An Automatic Comparison Approach to Detect Errors on 3D City Models

Benjamin Gorszczyk, Guillaume Damiand, Sylvie Servigne, Abdoulaye Abou Diakite, Gilles Gesquière

An Automatic Comparison Approach to Detect Errors on 3D City Models

B. Gorszczyk[1], G. Damiand[1], S. Servigne[1], A. Diakité[2] and G. Gesquière[1]

[1]Univ Lyon, CNRS, LIRIS, UMR5205, F-69622, Lyon, France
[2]3D GeoInformation, Delft University of Technology, Jaffalaan 9, 2628 BX Delft, The Netherlands

Abstract

3D building models are needed in several professional domains. To provide better results, these models must be errors-free and that is why it is required to have a way to detect and to correct errors. These errors can be geometric, topological or semantic. By using a topological structure called EBM-LCC that allows to model buildings, we create a new tool that allows to detect these three type of errors in 3D city models. The solution we propose is an algorithm that compares two EBM-LCC. This algorithm can be used to compare two different models, for example acquired with two different processes, or resulting from two different acquisition campaigns. It is also an interesting tool to compare and validate algorithms. In this work, we compare an EBM-LCC loaded directly from a CityGML model with an EBM-LCC reconstructed from a soup of polygons only. Then we can use the result of this comparison to outline possible differences or to correct one of the two models by using the information of the other one. This algorithm allowed to automatically detect and correct semantic errors on several models that are currently used by professionals. This shows the interest of EBM-LCC for the city modeling domain as it helps to reach an error-free model.

1 Introduction

Those last years, a lot of improvement has been made in the 3D data acquisition domain. These new technologies applied to city modeling made it possible to obtain 3D data that model up to an entire city. That includes buildings but also vegetation, roads and ground surfaces. The data that comes from acquisition process must go through a lot of treatments to be usable by professionals such as architects or land surveyors. Besides, professionals often need more information than the acquisition can provide (e.g. semantic). For this reason, it is needed to enrich models with semantic information.

Progress in the field of large-scale data acquisition as well as cost reductions have allowed many cities to have their "digital double". This is the case of the city of Lyon (France), which has at its disposal more than $500Km^2$ of data [Gra]. These 3D models are used by decision-makers in many fields ranging for example from urban planning to the simulation of physical phenomena (noise propagation, flood simulation, etc.) [BSL*15]. Innovative new tools can also be designed with these data, but for most of the applications, provided data in open repositories must be checked before.

Each piece of information contained in such a model must be certified to ensure the quality of the modeling. That also implies that the model must be error-free. Therefore, the aim of this project is to enhance 3D city models quality by detecting and correcting errors within the data. These errors can affect each piece of information contained in the model and can be geometric as well as topological or related to the semantic [OGC16, SK07]. One way to detect those errors is to compare two datasets that described the same scene, but which are obtained by two different processes. With such datasets, it is possible to extract information that can be used to build a unique model or to correct one of the two datasets.

Our approach is based on 3D combinatorial maps [DL14]. To be more specific, we use EBM-LCC (Enriched Building Model - Linear Cell Complex) which is a model introduced by Diakité et al. [DDG14] based on these 3D combinatorial maps. This model is build thanks to existing 3D dataset in order to describe its geometry, its topology and its semantic, while allowing efficient computation and modification operators. In this previous work, the authors used mainly the direction of normals to enrich semantics of the initial geometric model.

Based on this previous work, the main contribution in this paper is the proposal of a new EBM-LCC comparison tool allowing to compare geometry, topology and semantic of two different EBM-LCC. To illustrate the interest and the new possibilities offered by this tool, we use it in order to detect and correct errors in 3D city models.

In the following, we start in Sect. 2 by presenting preliminary work about 3D city object comparison. Then we introduce 3D combinatorial maps and the additional layer of EBM-LCC. In Sect. 3 we introduce our EBM-LCC comparison algorithm which is the main contribution of this work. This tool is generic and can be used to compare two models for different applications: for example to compare the results of two consecutive acquisition campaign, or to validate the result of an algorithm. As illustration, this tool is used in Sect. 4 in order to automatically detect and correct semantic errors in CityGML files. Lastly, Sect. 5 concludes and gives some future work.

2 State of the Art

Several works have been done around 3D model quality by the international community. We may mention the Open Geospatial Consortium quality experiment [OGC16]. In this context, many work have proposed some correction methods of 3D models, see for example [ZSL14, WAW*15], and some work have proposed to detect changes between two 3D models [PMG15].

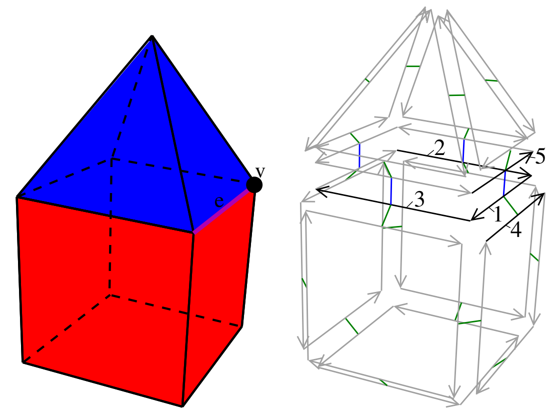
In all this work, there is a need to describe 3D models in term of topological subdivision. For example, a LoD2 (Level of Details 2) building is described by its faces (walls, doors, windows, roofs...), each face is described by its edges and each edge by its two vertices. Moreover, validity rules must be satisfied (for example a window must be rounded by walls).

To solve these needs, several works have shown that the use of a topological data-structure is a good solution [Bau75, DL89, dFMMP02]. In such a type of data-structure, 3D objects are described by their subdvision in *cells*: vertices, edges, faces and volumes. An i-cell is a cell in dimension i: vertices are 0-cells, edges 1-cells, faces 2-cells and volumes 3-cells.

The main interest of such topological data-structure is to have a precise mathematical definition, describing the *incidence* and *adjacency* relations between the cells. Two cells are incident if one belongs to the boundary of the second one; and two cells c_1 and c_2 are adjacent if they have the same dimension i and if it exists an $(i-1)$-cell incident to c_1 and c_2. Cells and incidence and adjacency relations are important because they are the key points in operations in order to iterate through parts of 3D models and to modify these objects.

Many topological data-structures exist in 2D but only few exist in 3D. Among the proposed solutions, we use 3D combinatorial maps, called 3-maps, which have many advantages [DL14]. It exists many efficient operations allowing to build, consult and modify 3-maps. Moreover, a free C++ library exists [Dam11] that proposes an efficient implementation of 3-maps and several operations.

The main principle of 3-maps is to describe 3D models by their boundary thanks to *darts*. A dart is a part of an oriented edge, which is linked with its neighbors darts thanks to 4 β links. An example of 3-map is given in Fig. 1(b). In this figure, darts are drawn by oriented segments. Each dart d describes a part of one vertex, one edge, one face and one volume of the represented 3D object. In our example, dart 1



(a) (b)

Figure 1: (a) A 3D object made of two adjacent volumes (a cube and a pyramid). (b) The 3-map describing this 3D object has 40 darts, 5 of them being numbered.

represents a part of vertex v, of edge e, of the face between the cube and the pyramid and a part of the cube itself. $\beta_0(d)$ gives the previous dart in the same face and the same volume than d, $\beta_1(d)$ gives the next dart in the same face and the same volume than d, $\beta_2(d)$ gives the other dart in the same edge and the same volume than d but not in the same face, and $\beta_3(d)$ gives the other dart in the same edge and the same face than d but not in the same volume. In the example given in Fig. 1(b), $\beta_0(1) = 2$, $\beta_1(1) = 3$, $\beta_2(1) = 4$ and $\beta_1(1) = 5$ (see [DL14] for more details and precise definitions). In figures, two darts linked by β_0 or β_1 are drawn consecutively, two darts linked by β_2 are linked by a small green segment and two darts linked by β_3 are linked by a small blue segment.

These 3-maps have shown their interest and their efficiency in building modeling, correction and enrichment in several previous work [HDDM15]. In [DDV14] the authors introduced an approach based on combinatorial maps to recover topological information from raw geometry of 3D building models. The approach is further extended in [DDG14] to allow automatic semantic enrichment of 3D building and city models on the basis of heuristic rules supported by the topology.

The resulting models from the latter approaches are the so-called *EBM-LCC*. Thanks to the combinatorial maps properties on which they rely, they simultaneously gather geometric, topological and semantic information. The combination of those three information basically makes it possible to perform advanced operations on the 3D models ranging from modification to correction and analysis. In this work, we use EBM-LCC in order to define our new method of 3D model comparison and error correction.

3 EBM-LCC Comparison

The main contribution of this work is the definition of a new algorithm allowing to compare two given EBM-LCC. This algorithm is generic and it allows to compare at the same time the geometry, the topology and the semantic of the two EBM-LCC. Moreover, at the end of our process, a list of similar and different parts of the two 3D objects is built. This list could then be useful in many further processes, as illustrated in the following section in order to detect and correct semantic errors in CityGML files.

Our method is based on a previous algorithm that determines if two connected components of two combinatorial maps are isomorphic or not [DSd*11]. Thus we start by recalling this algorithm before presenting our new comparison method.

3.1 Isomorphism of Combinatorial Maps

The isomorphism algorithm defined in [DSd*11] takes two 3-maps M and M' as input, and returns true or false depending if M and M' are isomorphic or not (two 3-maps are isomorphic if they have exactly the same topology, i.e. the same cells and the same incidence and adjacency relations).

Its main principle consists in starting from two darts $d \in M$ and $d' \in M'$, and in traversing simultaneously the two 3-maps by using the same β links. During the traversal, a mapping is built between the darts of M and M'. Then is it enough to test during the traversal if the neighbors darts of the two current darts are compatible (i.e. they are in relation by the mapping) in order to test the isomorphism.

This traversal is done by considering successively each dart of M as starting point. M and M' are isomorphic if one run succeed to traverse all darts of the two 3-map, and they are non-isomorphic otherwise.

This algorithm has the main advantage to be very simple, and to have a good complexity which is quadratic[1] in number of darts of M. However, several limits prevent us from using directly this algorithm. First, the algorithm returns true or false, which is not what we need. Our comparison tool aims to locate differences and to know what type of difference has been spotted. Second, this algorithm works only if one of the two 3-map is connected, which is not the case of our application: a 3-map describing a city can have several connected components.

We show in the next two sections how the original algorithm is modified in order to solve these two limitations.

[1]The quadratic complexity is the worst case of the isomorphism algorithm based only on topological information. This complexity can be improved by using some additional geometrical information.

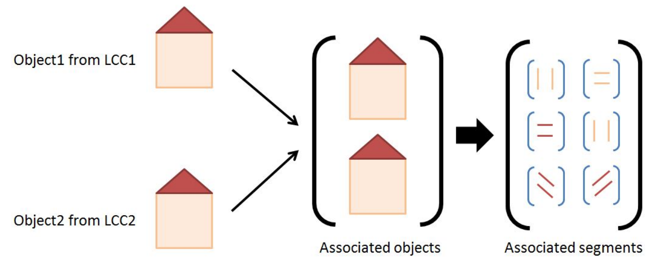


Figure 2: Illustration how objects and segments are associated between two EBM-LCC.

3.2 Extension for Non Connected 3-maps

If both 3-maps M and M' are non connected, it is mandatory before starting the comparison process to associate each connected component of M with its "corresponding" connected component in M'. Moreover, since our goal is to compare the whole 3D models, we need to associate each dart of M with a dart in M' in order to be able to restart a new traversal when the previous run failed due to a difference.

Therefore, the first step of the comparison process is objects association. Our goal is to associate each dart of the first 3-map with a possible corresponding dart in the second one. The difficulty consists thus in defining what is the "corresponding" relation. Using geometric information for association is mandatory as it is the only way to have an association between two 3-maps that are similar but non isomorphic. Moreover, if association were made using topology, there will be too much possible corresponding darts since there are many configuration which are locally isomorphic in two EBM-LCC (for example each wall represented by a rectangular cuboid in the first 3-map can be associated with all the similar walls in the second 3-map which is not discriminant).

Nevertheless, there are a lot of ways to use the geometric information for the association purpose. We propose to build this association in two steps: connected components of the two 3-maps are associated first, then pair of darts in each compatible connected components are associated.

We consider two connected components compatible if their two axis-aligned bounding boxes share at least one face, or if they share a certain amount of points (the threshold is a parameter of our method). The first case implies that the two connected components start from the same part of the 3D space. The second case allows to consider as similar two connected components having a certain number of differences (for example in order to match a building with the same building modified by adding a new room or a new level). In both cases, our method requires that the two EBM-LCC have two close enough geometry (same scale, close bounding boxes, etc...).

This first step creates a list of pairs of compatible connected components. The second step consists in associating segments with the same geometry within two associ-

Figure 3: Example of association between darts: darts with the same color are associated. Two black darts of LCC2 are associated with only one dart of LCC1. Dotted darts are not associated because 3D points on their origins are different.

ated connected components. This process produces a list of associated darts that have the same geometric meaning on both EBM-LCC. The association structure resulting from this process is shown in Fig. 2.

Segment association is actually "sets of darts that describe the same segment" association. In most cases those sets would have only 2 darts associated. These darts must have the same geometry (same points on their origins) and belong to the same face (same face normals). But it can happen that one segment is described by one dart in the first 3-map and by several darts in the second 3-map (or reciprocally). In that case, the first dart will be associated with two darts in the second 3-map (see example in Fig. 3).

Three observations must be done. First, we need georeferenced models in order to be able to compare the geographic coordinates of the two model. Second, the geometric comparisons could be done here by using some tolerances (epsilon geometry) in order to be robust to some small changes. Third, as we will see in the next section, the choice of the initial first matching is not so important thanks to the full list of association between darts. Indeed if the first match failed, the second pair is considered, then the third one...

3.3 Modification to Compare Geometry, Topology and Semantic

The original isomorphism algorithm is mainly based on topological information, which implies that if a topological difference is detected, the traversal cannot be continued because it is not ensure that the next darts will have the same meanings. For this reason, every time a topological difference is spotted the run stops. Then another starting pair of darts is used in order to restart a new traversal that has not been reached by previous runs. Thanks to the association built in the previous step, every possible starting pairs of darts are known and can be used to start a new traversal. The comparison between LCC1 and LCC2 is finished when each dart contained in the list of possible pairs of starting darts has been processed.

During one traversal of the two 3-maps, dart d is the current dart for the first 3-map, and dart d' is the current dart for the second 3-map. Three types of differences are detected:

1. if dart d and d' do not have compatible neighborhoods;

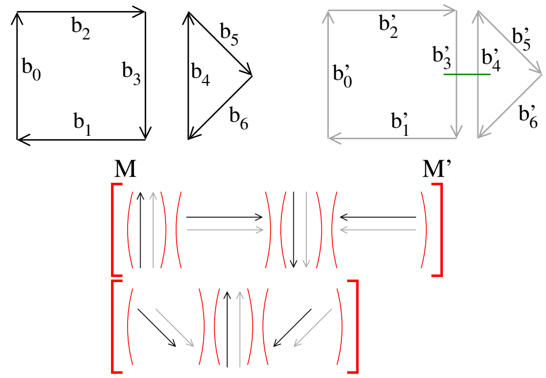


Figure 4: Example of two EBM-LCC comparison: M_1 is composed with a square and a triangle which are not adjacent, while M_2 has also one square and one triangle but adjacent (thus only one connected component). M_1 darts are drawn in black, while M_2 darts are drawn in gray.

2. if dart d and d' do not have similar geometry;
3. if dart d and d' do not have same semantic.

The first case is the same than in the original algorithm: all neighbors darts of d and d' for each β link are tested to verify if their association are compatible. When this is not the case, dart d and d' are marked as topological difference and another pair of compatible darts is taken to start a new traversal.

For the second case, the two geometry of the maximal segments containing darts d and d' are compared. When these two geometries are different, dart d and d' are marked as geometric difference, but in this case the same run can continue. Considering maximal segment (i.e. the maximal set of collinear darts) allows to solve the case shown in Fig. 3 where a same segment is represented by two different set of darts.

The last case consists only to compare the semantic associated with both darts and mark dart d and d' as semantic difference when they differ.

An example of comparison of two EBM-LCC is given in Fig. 4. On the bottom, the list of all pairs of possible starting darts is shown. Note that each connected component of M_1 has its own sub-list of pairs of darts. If the first starting pair of darts considered is b_0 and b'_0, then the first traversal will match darts b_0 and b'_0, b_1 and b'_1, b_2 and b'_2 and b_3 and b'_3. Here a topological difference is spotted because b_3 does not have any β_2 link whereas b'_3 has one. Therefore the run stops and the four pairs of darts (b_0, b'_0) to (b_3, b'_3) are deleted from the list because they have been reached. Another run is started using a next pair, (b_4, b'_4) for example. This run will match all the darts of the two triangles, marking darts b_4 and b'_4 with a topological difference. The list of starting pairs is then empty and the comparison is over.

At the end of the comparison algorithm, each element where a difference has been found is marked accordingly and a list of each difference detected is created. This list is a set of pairs of darts on which differences has been detected. This way we have a comprehensive list of every difference detected on which it is possible to run through easily.

Note that the list contains the three type of difference, geometric, topological or semantic, each dart being marked with the type of the difference. Note also than a same dart could be marked with one, two or three type of differences.

The result for Fig. 4 is a list containing (b_3, b'_3) and (b_4, b'_4), the four darts b_3, b'_3, b_4 and b'_4 being marked as topological difference.

4 Experiments

4.1 Comparison Tool Applied to Errors Detection

A way to detect errors using the comparison tool is cross-referencing. We conducted an experiment in order to compare the semantic of two different EBM-LCC. The first EBM-LCC is coming directly from a CityGML file (the semantic is thus contained in the file) while the second one is reconstructed from a soup of polygons only given in an OBJ file[2]. Note that in this experiment, the OBJ file was created from the CityGML, and thus has lost all its semantic information. For the second EBM-LCC, its semantic is automatically labeled thanks to the method proposed in [DDG14].

Then by comparing the two EBM-LCC, what is actually compared is the way semantics the CityGML file has been determined to the way it is done using the automatic labeling method. This is a way to estimate the quality of the results of the automatic labeling method. For this reason, we do not made other modification on the input OBJ file (geometrical or topological modification) in order to study here only the semantic comparison.

When a semantic difference is spotted, the software color faces compared in purple so that the user can easily locate the difference (see Fig. 6, on the left side). Besides a list that contains every differences spotted is created.

4.2 Data Used

Experimentations were made using Lyon city data [Gra]. These are CityGML files containing a comprehensive description of the city. It includes buildings but also roads, grounds, vegetation,... Each object of the scene is described by polygons with an associated semantic information.

To run the comparison tool, only buildings among data are loaded. Moreover, in order to avoid to load the entire city on each run, data has been cut into slab of 500 by 500 meters

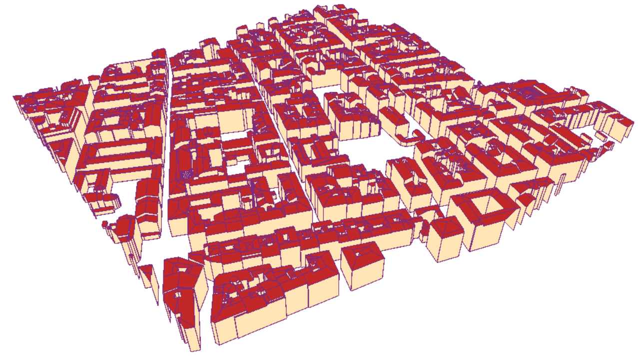


Figure 5: Example of a slab use for tests. This EBM-LCC has been built using a CityGML file coming from the Grand Lyon data.

(see an example of a slab in Fig. 5). Tests were made on 9 slabs which represent a total of 1113 tested buildings (having in average 81000 vertices, 116000 edges and 35000 faces). Slabs used have different density types (slabs are not only center-town data, but also suburbs, villages, etc...). Finally, these datasets contains LoD2 objects only. The reason is that EBM-LCC algorithms were developed for such LoD. Higher LoD can be loaded and the topological reconstruction can be applied. However the automatic semantization method must be modified in order to deal with new semantic and to consider new geometry contained in LoD3 and 4.

4.3 Results

On 1113 tested buildings 1083 were alike, which tents to show that both semantic labeling methods are good. A topological difference has been spotted on 20 buildings but these differences are induced by one pre-processing of EBM-LCC construction. Indeed, during this pre-processing, faces that are co-planar are merged (in order to decrease the number of faces and thus to improve the computation time). Since the order in which faces are considered is not the same for the two processes (due to different input file formats), this implies possible different faces after the merging process. For these reasons, topological differences spotted are not considered as relevant. Finally, on 10 buildings a semantic difference has been spotted. After watching case by case these 10 differences, we were able to certify that these errors come from CityGML files (see one such error in Fig. 6). These results are sum up on Table 1.

Errors on CityGML files were spotted by comparing their semantics to the semantics computed by the automatic labeling process. This implies that semantics computed on EBM-LCC is better. Besides this, correction of CityGML semantic is directly possible thanks to the list of detected errors that has been built.

Finally, execution times have been observed. On Table 2 execution times for the 3 first Slab are given. What takes the

[2]Both our CityGML and OBJ files are triangulated; however our method works both for triangulated and non-triangulated polygons.

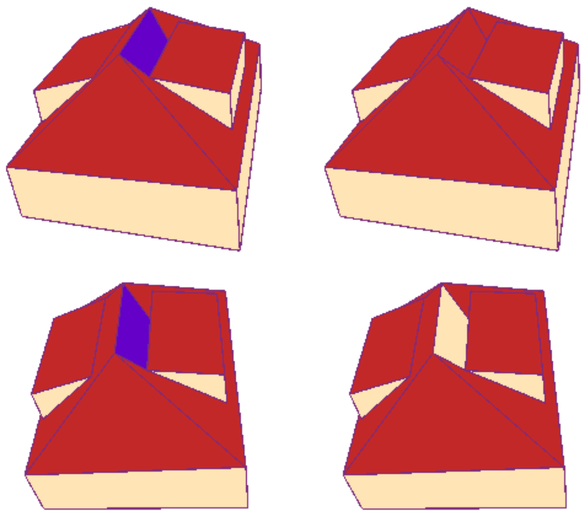


Figure 6: Example of a semantic difference spotted by the comparison algorithm. The top view is the EBM-LCC that comes from the OBJ file, the bottom view is the EBM-LCC that comes from the CityGML file. On the left this is a view that color semantic differences in purple so that the user can spot it easily. On the right this is the normal view with regular semantic associated color.

	#SD	#TD	#AL	#Total
Slab 1	0	0	48	48
Slab 2	1	1	79	81
Slab 3	1	3	135	139
Slab 4	0	1	122	123
Slab 5	0	0	209	209
Slab 6	4	5	119	128
Slab 7	2	0	197	199
Slab 8	0	7	122	129
Slab 9	2	3	52	57
Total	10	20	1083	1113
Percentage	0.90%	1.80%	97.30%	100%

Table 1: Result of our comparison method on 9 slabs. #SD: number of buildings with a semantic difference; #TD: number of buildings with a topological difference; #AL: number of buildings that are alike; #Total: total number of buildings.

	Loading	Comparison	Rest
Slab 1	0.96 (69.5%)	0.37 (26.9%)	0.05 (3.6%)
Slab 2	4.39 (59.5%)	2.26 (30.7%)	0.72 (9.8%)
Slab 3	1.48 (66.9%)	0.65 (29.7%)	0.07 (3.4%)

Table 2: Execution time in seconds for the 3 first Slab. Rest column is other process execution time, it includes preprocesses, display, semantic labeling and topological reconstruction.

greatest among of time (60-70% of the entire execution time) is the loading but this is mainly due to specific libraries used to load CityGML files. The comparison process is quite slow as well (26-31% of the entire execution time). This is mainly due to the association process that require to run through each object of both EBM-LCC (complexity in $O(n^2)$). Nevertheless, this step can still be optimized by using an accelerating structure so that the process ends faster.

5 Conclusion

The aim of this project is to detect and correct errors within existing model of 3D cities by using a new structure called EBM-LCC. To reach this goal, we have proposed in this work a generic comparison tool allowing to compare two 3D models and to compute the geometric, topological, and semantic differences. This tool has been used in a first experiment that allowed us to automatically detect and correct errors in the grand Lyon data. This shows that the EBM-LCC is a good structure that can support the definition of higher level tools for the city modeling domain.

Since our comparison tool is generic, it could be used in different applications, for example to study the evolution of a city by comparing one old data to a new data. Moreover, several error corrections could be defined thanks to the topological description and the coherence rules defined for combinatorial maps. We can for example define a hole filling algorithm in order to guarantee that each volume has no boundary.

In order to reach an error-free model, many things still need to be done. The comparison tool helps detecting some errors but it can happen that two models have the same error, and in that case nothing would be detected. The topological structure of EBM-LCC allows to define new rules on semantic labeling. Such new rules should help to get more and more closer to an error-free model. This way the topology could be used to validate the data. This is one of the many possibilities that the EBM-LCC model offers.

References

[Bau75] BAUMGART B.: A polyhedron representation for computer vision. In *Proc. of AFIPS Na-*

tional Computer Conference (1975), no. 44 in AFIPS '75, pp. 589–596.

[BSL*15] BILJECKI F., STOTER J., LEDOUX H., ZLATANOVA S., ÇÖLTEKIN A.: Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information 4*, 4 (2015), 2842–2889. doi:http://dx.doi.org/10.3390/ijgi4042842.

[Dam11] DAMIAND G.: Combinatorial maps. In *CGAL User and Reference Manual*, 3.9 ed. 2011. http://www.cgal.org/Pkg/CombinatorialMaps.

[DDG14] DIAKITÉ A. A., DAMIAND G., GESQUIÈRE G.: Automatic semantic labelling of 3d buildings based on geometric and topological information. In *Proc. of 9th International 3DGeoInfo Conference* (Dubai, United Arab Emirates, November 2014), 3DGeoInfo conference proceedings series, Karlsruhe Institute of Technology, pp. 49–63.

[DDV14] DIAKITÉ A. A., DAMIAND G., VAN MAERCKE D.: Topological reconstruction of complex 3d buildings and automatic extraction of levels of detail. In *Proc. of 2nd Eurographics Workshop on Urban Data Modelling and Visualisation* (Strasbourg, France, April 2014), Eurographics Digital Library, The Eurographics Association, pp. 25–30.

[dFMMP02] DE FLORIANI L., MESMOUDI M., MORANDO F., PUPPO E.: Non-manifold decomposition in arbitrary dimensions. In *Proc. of 10th International Conference on Discrete Geometry for Computer Imagery* (Bordeaux, France, april 2002), no. 2301 in Lecture Notes in Computer Science, pp. 69–80.

[DL89] DOBKIN D., LASZLO M.: Primitives for the manipulation of three-dimensional subdivisions. *Algorithmica 4*, 1 (1989), 3–32.

[DL14] DAMIAND G., LIENHARDT P.: *Combinatorial Maps: Efficient Data Structures for Computer Graphics and Image Processing*. A K Peters/CRC Press, September 2014.

[DSd*11] DAMIAND G., SOLNON C., DE LA HIGUERA C., JANODET J.-C., SAMUEL E.: Polynomial algorithms for subisomorphism of nd open combinatorial maps. *Computer Vision and Image Understanding (CVIU) 115*, 7 (July 2011), 996–1010.

[Gra] GRANDLYON: http://data.grandlyon.com/.

[HDDM15] HORNA S., DAMIAND G., DIAKITÉ A. A., MENEVEAUX D.: Combining geometry, topology and semantics for generic building description and simulations. In *Proc. of 3rd Eurographics Workshop on Urban Data Modelling and Visualisation* (Delft, the Netherlands, November 2015), Eurographics Digital Library, The Eurographics Association, pp. 13–18.

[OGC16] OGC: *OGC CityGML quality interoperability experiment*. Tech. Rep. Document OGC 16-064r1, 2016. doi:http://www.opengeospatial.net/doc/PER/citygml-quality-ie.

[PMG15] PEDRINIS F., MOREL M., GESQUIÈRE G.: Change Detection of Cities. *Lecture Notes in Geoinformation and Cartography The Selected Papers of the 3D GeoInfo 2014* (2015), 123–139.

[SK07] STADLER A., KOLBE T. H.: Spatio-semantic coherence in the integration of 3D city models. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences. Proceedings of the WG II/7 5th International Symposium Spatial Data Quality 2007 with the theme: Modelling qualities in space and time* (Enschede, the Netherlands, 2007), Stein A., (Ed.), p. 8.

[WAW*15] WAGNER D., ALAM N., WEWETZER M., PRIES M., COORS V.: Methods for geometric data validation of 3d city models. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-1/W5* (2015), 729–735.

[ZSL14] ZHAO J., STOTER J., LEDOUX H.: A framework for the automatic geometric repair of CityGML models. In *Cartography from Pole to Pole*, Buchroithner M., Prechtel N., Burghardt D., (Eds.), Lecture Notes in Geoinformation and Cartography. Springer Berlin Heidelberg, 2014, pp. 187–202.