.angusj.com/delphi/clipper.php.

Ladenhauf, D., K. Battisti, R. Berndt, E. Eggeling, D. W. Fellner, M. Gratzl-Michlmair, and T. Ullrich (2016). Computational geometry in the context of building information modeling. *Energy and buildings 115*, 78–84.

Lilis, G. N., G. Giannakis, K. Katsigarakis, and D. Rovas (2018). A tool for IFC building energy performance simulation suitability checking. In *Proc. 12th European Conference on Product and Process Modelling (ECPPM)*, pp. 57–64.

Lilis, G. N., G. Giannakis, K. Katsigarakis, and D. Rovas (2019). Space boundary topology simplification for building energy performance simulation speed-up. In *Proceedings of Building Simulation 2019: 16th Conference of IBPSA*, pp. 175–181. International Building Performance Simulation Association (IBPSA).

Lilis, G. N., G. I. Giannakis, and D. V. Rovas (2017). Automatic generation of second-level space boundary topology from IFC geometry inputs. *Automation in Construction 76*, 108–124.

Pinheiro, S., R. Wimmer, J. O'Donnell, S. Muhic, V. Bazjanac, T. Maile, J. Frisch, and C. van Treeck (2018). MVD based information exchange between BIM and building energy performance simulation. *Automation in Construction 90*, 91–103.

Rose, C. M. and V. Bazjanac (2015). An algorithm to generate space boundaries for building energy simulation. *Engineering with Computers 31*(2), 271–280.

Solibry (2019). Solibri model checker, v9.9.5.113. https://www.solibri.com/solibri-office.

Vatti, B. R. (1992). A generic solution to polygon clipping. *Communications of the ACM 35*(7), 56–63.

Ying, H. and S. Lee (2019). An algorithm to facet curved walls in IFC BIM for building energy analysis. *Automation in Construction 103*, 80–103.

Ying, H. and S. Lee (2021). Generating second-level space boundaries from large-scale IFC-compliant building information models using multiple geometry representations. *Automation in Construction 126*.

https://doi.org/10.26868/25222708.2021.30888