SECOND-LEVEL SPACE BOUNDARY TOPOLOGY GENERATION FROM CITYGML INPUTS

G.N. Lilis[1], D.V. Rovas[2], I. Prieto[3]

[1]Department of Production Engineering and Management, Technical University of Crete, Chania, Greece

[2]Institute for Environmental Design and Engineering, University College London, London, UK

[3]Construction Division, Tecnalia, Spain

ABSTRACT

Although CityGML geometrical data exported either from GIS data or from compatible design software are suitable for scene rendering and navigation, they are not directly usable for energy simulation purposes because the second-level space boundary information, essentially surface pairs through which thermal energy exchange among buildings or building rooms or among a building room and its outside environment occurs, is missing. In order to address this need, a district space boundary topology generation algorithm, that takes as input data formatted according to the CityGML standard, is introduced. The algorithm is based on four main processes and special operations which are designed according to specific input data scenarios. The algorithm is demonstrated with successful results on examples with Level Of Detail 2, 3 and 4, as defined in the CityGML standard. Also certain cases requiring further investigation are discussed.

INTRODUCTION

Nowadays, 3D City models can be obtained in many forms using a variety of approaches on different data sets, such as: vector map data combined with digital elevation models and aerial images; laser scanning data combined with high resolution satellite images; and terrestrial images combined with digital surface models using close range photogrammetry and texture mapping (Singh et al., 2013). The geometric data of these models, after appropriate processing can be merged into planar patches (Dorninger and Pfeifer, 2008) and finally transformed into semantically enriched 3D data models (Prieto et al., 2012), such as the popular City Geography Markup Language (CityGML) standard defined by the Open Geospatial Consortium (Open Geospatial Consortium, 2012).

The need for reduction of the building sector's total energy consumption, highlighted the importance of accurate energy demand estimation in a district setting; which in turn, requires detailed energy models of the buildings in the district (Kaden and Kolbe, 2013). Although CityGML-based energy models have been developed in the past Nouvel et al. (2013), detailed district energy models supporting minute scale energy simulations, cannot be generated directly from CityGML geometric data. Such models require the second-level space boundary topology of the buildings in the district, which is not supported by the CityGML schema.

As in the case of a single building (Bazjanac, 2010), the second level space boundary topology of a district consists of surfaces through which thermal energy flows, either from one room to another in the same building, or from a building to another building, or from a building/room to the outside environment air/terrain. Conclusively, the second-level space boundary topology of a district provides the necessary geometric data in order to assess the total thermal energy exchange among the conditioned spaces of a district and their environment and as a result, is a prerequisite of the district's total energy demand estimation. Although second-level space boundary generation algorithms for individual buildings based on their BIM data have been developed in the past (Rose and Bazjanac, 2015),(Lilis et al., 2014), there is no such algorithm for a district based on GIS data.

Aligned to the previous discussion, the present work provides the necessary algorithmic steps to transform the geometric data in CityGML files, with Levels of Detail (LoD) 2,3 and 4, into a second level space boundary topology. This topology will enable the automatic generation of a district energy model and facilitate the use building-scale thermal simulation programs at district level.

The rest of the paper is structured as follows: initially, the necessary input and desired output are presented in the preliminaries section. Then the description of the proposed algorithmic structure, which includes two primary operations (1.1 Common boundary definition and 2.1 Boundary intersection projection) each one followed by a secondary operation (1.2 Environment surface update and 2.2 Second order projection), is explained. Certain CityCML data require special operations, which involve processing of single and multi-surface openings and the estimation of ground slabs of buildings from their footprint, as described in separate section after the description of the main algorithm. Finally, the algorithm is demonstrated on a LoD2 example referring to a real city district and on two LoD3, LoD4 examples referring to the same reference building. Certain cases which require further operations not supported by the proposed method are discussed in the conclusions as extensions of the present work.

PRELIMINARIES

Before presenting the proposed algorithmic structure, the input (CityGML data) and the desired output (second-level space boundary topology) are defined. Certain geometric entities (building openings and definitions of the building footprint) require special operations which are described in separate sections.

CityGML data

CityGML data represent building geometries in various levels of detail (LoDs) as surface sets, ranging from LoD 1, 2 and 3; where only the envelopes of the buildings are defined as closed shell geometries; to higher levels of LoDs (LoD 4), where the interior building room volumes are defined as closed shell geometries, as well. Figure 1 displays LoD1, LoD2, LoD3 and LoD4 geometric representation examples of the same building. LoD 1 is the simplest geometric representation, where buildings are modelled as rectangular boxes (example of figure 1 I). Moving to higher level of detail, LoD2 includes representations of tilted roof surfaces in the building shell, which are not contained in the LoD1 rectangular box approximations (example of figure 1 II). LoD 1 and LoD 2 do not contain external opening descriptions and external roof overhangs, which are contained in LoD3 and 4 representations (examples I,II compared to examples III and IV of figure 1). Finally, LoD4 contains geometric representations of internal rooms as closed shell objects (figure 1 IV). LoD 4 data differ from LoD 1,2 and 3 (LoD1-3) in the following sense: In LoD 4 data, the thickness of a building construction can be obtained from the distance between two polygons which belong to two different closed shells, refer to the space building construction and face each other.

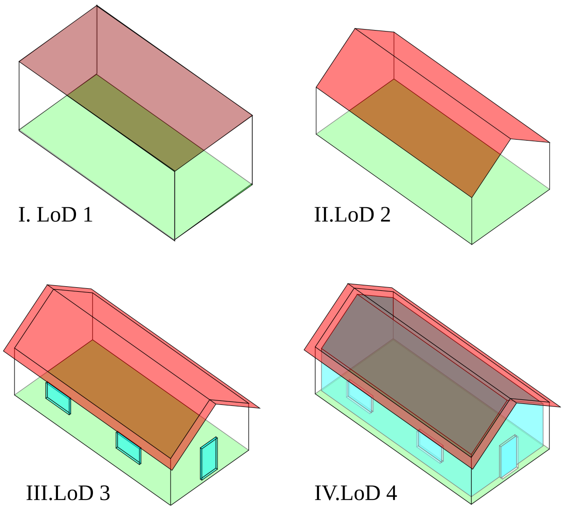


Figure 1: CityGML LoD 1-4 representations of the same building

On the contrary, in LoD 1,2 and 3 data the construction thicknesses are unknown and cannot be inferred from the outer shell geometry alone. Such differentiation alters slightly the proposed algorithmic process which is described in the following sections.

Second-level space boundary topology

The second level space boundary topology of a building group, consists of surface pairs, as demonstrated for two neighbor buildings in figure 2 B. Each surface pair is form by two second-level space boundaries of either, first or second order. First order space boundary pairs are formed by surface pairs, where either: both pair surfaces are attached to two spaces belonging to the same building, or one of the pair surfaces is attached to a building space and the other to the building's environment (solid double arrow in figure B). Second order space boundary pairs are formed by surface pairs, where the pair surfaces are attached to two spaces belonging to two different buildings (dashed double arrow in figure 2 B).

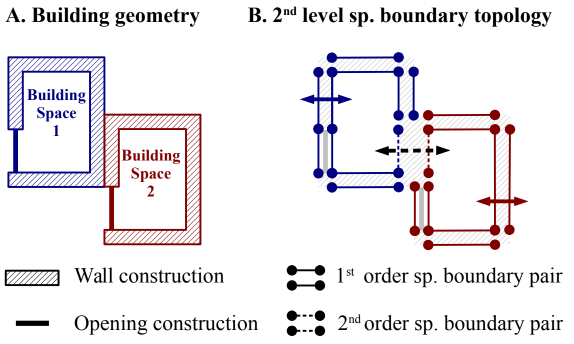


Figure 2: Building geometry (A) and its 2^{nd} level space boundary topology (B) example.

ALGORITHM STRUCTURE

Basic operations

In order to generate the second level space boundary topology of a district from its CityCML description, two basic processes must be performed. The first process is called common boundary definition because, during this process, the common boundary surfaces (CBs) are defined by the polygons contained in the CityCML file. The CBs are common polygonal surfaces shared among: building constructions and internal spaces or building constructions and their environment air or ground. The CBs obtained from the first stage are projected to each other and the projections are intersected with the original CBs, in the second stage, in order to generate the second level space boundary surface pairs of the first order. This process is called boundary intersection projection. Buildings attached to each other require two additional special operations apart from the common boundary definition and the intersection projection processes described previously. These operations must be performed after the common boundary definition process. In the first operation, called environment surface update, all the environment boundary surfaces of the buildings are updated by removing common boundary surfaces between buildings. In the second operation, called second order projection, the second order space boundary surface pairs are generated.

1.1 Common boundary definition

In this stage, CBs between building constructions (walls, roofs, ...) and building space or room volumes are defined using the CityGML polygons and classified to appropriate surface classes. For LoD1-3 cases, for every polygon of a building shell, two CBs are defined. One CB is described by a polygon with orientation towards the building shell exterior and is placed in the Construction-Environment (Cn-En) surface class (black segments in figure 3 A1). The other CB is described by the same polygon with reverse point order and orientation towards the building shell interior and is placed in the Construction-Building space (Cn-Bs) surface class (red segments in figure 3 A1). The orientation of every polygonal surface is defined by its normal vector following the right hand rule.

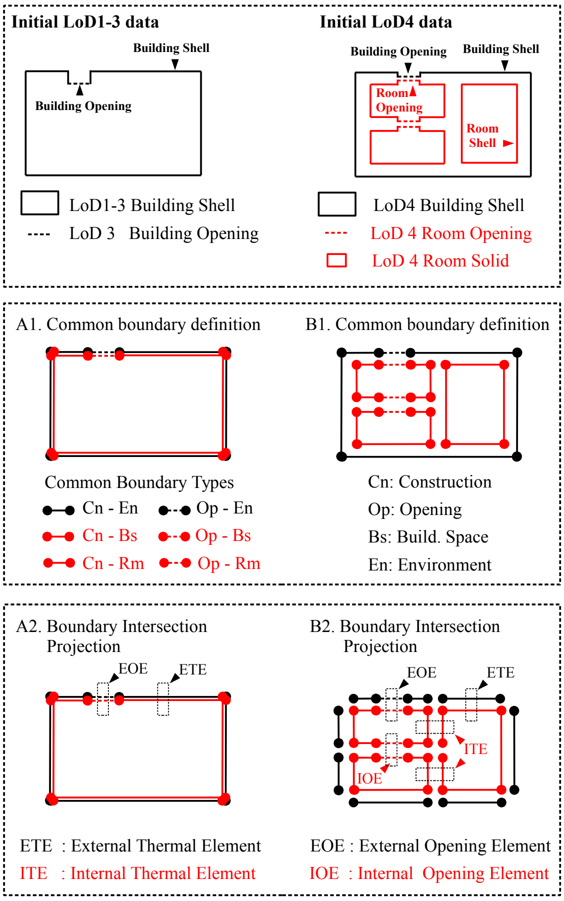


Figure 3: Illustration of the two algorithmic steps required for the conversion of an CityCML topology to its space boundary topology. The process is illustrated for LoD1-3 data in parts A1 and A2 and for LoD4 data in parts B1 and B2.

On the other hand, in LoD4 geometries, as in LoD1-3 cases, the CBs defined by building shell polygons are placed in the Cn-En class of surfaces (black segments of in figure 3 B1). The CBs defined by internal room shell polygons are placed in the Construction-Room (Cn-Rm) surface class (red segments in figure 3 B1).

Polygon surfaces referring to building openings require special handling as described in a separate section. The obtained opening polygons, after this processing, are used to generate CBs, as in the case of constructions, with the following change: in the names of the respective surface classes the term ”Construction” is replaced by the term ”Opening” i.e. Opening-Environment (Op-En), Opening-Building space (Op-Bs) and Opening-Room (Op-Rm) (dashed lines in figure 3 A1 and B1).

1.2 Environment surface update

In case neighbor buildings are attached to each other, their environment surfaces (wall-air or wall-site) are updated by removing the common boundary surfaces of their building shells, as illustrated in figure 4. To take into account small building distance inaccuracies and to identify the common boundary surfaces of the building shells in a robust manner, the following steps are performed.

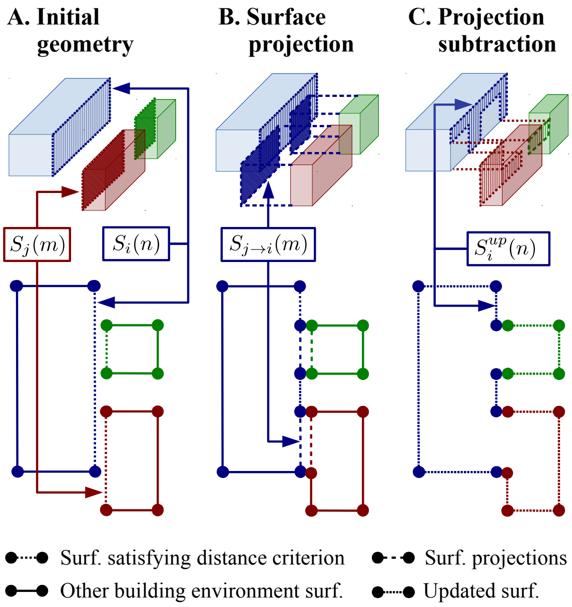


Figure 4: Environment surface update example.

Firstly, building i and its adjacent buildings with indexes j contained in the set M_i ($j \in M_i$), are considered. For every environment surface n of building i ($S_i(n)$), all proxy environment surfaces $S_j(m)$ with indexes $m \in D_j$ of building $j \neq i$, are collected. The set D_j contains the indexes m of the proxy, to $S_i(n)$, surfaces $S_j(m)$, which belong to, the adjacent to i, building j and whose point's distances from the plane of $S_i(n)$ are within some distance limits $[d_{min}, d_{max}]$. The proximity criterion of the plane of $S_j(m)$ to the plane of $S_i(n)$ can be adjusted using the min/max distance parameters d_{min} and d_{max}. These proxy surfaces $S(m)_j$ are projected to the plane of $S_i(n)$. These projections are denoted as $S_{j\to i}(m)$. Finally, the surface $S_i(n)$ is updated by a surface obtained by subtracting from $S_i(n)$ the union of the projections $S_{j\to i}(m)$, with $m \in D_j$:

$$S_i^{up}(n) \leftarrow S_i(n) - \bigcup_{\substack{\forall m \in D_j \\ \forall j \neq i}} S_{j \to i}(m) \qquad (1)$$

The subtraction and union operations in (1) are applied on polygons which belong to the same plane using polygon clipping functions (Vatti, 1992). The environment surface update process is illustrated in figure 4, where the environment surfaces of three buildings, satisfying the distance criterion, are projected to each other (figure 4 part B) and the projections are removed from the original surfaces (figure 4 part C), yielding the new updated environment surfaces.

2.1 Boundary intersection projection

The common boundary surfaces defined in the previous stage are used here, in order to generate surface pairs which are the elements of the desired second order space boundary topology. A surface pair is formed using a boundary intersection operation applied on two CBs, which are close enough and face each other (CB_1 and CB_2 in figure 5 A).

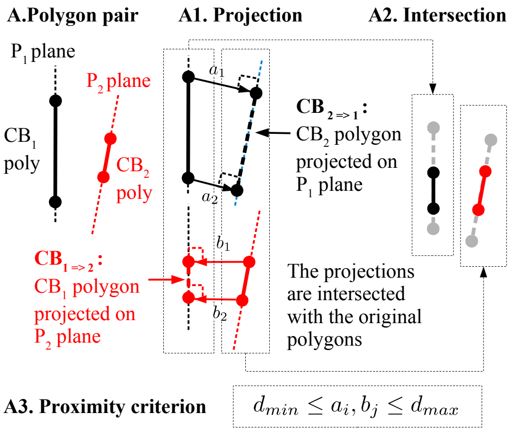


Figure 5: Boundary intersection projection example.

The proximity of the CBs is tested using the following condition : two CBs are considered to be close enough if all the distances of the points of one CB polygon from the plane P of the other CB polygon are within certain distance limits $[d_{min}, d_{max}]$, as illustrated for distances a_i, b_j in figure 5 A3. If the previous condition holds, every CB polygon of the pair is projected to the plane of the other (projection step : $CB_{1\Rightarrow2}$ and $CB_{2\Rightarrow1}$, in figure 5 A1). Then, the projections are intersected with the original polygons (intersection step: $CB_{1\Rightarrow2} \cap CB_2$ and $CB_{2\Rightarrow1} \cap CB_1$, in figure 5 A2). The final polygons after the intersection step, form a first order space boundary surface pair. The tolerances d_{min} and d_{max} are set according to the maximum construction thickness. The intersection operations are performed on coplanar polygons, using the polygon clipping functions (Vatti, 1992).

In LoD1-3 cases, the boundary intersection projection process generates a first order space boundary surface pair, which usually is the same as the original polygon pair since the CB polygons are the same (in figure 3 A2). In LoD4 cases however, the first order space boundary pair, obtained from the boundary intersection projection stage, usually differs from the original CB pair, because the CB surface polygons are different and belong to different planes (figure 3 B2). The boundary intersection projection process identifies three types of boundary surface pairs:

(a) **Thermal elements**. This type of boundary surface pairs describe parts of architectural building constructions - excluding building openings - which impede the flow of thermal energy, either between adjacent internal building spaces or between a building space and the outside environment air or ground. Two cases can be distinguished:

Internal thermal elements. These elements appear only in LoD4 cases and are obtained from projections of two construction-room CBs : Cn-Rm1 / Cn-Rm2. The internal thermal elements are denoted by the triplet Rm1-Cn-Rm2.

External thermal elements. These elements appear in LoD1-3 cases between a building space and its outside environment and in LoD4 cases between a building room and its outside environment. In LoD1-3 cases the surfaces of an external thermal element are obtained by projection of a Cn-Bs CB to a Cn-En CB and vice versa and the resulting element is denoted by the triplet Bs-Cn-En. In LoD4 cases the surfaces of an external thermal element are obtained by projection of a Cn-Rm CB to a Cn-En CB and vice versa and the resulting element is denoted by the triplet Rm-Cn-En. The CBs, used to generate a thermal element, refer to the same construction (Cn). A spacial case of an external thermal element is the site boundary, where the environment En is the building's terrain volume.

(b) **Opening elements** Similar to the thermal element type this type of boundary surface pairs is related to building opening constructions which impede, under certain conditions, the flow of thermal energy either between adjacent internal building spaces or between a building space and the outside environment air. Two cases can be distinguished, namely: *External opening elements* and *Internal opening elements*. The respective surface pairs, are obtained as in the thermal element case, by projection of the same CBs but with the Op notation instead of the Cn notation. The CB pairs, used to generate an opening element, refer to the same opening (Op).

(c) **Shading elements**. This element type, refer to constructions which impede sunlight rays. As in the other element types two cases are considered: *External shading elements* obtained from two Construction-Environment CBs (Cn-En /

Cn-En) denoted by the triplet En-Cn-En. *Internal shading elements* obtained either from projection of two Construction-Building space CBs (Cn-Bs/Cn-Bs) in LoD1-3 or from projection of two Construction-Room CBs (Cn-Rm/Cn-Rm) in LoD4 referring to the same room (Rm) denoted by the triplet Rm-Cn-Rm. The surfaces of the shading elements are not in the set of the building or room envelope shell surfaces.

2.2 Second order projection

The second order space boundaries among attached buildings illustrated in figure 2 B, cannot be identified only by steps 1 and 2 alone. These boundaries are obtained from construction-building space (Cn-Bs) CB pairs which refer to different constructions and different building spaces (Cn1-Bs1 and Cn2-Bs2 figure 6 A). If the projection condition is satisfied for a CB pair (Cn1-Bs1 / Cn2-Bs2), a second order space boundary pair is obtained by the boundary intersection projection operations on these CBs, and is denoted as Bs1-Cn1-Cn2-Bs2. This process is called second order projection and is illustrated in figure 6. Conclusively, the second order space boundary pair obtained from the above process determines the surface through witch thermal energy flows among building spaces Bs1 and Bs2 passing through their two external constructions Cn1 and Cn2.

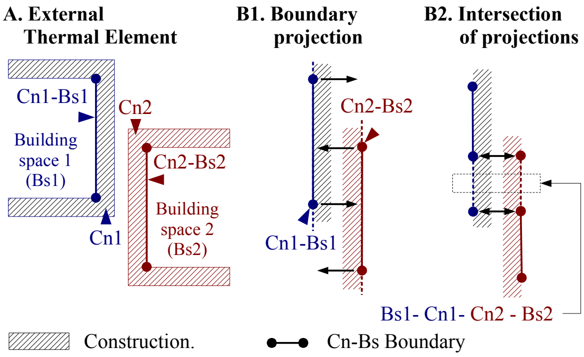


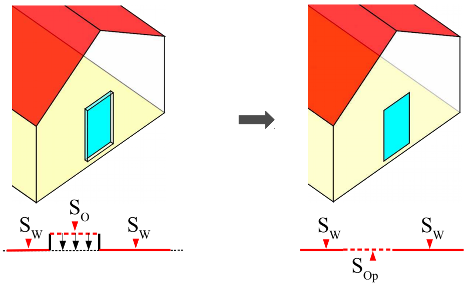
Figure 6: Illustration of the second order projection process on two Cn-Bs common boundary surfaces referring to two different constructions (Cn1,Cn2) and building spaces (Bs1,Bs2).

Special operations - Openings

In (LoD3 or higher) geometries, openings described by a single or multiple surface polygons are processed differently, in order to be transformed into surfaces suitable for simulations, as explained in the following sections.

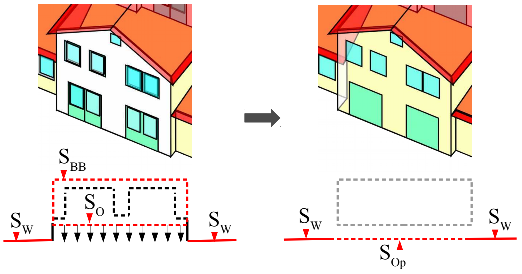
Single surface openings : If an opening in LoD3-4 cases is described by a single surface polygon, then this polygon is projected on the wall surface, which has the maximum area among all the surfaces of the wall, to which the opening belongs to. This process is illustrated in figure 7 A, where the single surface openings (S_O) are projected on the wall surface with maximum area (S_W). The projected opening surface S_{O_p} and the wall surface with the maximum area S_W are the only surfaces which are finally retained.

A. Single opening surface case



1. From the surfaces of the wall containing the opening the one with maximum area is selected (S_W), the others are omitted.
2. Opening surface (S_O) is projected on the selected wall surface.

B. Multiple opening surface case



1. The minimum volume bounding box of the opening surfaces is determined (S_{BB}).
2. From the surfaces of the wall containing the opening the one with maximum area is selected (S_W), the others are omitted.
3. The maximum area surface (S_O) of the minimum volume bounding box is projected on the selected wall surface.

Figure 7: Illustration examples of the algorithmic steps referring to the processing of opening surfaces of LoD3 buildings.

Multiple surface openings : In case an opening is described by multiple surfaces referring to different opening parts such as frames and dividers, the minimum volume bounding box of the opening is determined first (dashed S_{BB} line in figure 7 B). The surface of the bounding box with maximum area is then extracted as the surface of the opening (S_O surface in figure 7 B). This opening surface is projected, as in the single surface opening case, on the on the wall surface which has the maximum area among all the surfaces of the wall to which the opening belongs to (S_W surface in figure 7 B).

The projected opening surface (S_{O_p}) and the respective wall surface (S_W) are the only surfaces which are finally retained.

Special operations - Ground slab estimation

Some CityGML LoD1-3 data files contain the footprint of buildings, defined by the building perimeter points at terrain level, instead of the buildings' ground slab surfaces, required as boundary conditions for simulations. Consequently, in such cases the required ground attached second level space boundaries should be estimated, as indicated in the modelling guidelines of (SIG$_3$D Quality Working Group, 2014). The estimation of the ground slab of a building from its footprint is based on the footprint point with minimum z coordinate (z_{min} point). The z_{min} point, determines the minimum z plane (z_{min} plane) which is a plane parallel to the xy plane at $z = z_{min}$. The ground slab polygon points are obtained from the projections of the footprint points on the minimum z plane. This process is illustrated with green color for a LoD2 building in figure 8.

After the ground slab generation, the boundary building walls are extended downwards to reach the ground slab defining new wall site boundaries. These boundaries surfaces are indicated with blue color in the example of figure 8. Since the new wall-site boundaries are building boundary surfaces, they are updated using the environment surface update process mentioned previously.

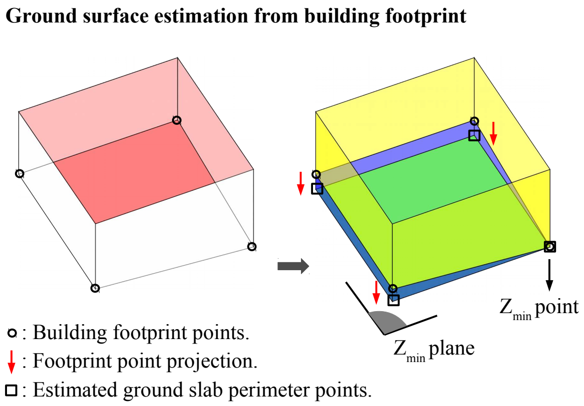


Figure 8: LoD2 example of ground slab estimation from building footprint.

Additionally this wall-environment downwards extension results to a downward extension in the internal wall-building space CB as well. The new wall-site CBs and the extended wall-building space CBs are then used as input to the intersection projection operations in order to generate the second level boundary topology.

EXAMPLES

LoD2

The proposed second level topology generation process is demonstrated on a CityGML LoD2 model referring to a historic district located in Santiago de Compostela city in Spain. The aerial view of the district is presented in figure 9 A.

The district consists of 79 building spaces contained in 65 buildings of various types, ranging from single and multifamily houses to apartment blocks and non-domestic buildings that cover an area of 16 thousand square meters. The LoD2 geometric data consisted of external building wall and roof polygons, presented in figure 9 B, with white and red colors respectively.

During the common boundary definition stage, wall-environment (Cn-En) boundaries were detected. The building ground slabs were estimated based on the building's footprint. The estimated ground slabs generated two new common coinciding boundary surfaces: a Cn-Bs surface facing inside the building and a Cn-En surface facing towards the building's site (green color surfaces in figure 10 A). As the wall-environment surfaces are extended downwards to reach the ground slab polygons new wall-site (Cn-En) boundaries are generated, displayed with blue color in figure 10 A.

Both wall-environment and wall-site boundaries (Cn-En) facing outside the building towards the environment, generated new respective Cn-Bs common boundaries with the same polygon points in reversed order, facing inside the building. Additionally the wall-environment and wall-site boundaries boundaries were updated by removing the common surfaces shared among neighbor buildings using the environment surface update algorithm.

During the intersection projection stage the Cn-Bs common boundaries were projected and intersected with the Cn-En boundaries (either wall-environment or wall-site), generating external thermal elements (Bs-Cn-En) defined as first order space boundary surface pairs (yellow and blue and green colored surfaces in 10 A).

Finally, second order space boundaries were generated by projecting and intersecting wall-building space boundaries (Cn-Bs) belonging to different buildings. The second order space boundaries are indicated with black color in figure 10 B.

LoD3

The proposed algorithm is demonstrated on a LoD3 building described by a single closed shell displayed in figure 11 A. The building has three openings (two windows and one door) recessed in the walls of the building indicated with gray color in figure 11 A.

Three kinds of elements were generated which included: seven external thermal elements of Bs-Cn-En type (two roof boundary surface pairs displayed with red color in figure 11 B, one ground slab boundary surface pair indicated with yellow color in figure 11 B and four external wall boundary surface pairs indicated with white color in figure 11 C); three external opening elements of type Bs-Op-En (opening boundary surface pairs displayed with cyan color in figure 11 C) and two external shading elements of type En-Cn-En (roof boundary surface pairs indicated with black color in figure 11 D).

Santiago de Compostela district

A. Aerial view.

B. CityGML LoD 2 data.

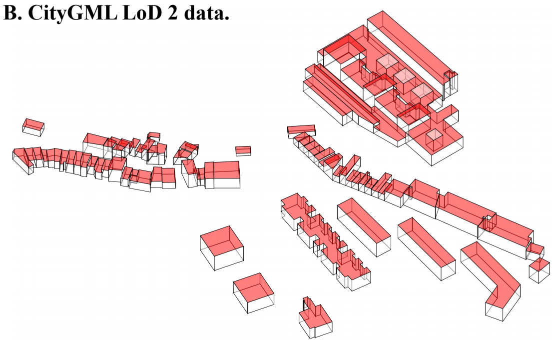
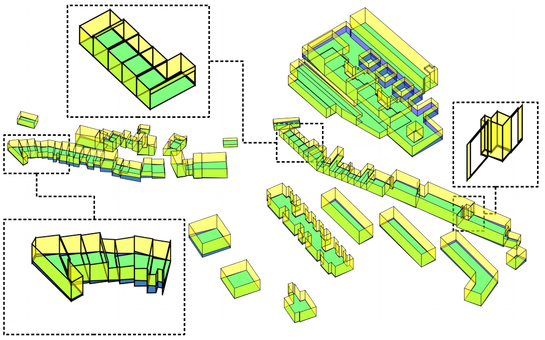


Figure 9: Aerial view and CityGML LoD2 representation of Santiago de Compostela district.

Santiago de Compostela district results

A. Updated environment surfaces (air-yellow / ground-green)



B. Second order space boundaries (gray)

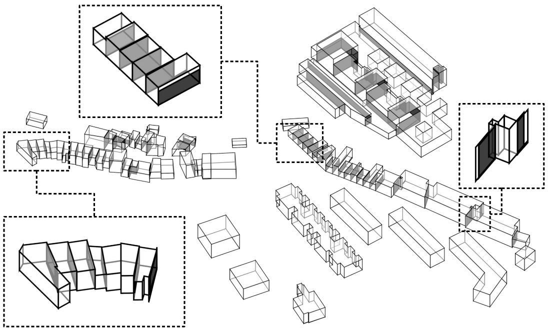
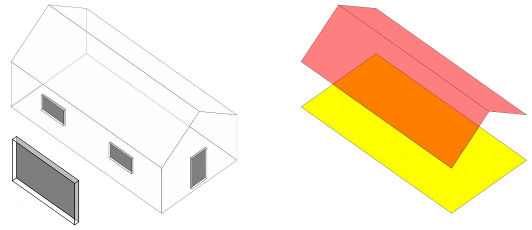


Figure 10: CityCML LoD2 processing results of Santiago de Compostela district.

In order to obtain the external opening elements the respective surfaces were processed using the operations for the single surface openings.

A. Initial building shell. **B. Roof and slab boundary surface pairs.**



C. Wall and opening boundary surface pairs. **D. External shading surface pairs.**

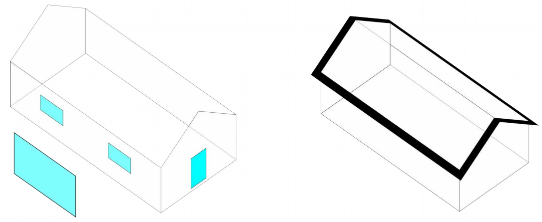
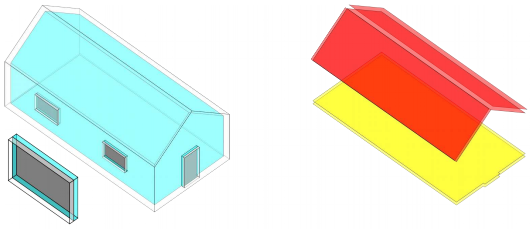


Figure 11: Space boundary results of CityCML LoD3 example.

A. Initial building and room shell. **B. Roof and slab boundary surface pairs.**



C. Wall and opening boundary surface pairs. **D. External shading surface pairs.**

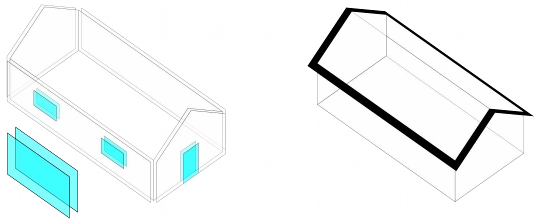


Figure 12: Space boundary results of CityCML LoD4 example.

LoD4

As in the previous LoD3 case, the algorithm is tested on a LoD4 version of the previous LoD3 building. The building is defined by two closed shells, one contained inside the other: an outer building shell (indicated with white color in figure 12 A) and an inner room shell (indicated with cyan color in figure 12 A). Both shells have as common surfaces, three opening surfaces, which refer to two windows and one door highlighted with gray color in figure 12 A.

As in the LoD3 case, the algorithm generated three types of elements: seven external thermal elements of type Rm-Cn-En, three opening elements of type Rm-Op-En and two external shading elements of type En-Cn-En indicated in parts B,C and D of figure 12, respectively.

Theses elements are described geometrically by surface pairs in which the outer surface (building shell surface) is the same as in the LoD3 case. The inner surface (room shell surface) is slightly offset with respect to the outer surface, due to the thickness of the element's construction.

As in the LoD3 case the opening surfaces of the building and room shells, were processed using the operations for the single surface openings, in order to generate the surface pairs of the external opening elements (indicated with cyan color in figure 12 C).

CONCLUSIONS

The required operations for generating second level space boundary topology of a district from its CityGML geometric data of 2,3 and 4 LoD, were presented. These include: the *common boundary definition*, where the building and room CityGML surfaces are extracted and characterized; the *environment surface update*, where the environment boundaries of buildings are updated and the *boundary intersection projection* / *second order projection* which are used to identify the first / second order space boundaries, respectively.

Special operations are required for LoD3 and higher cases, where openings in wall recesses defined by single or multiple surfaces. Further operations are also needed in cases where building ground slabs are missing and should be estimated from the building footprint.

The proposed process was applied on a LoD2 case referring to a city district, and on two LoD3 and LoD4 representations of the same demonstration building. In all cases, the respective first and second order space boundaries as well as the shading elements of all buildings were identified correctly.

Surfaces which do not belong to the building or room envelopes are used to generate the shading elements (internal or external surface pairs), which play an indirect role in a energy simulation as they are involved in solar shading calculations. If the set of envelope surfaces is not defined in a CityGML file, the shading surfaces cannot be extracted from the CityGML data. In such cases shading surfaces should be identified using additional geometric operations which is a topic of further research.

Additionally, the proposed methods, can be applied in a straightforward manner, to the geometric models of Cellular Automata (CA), which have been used in order to simulate urban growth (Batty et al., 1999). In these models the suggested algorithms can be used in order to identify the common boundary surfaces among neighbor cells, where the inter-cell interactions are taking place.

Finally, in the worst case scenario, the computation time of the overall process grows quadratically with the total number of building surfaces, since during the environment surface update stage, every external surface must be compared against all others. This time can be reduced significantly, if building proximity criteria are introduced.

ACKNOWLEDGMENTS

This work is supported by an E.U. project: Innovative design tools for refurbishing of buildings at district level H2020-EeB-05-2015 #680676 (OptEEmAL).

REFERENCES

Batty, M., Xie, Y., and Sun, Z. 1999. Modeling urban dynamics through GIS-based cellular automata. *Computers, environment and urban systems*, 23(3):205–233.

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

Dorninger, P. and Pfeifer, N. 2008. A comprehensive automated 3D approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds. *Sensors*, 8(11):7323–7343.

Kaden, R. and Kolbe, T. 2013. City-wide total energy demand estimation of buildings using semantic 3D city models and statistical data. *Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2-II/W1:163–171.

Lilis, G. N., Giannakis, G., Kontes, G., and Rovas, D. 2014. Semi-automatic thermal simulation model generation from IFC data. In *proc. of European Conference on Product and Process Modelling*, pages 503–510, Vienna, Austria.

Nouvel, R., Schulte, C., Eicker, U., Pietruschka, D., and Coors, V. 2013. CityGML-based 3D city model for energy diagnostics and urban energy policy support. pages 218–225.

Open Geospatial Consortium 2012. OGC City Geography Markup Language (CITYGML) Encoding Standard. <https://portal.opengeospatial.org/>.

Prieto, I., Izkara, J. L., and Delgado, F. 2012. From point cloud to web 3D through CityGML. In *proc. of 18th Virtual systems and multimedia (VSMM)*, pages 405–412.

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

SIG$_3$D Quality Working Group 2014. Modeling guide for 3D Objects Part 2: Modeling of Buildings (LoD1, LoD2 and LoD3). <http://en.wiki.quality.sig3d.org/>.

Singh, S. P., Jain, K., and Mandla, V. R. 2013. Virtual 3D city modeling: techniques and applications. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(2):73–91.

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