

DETECTION AND EVALUATION OF TOPOLOGICAL CONSISTENCY IN CITYGML DATASETS

Anna Giovanella, Patrick Erik Bradley, Sven Wursthorn

Karlsruhe Institute of Technology (KIT), Institute of Photogrammetry and Remote Sensing (IPF), Karlsruhe, Germany
{anna.giovanella, bradley, sven.wursthorn}@kit.edu

Commission IV, WG IV/1

KEY WORDS: Topology, Topological Consistency, CityGML, 3D Model, Spatial Database, Polygon, Geometry

ABSTRACT:

The topological consistency of Boundary-Representation models, meaning here that the incidence graph is homeomorphic with the underlying topology of geographical data, is checked for several CityGML datasets, and a first classification of topological inconsistencies is performed. The analysis is carried out on a spatial database system into which the datasets have been imported. It is found that real-world datasets contain many topologically inconsistent pairs of intersecting polygons. Also data satisfying the ISO/OGC standards can still be topologically inconsistent. In the case when the intersection is a point, topological inconsistency occurs because a vertex lies on a line segment. However, the most frequent topological inconsistencies seem to arise when the intersection of two polygons is a line segment. Consequently, topological queries in present CityGML data cannot rely on the incidence graph only, but must always make costly geometric computations if correct results are to be expected.

1. INTRODUCTION

Topological queries like "find all objects at the boundary of object A " or "how near are objects A and B topologically", where topological nearness of A and B means that there is a short path connecting A and B in the incidence graph, can be expected to be most efficiently answered by using the incidence graph of the topological model for given spatial data. The incidence graph is a structure which models the relation "is bounded by", and answering those queries ideally need not resort to the application of geometric operations like intersection, because the incidence graph correctly models the topology. Geometric operations become costly especially when many objects are geometrically near, but topologically not. On the other hand, if objects are topologically near but not geometrically, then they are not considered for the topological query if geometry is used as a basis. Index structures based on Euclidean geometry (like e.g. R -tree) become sub-optimal because they need to take into account objects which are further away than necessary. So, a desideratum is a topological index which relies on the topological model only, ignoring the underlying geometry. This is the topic of ongoing work. A necessary condition for the correctness of such an approach is that the topology underlying the geometrical model coincides with that of the topological model. In other words, it is assumed that the model is *topologically consistent*, a notion which will be made more precise in this article. Possible applications of this are the calculation of the volume, the volume-adjacency graph for path queries, or heat propagation in buildings.

CityGML has become a widespread format for urban building data in various levels of detail (LoDs). Biljecki et al. (2015b) give an overview of different applications of 3D city models. If the data stored in CityGML is to be used for efficient analysis beyond visualisation, they are necessary to be topologically consistent. Otherwise, topological queries yield incorrect results. However, in this present study it turns out that real-world CityGML datasets mostly have different kinds of topological inconsistencies.

After the following Section 2 on related work, we explain in Section 3 first how topology and geometry are modelled in CityGML, followed by a detailed introduction to our notion of topological consistency. The intersection matrix is then introduced as a first means for recognising topological consistency and distinguishing between different types of topological inconsistencies when the configuration consists of two polygons. Section 4 contains a discussion of our results for a collection of CityGML datasets. This is followed by a conclusion and outlook in Section 5.

2. RELATED WORK

The incidence graph is a finite representation of the topology of a spatial model. It has a simple relational database representation through one table for the objects, and another for the topology-defining relation (Bradley and Paul, 2010). It is also shown that its storage complexity is quadratic in the number of objects, and this is in general the most efficient to be expected (Paul, 2008). Furthermore, this data model is universal in that it captures any possible finite topological representation of data (Bradley and Paul, 2010). The literature contains various differing notions of topological consistency, cf. e.g. (Dušan and Branislav, 2004; Li, 2006; Kang and Li, 2005; Rodriguez et al., 2010). Bradley (2015) gives an overview of topological data models and introduces topological consistency in the context of smart cities. Jahn et al. (2017) give a first definition of topological consistency which relates geometry and the incidence graph in the context of distributed big geographical data, and define a measure for topological inconsistency based on Betti numbers of finite partially ordered sets. Alam et al. (2014) have a list of consistency rules for topology and semantics in which they do not allow more than two polygons to have a common edge. Gröger and Plümer (2011) require a consistent model to represent a finite tessellation of \mathbb{R}^3 . This excludes polygons not bordering a solid, like e.g. a building with free walls. Ledoux and Meijers (2011) define a notion of topological consistency which is a special case of the one considered here. They e.g. do not allow polygons with holes or punc-

tures, and they develop an algorithm for extruding planar polygons for to serve as building models. Biljecki et al. (2015a) reduce redundancies in synthetic CityGML data and thus improve the topological consistency.

Applications of such topological consistency are shown e.g. in Steuer et al. (2015), where the volume of buildings in CityGML is approximated by overcoming topological errors. This approach is useful for indoor routing and healing of building models. In general, we emphasise that any topological query in one way or the other makes use of the underlying topology and thus naturally can be applied to the incidence graph in the case that the data are topologically consistent in our sense.

3. METHODOLOGY

3.1 Topology in CityGML

The geometrical and topological models of CityGML are closely related (Gröger and Plümer, 2012; Gröger et al., 2012). The spatial properties of CityGML objects are represented by objects of the geometry model of the Geography Markup Language (GML3) (Cox et al., 2002). This model is based on the ISO Standard 19107 "Spatial Schema", which represents three-dimensional geometries according to the well-known *Boundary Representation* (Foley et al., 1996). The GML3 geometry model consists of primitives that can be combined to form complexes, composite geometries, or aggregates. For every dimension there is a geometrical primitive, such as *Point*, *Curve*, *Surface* and *Solid*. The representation of surfaces and curves is restricted to planar polygons: all coordinates of the outer boundary and of the optional interior boundaries (forming holes in the polygon) must be located in the same plane. Similarly, only straight lines (complying with the GML3 class *LineString*) are allowed. CityGML provides the explicit modelling of topology, for example the sharing of geometry objects between features or other geometries. One part of space should be represented only once by a geometry object and be referenced by all features or more complex geometries which are defined or bounded by this geometry object. So redundancy should be avoided and explicit topological relationships between the parts should be preserved. Instead of implementing topology with own XML-tags, CityGML uses the XML concept of *XLinks*, which is provided by GML3. However, there is no need to model the topology in this way to get a valid CityGML file.

3.2 Topological consistency

Consider a topological model of spatial objects modelled as a polytope complex, i.e. a cell complex whose cells are polytopes of various dimensions. Assume that all vertices are given coordinates. The incidence graph represents the topology of the model correctly, if and only if the intersection of two distinct open cells is empty. The topology of the incidence graph is that of a finite partially ordered set X , where the partial order is given by the "bounded-by"-relation:

$$x \leq y \Leftrightarrow y \text{ is bounded by } x$$

This is a so-called T_0 -topology. It is well-known that the T_0 -topologies on a finite set are in one-to-one correspondence with the partial orders on that set (Alexandrov, 1937). We say that the model is *topologically consistent*, if for all pairs of closed cells A , B it holds true that the intersection of a boundary object of A with a boundary object of B is a common boundary object of A and B .

The definition of topological consistency here extends the definition of Bradley (2015) and differs from that of Jahn et al. (2017). Observe that our definition of topological consistency can be also applied to the situation where the 'cells' of the complex are allowed to have polytope-shaped holes. In that case, it is the topology of the incidence graph which is correctly represented by the model, if and only if it is topologically consistent. Notice that the model can consist of a single, a few, or many objects which may or may not form one or several buildings. We will show that following the ISO/OGC standards does not necessarily mean that the model will be topologically consistent.

3.3 Intersection matrix

Based on the definition of topological consistency in section 3.2, it becomes clear that it is necessary to intersect each polygon with each other to check the topological consistency. For this purpose an intersection matrix I of the form

	V	E	S
V	a	x	y
E	x	b	z
S	y	z	c

with $a, b, c, x, y, z \in \{0, 1\}$ was defined. An entry $I(O, O') = 1$ means that there exists a non-empty intersection between the respective geometric objects

$$O, O' \in \{\text{vertex } (V), \text{edge } (E), \text{surface } (S)\}$$

when intersecting two polygons P and P' , whereas $I(O, O') = 0$ means there is no intersection between geometric objects of the prescribed type. Notice that our intersection matrix is not related to Egenhofer's 9-intersection matrix from (Egenhofer, 1991).

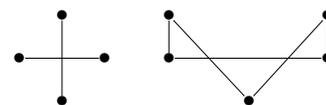


Figure 1. Two topologically inconsistent situations.

As an example, consider the situation in Figure 1. Both configurations of points and line segments are topologically inconsistent, as in each case there are two line objects whose intersection is a point which is not an object of the configuration. The intersection matrices for the configurations, viewed as consisting of line segments, are

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The first intersection matrix can be realised with two distinct polygons in 3D intersecting only in one pair of edges, which represents a topologically inconsistent configuration. See an example for this in Figure 3(d).

The configuration on the right of Figure 1 can be viewed as the boundary of a topologically inconsistent polygon. Another type of topologically inconsistent polygon is given when one vertex lies in the interior of an edge. Then the intersection matrix of the boundary configuration is

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

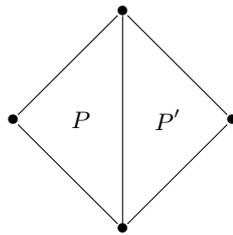


Figure 2. A topologically consistent configuration of two distinct triangles.

depending on whether two edges intersect in their interiors or not. A topologically consistent configuration of two distinct triangles is shown in Figure 2. The corresponding intersection matrix is

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \text{diag}(1,1,0)$$

In Figure 3(e) a three-dimensional constellation of two polygons is depicted that has the same corresponding intersection matrix.

When intersecting two topologically consistent planar polygons P and P' in 3D, there are forty-nine ways in which the intersection matrix I can be populated (Giovanella, n.d.). Of these, four possible configurations can be topologically consistent. In this case, I is a diagonal matrix. If I is not a diagonal matrix, this means that geometric objects of different dimensions intersect. This however means that the intersection geometry can not possibly be a union of vertices and edges of both polygons, i.e. the configuration is not topologically consistent. Conversely, if I is a diagonal matrix, it follows that only geometrical objects of the same dimension intersect. This again means, with two exceptions, that the two intersecting objects O and O' must be identical, i.e. $O \cap O' = O = O'$. In that case, the configuration is topologically consistent. A first exception is the case $I = \text{diag}(0,1,0)$ which is a possible result of the intersection of two three-dimensional polygons. Descriptively, this would mean that two boundary edges of the polygons intersect at a point that is not the vertex of one of the two polygons. This case again corresponds to the constellation illustrated in Figure 3(d). This configuration is not topologically consistent. The other exception is the matrix $I = \text{diag}(1,1,1)$ which means either $P = P'$ or a topologically inconsistent configuration as e.g. in Figure 3(f). The case $I = \text{diag}(0,0,1)$ can not occur in 3D, as this would mean that two surfaces intersect in their interiors. This yields only a valid intersection matrix if both polygons P and P' are identical. If P and P' are identical, then I is equal to $\text{diag}(1,1,1)$ and this leads to a contradiction to $I = \text{diag}(0,0,1)$. The same applies to $I = \text{diag}(0,1,1)$ and $I = \text{diag}(1,0,1)$. Thus, there remain exactly three topologically consistent constellations of I , namely, $\text{diag}(0,0,0)$, $\text{diag}(1,0,0)$, and $\text{diag}(1,1,0)$; and also $\text{diag}(1,1,1)$ which may or may not be topologically consistent. If $I = \text{diag}(0,0,0)$, the two topologically consistent polygons do not intersect, $I = \text{diag}(1,0,0)$ means P and P' share a vertex (see Figure 3(a)), $I = \text{diag}(1,1,0)$ means they share an edge (see Figure 3(e)).

If the intersection of two polygons is a point, then there can occur four different intersection matrices. These four matrices can be given the following descriptive names:

$$\begin{aligned} \text{point-point} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \text{point-line} &= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ \text{point-area} &= \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} & \text{line-line} &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

As you have seen above 'point-point' describes a topologically consistent, whereas 'point-line', 'point-area' and 'line-line' describe topologically inconsistent configurations of two distinct polygons. These four intersection constellations are depicted in Figures 3(a)-3(d), where you can see a simple synthetic example of a house with different kinds of topological inconsistencies.

3.4 Implementation

In order to make topological and geometric queries, the CityGML data was imported into a 3d City Database schema (3DCityDB, 2018; Stadler et al., 2009). 3DCityDB is a free Open Source package consisting of a database schema and a set of software tools to import, manage, analyse, visualise, and export virtual 3D city models according to the CityGML standard. The database schema results from a mapping of the object oriented data model of CityGML 2.0 to the relational structure of a spatially-enhanced relational database management system (SRDBMS). The 3DCityDB supports the commercial SRDBMS Oracle (with 'Spatial' or 'Locator' license options) and the Open Source SRDBMS PostgreSQL which is an extension to the free RDBMS PostgreSQL and which was used for this work. 3DCityDB is in use in real life production systems in many places around the world and is also being used in a number of research projects. As an example, consider Chaturvedi et al. (2015). According to 3DCityDB (2018), the cities of Berlin, Potsdam, Munich, Frankfurt, Zurich all keep and manage their virtual 3D city models within an instance of 3DCityDB. The included Importer/Exporter software tool allows for high performance importing and exporting of CityGML datasets according to CityGML versions 2.0 and 1.0. The tool allows the processing of very large datasets, even if they include XLinks between CityGML features or XLinks to three-dimensional GML geometry objects (Kunde, 2012; Kunde et al., 2013).

The implementation uses SFCGAL functions (SFCGAL, 2018). SFCGAL is a wrapper around the Computational Geometry Algorithms Library (CGAL, 2018) that intends to implement 2D and 3D operations on OGC standard models (Simple Feature Access, CityGML, ...). Using the C API of SFCGAL, PostGIS exposes some of SFCGAL's functions in spatial databases and can be patched for more functions.

The first part of the intersection analysis was done directly in the database using SQL queries, taking advantage of spatial indices. In order to effect this, the Procedural Language/PostgreSQL Structured Query Language (PL/pgSQL) (Eisentraut, 2003) was used to write a function. PL/pgSQL was introduced to extend PostgreSQL's SQL capabilities. PL/pgSQL code can be stored as a *Stored Procedure* in the database itself. It supports variables, conditions, loops, functions, database cursors, and exception handling. PL/pgSQL code can be called from both SQL commands and database triggers. For each intersecting pair of polygons, the intersection geometry was calculated and the geometry type of the intersection geometry was determined and the results were

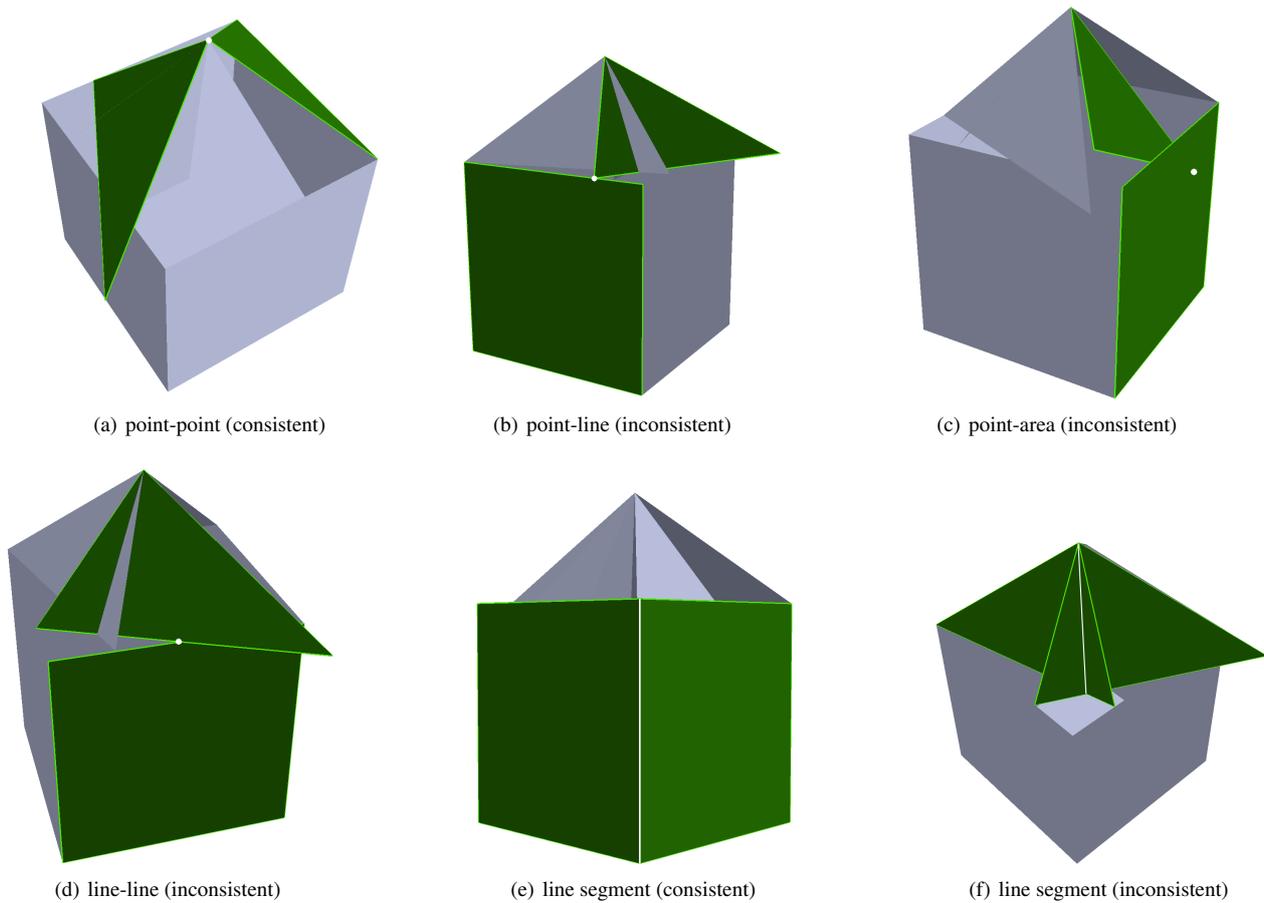


Figure 3. Simple synthetic example of a house with different kinds of topological inconsistencies. The green geometries depict the different types of intersection constellations.

written into a newly created table in the database. For this purpose, corresponding SFCGAL functions were used, which are provided by PostGIS. PostGIS aims to support the SQL option of the OGC Simple Features Access standard (Herring, 2010). Previously, all polygons were checked for validity, i.e. they were tested for planarity and self-intersection, and the position and orientation of interior rings were checked. For the validity check, the SFCGAL function *isValid3d* was used, which had to be patched to PostGIS, as the *stIsValid* function provided by PostGIS can only process two-dimensional geometries.

The intersection matrix operators were then implemented directly in C++ within the SFCGAL framework, since the SFCGAL functions provided by PostGIS were not sufficient to perform the necessary queries directly on the database. For this purpose, the pairs of polygons whose intersection geometry type is *Point* or a line segment (i.e. *LineString* consisting of only two points) were first exported from the database, and then further processed by a C++ function. For now, only these two types of intersection geometries have been considered, as they occur most often in CityGML datasets and it is quite easy for them to determine the intersection constellation. If the intersection geometry is a point, then, as described in Section 3.3, there are four possible intersection matrices, of which exactly one comes from a topologically consistent configuration. To determine the intersection matrix, it is first checked if the point of intersection is equal to one of the vertices of one or both the intersected polygons. If a matching vertex is found on both polygons, it means that both polygons intersect

at that point. For this configuration, the intersection matrix corresponds to 'point-point'. This case is topologically consistent. If no matching vertex is found on either of the two polygons, then the intersection matrix corresponds to 'line-line', as this is only possible when two edges of the polygons intersect. If the intersection point is identical to a vertex of one of the two intersected polygons, then it is further tested whether it lies on one edge of or inside the other polygon. If it lies on one edge of the other polygon, then the intersection matrix corresponds to 'point-line' and if it lies within the interior of the other polygon, the intersection matrix corresponds to 'point-area'.

For the intersection geometry type line segment a distinction was made only between consistent and inconsistent, since it would be very costly to determine the exact intersection matrices for all possible configurations. In fact, the set of all possible configurations of two distinct polygons for a given intersection matrix has not yet been found, except in the case when the intersection is a point. To distinguish between consistent and inconsistent intersection constellations of a line segment, it is sufficient to check if both polygons contain the intersection geometry, i.e. whether the line segment is identical to an edge of both polygons. If so, the configuration is topologically consistent, otherwise inconsistent.

4. EXPERIMENTAL RESULTS AND DISCUSSION

For this study, nine real-world datasets and four synthetic datasets were used. The largest dataset contains the whole city of Delft in

name	polygons	valid [%]	intersections	buildings	max _{cons.}	val3d
Delft	112,888	100	1,168,248	7,970	21	7,970
Alexanderplatz	57,831	99.7	1,316,110	1,123	1,033	894
Pariser Platz	6,008	99.7	52,368	18	2	2
Karlsruhe 1	745	86.7	4,564	95	*	95
Karlsruhe 2	643	92.5	3,530	67	*	62
Karlsruhe 3	624	90.4	4,410	64	*	64
Karlsruhe 4	918	80.8	5,066	104	*	102
Potsdam	1,547	100	9,074	97	97	97
Waldbrücke	4,038	93.6	20,644	491	*	484
Random3DCity 1	7,890	88.9	36,710	900	*	900
Random3DCity 2	40,263	88.7	8,212	900	*	885
Random3DCity Error 1	679	99.6	8,642	100	*	100
Random3DCity Error 2	4,663	88.8	2,206	100	*	98

Table 1. List of CityGML datasets used in this study. The column "valid" gives the proportion of valid polygons within the dataset. The column "max_{cons.}" gives the number of buildings consisting of valid polygons and not containing a topologically inconsistent point or line segment intersection, or an intersection of types *TIN* or *Triangle*. And "val3d" gives the number of valid solids according to val3dity. * Assignment of polygons to buildings was not possible.

LoD1. It was downloaded from (TUDelft 3D geoinformation, 2018). Two of the datasets contain small parts of Berlin in LoD2. These are the datasets Alexanderplatz and Pariser Platz (the place where the "Brandenburg Gate" is located). Both datasets as well as the likewise available CityGML model from all of Berlin were generated from extracted cadastral data. All available CityGML datasets from the city of Berlin can be downloaded from (Berlin Business Location Center, 2018). Also, four datasets from Karlsruhe were examined. These datasets come from the "Liegenchaftsamt" of the city of Karlsruhe. They contain single streets or small residential areas of the city of Karlsruhe, which were generated from LIDAR data and modelled in LoD2. The two other real-world datasets are available in LoD1. These are the whole city of Potsdam and the village Waldbrücke, which is part of the municipality Weingarten near Karlsruhe. The Potsdam dataset is included in the download package of the 3DCityDB and Waldbrücke was downloaded from the CityGML homepage. In addition, four synthetic datasets were used, which were generated with the tool Random3DCity (Biljecki et al., 2016a) and which can be downloaded from the project homepage (Biljecki, 2018). Two of these datasets contain buildings in LoD2 and the others in LoD3. Two of these datasets (one in LoD2, one in LoD3) were modelled with errors (overlapping buildings). Table 1 shows a list of the CityGML datasets used in this study together with the number of polygons and non-empty intersections of distinct polygons. It can be seen that the vast majority of polygons are valid, i.e. are both planar and without self-intersections and if there exists an interior ring, its position and orientation are correct. In particular, they are topologically consistent.

An unsolved problem with the first part of the implementation is the PostGIS-provided SFCGAL function *st_3dintersection*, which for certain polygon configurations leads to an interruption of the connection to the database due to a crash of the responsible code of the function *st_3dintersection*. This leads to the fact that some mostly larger datasets could not be completely processed. The only way to avoid these crashes was to drop the respective polygons from the database. For this reason, 6 polygons were deleted from the dataset Delft and 8 polygons from the dataset Pariser Platz. Another problem, which also leads to crashes, occurred with two synthetic datasets with "topological errors". This concerned the PostGIS function *st_3dintersects*, which first tests if two polygons intersect before the actual intersection analysis is

started. This problem could not be solved so far, that is why these two datasets could not be used.

Figure 4 shows the percentages of polygon intersection types when the intersection is non-empty and the two polygons are distinct. A *MultiLineString* is a union of *LineStrings* with at least two components. *TIN* stands for *Triangulated Irregular Network* and means that the configuration is topologically inconsistent, as the intersection contains a surface strictly contained in the faces of both polygons. The other types may or may not be topologically consistent. It can be seen that the intersection is, in the vast majority, either a point or a line segment (an exception being Random3DCity Error 2). That is the reason why these two types of intersections were further investigated. Thus, between 48.4% and 98.8% of the intersections could be analysed for the examined datasets. It is also striking that the distribution in the dataset Alexanderplatz deviates significantly from the other datasets. Also the datasets from Karlsruhe, as well as the two synthetic datasets have a higher proportion of intersections of the type *Point*. In the case of the two erroneous synthetic datasets, the proportion of the intersection types *Point* and *LineString* is lower and the proportion of *MultiLineString* and *TIN* is higher.

Figure 5 shows the relative frequencies of occurrences of the four intersection matrices when the intersection of two distinct polygons is a point. Except for Delft, the majority consists of the topological consistent case of 'point-point'. However, most datasets have a large proportion of topologically inconsistent intersection matrices of type 'point-line'. Only the synthetic datasets with exception of Random3DCity Error 2 do not contain topologically inconsistent intersections of the type *Point*.

Figure 6 shows the proportion of topologically consistent or inconsistent polygon pairs when the intersection is a point or a line segment (special case of the type *LineString*). These are by far the most frequent intersection types, at least in the real-world datasets. For the synthetic datasets Random3DCity 1 and Random3DCity 2 it turns out that all those configurations are topologically consistent. For the real-world datasets the majority, but by no means all, of the configurations are topologically consistent. The most frequent inconsistent intersection type for these data is *LineString*.

Looking at all the results, it is noticeable that there is a high in-

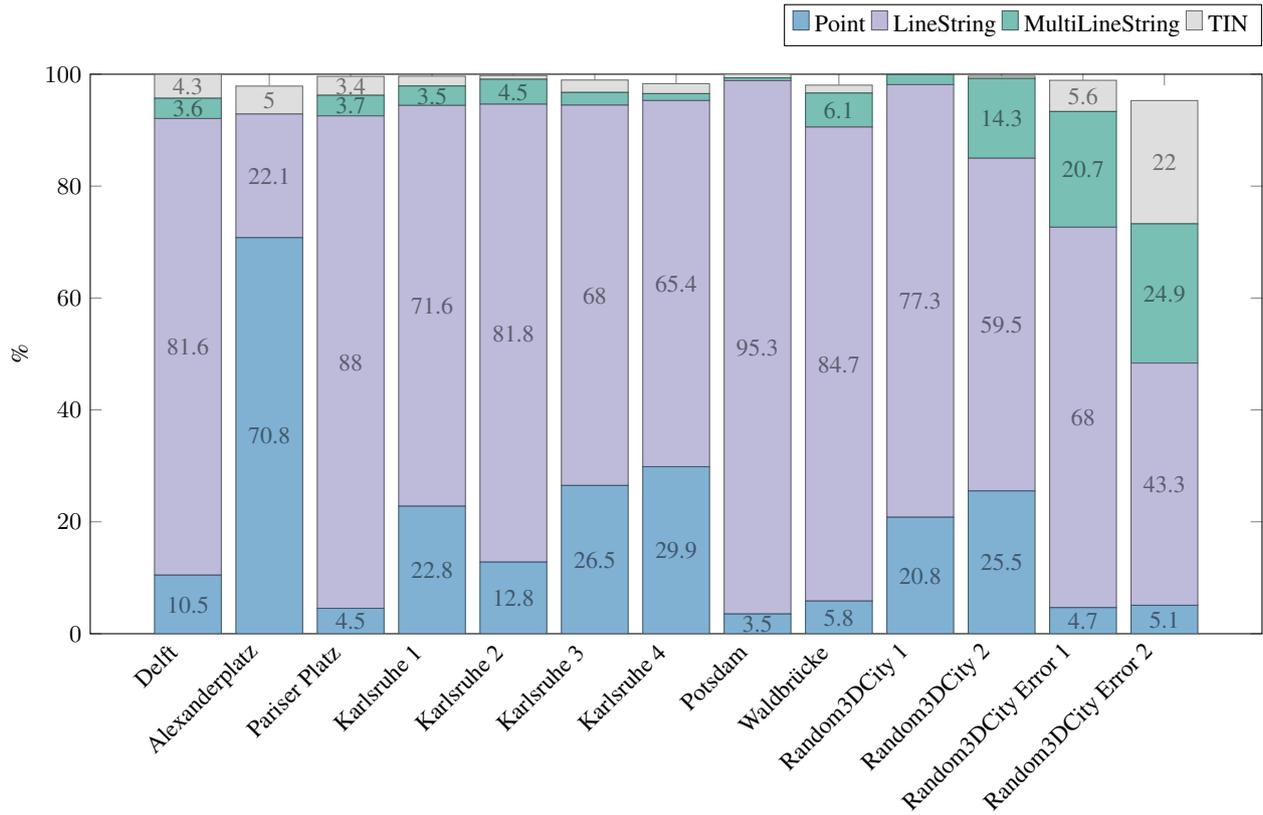


Figure 4. Proportions of the most frequent non-empty polygon intersection types

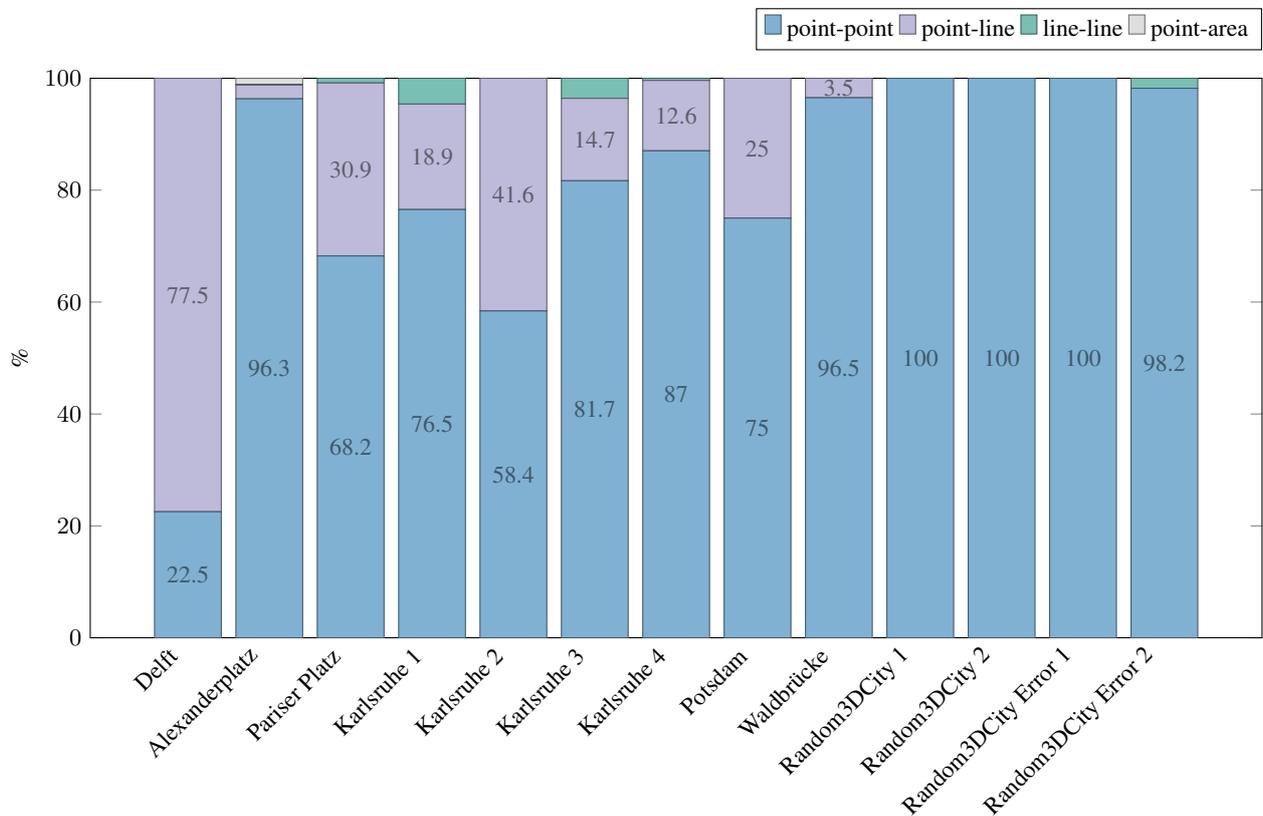


Figure 5. Intersection matrices when intersection is a point.

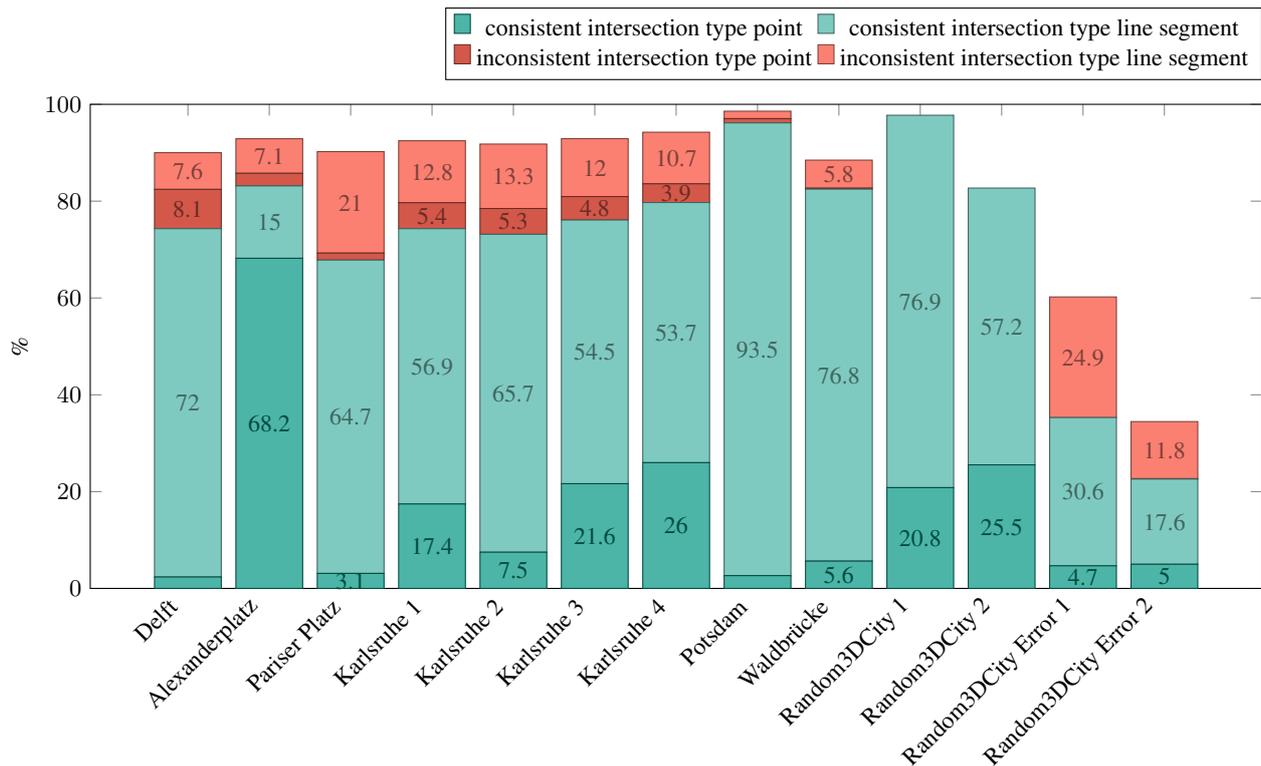


Figure 6. Topologically consistent and inconsistent configurations with point and line segment intersections.

homogeneity between the datasets. The distribution of the intersection types and the proportion of topologically inconsistent constellations seem to depend on the type of data collection and the LoD.

In Ledoux (2013) is described a tool (val3dity) for validating solids against the ISO/OGC specifications. This tool also checks if pairs of distinct solids intersect in their interiors. However, it does not verify if they intersect in a topologically inconsistent way in their boundaries, because it is not required by the standard. This would necessitate the check of intersecting polygon pairs. In difference to val3dity, the aim of this work is to check topological consistency regardless of the conformity to the corresponding standards. For example, the topologically inconsistent example house used here for illustration purposes has been run through val3dity and has been found 'valid', when it is modelled as a combination of *MultiSurfaces*, which is correct according to the CityGML standard. The comparison of val3dity to our methodology is only possible if building shells are modelled as *Solid*, which means one exterior shell minus possible interior shells. The assignment of polygons to buildings becomes problematic if the building shell is modelled as *MultiSurface* geometries instead of *Solid*. For this reason, buildings with inconsistent polygons and polygon pairs can only be determined in case they are modelled as *Solid* (cf. Tab. 1).

In (Biljecki et al., 2016b), the most common geometric and semantic errors in CityGML data are analysed. They find that the most common topological errors are that polygons are not properly oriented, and that geometries are not properly "snapped". From what is stated there, one can see that our approach is on the one hand a further differentiation of that error type, and, on the other hand (unlike *loc. cit.*) we do not require a building to consist of solids only, as long as the polygons intersect in common boundary elements. E.g. balconies, porches, and shelters often

have geometries which do not form a shell, i.e. are non-closed surfaces.

5. CONCLUSION AND OUTLOOK

Using the B-Rep model means to rely on the correctness of its incidence graph. This is only the case if the topology underlying the geometric model coincides with the topology underlying the B-Rep model—in other words, if the data are topologically consistent in our sense. In the case of CityGML it is possible to model correctly according to the standard and still have a topologically inconsistent model. Towards distinguishing between different forms of topological inconsistency, the intersection matrix defined here is a first indicator. However, some matrices are ambiguous: they can come from both, consistent and inconsistent configurations. A classification of these matrices is work in progress.

Among the many different notions of topological consistency, the one considered here relates geometry and the incidence graph in such a way that topological consistency means that the incidence graph models the topology underlying the geometric model. Nine real-world and two synthetic CityGML datasets were examined with the scope of checking topological consistency in this sense and to classify the most frequent topological inconsistencies. It turns out that real CityGML data are topologically inconsistent and the distribution of their inconsistency types varies. The most frequent inconsistent case is when the intersection of two polygons is a line segment. In the case that the intersection is a point, the most frequent inconsistency is when a vertex lies in the interior of a line segment. Hence, the data are not suitable for efficient analysis beyond visualisation, as topological queries are bound to yield incorrect results, if they rely on the incidence graph only, in order to avoid costly geometric computations. This means that

in the process of producing a geometry model in CityGML from point cloud data, it is necessary to include a check for topological consistency in the sense of this article. Finding ways of healing such data with the aim of storing only topologically consistent datasets in topological databases is work in progress. Furthermore, to distinguish the different types of data collection by their types of topological inconsistency is the topic of future work.

ACKNOWLEDGEMENTS

Stefan Hinz is thanked for valuable discussions and suggestions. The Liegenschaftsamt of the City of Karlsruhe is thanked for some datasets. The anonymous reviewers are thanked for criticism which helped to significantly improve this paper. This work is partially supported by the Deutsche Forschungsgemeinschaft under grant number BR 3513/12-1.

REFERENCES

- 3DCityDB, 2018. 3DCityDB database schema for CityGML. <https://www.3dcitydb.org/3dcitydb/3dcitydbhomepage/>. Last visited on 15/05/2018.
- Alam, N., Wagner, D., Wewetzer, M., von Falkenhausen, J., Coors, V. and Pries, M., 2014. Towards automatic validation and healing of CityGML models for geometric and semantic consistency. In: U. Isikdag (ed.), *Innovations in 3D Geo-Information Sciences*, Springer, pp. 77–91.
- Alexandrov, P. S., 1937. Diskrete Räume. *Matematicheskii Sbornik (N.S.)* 2, pp. 501–518.
- Berlin Business Location Center, 2018. Berlin 3D - Download Portal. <http://www.businesslocationcenter.de/en/downloadportal>. Last visited on 15/05/2018.
- Biljecki, F., 2018. Random3Dcity - Synthetic CityGML data and procedural modelling engine. <https://3d.bk.tudelft.nl/biljecki/Random3Dcity.html>. Last visited on 15/05/2018.
- Biljecki, F., Ledoux, H. and Stoter, J., 2015a. Improving the consistency of multi-LOD CityGML datasets by removing redundancy. In: M. Breunig, M. Al-Doori, E. Butwilowski, P. V. Kuper, J. Benner and K. H. Haefele (eds), *3D Geoinformation Science: The Selected Papers of the 3D GeoInfo 2014*, Lecture Notes in Geoinformation and Cartography, Springer, pp. 1–17.
- Biljecki, F., Ledoux, H. and Stoter, J., 2016a. Generation of multi-LOD 3D city models in CityGML with the procedural modelling engine Random3Dcity. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences IV-4/W1*, pp. 51–59.
- Biljecki, F., Ledoux, H., Du, X., Stoter, J., Soon, K. H. and Khoo, V. H. S., 2016b. The most common geometric and semantic errors in CityGML datasets. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences IV-2/W1*, pp. 13–22.
- Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S. and Çöltekin, A., 2015b. Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information* 4(4), pp. 2842–2889.
- Bradley, P. E., 2015. Supporting data analytics for smart cities: An overview of data models and topology. In: A. Gammerman, V. Vovk and H. Papadopoulos (eds), *Statistical Learning and Data Sciences SLDS 2015*, LNCS 9047, Springer, pp. 406–413.
- Bradley, P. E. and Paul, N., 2010. Using the relational model to capture topological information of spaces. *The Computer Journal* 53, pp. 69–89.
- CGAL, 2018. Computational Geometry Algorithms Library. <http://www.cgal.org/>. Last visited on 15/05/2018.
- Chaturvedi, K., Yao, Z. and Kolbe, T. H., 2015. Web-based exploration of and interaction with large and deeply structured semantic 3D city models using HTML5 and WebGL. In: 35. *Wissenschaftlich-Technische Jahrestagung der DGPF*, Deutsche Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V.
- Cox, S., Daisey, P., Lake, R., Portele, C. and Whiteside, A., 2002. Geography Markup Language (GML) encoding specification. Standard, OGC.
- Dušan, J. and Branislav, B., 2004. Elements of spatial data quality as information technology support for sustainable development planning. *Spatium* 11, pp. 88–83.
- Egenhofer, M. J., 1991. Reasoning about binary topological relations. In: O. Günther and H.-J. Schek (eds), *Advances in Spatial Databases*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 141–160.
- Eisentraut, P., 2003. *PostgreSQL: Das offizielle Handbuch*. verlag moderne industrie.
- Foley, J. D., van Dam, A., Feiner, S. K. and Hughes, J. F., 1996. *Computer Graphics: Principles and Practice in C, 2nd Edition*. Addison-Wesley.
- Giovannella, A., n.d. Automatische redundanzfreie Generierung von Topologie aus multidimensionalen geographischen Datenbeständen. PhD thesis in progress.
- Gröger, G. and Plümer, L., 2011. How to achieve consistency for 3D city models. *Geoinformatica* 15(1), pp. 137–165.
- Gröger, G. and Plümer, L., 2012. CityGML – interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing* 71, pp. 12–33.
- Gröger, G., Kolbe, T. H., Nagel, C. and Häfele, K.-H., 2012. OGC City Geography Markup Language (CityGML) encoding standard. Standard, OGC.
- Herring, J. R., 2010. Opendata implementation standard for geographic information—simple feature access—part 2: SQL option. Technical Report OGC 06-104r4, Open Geospatial Consortium Inc.
- Jahn, M. W., Bradley, P. E., Al-Doori, M. and Breunig, M., 2017. Topologically consistent models for efficient big geo-spatio-temporal data distribution. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences IV-4/W5*, pp. 65–72.
- Kang, H. and Li, K., 2005. Assessing topological consistency for collapse operation in generalization of spatial databases. In: J. Akoka (ed.), *Perspectives in Conceptual Modeling*, LNCS 3770, Springer, Berlin.
- Kunde, F., 2012. CityGML in PostGIS – Portierung, Anwendung und Performanz-Analyse am Beispiel der 3DCityDB Berlin. Master's thesis, Universität Potsdam.
- Kunde, F., Nagel, C., Herrerruela, J., Ross, L. and Kolbe, T. H., 2013. 3D-Stadtmodelle in PostGIS mit der 3D City Database. In: *Tagungsband der FOSSGIS-Konferenz*.
- Ledoux, H., 2013. On the validation of solids represented with the international standards for geographic information. *Computer-Aided Civil and Infrastructure Engineering* 28(9), pp. 693–706.
- Ledoux, H. and Meijers, M., 2011. Topologically consistent 3D city models obtained by extrusion. *International Journal of Geographical Information Science* 25(4), pp. 557–574.
- Li, S., 2006. On topological consistency and realization. *Constraints* 11(1), pp. 3151.
- Paul, N., 2008. Topologische Datenbanken für Architektonische Räume. PhD thesis, Universität Karlsruhe, Germany.
- Rodriguez, M., Brisaboa, N., Meza, J. and Luaces, M., 2010. Measuring consistency with respect to topological dependency constraints. In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 182–191.
- SFCGAL, 2018. Simple Features wrapper library for CGAL. <http://www.sfcgal.org>. Last visited on 15/05/2018.
- Stadler, A., Nagel, C., König, G. and Kolbe, T. H., 2009. Making interoperability persistent: A 3D geo database based on CityGML. In: J. Lee and S. Zlatanova (eds), *3D Geo-Information Sciences*, Lecture Notes in Geoinformation and Cartography, Springer, pp. 175–192.
- Steuer, H., Machl, T., Sindram, M., Liebel, L. and Kolbe, T. H., 2015. Voluminator—approximating the volume of 3D buildings to overcome topological errors. In: F. Bacao, M. Santos and M. Painho (eds), *AG-ILE 2015*, Lecture Notes in Geoinformation and Cartography, Springer, Cham.
- TU Delft 3D geoinformation, 2018. Open datasets created with 3dfier. <https://3d.bk.tudelft.nl/opendata/3dfier/>. Last visited on 15/05/2018.