

Provably correct and complete transaction rules for updating 3D city models

Gerhard Gröger · Lutz Plümer

Received: 8 March 2010 / Revised: 28 March 2011
Accepted: 4 May 2011 / Published online: 25 May 2011
© Springer Science+Business Media, LLC 2011

Abstract The shapes of our cities change very frequently. These changes have to be reflected in data sets representing urban objects. However, it must be assured that frequent updates do not affect *geometric-topological consistency*. This important aspect of spatial data quality guarantees essential assumptions on which users and applications of 3D city models rely: viz. that objects do not intersect, overlap or penetrate mutually, or completely cover one another. This raises the question how to guarantee that geometric-topological consistency is preserved when data sets are updated. Hence, there is a certain risk that plans and decisions which are based on these data sets are erroneous and that the tremendous efforts spent for data acquisition and updates become vain. In this paper, we solve this problem by presenting efficient *transaction rules* for updating 3D city models. These rules guarantee that geometric-topological consistency is preserved (*Safety*) and allow for the generation of arbitrary consistent 3D city models (*Completeness*). Safety as well as completeness is proven with mathematical rigor, guaranteeing the reliability of our method. Our method is applicable to 3D city models, which define—besides the terrain surface—complex spatial objects like buildings with rooms and storeys as interior structures, as well as bridges and tunnels. Those objects are represented as aggregations of solids, and their surfaces are complex from a topology point of view. 3D GIS models like *CityGML*, which are widely used to represent cities, provide the means to define semantics, geometry and topology, but do not address the problem of maintaining consistency. Hence, our approach complements *CityGML*.

Keywords Transaction rules · Consistency · Surfaces · 3D models · Handles · 2-manifolds

G. Gröger (✉) · L. Plümer
Institute for Geodesy and Geoinformation, University of Bonn, Meckenheimer Allee 172, 53115 Bonn,
Germany
e-mail: groeger@igg.uni-bonn.de

L. Plümer
e-mail: pluemer@igg.uni-bonn.de

1 Introduction

The shapes of our cities change with a very high frequency. For transportation objects alone, it is estimated that 10–15% of all objects change every year. In populous areas, changes of about 40% of all objects are observed [34]. For 3D city models in general, these figures will be higher. These changes of urban objects have to be reflected in data sets representing cities, for example in vehicle navigation data, land use data or 3D city models. The quality of plans and decisions such as route planning or noise control planning, highly depends on the up-to-datedness of data. On the other hand, it must be avoided that updates affect spatial data quality [18]. If this last requirement is not met, the tremendous efforts spent for data acquisition and updates will be in vain and the return of investment would be questionable.

An important aspect of spatial data quality is *geometric-topological consistency* [12, 15, 22], which guarantees many important real world assumptions users have about their spatial data. Examples for such assumptions are that objects do not intersect, overlap or penetrate mutually, or completely cover one another, e.g. rooms and buildings. The topic of this paper is a method for updating three-dimensional spatial data sets, guaranteeing that geometric-topological consistency once given is preserved in a reliable way. The method consists of a set of *transaction rules* for updating 3D city models.

Consistency checking has *static* and *dynamic* aspects. The static aspect deals with methods to check whether a spatial dataset *as a whole* is consistent. In [15] and [14], we have presented *axioms* fulfilling that task efficiently and reliably. The transaction rules presented in this paper cover the *dynamic* aspects of consistency checking. Updates of data sets (the insertion of a building or a bridge, for example, or the deletion of a building) typically have *local* effects. Hence, globally checking local changes would be sufficient with regard to correctness, but not in terms of efficiency. Transaction rules, which emerged from the field of active databases [38], locally check local updates, by restricting the axioms to that part of the data set which is actually affected by the changes. Furthermore, incomplete update requests are completed in a consistent way and inconsistent updates are rejected.

The transaction rules are based on a specific consistency notion which is given by the concept of a *geometrical-topological 3D city model*, or in short, *3D city model* [15]. Section 2 will introduce this model, which is sufficiently general to cover most common 3D GIS models [28, 33] or 3D city models, in particular those represented in *CityGML* [9, 24]. The model integrates *surfaces* that represent the terrain and the outer hull of buildings with *solids* or *aggregations* of solids that model volume objects (buildings, for example) or interior structures (rooms or other building parts). Other spatial objects covered by the model are bridges, tunnels or arcades, which are different from a topological point of view. Those objects are mathematically captured by the notion of a *handle*. The rules can insert or delete arbitrary handles in order to insert or delete objects like tunnels, bridges and arcades.

The focus of the transaction rules, presented in this paper, is set on topology and geometry. Semantic aspects, such as the object type (residential building or industrial building, for example), or attributes (like the year of construction or the owner's name) are dealt with in semantic models like *CityGML*. *CityGML* is an OGC standard for the definition of the semantical, geometrical and topological aspects of cities but does not address the problem of maintaining consistency. Hence, our approach complements *CityGML*.

A crucial property of transaction rules is that consistency is preserved for *arbitrary applications* of the rule. Such rules are called *safe*. Another important property is the

usability and flexibility of the rules in the sense that each (consistent) 3D city model can be generated and deleted by application of the rules. Such a set is called *complete*. All rules presented in this paper are safe and the whole set is complete. Both properties are proven mathematically.

The set of rules, introduced in this paper, extends and generalizes earlier work which covers the 2D case [11]. Some approaches provide transaction rules for surfaces embedded in 3D [8, 28], particularly the well-known *Euler Operators* [28] and its application to GIS [35]. This approach has been extended to solid models [3]. However, these approaches do not guarantee consistency between geometry and topology. More details will be provided in Section 5.

The contribution of this paper is the provision of rules for updating complex 3D city models which are represented in formats like CityGML. The rules maintain the consistency of these models in a reliable way (safety). At the same time, the rules provide the flexibility to realize any consistent update (completeness). The rules allow for complex updates in 3D City models, including aggregated objects like buildings and topologically complex objects like tunnels and bridges.

The difficult problems solved in this paper are mainly caused by handles in different respects: Regarding the safety of the rules, the occurrence of handles may cause non-manifold boundaries even if constructing non-handle objects. This phenomenon is not obvious. Furthermore, handles entail difficulties proving that any 3D model can be generated by the rules (completeness): Due to the occurrence of handles, there are consistent 3D models which cannot be generated in a straightforward way by any rules maintaining consistency. More details about the solutions of both problems can be found in Sections 3.2 and 3.3.

The rest of this paper is organized as follows: The second section presents the notion of consistency for surfaces and 3D city models which provides the base of the rules. Consistency is specified declaratively as well as axiomatically. Furthermore, the relation to CityGML is discussed. The transaction rules are introduced in the third section. After an outline, the rules for updating 3D city models are given in detail. First, rules which split and merge solids and hereby preserve the number of handles are introduced. To insert or delete handles, an extended version of the rules is provided. The safety of each of the rules is proven formally, as well as the completeness of the whole set. After presenting a concept for the implementation of the rules on top of a spatial data base in Section 4, Section 5 recalls related work on static and dynamic consistency checking in GIS. This paper ends with concluding remarks and a discussion of open questions and further work. Mathematical definitions that are used in the paper are provided in Appendix A.

2 Consistency of 3D city models

Three-dimensional city models can be viewed as the composition of two types of components: first, composition of *solids* and *aggregations of solids* which model volume objects like buildings and their interior structure, and second composition of *surfaces* representing the terrain that surrounds objects like buildings. Since each solid is bounded by a closed surface, the atomic entity of a 3D city model is a (composed) surface, which is either closed (bounding solids) or not closed (representing the terrain). These surfaces may have *handles* in order to represent tunnels and bridges. The next subsection will shortly introduce the components of 3D city models (surfaces, solids and aggregations of solids). An axiomatic characterisation of those models providing an effective and efficient

procedure to check consistency is given in subsection 2.3. Standard definitions and notations from graph theory [19] to mathematical topology [1, 2, 20, 28] are being used. A definition of the most important concepts can be found in Appendix A.

2.1 3D city models

For modelling the terrain, special surfaces, called *2.8D maps* [12], are suitable. A 2.8D map is a connected cell complex embedded in 3D space that is single 2-manifold.¹ A 2.8D map can be imagined as a cloth draped over the terrain, including man-made objects like buildings. In contrast to 2.5D terrain models [31], a 2.8D map allows for vertical walls and overhangs. A special, unbounded face *Out* surrounds the bounded faces of the map. This face extends to infinity, since it surrounds all other faces, edges and vertices. In contrast to all other faces, *Out* does not have to be planar. Figure 1 depicts the outer hull of a saddle roof building (walls, roof) as an example for a 2.8D map. It contains vertical walls and overhangs (roof overhangs, for example).

Volume objects are modelled by *solids* [21, 28]. A solid is bounded by a single *closed composite surface* [21]. A closed composite surface is a connected cell complex consisting of 0-, 1- and 2-cells and being topologically equivalent to a sphere, i.e. is a closed 2-manifold. Since solids are bounded by a single connected 2-manifold surface (exterior shell), enclaves ('solids in solids') which are represented by *interior shells* [21] are excluded.

A closed composite surface is a structure similar to a 2.8D map. Both concepts differ from each other only by the existence of the unbounded face *Out*, which is missing in closed composite surfaces. Instead, a closed composite surface encloses an object completely.

The integration of a 2.8D map (terrain, outer hull of buildings) with solids and aggregations of solids (volume objects) defines a *3D city model* [15]. A 3D city model is a three dimensional, connected cell complex, with the additional property that all solids cover the space completely: it forms a three-dimensional *tessellation* of space. Hence, unwanted voids like for example inside buildings, are avoided. A simple example for a 3D city model is given in Fig. 2: Two saddle roof buildings, one of them with a neighbouring garage, are integrated in the terrain (grey face *Out*).

In order to achieve a complete space coverage, two special solids are introduced: an *air solid* representing the air mass, and an *earth solid* representing the earth's mass. The air solid and the earth solid are partially bounded by a 2.8D map. Again, due to the connectedness of the model, enclaves are not allowed.

Figure 3 depicts a cross profile of a 3D city model: the air solid is bounded partially by a 2.8D map representing the terrain and the outer hull of two buildings, including a tunnel. The earth solid models again the terrain including the tunnel and the outer hull of the cellars of both buildings.

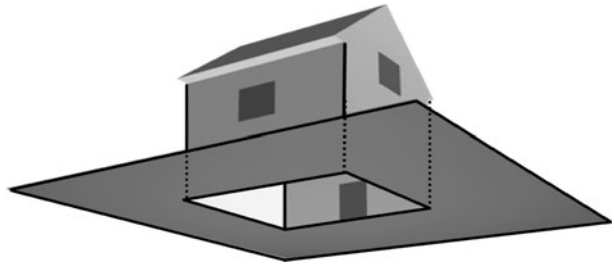
Partially bounded solids in some sense generalize the 2D concept of the unbounded face *Out* to 3D. The aim is the same in both cases: to achieve a complete coverage of the plane/the 3D space.

2.2 Relation to CityGML

The concept of a 3D city model presented in this paper has been particularly designed to represent CityGML data sets [9]. CityGML covers buildings and other constructions as well as the terrain in different Levels of Detail (LoD). The other constructions embrace bridges

¹ See Appendix A for a definition of (connected) cell complexes and 2-manifolds.

Fig. 1 Example of a 2.8D map modelling the outer hull (wall, roof surfaces) of a saddle roof building and the surrounding terrain. The map is depicted from below. The face *Out* (not shown) extends the terrain face to infinity



and tunnels, which currently are represented by generic concepts. A semantic characterization will presumably be provided in the next version 1.1 of CityGML. In LoD1 to LoD3, buildings are geometrically represented either by a single solid or—in the case of distinguishable building parts with different height or semantical attributes—by an aggregation of solids. Each building part is represented by a solid. The outer hull of that aggregation forms again a solid, which represents the building. Hence, buildings in LoD1 to LoD3 can be represented by 3D city models. For LoD4, which is concerned with interior structures like rooms or staircases, there are several modelling options. In the simple case, a building (or a building part) is an aggregation of solid rooms (c.f. Fig. 4a and b). Walls and slabs are represented by surfaces in that case. In more detailed models, walls and slabs are volume objects. Since each room must be accessible from outside and from other rooms, doors or windows must connect rooms to the exterior hull and doors must connect rooms with other rooms. Hence, doors must be represented by solids that are incident to both room solids (c.f. Fig. 4c and d) and windows must be represented by solids that connect room solids and the air solid. Alternatively, doors can be represented by a surface. In that case, both solids have to be enlarged in order to touch. An example is the modified scene in Fig. 4c and d, where s_1 and s_4 (as well as s_2 and s_5) are merged to a single room solid by omitting the rectangular surface separating both solids. In any case, a building where all rooms are connected by doors and staircases can be represented by an aggregation of solids, where each solid is bounded by a single closed composite surface (exterior shell). However, as stated in the last section, some standards (e.g., [21] or [33]) allow for the representation of solids which have interior shells. Such interior shells enclose other solids completely, but there is no connection or access to those solids. An example is depicted in Fig. 4e and f: The enclosing solid s_3 has two interior shells, which coincide with the exterior hulls of s_1 and s_2 . Although LoD4 in principle covers such models, such a representation is not

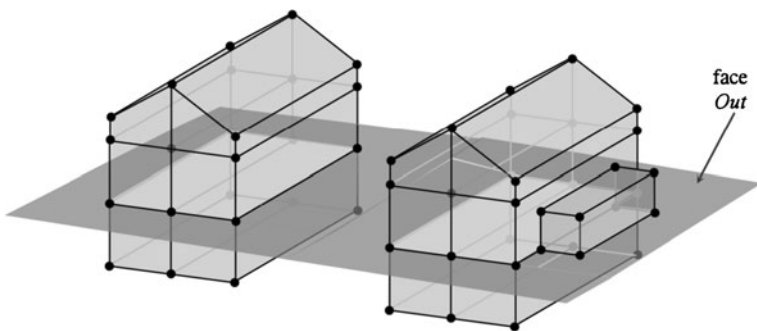


Fig. 2 Simple example for a 3D city model. The left building consists of six bounded solids, the right one of seven. In addition, there is an air solid and an earth solid. Both are partially bounded by 2.8D maps

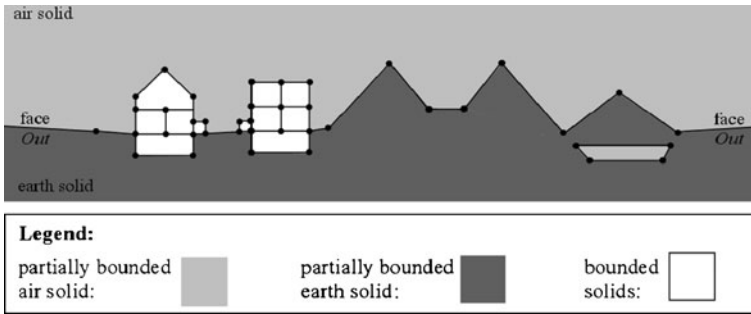


Fig. 3 Sketch of a cross profile of a geometric-topological 3D city model. Each bounded solid is bounded by a single closed composite surface and each of the two partially bounded solids is (partially) bounded by a 2.8D map

adequate since reachability and accessibility, which are crucial for indoor navigation, robot navigation [39] or escape route determination [27] as applications of indoor models, are not supported. Hence, such representations are outside the scope of 3D city models discussed in this paper (solids have exterior shells only but no interior shells). All other LoD 4 models, as well as LoD1–LoD3, can be represented by 3D city models.²

2.3 Axioms for 3D city models

The mathematical definition of 3D city models given in Section 2.1 doesn't provide an effective and efficient method to check consistency. In [15] and [14], we have defined so called *axioms* as an alternative mathematical formalization of 3D city models. In [12] we provide corresponding axioms for 2.8D maps. These axioms are complete (each violation of the definition is detected by the axioms), correct (each violation of the axioms is in fact an error) and efficient. In particular, the test whether two solids penetrate, which is very elaborate and expensive from an algorithmic and running time point of view, is not necessary. This property has already been implied by other axioms which can be checked in a much cheaper way.

The systems of axioms for 3D city models is composed of three parts. First, axioms for 2.8D maps referring to the consistency of a single 2.8D map are defined (see Table 1). Second, axioms for closed composite surfaces are given. These differ slightly from the axioms for 2.8D maps: Axiom S11 is replaced by axiom

S'11: Each face is bounded

since there are no unbounded faces in a closed composite surface. Furthermore, axiom S13 can be omitted for closed composite surfaces since that property is already implied by the other axioms. For details see [15] or [14].³

Third, axioms for 3D city models, i.e., the (space covering) aggregation of solids and their integration into the terrain surface, are defined (see Table 2). The axioms M1, M2 and M3 refer to the axioms for 2.8D maps and closed composite surfaces (Table 1) and induce the application of these axioms to well defined surface components of the 3D city model: to the boundary of the air solid (M1), to the boundary of the earth solid (M2) and to the

² This is true as far as solid models are concerned. Purely areal or line objects (roof overhangs, antennas, for example) currently are not in the scope of the model, see section 6.

³ In [15], surfaces without handles are considered. We have extended the axioms to surfaces with handles in [14].

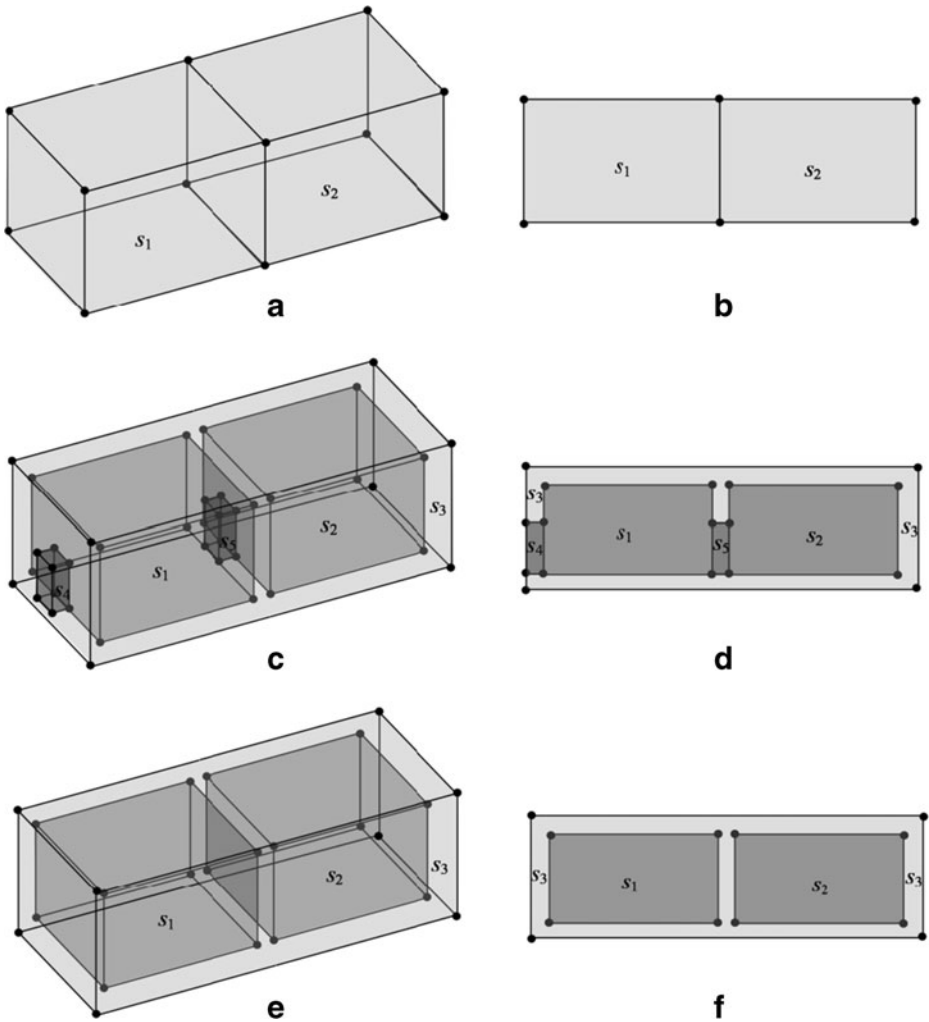


Fig. 4 Representation of interior building structures (LoD4 in CityGML) by 3D city models, case of a building solid composed of two room solids s_1 and s_2 . **a, b** Solid is aggregation of s_1 and s_2 . **c, d** walls and slabs are modeled as solid (s_3), and s_1 and s_2 are connected by doors (solids s_4 and s_5). **e, f** solids s_1 and s_2 are enclaves in solid s_3 and not accessible from outside. **b, d** and **f** provide lateral views of the models in **a, c** and **e**

boundary of each bounded solid (M3). The application of these axioms is restricted to the vertices, edges and faces of the corresponding surface component. Consider axiom S7 (two incident faces for each edge) as an example. The validity of this axiom can be checked with regard to a solid boundary, whereas it is not valid with regard to all faces of a 3D city model (an edge can have more than two incident faces).

The axioms M4 and M5 refer to all vertices, edges and faces of the space covering model.

As consistency notion that underlies the transaction rules, the declarative mathematical definition is as suitable as the equivalent axiomatic characterization. In this paper, both concepts are used.

Table 1 Complete and correct axioms for 2.8D maps [14, 15]

S1	Different vertices have different coordinates.
S2	Each vertex has at least two incident edges (vertex degree ≥ 2).
S3	Each vertex is surrounded by exactly one alternating sequence of edges and faces. These faces do not penetrate pairwise. (Umbrella-Axiom).
S4	Each edge has exactly two distinct vertices as end points.
S5	Edges are straight line segments geometrically.
S6	Edges intersect only at common vertices (intersection-free edges).
S7	Each edge has exactly two distinct incident faces.
S8	Each face is bounded by exactly one simple cycle of edges.
S9	Bounded faces are planar.
S10	No point of an edge touches the interior of a face.
S11	There is exactly one unbounded face ‘OUT’.
S12	The underlying graph is connected.
S13	The map is orientable.

3 Transaction rules

The axioms for 3D city models check consistency of a dataset as a whole. From an efficiency point of view, it is too expensive to completely check the consistency of the a data set once more in total after an update. To guarantee that an update does not violate consistency, we now introduce so called transaction rules. These rules are in some sense a specific version of the axioms, which are tailored for a specific update operation and restrict the checks locally to the region or space that is affected by the update. All rules presented in this section are safe and complete for 2.8D maps and 3D city models. Hence, consistency is preserved and all allowed modifications are realisable. We first describe the structure of the rules.

A transaction rule consists of three parts: *transaction*, *condition*, and *action*. The transaction part is the header of the rule. It describes the operation that the rule performs, the input and the effect of the rule. The condition part plays the role of a guard [36]. It specifies the conditions under which the action part transforms the input of the rule (a consistent data set and a surface to be added or deleted) to a new consistent state. If the input of a rule does not yield a consistent state, the transaction will be rejected by the guard. The action part is a sequence of actions performed in order to fulfill the user’s request. These rules are in a sense similar to ‘Event-Condition-Action-rules’ (ECA-rules: On E if C do A), which make up the main mechanism of active databases [38].

To specify the input of the transaction rules, the concept of a *composite surface* [21] is required. It is similar to a closed composite surface but has one boundary which is a simple

Table 2 Axioms for geometric-topological 3D city models, which are complete and correct

M1	The boundary of the air solid is a 2.8D map.
M2	The boundary of the earth solid is a 2.8D map.
M3	Each bounded solid is bounded by a closed composite surface.
M4	Each face delimits two solids on different sides.
M5	Each vertex and each edge is incident to at least one face.
M6	Besides the air and the earth solid, all solids are bounded.

cycle. Hence, it does not enclose a volume. Unless stated otherwise, a composite surface has exactly one boundary. In this paper, composite surfaces with two or more boundaries are used for dealing with handles. However, each boundary is still a simple cycle, and all boundaries are pairwise disjoint.

3.1 Outline of the rules

Following, an outline of the transaction rules for updating 3D city models is given. For reasons of completeness, we first give rules for updating surfaces. Those rules have been sketched in [13] and will be the topic of a subsequent paper [16]. The focus of the paper at hand is set on rules for updating 3D city models. These rules will be described in detail.

The rules for surfaces are as follows:

- TR1: Initial insertion of a surface
- TR2: Final deletion of a surface
- TR3: Splitting of a face
- TR4: Merging of two faces
- TR5: Changing the geometry of vertices
- TR6: Replacing a composite surface
- TR7: Insertion of a handle
- TR8: Deletion of a handle

The first and the second rule are used only once to create/delete an initial model (a surface that consists of one bounded face only). The rules TR3 and TR4 are used to add or to delete faces. Rule TR5 changes the geometry of vertices (cf. Fig. 5a), and rule TR6 replaces one composite surface by another (typically being more complex) with the same boundary. For example, the footprint polygon of a building (as part of a larger surface) is replaced by a composite surface representing the walls and the roof of the building (cf. Fig. 5b). The rule may be used to add a roof to a building, or to add a balcony, a dormer or a chimney to a roof, for example. Rule TR7 adds a handle to a surface (cf. Fig. 5c). In contrast to TR6, the surface to be inserted has two boundaries and touches the terrain surface in two simple cycles. Hence, two disjoint surfaces are replaced by one single surface with two boundaries. TR8 is the inverse rule of TR7, since it removes a handle from a surface. One surface with two boundaries is replaced by two surfaces with one boundary.

Based on 2.8D maps or surfaces created by TR1 to TR8, 3D city models can be updated by the following rules⁴:

- TR9: Splitting a solid by inserting a composite surface
- TR10: Merging two solids by deleting a composite surface
- TR11: Splitting a solid by inserting multiple composite surfaces with multiple boundaries (changing the number of handles)
- TR12: Merging two solids by deleting multiple composite surfaces with multiple boundaries (changing the number of handles)

⁴ The rules TR11 and TR12 presented in Gröger et al. [13] differ from the version discussed here. There are two reasons for the reformulation: First, the new rules allow for explicit addition or removal of handle objects, which was possible with the former version as well but required multiple rule applications. Besides that semantic reason, the completeness proof (Section 3.3) can be presented in a more concise and elegant way based on the modified version of TR11 and TR12.

Fig. 5 Examples illustrating the transaction Rules **a** changing geometry of vertices (TR5), **b** replacing one surface by another (TR6), **c** insertion/deletion of a Handle (TR7, TR8), **d** splitting/merging of solids (TR9, TR10), **e–g** splitting/merging of solids, deleting/introducing a handle (TR11, TR12)

TR9 inserts a composite surface into a solid, which is part of a 3D city model, and splits the solid into two solids (cf. Fig. 5d). Applications of TR9 cover the generation of a storey by splitting a building's hull into a storey solid and a solid representing the remaining building part. Another application is the generation of a room by splitting a storey solid into a room solid and a solid representing the remaining storey part. Furthermore, a roof, a balcony or a chimney can be added to a building by applying TR9. In that case, the air space solid is split into a balcony/roof/chimney solid and an updated, reduced air space solid.

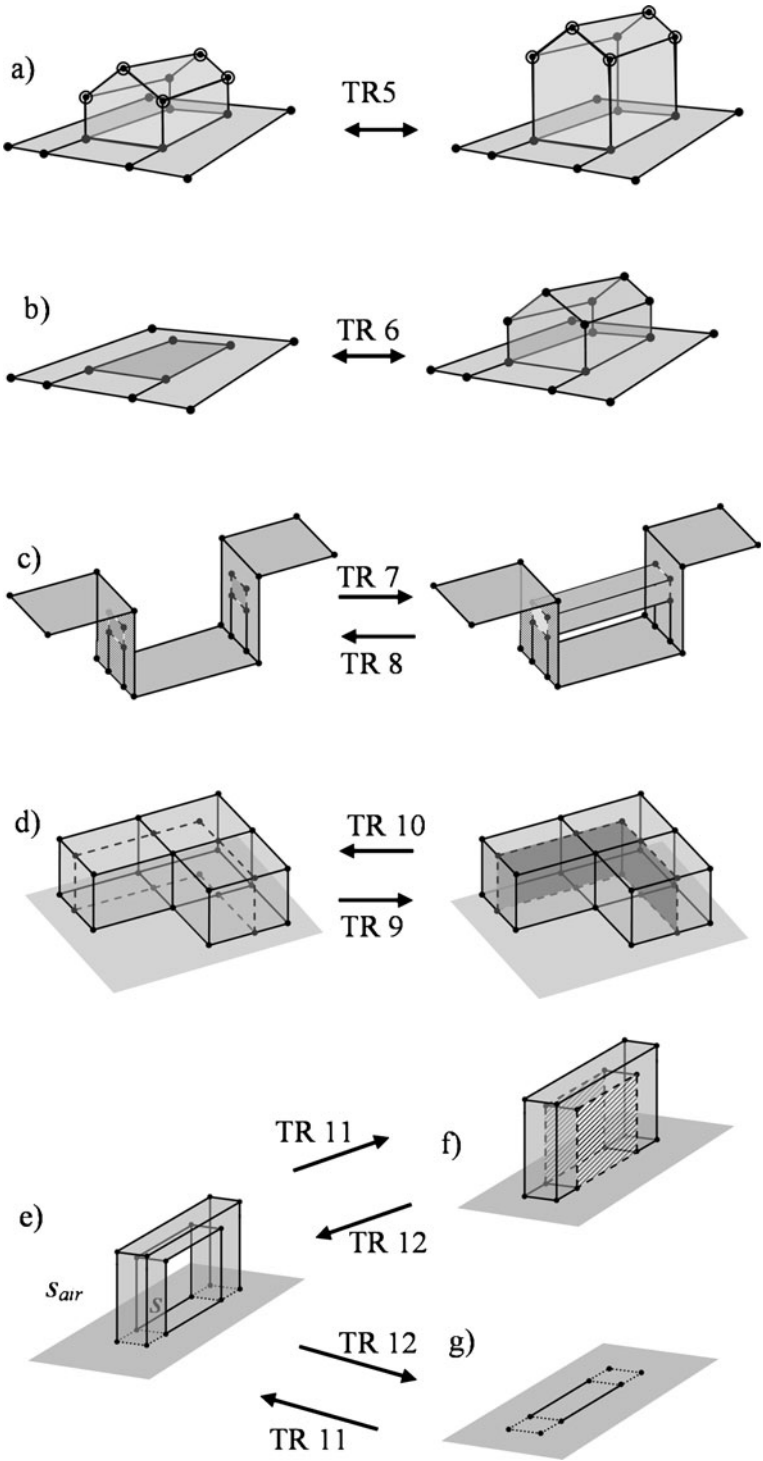
Rule TR10 is the inverse rule of TR9: it merges two neighbouring solids (see Fig. 5d). This rule may be used to remove rooms, storeys, balconies, roofs or chimneys from a 3D city model. The precondition for the application of TR10 is that the two solids share a single composite surface (with one boundary).

The rules TR9 and TR10 cover the task of splitting or merging two solids when both meet in a single composite surface with one boundary. In that case, the number of handles is retained. If the number of handles shall be either reduced or increased, both these rules won't be sufficient. Consider for example the bridge in Fig. 5e. If the bridge as a whole shall be removed, the composite surface representing the outer hull of the bridge has to be removed, and the solid s representing the bridge and the airspace solid s_{air} have to be merged. However, s and s_{air} meet in a composite surface with two boundaries (depicted dotted in Fig. 5e). Hence, TR9 cannot be applied. A similar case is the 'removal' of a handle by sealing the hole. For example, the number of handles in Fig. 5e can be decreased by inserting a surface sealing the hole (c.f. Fig. 5f). If rule TR 8 is applied, only a *single* surface is inserted. The result is that the hole is sealed, but the air solid is not longer bounded by a 2-manifold: The inserted surface delimits the air solid on both sides. What is needed to seal the hole is a rule that inserts two composite surfaces simultaneously.

The new rule that increases the number of handles (either by adding a handle or by opening the holes of a handle) is TR11, and the new rule that decreases the number of handles (either by deleting the handle or by sealing the holes of a handle) is TR12. Both rules deal with surfaces that either have multiple boundaries or consist of multiple disjoint surfaces. Hence, there is a clear distinction whether TR9/TR10 or TR11/TR12 are applicable to split or to merge two solids: if both touch in a *single* composite surface with a *single* boundary, TR9/TR10 are applied. If both touch either in *more than one* composite surface (with an arbitrary number of boundaries) or in a single composite surface with *more than one* boundary, TR11/TR12 are applied.

In Fig. 5, TR11 is used to insert two disjoint surfaces (depicted hatched, hatched boundary) to seal the hole in the bridge, decreasing the number of handles (Fig. 5f based on Fig. 5e). Likewise, TR11 is used to insert a bridge completely (with two boundaries, which are depicted dotted), see Fig. 5e based on Fig. 5g. Inversely, TR12 is employed to open the holes of a bridge (Fig. 5e based on Fig. 5f) or to delete the bridge completely (Fig. 5g based on Fig. 5e).

The examples in Fig. 5e–g illustrate how to change the number of handles by splitting the partially bounded air solid into a bounded solid and an updated air solid, or by merging the air solid and a bounded solid. The rules can also be applied to two bounded solids, for example to remove a pillar inside a room or to insert such a pillar. Illustrations of that case will be given later in Section 3.2.2



In the rules discussed in the rest of this paper, two input structures, touching in one, two or multiple simple cycles, will be considered. According to the well-known *4-intersection-model* [6] and its extension to 3D [40], *touch* means that both meet only in the simple cycle and are disjoint everywhere else. We assume that the corresponding cycles in both structures have been pre-processed and hence are identical with regard to topology and geometry. We further assume that after both have been merged, duplicates are removed and corresponding vertices and edges are connected to yield a cell complex. This mutual adoption of the corresponding cycles can be performed by standard procedures. This topic is not discussed any further in this paper.

By using the rules TR9 to TR12, any arbitrary 3D city model can be constructed, based on the assumption that any surface can be generated by TR1 to TR8. Whereas the rules for surfaces are presented in [16], we now will focus on the rules TR9 to TR12 for 3D city models.

3.2 Safe rules for updating 3D city models

As mentioned in the introduction of Section 3, the update of 3D city models can be reduced to two operations: the splitting of a solid obtaining two solids and its reverse operation, the merging of two neighbouring solids. The next subsection introduces the simple case where one single surface with one boundary is inserted or deleted. The more complex case of inserting or deleting multiple surfaces with multiple boundaries (where the number of handles is changed) will be discussed afterwards.

3.2.1 Splitting and merging of solids (retaining the number of handles)

The rule TR9 splits a solid by the insertion of a single composite surface. The purpose of the rule is versatile: It may be used to split a room into two rooms, to add a dormer, a balcony or a cellar to a building, or to add a building to a terrain model. The applications mentioned last are covered by the rule, since the air space is modelled as a solid. A new bounded solid is split from this solid by inserting a composite surface. This new solid may represent a dormer, a balcony or a building. In the case of a cellar, the earth solid is split into a new bounded solid and an updated earth solid.

The rule is defined as follows:

Transaction Rule TR9: Splitting of a solid by inserting a composite surface

Transaction: Splitting a solid s of a 3D city model m by inserting a composite surface cs

Condition:

1. m is a 3D city model
2. cs is a composite surface with one boundary
3. The boundary of s and cs touch in a simple cycle c
4. At least one vertex, edge or face of cs is in the interior of s
5. c is a Jordan cycle with regard to the boundary of s

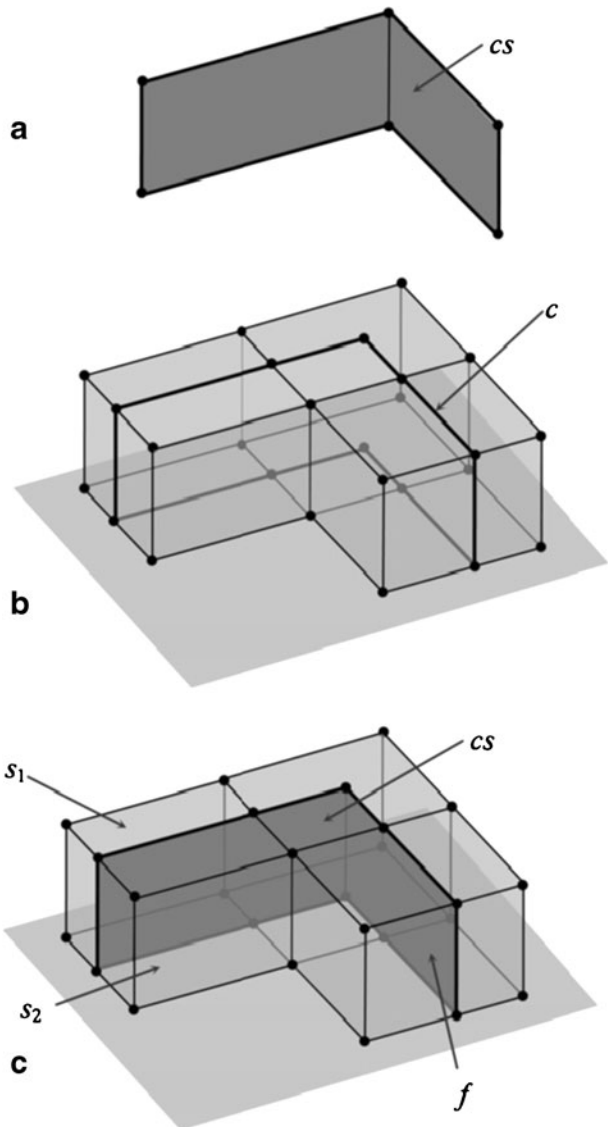
Action:

1. Merge m and cs , yielding m'
2. Create two new solids s_1 and s_2 in m'
3. For all faces in cs , set the reference of one side to s_1 and of the other to s_2
4. c partitions the faces in the boundary of s in two disjoint sets, F_1 and F_2 . Let F_1 be the set on the s_1 -side of cs , and F_2 the set on the s_2 -side. Replace in F_1 the reference to s by the reference to s_1 and in F_2 to s_2

The first condition just states that the rule is based on the assumption that the data set is consistent before the update takes place. The geometry cs to be inserted must be a composite surface with one boundary, which touches the solid to be split in a simple cycle. To avoid that the ‘wrong side’ of the solid is split, it is stated explicitly that cs (at least one vertex of cs) must be inside the solid. Condition 5 will be reasoned and explained later on after the rule has been illustrated by giving two examples. The action part of the rule splits the solid and restores topology.

Figure 6 depicts an example for the application of the rule. The solid s in b) is split into two solids s_1 and s_2 (c) by inserting the composite surface cs (a). The boundaries of s and cs touch in a simple cycle c (cond. 3). Since a face f of cs is inside s (cond. 4), the whole surface cs is inside s . In the action part, the references of faces to s_1 and s_2 are set accordingly.

Fig. 6 Illustration of TR9: Splitting a solid (b) by inserting a composite surface cs (dark grey, a), yielding solids s_1 and s_2 (c)



Note that *touch* means that cs and the boundary of s meet only in a simple cycle and are disjoint everywhere else.

Figure 7 depicts another example for the application of TR9: A balcony is added to a box-shaped building. The balcony, represented by a composite surface cs with one boundary, touches the building in a simple cycle c (depicted hatched). The solid s to be split is the partially bounded air solid. It is split into the balcony solid s_1 and an updated air solid s_2 .

Note that Cond. 4 of TR9 is necessary: a composite surface can touch the boundary of a solid in a simple cycle without being located inside the solid. As a consequence, the wrong solid is split. This is prevented by Cond. 4.

To recognize why Cond. 5 of the rule TR9 is required, regard the example in Fig. 8. The solid s in a) (which may represent a room with a large pillar in the middle) should be split by inserting a composite surface cs . This surface touches the boundary of s in a simple cycle c (depicted hatched) and cs is inside s . Hence, the preconditions 1 to 4 of the rule TR9 are fulfilled. However, cs does not split s into two solids (see Fig. 8b). Instead, a singularity is introduced, since cs delimits the solid s on both sides (s has no longer a 2-manifold boundary). The problem stems from the fact that the boundary of solid s has a handle (s is topologically equivalent to a torus) and that the cycle c is involved in that handle. On such surfaces, Jordan's Curve Theorem [20], which implies the partitioning of a surface into two parts by cutting out a simple cycle, is not valid. This problem has recently been identified and solved in [14] by differentiating between *Jordan cycles* (see cycle c in Fig. 6 as an example) and *handle cycles* (cycle c in Fig. 8b). Jordan cycles partition a surface and are suitable for splitting solids, whereas handle cycles are not. An efficient method of employing standard graph search algorithms for differentiating between both types of cycles is also introduced in [14]. This method can be used to implement Cond. 5 in TR9 preventing error cases that result from splitting solids along a handle cycle.

Another error case detected by the precondition of the rule TR9 is given in Fig. 9. In contrast to the error case discussed in the last paragraph, the solid s to be split is the air solid. The surface to be inserted touches the boundary of s in a simple cycle. Due to the tunnel, a non-manifold surface boundary of the air solid would be created. This is detected by Cond. 5 of TR9, since c is a handle cycle.

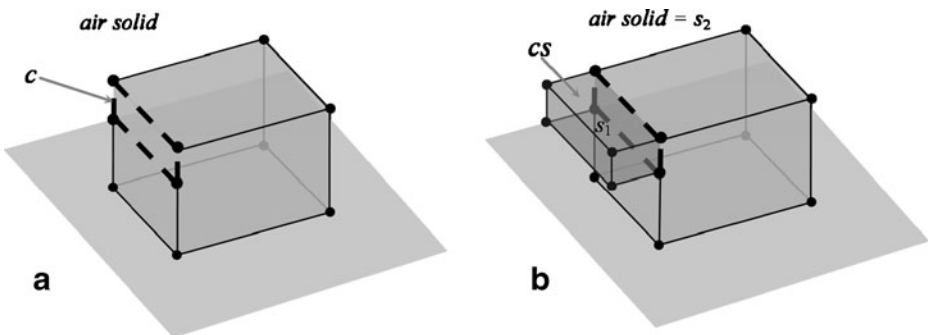


Fig. 7 Example for the application of TR9: Adding a balcony to a building by inserting a composite surface cs . The partially bounded air solid is split by cs in a bounded solid s_1 and an updated air solid s_2

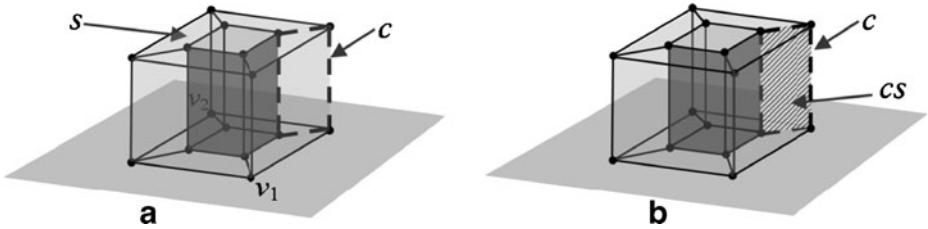


Fig. 8 Error case when inserting a composite surface cs by TR9. **a** initial situation: a box-shaped pillar (depicted *dark grey*) inside a box-shaped solid s . This solid surrounding the pillar is a torus. **b** error case: a surface cs (hatched, hatched boundary) has been inserted into solid s (touching the boundary of s in the hatched cycle c), but s is not split by cs into two solids. cs delimits the same solid on both sides. Hence, the boundary of the solid is not 2-manifold

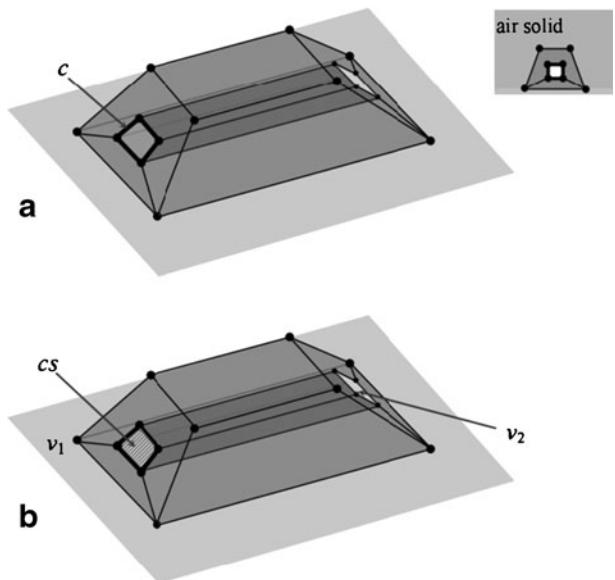
We now show that each application of TR9 preserves consistency:

Proposition 1 Transaction Rule TR9 is safe.

Proof First we have to show that in Action 4 the references to s_1 and s_2 can uniquely be determined. The surface cs and the boundary of s touch in a simple cycle c (Cond 3), c is a Jordan-Cycle with regard to the boundary of s (Cond. 5), and cs is inside s (Cond. 4). Hence, c partitions the boundary of s in two disjoint parts. The solid reference of the faces in the first part is set to s_1 , and the solid reference of the faces in the second is set to s_2 .

Now we have to show that Axioms M1 to M6 are valid for m' under the precondition that these axioms are valid for m . This is sufficient, due to the completeness of the axioms [15]. Axioms M5 and M6 cannot be violated by the action part of the rule and therefore are valid. Since the inserted surface cs is completely inside the solid s (this is implied by Cond. 3 and 4), we can restrict our check to the space covered by s and the

Fig. 9 Error case: Insertion of the surface cs (hatched) in the model in (a) yields a solid (air solid) with a non 2-manifold boundary (b)



boundary of s , regardless of s being a partially bounded air or earth solid or a bounded solid. Since s_1 and s_2 are new solids, and since all faces in cs delimit s_1 on one and s_2 on the other side, axiom M4 is valid as well. To check axioms M1 to M3, we have to consider the axioms S1 to S13 for s_1 and s_2 . Whether the axioms for 2.8D maps or for closed composite surfaces are relevant, depends on s : if s is bounded, so will be s_1 and s_2 . If s is only partially bounded, one of s_1 and s_2 will be also partially bounded, and the other will be bounded. The axioms S1, S2, S4, S5, S9, S11, S'11 and S13 cannot be violated by the transaction. The validity of the other axioms remains to be proven:

- Axiom S3: Since cs and the boundary of s touch in a simple cycle, there is no touch in a single vertex, hence the axiom is valid.
- Axiom S8: This property is inherited by validity of the corresponding axiom for cs and the boundary of s .
- Axioms S6 and S10: By definition, *touch* means that cs and the boundary of s do not penetrate.
- Axiom S12: since cs and the boundary of s touch, both boundaries of s_1 and s_2 are connected.
- Axiom S7: For the edges in s_1 and s_2 which are either in the interior of cs or which have two incident faces with references to s in m , validity of Axiom S7 is inherited from the validity of the axioms for s and for cs . The remaining edges in s_1 or s_2 are in c . By construction, c partitions the faces in the boundary of s in two parts, one with reference to s_1 and one with reference to s_2 . Hence, each edge in c has two incident faces. ■

The next rule TR10 is the inverse rule of TR9. It merges two solids by deleting a composite surface which separates both solids:

Transaction Rule TR10: Merging two Solids

Transaction: Merging two solids s_1 and s_2 of a 3D city model m by deleting a composite surface cs

Condition:

1. m is a 3D city model.
2. cs is a composite surface with one boundary
3. the structure where s_1 and s_2 touch is cs : $cs = \text{boundary}(s_1) \cap \text{boundary}(s_2)$
4. s_1 or s_2 is a bounded solid

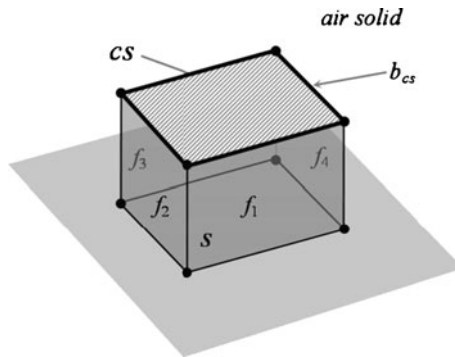
Action:

1. delete all faces, edges and vertices in cs besides the one in the boundary of cs , yielding a model m'
2. replace in m' all references to s_1 or s_2 by references to s

To illustrate the rule, the example in Fig. 6 is used again. The composite surface cs in c) is deleted, and both solids s_1 and s_2 are merged (b). The balcony in Fig. 7b can be deleted by TR10 as well. The balcony solid and the air solid are separated by the composite surface cs (b). After deletion of cs , the air solid has been extended accordingly.

An error case for TR10 is depicted in Fig. 10. The air solid and a solid s are separated by a composite surface cs (hatched). After deletion of cs , the four faces f_1, f_2, f_3 and f_4 remain. The air solid has a non manifold boundary. This error is prevented by Cond. 3 of TR10, since the

Fig. 10 Error case when removing a composite surface cs (depicted hatched) merging two solids (s and the air solid). The resulting modified air solid has a non-manifold boundary. This error case is prevented by Cond. 3 of TR10



air solid and s touch in five faces, not only in cs which has been deleted. Hence, $cs \neq boundary(s_1) \cap boundary(s_2)$ is valid.

A similar error case for TR10 is given in Fig. 11. Solid s_2 and the air solid should be merged by deleting the composite surface cs with boundary b_{cs} . The result is a non-manifold boundary of the air solid, due to edge e . Since $cs \cup \{e\} = boundary(s_1) \cap boundary(s_2)$ holds, $cs \neq boundary(s_1) \cap boundary(s_2)$ is implied: the intersection of both boundaries is a non-manifold structure, due to edge e .

The following proposition states that TR10 preserves consistency:

Proposition 2 Transaction Rule TR10 is safe.

Proof We have to show that the structure m' , generated by the action part of the rule, is a 3D city model. We can safely restrict our considerations to the solids, faces, edges and vertices in the boundary and interior of s_1 and s_2 .

To check axioms M1 to M3, we have to show that Axioms S1 to S12 are valid for s_1 and for s_2 . Axioms S1, S5, S6, S8, S9, S10 cannot be violated by the transaction and are therefore valid.

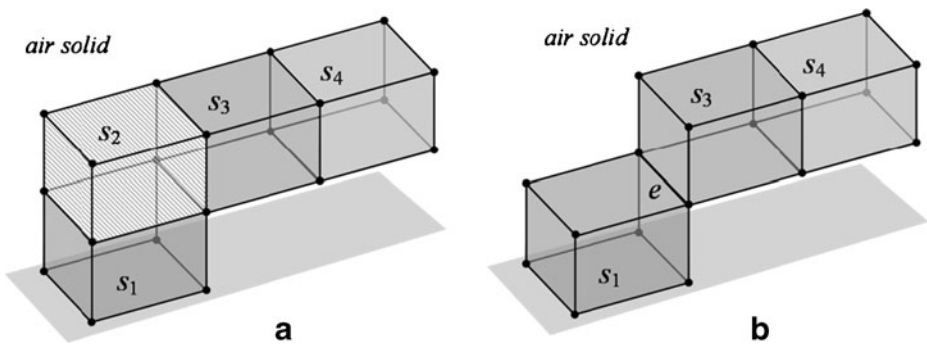


Fig. 11 Error case when deleting a composite surface (depicted hatched) in **a** the air solid has a non-manifold boundary **b** Note that solid s_2 and the air solid don't touch in a composite surface

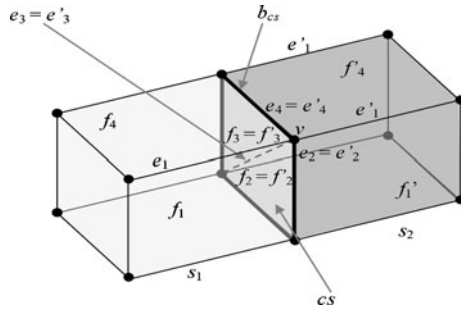
- S2 (vertex has two edges): edges and vertices in the boundary of cs are not deleted by construction.
- S3 (umbrella axiom): We can restrict the proof to vertices v in m' which are in the boundary b_{cs} of cs . Other vertices are not affected by the transaction. Let the umbrella of v in m be $u_1=e_1 f_1 e_2 f_2 e_3 f_3 \dots e_1$ for the boundary of s_1 and $u_2=e_1' f_1' e_2' f_2' e_3' f_3' \dots e_1'$ for the boundary of s_2 (see Fig. 12 for example). We assume that both are consistently oriented, i.e. that common sequences of vertices and faces have identical order. Since s_1 and s_2 touch in a single composite surface cs with a single, simple cycle as boundary (Cond. 2), a maximal subsequence u_1' of u_1 and a subsequence u_2' of u_2 exist and therefore $u_1'=u_2'$. The umbrellas are disjoint everywhere else. Let $u_1'=e_1 \dots e_j$ and $u_2'=e'_k \dots e'_l$. Then the edges $e_i=e'_k$ and $e_j=e'_l$ are on the boundary b_{cs} of cs . In m' , both sequences u_1 and u_2 are merged and u_1' and u_2' (without the edges on the boundary of cs) are deleted. The corresponding umbrella of v for s is $e_1 f_1 e_2 f_2 \dots (e_i=e'_k) f'_{k+1} e'_{k+1} \dots e'_{l-1} f'_{l-1} (e_j=e'_l) f'_{j+1} \dots e_1$, fulfilling axiom S3.
- S4 (edge has two vertices): Suppose there is an edge e which does not fulfill axiom S4. Then one endpoint v_e of e must be in the interior of cs , but e is not in the interior or boundary of cs . Hence, e must be in the boundary of one of the solids s_1 or s_2 , say s_1 , and e must be in the boundary of another solid s_3 , $s_3 \neq s_1$ and $s_3 \neq s_2$ (if e is in the boundary of s_1 and s_2 , it would have been deleted). Hence, v is incident to three solids: s_1 , s_2 and s_3 . There must be an edge e' incident to v , which is in the interior of cs (otherwise, s_1 and s_3 would touch in v in a non-manifold way). There must also be a face incident to e' which is in the interior of cs and which is in the boundary of s_2 and of s_3 (otherwise, s_1 and s_3 would touch in e' in a non-manifold way). But in that case, e' is in the boundary of cs . This contradicts the assumption that v is not in the boundary of cs .
- S7 (edge has two different faces): Suppose there exists an edge e in m' which has only one incident face f (By deletion of faces, the number of incident faces cannot increase. Hence, we safely can restrict our considerations to one incident face). Since the axioms were valid in m , there is a face f' which is incident to e , which was incident to s_1 and s_2 in m and which has been deleted by the transaction. Hence, f is also in the common boundary of s_1 and s_2 . But in that case, f must have been deleted by the transaction rule, which is a contradiction.
- S11 (one unbounded face): Since a composite surface is bounded by definition, an unbounded face cannot be deleted by the transaction rule.
- S12 (connectedness): The proof of connectedness of m' is similar to the proof of axiom S4. Suppose m' is not connected. The composite surface cs has only one boundary which is connected. Hence, there must be an edge e , which is not in the interior of cs , but which has an endpoint in the interior of cs (which is deleted by the transaction). In the proof of validity of S4 it has been shown that such an edge e cannot exist.

The proof of axiom M4 is similar to the proof of S7: If a face f delimits two identical solids, then this solid must be s . Therefore, f delimits s_1 and s_2 in m . But then f must have been deleted by the transaction, yielding a contradiction. M5 cannot be violated by TR10. Axiom M6 is valid, since m' still contains two partially bounded solids (Cond. 4). Only if both s_1 and s_2 are partially bounded, the number of partially bounded faces will decrease. ■

3.2.2 Splitting and merging of solids (changing the number of handles)

The transaction rules, dealt with in the last section, generate or modify models where the number of handles is not changed. This is due to the fact that a single composite

Fig. 12 Illustration of the proof of axiom S3 (Umbrella-Axiom) in Proposition 2. Vertex v has one umbrella for s_1 ($e_1 f_1 e_2 f_2 e_3 f_3 e_4 f_4 e_1$) and one for s_2 ($e'_1 f'_1 e'_2 f'_2 e'_3 f'_3 e'_4 f'_4 e'_1$). After merging s_1 and s_2 to s , v has a single umbrella ($e_1 f_1 e_2 f_1 e'_1 f_4 e_4 f_4 e_1$) for s . The touching composite surface and the edge e_3 in cs are shown hatched



surface with a single boundary is considered in the rules. For changing the number of handles:

- *multiple disjoint surfaces* have to be deleted or inserted simultaneously, or
- one surface with *multiple boundaries* has to be deleted or inserted.

An example for the first case has already been presented in Fig. 5e and f, whereas in e) and g) the second case has been illustrated.

Figure 13 shows another example: a pillar is added to a room s (a) by inserting a composite surface cs with two boundaries, b_1 and b_2 . Solid s is split in one solid s_2 representing the pillar, and an updated room solid s_1 (b). In Fig. 14a and b, a connecting bridge between two simple buildings is added by inserting a composite surface with two boundaries (depicted hatched). In that case, the air solid is split into a new solid s_3 and an updated air solid. An important precondition is that the two solids to be merged touch only in composite surfaces with simple cycles as boundaries.

There are complex cases where multiple surfaces, each having multiple boundaries, have to be deleted or inserted. Suppose, for example, having a room with multiple pillars. If each pillar is represented by its own solid, each one will be removable by subsequently applying the rule to merge two solids (by removing a single surface with two boundaries). However,

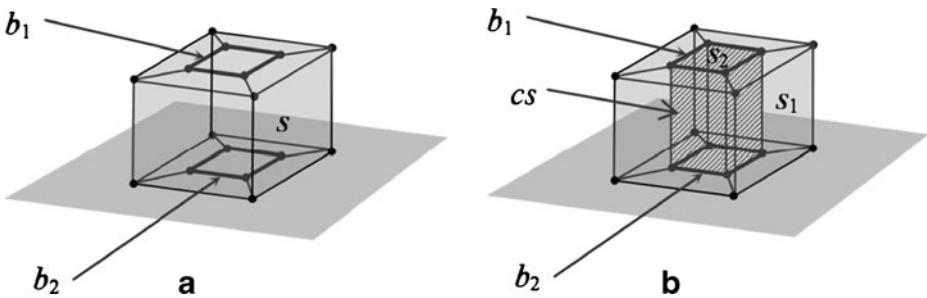


Fig. 13 Insertion of a pillar into a room: box-shaped room/solid s **a** is split by inserting the composite surface cs (depicted hatched) with two boundaries b_1 and b_2 (thick lines) into s **b** The boundaries b_1 and b_2 are depicted as *thick lines*

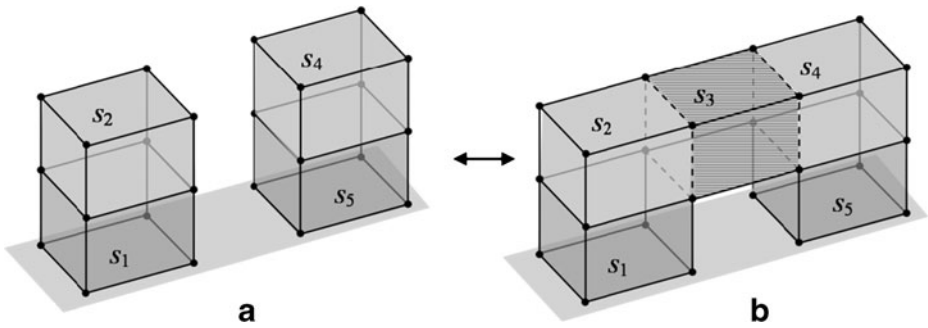


Fig. 14 Example for the insertion of a composite surface (*light grey, hatched boundaries*), splitting the air solid and introducing a new solid s_3

consider the case where all pillars are represented by the same solid by connecting all pillars with structures above and below the room. In that case, a single pillar cannot be removed. Hence, all pillars must be removed simultaneously, by a rule which deletes multiple surfaces with multiple boundaries each. Our rules and the correctness and completeness results cover that case.

The transaction rule for splitting a solid (changing the number of handles) is the following:

Transaction Rule TR11: Splitting of a solid (reducing/increasing the number of handles)

Transaction: Splitting a solid s of a 3D city model m in two solids s_1 and s_2 by inserting multiple composite surfaces cs_1, \dots, cs_k , where each cs_i may have multiple boundaries

Condition:

1. m is a 3D city model.
2. $cs_i, 1 \leq i \leq k$, are composite surfaces which are pairwise disjoint
3. Each cs_i and the boundary of s touch in a simple cycle.
4. At least one vertex, edge or face of each cs_i is in the interior of s
5. For each cs_i , each of both sides can be assigned uniquely a s_1/s_2 -orientation, such that each s_1 (or s_2)-side of a cs_i , can be connected with each other s_1 (or s_2)-side of a cs_i , by the use of paths. These paths do not cross any cs_i
6. All boundaries of all cs_i are Jordan cycles with regard to $boundary(s) \cup cs_1 \cup \dots \cup cs_k$.

Action:

1. Merge m and cs_1, \dots, cs_k yielding m'
2. Create two new solids s_1 and s_2 in m'
3. Assign the references of faces in all cs_i to s_1 or s_2 , according to the orientation in Cond. 5.
4. The cs_i partition the faces in the boundary of s in two disjoint sets, F_1 and F_2 . Let F_1 be the set on the s_1 -side of the cs_i , and F_2 the set on the s_2 -side. Replace in F_1 the reference to s by the reference to s_1 and in F_2 to s_2 .

The rule is a generalization of TR9: Instead of dealing with a single composite surface cs , multiple surfaces cs_i , with multiple boundaries each are considered. The touching of cs

and the boundaries of s in a Jordan cycle is not sufficient for defining two new solids. Instead, it has to be explicitly assured that each of the two solids is connected (Cond. 3) and that both are separated (Cond. 6).

The following proposition states that the result of applying TR11 to a 3D city model is a 3D city model again:

Proposition 3 Transaction Rule TR11 is safe.

Proof The proof is a generalization of the proof for Proposition 1. Again we first have to show that the surfaces cs_i partition s in two disjoint parts, s_1 and s_2 . Due to Cond. 4, all cs_i are inside s . According to Cond. 6, all cycles in the boundaries of all cs_i are Jordan-cycles, i.e. each cs_i separates the space of s in at least two solids. Because of Cond. 5, there are exactly two solids, s_1 and s_2 , each of which is connected.

The proof of the validity of the axioms M1 to M6 and S1 to S13 can be conducted analogously to the proof of Proposition 1. All propositions related to a single composite surface are still valid for multiple, disjoint surfaces with multiple boundaries. ■

The last rule presented in this paper, TR12, is the inverse rule of TR11. It merges two solids, changing the number of handles:

Transaction Rule TR12: Merging two solids, reducing/increasing the number of handles

Transaction: Merging two solids s_1 and s_2 of a 3D city model m by deleting multiple composite surfaces cs_1, \dots, cs_k , where each cs_i may have multiple boundaries

Condition:

1. m is a 3D city model.
2. $cs_i, 1 \leq i \leq k$, are composite surfaces with multiple boundaries which are disjoint pairwise
3. The structure where s_1 and s_2 touch is the union of all cs_i : $cs_1 \cup \dots \cup cs_k = \text{boundary}(s_1) \cap \text{boundary}(s_2)$
4. s_1 or s_2 are bounded solids
5. The structure remains connected after deleting $cs_1 \cup \dots \cup cs_k$

Action:

1. Delete all faces, edges and vertices in each cs_i besides the one in the boundary of cs_i .
2. Replace all references of faces to s_1 or s_2 by references to s

To illustrate TR12, all examples for rule TR11 (splitting a solid) discussed in that section are suitable. In Fig. 13b, the composite surface cs with two boundaries (depicted hatched) is deleted and solids s_1 and s_2 are merged to a new solid s , representing a room without pillar. In Fig. 14b, a composite surface with two boundaries (depicted hatched) is deleted. The air solid and the solid s_3 are merged to an updated air solid (a).

An error case for TR12 is illustrated in Fig. 15: If in a) the composite surface with two boundaries (delimiting solid s_3) is deleted, the resulting model (b) is not consistent. This is due to the fact that the model is not connected. To be more precise, the 2.8D map partially delimiting the air solid is not connected, violating axiom S12 for that 2.8D map. This error is prevented by Cond. 5 of TR12.

Again we proof that consistency is maintained by rule TR12:

Proposition 4 Transaction Rule TR12 is safe.

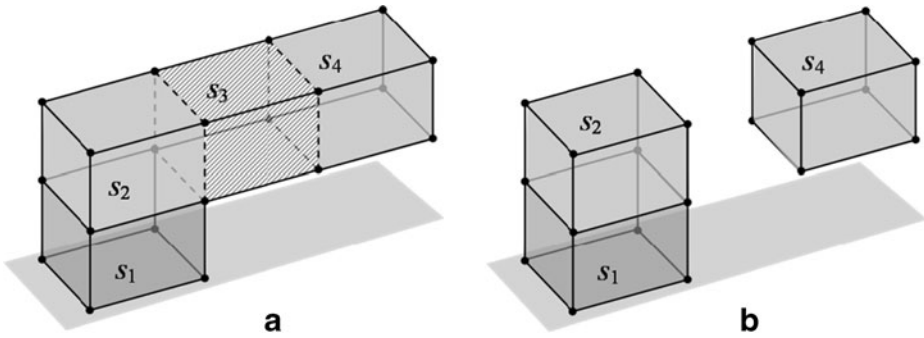


Fig. 15 Error case for TR12: Deleting the composite surface with two boundaries (depicted hatched) destroys connectivity (violation of axiom S13, detected and prevented by Cond. 5 of TR12)

Proof This proof is a generalization of the proof of Proposition 2 and can be conducted analogously. All propositions related to a single composite surface are still valid for multiple, disjoint surfaces with multiple boundaries, only the proof of connectedness (S11) depends on the connectedness of the boundary. But this property of the model is explicitly stated in Cond. 5. ■

3.3 Completeness

We now prove that the rules TR9 to TR12 presented in the previous sections are complete, i.e. that each arbitrary 3D city model can be constructed by subsequent rule applications. We assume the completeness of the rules TR1 to TR8 for composite surfaces and 2.8D maps, which will be the topic of a subsequent paper.

The completeness proof for 3D city models proceeds by induction on the number of bounded solids. We will show that in each arbitrary 3D city model m , two solids can be merged by applying rule TR10 or TR12. By inductive assumption, the claim holds for the resulting model m' , and by applying the corresponding inverse transaction rule, it also holds for m . This proof is based on the assumption that in each 3D city model there exists at least one pair of solids which can be merged by TR10 or TR12, and accordingly be split by the corresponding inverse rules. This is the case in typical models as they occur in practice. However, pathological models without that property do exist: the merging of each pair of neighbouring solids is blocked by a non-manifold structure shared by both of them. An example for such a shared structure between only two solids has already been given in Fig. 11a, where merging s_2 and the air solid yields a non-manifold solid boundary. Dealing with such pathological models is the crucial point in proving completeness of the transaction rules.

For an impression of those pathological models where each merging of two solids is blocked we describe briefly how to construct them: for each pair of solids touching in at least one face (and hence being potential candidates for merging), an additional touch in a single vertex or a single edge is introduced, by adding corresponding faces to the solid boundary. This touch causes a non-manifold solid boundary of the merged solid and hence the precondition of rule TR10 or TR12 prevents the application of the rules. This touch in a vertex or edge mustn't violate the requirement of each solid

boundary having to be a 2-manifold. Hence, the touch must be surrounded or ‘masked’ by two other solids. Therefore, such a non-manifold edge or vertex must be incident to at least four solids: two which are prevented from merging to two others to mask the non-manifold touching. Figure 16 depicts a model where both merging s_1 and the earth solid as well as merging s_2 and the earth solid are being blocked by a prism which touches the earth solid in a vertex (depicted as large dot). Symmetrically, the merging of s_1/s_2 with the air solid and the merging of s_1 with s_2 can be blocked by adding corresponding prisms to solids.

The first step in proving completeness is to show that mutual blocking of merging solids can be released by splitting and merging of faces, preserving the number of solids in the model:

Proposition 5 Let s_1 and s_2 be two solids in a 3D city model m , which touch in at least one composite surface. If s_1 and s_2 touch in a non-manifold way, i.e. if both touch in a graph $g_{1,2}$ which contains vertices and/or edges that are not part of a composite surface of $g_{1,2}$, then m can be transformed to a model m' with the same number of solids, where two solids s_1' and s_2' touch in composite surfaces only, i.e. in a manifold way.

Proof The main idea of the proof is to split one of both solids in such a way that a non-manifold touch doesn’t exist any longer. The parts which are split are infinitesimally small, even small enough to eliminate the touching in vertices or edges. The resulting small solids are merged with another adjacent solid, not causing any additional non-manifold touch due to the small size of the solids.

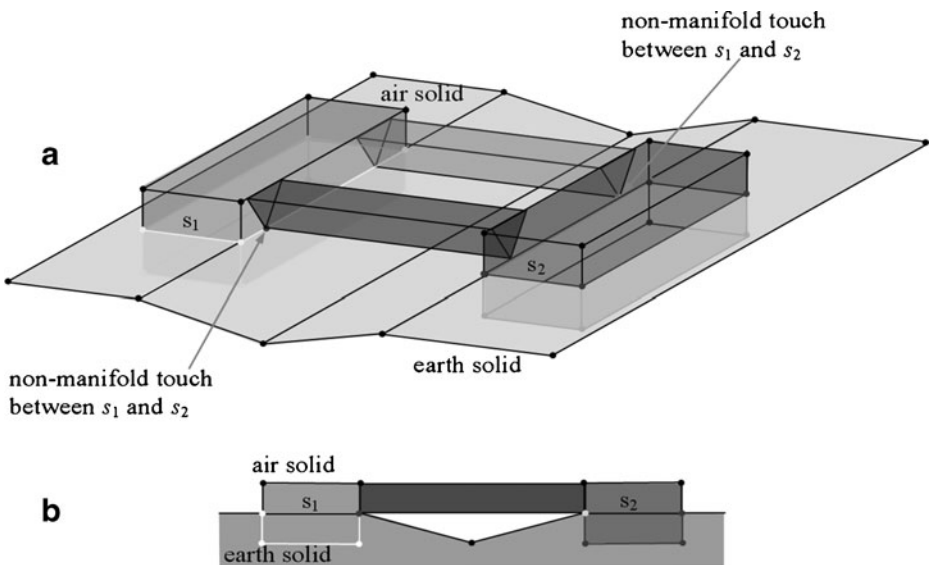


Fig. 16 3D city model where merging solid s_1 or s_2 with the earth solid is blocked by an additional touch in a vertex, introduced by adding a prism. The merging of a pair of solids will not be possible if blocking prisms between s_1 or s_2 and the earth solid as well as between s_1 and s_2 are added analogously. **a** Oblique view on the model, **b** Frontal view on the model

To be more precise, let $g_{1,2}$ consist of $d > 0$ connected components cc_1, \dots, cc_d . Each cc_i is composed of at most one composite surface (perhaps with multiple boundaries), and additional vertices and edges which are not part of a composite surface. Those sets are denoted by V_i and E_i , $1 \leq i \leq d$. Note that the edges in E_i can form branching graphs or even cycles. We consider each cc_i separately. The following steps are performed for each cc_i , $1 \leq i \leq d$:

We construct an infinitesimal small buffer around the vertices and edges in V_i and E_i on the boundary of s_1 . From the edges and faces forming that buffer, we construct a new solid s' by generating new faces and closing the solid. The volume of s' is infinitesimal small. Now s_2 is split by TR9 or by TR11 into s' and the remaining part s_2' of s_2 . TR11 is used in the case that the edges in E_i form a cycle, TR9 otherwise. In Fig. 17 two examples are depicted. A non-manifold touch in a vertex in V_i is treated by splitting solid s_2 in s and s_2' by inserting a triangle. In b), a rectangle separates an edge from E_i , splitting a solid in s to s_2' . The next step is to merge s' with one of the solids which on one hand is incident to an edge in E_i or a vertex in V_i , and on the other hand is different from s_1 or from s_2 . Such a solid must exist, since otherwise a solid would have been bounded by a non manifold structure (see discussion above). The resulting structure is a 3D city model (it is the result or the application of safe transaction rules to 3D city models) which has the same number of solids as the initial model and contains a pair (s_1, s_2') of solids touching in composite surfaces only.

Since any non-manifold touch is eliminated by splitting s_1 , the new solid s_1' touches other solids in composite surfaces only. The number of solids remains the same, since any new solid resulting from splitting has been merged with another adjacent solid. ■

By means of Proposition 5 we now are able to proof that each 3D city model contains a pair of solids which can be merged safely:

Proposition 6 In each 3D city model m with $n+1$ bounded solids ($n > 0$), there exists a pair (s_1, s_2) of solids which can be merged by application of the transaction rules, yielding a 3D city model m' with n bounded solids.

Proof Let s_1 and s_2 be two arbitrary solids in m which touch in at least one composite surface. Such a pair must exist since each solid is bounded by a composite surface and the

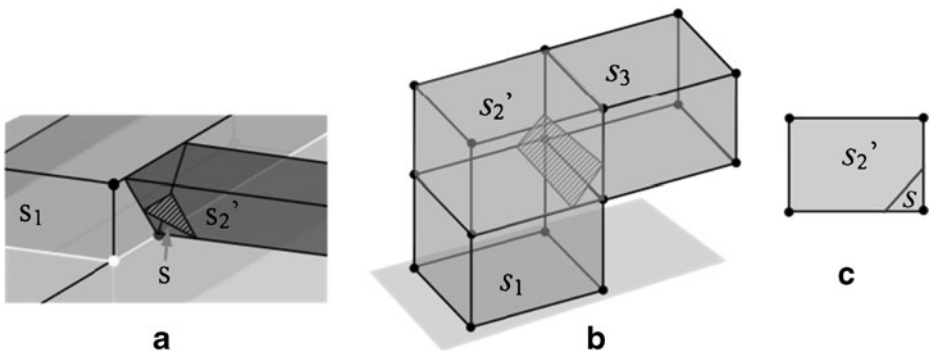


Fig. 17 Illustration of healing a non-manifold touch in the proof of Proposition 5. **a** non-manifold touch in a vertex, not connected to the composite surface (Figure is an amplification of Fig. 16a) **b** non-manifold touch in an edge, connected to composite surface. **c** front view of two solids in (b)

space is completely covered by solids. s_1 and s_2 touch in a graph $g_{1,2}$. We have to consider all possible cases for $g_{1,2}$, reflecting all possible touching patterns of two solids s_1 and s_2 :

Case 1: $g_{1,2}$ is exactly one composite surface with a single boundary.

In that case the precondition of TR10 is satisfied: s_1 and s_2 can be merged, yielding a 3D city model with n bounded solids.

Case 2: $g_{1,2}$ is the union of k composite surfaces, each having multiple boundaries.

We have to consider two subsidiary cases since rule TR12 can only be applied if the resulting model remains connected after merging both solids:

Case 2.1: After merging s_1 and s_2 the model remains connected.

The preconditions of TR12 are fulfilled. Therefore, s_1 and s_2 can be merged yielding a model with n bounded solids.

Case 2.2: The model is not connected after merging s_1 and s_2 .

This case is more complicated, since the precondition of TR12 is not satisfied by s_1 and s_2 and hence they cannot be merged. Suppose the composite surfaces in $g_{1,2}$ are deleted virtually. Then the 3D city model is separated in at least two parts, say p_1, \dots, p_t ($t > 1$). To illustrate this separation, we refer to Fig. 15 where the deletion of s_3 yields a separation in two parts: one containing s_4 and the other s_1, s_3 and the terrain surface. At least one part p_i must be isolated by construction and be completely inside one of both solids, say s_1 (in Fig. 15, p_i is the solid s_4 and s_1 the air solid). Choose one arbitrary solid s_{pi} in p_i and assume that s_{pi} is the new solid to be merged with s_1 , i.e. define $s_2 := s_{pi}$. Consider again the three cases 1, 2, or 3 for the new pair (s_1, s_2) , and repeat that process if Case 2.2 applies again. Since the number of solids in p_i strictly decreases in each step, due to the partitioning of the model, this process must terminate. Hence, either Case 1 or Case 2.1 must apply in the end.

Case 3: $g_{1,2}$ contains edges and/or vertices which are not part of a composite surface with multiple boundaries (non-manifold touching).

According to Proposition 5, m can be transformed to a model m' with two solids s_1' and s_2' which touch in a manifold way, i.e. in a graph $g_{1,2}'$ containing only composite surfaces. Hence, the preconditions of case 1 or case 2 are met and s_1' and s_2' are considered in one of the two cases (depending on the number of composite surfaces and their boundaries). Note that the number of solids does not change if Case 3 is applied.

In any case, the process terminates with either case 1 or case 2.1. In both cases, we get a 3D city model with n bounded solids. Since all possible cases of touching patterns of two solids are considered, the proposition is proven. ■

Now all prerequisites for the completeness proof are available:

Proposition 7 The transaction rules TR9 to TR12 are complete for 3D city models.

Proof We have to show that arbitrary city models can be generated by application of the rules TR9 to TR12. The proof proceeds by induction on the number n of bounded solids. For the case $n = 1$, the bounded solid s can be merged with either the air or the earth solid

according to Proposition 6. The city model is reduced to a single 2.8D map, which can be generated according to the completeness of rules TR1 to TR8 for surfaces. By applying the inverse transaction rules to that 2.8D map, the model with one bounded solid s can be generated.

As the inductive hypothesis we assume that the proposition holds for city models with n bounded solids. Let m_{n+1} be an arbitrary city model with $n+1$ bounded solids. According to Proposition 6, there exists a pair (s_1, s_2) of bounded solids in m_{n+1} which can be merged by applications of the transaction rules. This yields a 3D city model m_n' with n bounded solids. By the inductive hypothesis, the proposition is valid for m_n' . By applying the sequence of transaction rules, which are inverse to the one which transformed m_{n+1} to m_n' , in reverse order, we obtain m_{n+1} . Hence, arbitrary models m_{n+1} can be generated by the transaction rules TR9 to TR12. ■

4 Implementation issues

In this section we demonstrate that the rules presented in Section 3 can be mapped to the operators and functions of a spatial relational database. *Oracle Spatial Version 11g* [32] is taken as example since it provides spatial data types for the representation of 3D objects as well as spatial extensions of the query and data manipulation language SQL for the efficient selection of spatial objects. The (proprietary) language PL/SQL extends SQL by procedural concepts. The geometrical and topological aspects of 3D city models can be represented by the following (spatial) relational schema, which is derived from the UML diagram for 3D city models [15]. Since Oracle currently does not support 3D topology, the topological relations have been represented by standard relational tables (see also [17]):

```
vertex(NUMBER Id, SDO_GEOMETRY coordinates)

edge(NUMBER id, NUMBER startVertex, NUMBER endVertex,
      SDO_GEOMETRY geom)

edgeFace(NUMBER edgeId, NUMBER eaceId)

face(NUMBER id, NUMBER leftSolidId, NUMBER rightSolidId,
      SDO_GEOMETRY geom)

solid(NUMBER id, NUMBER isBounded, SDO_GEOMETRY geom)
```

An edge has a start vertex and an end vertex, and the table *edgeFace* relates the Ids of the edges in the boundary of a face to the Ids of that face. A face delimits two solids and the information whether a solid is bounded or not, is stored in the *isBounded* row of the solid table. The geometry of vertices, edges, faces and solids is represented by the geometry data type *SDO_Geometry*. These geometry attributes are to some degree redundant since they can be derived from the vertex geometry to the topological relations. However, geometrical queries are processed more efficiently by that representation.

The composite surface which is the input value of TR9 and 11 (and which is deleted in TR10 and 12) can be represented by a *Multi Polygon* being provided by the type *SDO_GEOMETRY*.

As an example, we now are going to discuss how the transaction rule TR9 can be implemented as PL/SQL module. The implementations of the other rules are similar:

```
1. create function TR9 (CS SDO_GEOMETRY, SolidID NUMBER)
2.     return boolean
3.     is
4.         SDO_GEOMETRY Cycle;
5.         B1 BOOLEAN;
6.         B2 BOOLEAN;
7.         B3 BOOLEAN;
8.         B4 BOOLEAN;
9.     begin
10.        B1 := isCompositeSurface(CS);
11.        B2 := TouchInSimpleCycle(CS, SolidID, Cycle);
12.        B3 := inInterior(CS, SolidID);
13.        B4 := IsJordanCycle(Cycle, SolidID);
14.        IF not B1 or not B2 or not B3 or not B4
15.            THEN return false;
16.        END IF;
17.        merge(CS);
18.        CreateSolid(SolidID, CS, NEWSolidID1, NEWSolidID2);
19.        CreateCompositeSurface(CS, NEWSolidID1, NEWSolidID2);
20.        UpdateReferences(SolidID,CS,NEWSolidID1, NEWSolidID2);
21.        return true;
22.    end;
```

The steps 10. to 16. correspond to the condition part of the rule, whereas the action part is implemented by steps 17. to 20. Both condition and action parts again can be implemented efficiently as PL/SQL procedures or functions. The condition in step 10 can be implemented by use of the axioms in Table 1 (see [12]) and the condition in step 11 by geometrical database operations (test for geometrical identity) and graph algorithms (simplicity of a cycle). Topological database queries (SDO_INSIDE) implement the condition in step 12, whereas the

procedure do discriminate Jordan and handle cycles [14] requires graph search methods, which can be implemented in PL/SQL. The action part (steps 17–20) again can be realized by PL/SQL statements that update the topology tables.

5 Related work

This section reviews related work on transaction rules for updates in GIS and on consistency and validation methods for GIS data, which are the base of the rules.

Transaction rules for 2D GIS have been developed in Gröger et al. [11], Kufoniyyi [25] and Kufoniyyi et al. [26], and for the special case of *simplicial complexes* (i.e., cell complexes consisting of triangles) in Egenhofer et al. [5]. For 3D models, Euler Operators [28] are used to construct surface models stepwise. However, Euler Operators only preserve topological consistency and do not focus on geometric-topological consistency: the geometry of objects is not considered and penetrations are not avoided explicitly. The granularity of Euler operators is very fine in contrast to the rules presented in this paper. Only single surfaces can be subject to an operation, not composite surfaces as in our case. Hence, an error case involving handles which is prevented by the conditions of TR6 cannot occur by applications of Euler operators.

In Gold [8] and Tse et al. [35], Euler Operators and GIS are coupled by applying the operators to triangulations. For some operators, the granularity becomes much coarser: There is, for example, one operator which inserts a handle. However, the handle must be geometrically restricted to a prism. Hence, the rule does not provide the flexibility of our TR7. The approach is restricted to triangulations, while our rules deal with planar faces of arbitrary shape. Furthermore, it covers only surfaces, while our approach copes with 3D models aggregating multiple solids. In [3] and [4], Euler operators are used to generate a single solid model, whereas extended Euler operators are mentioned which are capable of constructing models consisting of multiple solids. However, as it is the case with general Euler operators, the focus is on topological consistency only, and no proofs of completeness or correctness of the rules are provided.

The problem of consistency checking for 3D GIS has been addressed by several authors in the last decades. The methods presented in [28] and [30] are restricted to single solids as component of 3D models. However, not all violations of model assumptions are detected by these methods. The approach by Molenaar [29] is more comprehensive. He introduces *conventions* which check 3D models, but again this method provides necessary but not sufficient conditions. The approaches in the context of Euler Operators [28] for surface models and its application to GIS [8, 35] do not provide general rules to check consistency. The same is true for the extension of Euler operators to models with multiple solids [3, 4]. Consistency rules which check whether geometries are valid with regard to the language GML 3 (being the base of CityGML and an implementation of ISO 19107 ‘Spatial Schema’ [21]) are provided in [23]. The rules for composite surfaces allow non-orientable surfaces like the Möbius-strip (although non-orientable surfaces are excluded [21]) and non-manifold composite surfaces, e.g., four surfaces touching in an edge. The rules for the consistency of a single solid state that the interior should be connected but give no means how this condition should be checked. Furthermore, the rules for composite solids (aggregations of solids) force two solids which are part of a composite solid to have disjoint interiors and a connected interior. Again, no effective method to check the last-mentioned condition is presented.

6 Conclusions

In this paper, we have presented a set of transaction rules for updating 3D city models, which preserve geometric-topological consistency in a reliable way. We prove that the rules maintain consistency once given (safety), and that each consistent 3D city model can be generated by the rules (completeness). Up to now, transaction rules for 3D GIS which have been proven to be both safe and complete with regard to geometric-topological consistency have not been given. The rules target complex 3D city models [15] composed of surfaces representing the terrain and of solids which are aggregated to build complex semantic objects like buildings consisting of rooms and storeys. The rules allow for the construction and removal of handles representing semantic objects like bridges, tunnels and arcades outside buildings or pillars inside rooms.

We identify an error case that may occur if handles are involved in a transaction. We show that the classification of cycles in Jordan and handle cycles, which has been presented in a recent paper [14], can be integrated in the transaction rules to avoid that error. There are efficient algorithms employing standard methods from graph theory to discriminate both types of cycles. The safety results in the paper are proven mathematically, guaranteeing the reliability of our method. The completeness of the rules was difficult to prove. Again, the reason was the occurrence of handles causing potential non-manifold boundaries and therefore consistent models that cannot be generated by safe rules in a straightforward way. Local geometric changes of infinitesimal small size were necessary to solve that problem.

The contribution of the paper is a set of transaction rules for 3D city models. The rules are based on the axioms for 3D city models [12, 14, 15] and apply the method for differentiating between handles and Jordan cycles [14]. They have roughly been sketched in [13], where no details and no proofs of safety or completeness are given.

The next step will be the detailed description of transaction rules for surfaces (TR1 to TR8), complementing the rules for 3D city models as well as the proof of the corresponding safety and completeness results. This will be the topic of a subsequent paper which is in preparation [16].

Our approach currently deals with faces which are bounded by one exterior cycle. A generalization will be the consideration of interior cycles of faces, forming holes or enclaves in the face. This way the representation of free-standing buildings, for example, without introducing so-called *pseudo edges* is made possible. As a consequence, we have to deal with non-connected graphs, and our notion of reachability in (discrete) graphs has to be extended by considering paths in the interior of a surface. Particularly the algorithm to discriminate between handle and Jordan cycles has to be adjusted accordingly.

This paper focuses on surfaces delimiting solids or representing the terrain (partially delimiting the earth or air solid). Hence, a further extension of the model will be the incorporation of lines (to represent objects like antennas) and of surfaces that do not delimit a solid (to model roof overhangs, for example). Those cases will be handled by additional axioms, allowing that particular surfaces delimit the same solid on both sides. Additional transaction rules will be provided which insert, delete and modify such surfaces and lines. This will be the topic of another paper.

CityGML has strongly influenced our 3D city model and the transaction rules. The axioms and the rules are applicable to CityGML data sets and therefore assure its consistency. Particularly indoor (Level-of-Detail 4) models correspond to our aggregations of solids which represent interior rooms semantically (see Section 2.2). Line objects and surfaces not delimiting a solid (e.g., roof overhangs), which can be modeled in CityGML, are covered by the extension mentioned in the last paragraph.

Based on the implementation concepts presented in Section 4, an additional step will be the implementation of the rules in the context of a *CityGML* database [10, 17] on top of a spatial database. Such a database establishes the core of a *transactional Web Feature Service (WFS-T)* [37], that provides standardized access to data and a standardized way to update data, as part of a sustainable *spatial data infrastructure*.

Acknowledgements We would like to thank the anonymous reviewers for their constructive and very valuable comments which helped us to improve the paper significantly. We thank Rosemarie Schlager for proof-reading and improving the English language and style of this text.

Appendix A: Basic notions

In this paper we use standard notions from graph theory [19] to mathematical topology [1, 2, 20, 28]. In the following, we recapitulate the notions from mathematical topology which are necessary for defining the concepts presented in the paper.

A.1 Topology: 2-manifolds, boundaries, and genus

A *2-manifold* surface, which plays a decisive role in geometrical and topological modelling [7, 21, 28–30], is a topological space where each point has a neighbourhood which is topologically equivalent to an open two-dimensional disk [20]. Examples for 2-manifolds are the open disk, the open cylinder surface, the hull of a cuboid, or the sphere. Figure 18 depicts the open disk (a) and the hull of a saddle roof building (b) as examples for 2-manifolds, as well as three counterexamples: three rectangles meeting in a common edge (c), the hull of two cuboids meeting in a common edge (d), and the hull of two saddle roof buildings, where the roof of one buildings penetrates the wall of the other (e).

Another property of surfaces, which is also a topological invariant, is the *number of boundaries*. An open disk has one boundary, whereas an open cylinder surface has two and a sphere has zero boundaries (c.f. Fig. 19). Surfaces without boundaries are *closed*.⁵ They enclose a volume completely and hence are used in geometrical modelling to represent the boundary of *solids* [28, 30]. Some examples for a closed surface are the sphere, the hull of a cuboid, or the hull of a saddle roof building (c.f. Figs. 18b and 19b).

The number of *handles* is another topological invariant of surfaces. Handles in surfaces are used to model bridges, tunnels, arcades or similar phenomena. The number of handles of a surface is given by its *genus*.⁶ The genus is defined as the maximal number of closed, continuous, non self intersecting and mutually non-intersecting curves, the cutting of which preserves the connectivity of the surface. A sphere, for example, has genus zero and hence no handles, while the genus of a torus is one. It has one handle (see Fig. 20).

A.2 Cell complexes

The theory of cell complexes [20], a branch of Algebraic Topology, defines concepts for aggregating primitive objects to more complex ones in a topologically clean manner. Hence,

⁵ Note that the term *closed* is overloaded: It may denote a surface with boundary where the boundary points belong to the set (in contrast to *open sets*), or a surface which has no boundary at all. In this paper, the second meaning is used.

⁶ This proposition is valid for connected, orientable surfaces. We assume that all surfaces discussed in this paper have both properties.

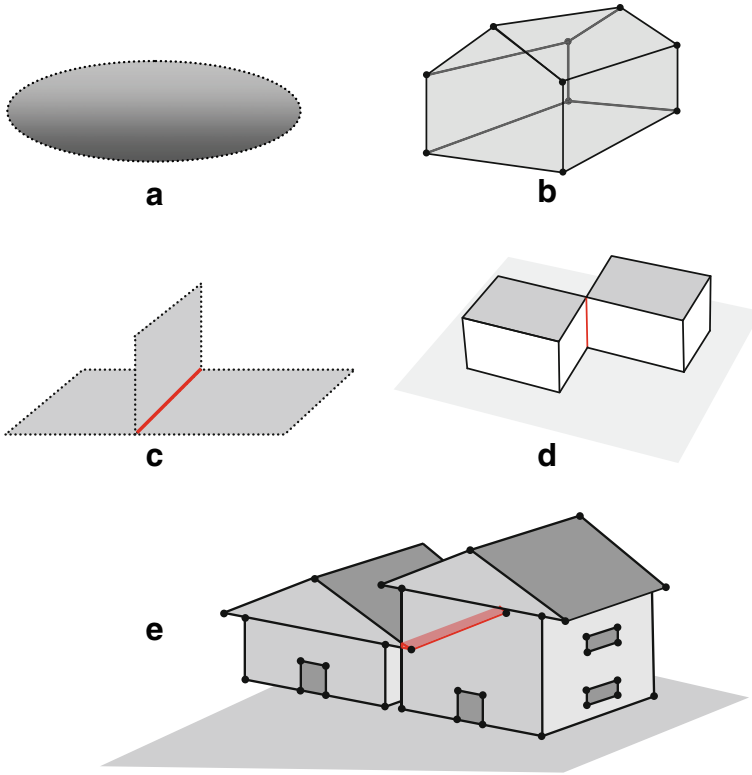
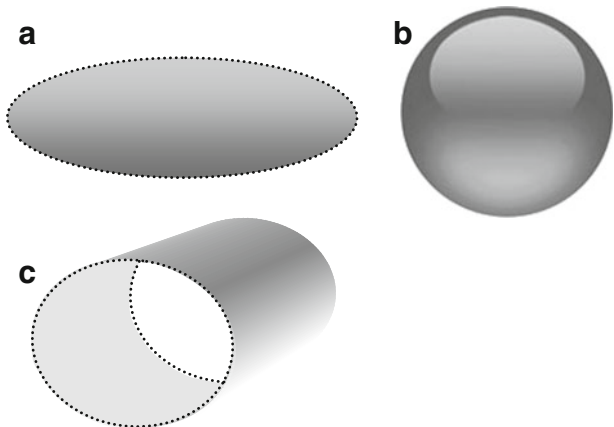


Fig. 18 Examples for 2-manifold surfaces (a, b) and non 2-manifold surfaces (c–e)

cell complexes are the base of many topological data models in GIS, CAD and Computer Graphics [7, 21, 29, 33]. Primitives are nodes, edges, faces and solids, which are also called *0-cells*, *1-cells*, *2-cells* and *3-cells*. Each *n-cell* is bounded by $(n-1)$ -cells, which are called the *boundary* of the *n-cell*. A *cell complex* is an aggregation of cells where the following condition holds true: The intersection of two cells in the cell complex is either empty or is a

Fig. 19 Surfaces with different numbers of boundaries: a disk, one boundary, b sphere: zero boundaries, c cylinder surface: 2 boundaries



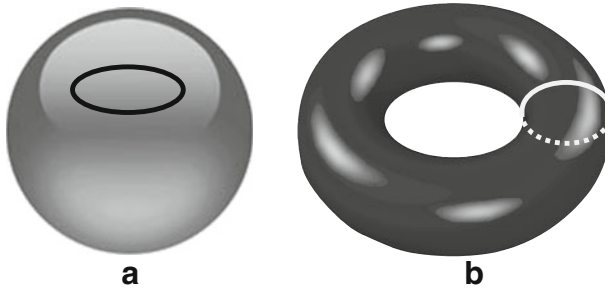
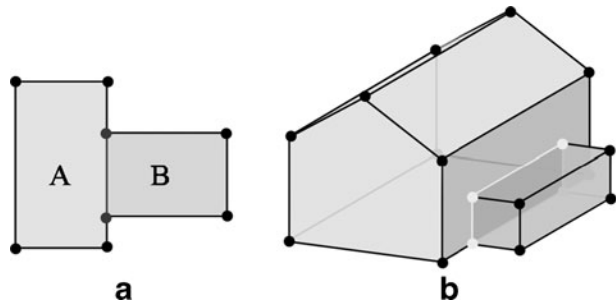


Fig. 20 The sphere **a** has no handle (genus zero), the torus **b** has one handle (genus one). Each cycle on the surface in **(a)** separates the surface, while the cycle on the surface in **(b)** does not. Each additional, non-intersecting cycle separates the surface

Fig. 21 Cell complexes consisting of two 2-cells **(a)** and of two 3-cells **(b)**



cell which is part of the boundaries of both cells. Thus, cell complexes avoid overlapping cells as well as penetrations of cells, and touching of cells is defined explicitly by commonly shared boundaries. To exemplify the concept of cell complexes, Fig. 21 depicts a cell complex consisting of two 2-cells **(a)** and one consisting of two 3-cells **(b)**. The touching of both cells is represented explicitly by a 1-cell and two 0-cells in **(a)** and by a 2-cell (depicted in dark colour), four 1-cells and four 0-cells (each depicted in light colour) in **(b)**.

References

- Alexandroff P (1961) Elementary concepts of topology. Dover New York
- Armstrong MA (1997) Basic topology. Undergraduate texts in mathematics. Springer, New York
- Boguslawski P, Gold C (2009) Construction operators for modelling 3D objects and dual navigation structures. In: Lee J, Zlatanova S (eds) 3D geo-information sciences. Lecture notes in geoinformation and cartography. Springer, Berlin, Heidelberg, pp 47–59
- Boguslawski P, Gold C (2010) Euler operators and navigation of multi-shell building models. In: Neutens T, Maeyer P (eds) Developments in 3D Geo-information sciences. Lecture notes in geoinformation and cartography. Springer, Berlin, Heidelberg, pp 1–16
- Egenhofer MJ, Frank AU, Jackson JP (1990) A topological data model for spatial databases. In: Buchmann AP, Günther O, Smith S, Wang Y (eds) Design and implementation of large spatial databases, first symposium SSD '89 Santa Barbara, California, July 17/18, 1989 Proceedings. Lecture notes in computer science, vol. 409. Springer, Berlin, Heidelberg, pp. 271–286
- Egenhofer MJ (1989) A formal definition of binary topological relationships. In: Litwin W (ed) Foundations of data organization and algorithms, 3rd International Conference, FODO 1989 Paris, France, June 21–23, 1989 Proceedings. Lecture notes in computer science, vol. 367. Springer, Berlin, Heidelberg, pp. 457–473

7. Foley JD, van Dam A, Feiner SK, Hughes JF (1995) *Computer graphics: principles and practice*, 2nd edn. Addison-Wesley systems programming series, Addison-Wesley Publishing Company, Boston
8. Gold C (2003) But is it GIS? *J Geosp Eng* 5(2):11–26
9. Gröger G, Kolbe TH, Czerwinski A, Nagel C (2008) OpenGIS City Geography Markup Language (CityGML) encoding standard. Version 1.0.0., Open Geospatial Consortium, OGC Doc. No. 08-007r1
10. Gröger G, Kolbe TH, Schmittwilken J, Stroh V, Plümer L (2005) Integrating versions, history and levels-of-detail within a 3D geodatabase. In: Gröger G, Kolbe TH (eds) *Proceedings of the 1st international ISPRS/EuroSDR/DGPF-Workshop on Next Generation 3D City Models*, Bonn, June 21–22. EuroSDR, pp. 35–40
11. Gröger G, Plümer L (1997) Provably correct and complete transaction rules for GIS. In: Laurini R, Bergougnoux P (eds) *Proceedings of the 5th International Workshop on Advances in Geographic Information Systems (ACM-GIS'97)*. ACM Press, Las Vegas, Nevada, pp. 40–43
12. Gröger G, Plümer L (2005) How to get 3-D for the price of 2-D-topology and consistency of 3-D urban GIS. *Geoinformatica* 9(2):139–158
13. Gröger G, Plümer L (2009) Updating 3D City Models—how to preserve geometric-topological consistency. In: Agrawal D, Aref W, Lu C, Mokbel MF, Scheuermann P, Shahabi C, Wolfsohn O (eds) *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2009)*, Seattle, Washington, November 4.-6., 2009. ACM Press, New York, pp. 536–539
14. Gröger G, Plümer L (2011) Topology of surfaces modelling bridges and tunnels in 3D-GIS. *Comput Environ Urban Syst*. 35:208–216. doi:10.1016/j.compenvurbysys.2010.10.001
15. Gröger G, Plümer L (2011) How to achieve consistency for 3D city models. *Geoinformatica* 15(1):137–165. doi:10.1007/s10707-009-0091-6
16. Gröger G, Plümer L (2011) Transaction rules for updating surfaces in 3D GIS. Submitted to *ISPRS Journal*
17. Gröger G, Reuter M, Plümer L (2004) Shall 3-D city models be managed in a commercial database? *Geoinformationssysteme (GIS)* 17(9):9–15
18. Guptill SC, Morrison JL (eds) (1997) *Elements of spatial data quality*, 1st edn. Elsevier Science Oxford. 202 p
19. Harary F (1969) *Graph theory*. Addison Wesley Publishing Company
20. Hatcher A (2001) *Algebraic topology*, 10th edn. Cambridge University Press, Cambridge
21. Herring J (2001) The OpenGIS abstract specification, topic 1: feature geometry (ISO 19107 Spatial Schema), open geospatial consortium, OGC document number 01–101. OGC
22. Kainz W (1997) Logical consistency. In: Guptill SC, Morrison JL (eds) *Elements of spatial data quality*. Elsevier Science, Oxford
23. Kazar BM, Kothuri RV, Oosterom P, Ravada S (2008) On valid and invalid three-dimensional geometries. In: Oosterom P, Zlatanova S, Penninga F, Fendel E (eds) *Advances in 3D geoinformation systems. Lecture notes in geoinformation and cartography*. Springer, Berlin, pp 10–46
24. Kolbe TH, Gröger G, Plümer L (2008) CityGML - 3D city models for emergency response. In: Zlatanova S, Li J (eds) *Geospatial information technology for emergency response. International society for photogrammetry and remote sensing book series*, vol 6. Taylor & Francis, London, pp 257–274
25. Kufoniya O (1995) Spatial coincidence modelling, automated database updating and data consistency in vector GIS. International Institute for Aerospace Survey and Earth Sciences (ITC). Enschede, The Netherlands, ITC Publication No. 28
26. Kufoniya O, Molenaar M, Bouloucos T (1995) Topologic consistency operations in vector maps. *GIS* 8(4):7–13
27. Lee J, Zlatanova S (2009) A 3D data model and topological analyses for emergency response in urban areas. In: Zlatanova S, Li S (eds) *Geospatial information technology for emergency response. ISPRS book series*. Taylor & Francis, London, pp 143–168
28. Mäntylä M (1988) *An introduction to solid modeling. Principles of computer science*, Computer Science Press
29. Molenaar M (1992) A topology for 3D vector maps. *ITC Journal* 1992–1
30. Mortenson ME (1997) *Geometric modelling*. Wiley
31. Okabe A, Boots B, Sugihara K, Nok Chiu S (2000) *Spatial Tessellations, Concepts and applications of Voronoi diagrams*, 2nd ed. Wiley series in probability and statistics, Wiley Chichester
32. Oracle Cooperation (2009) *Oracle spatial developers guide 11 g release 1 (11.1)*
33. Portele C (2007) OpenGIS Geography Markup Language (GML) encoding standard, version 3.2.1, OGC Doc. No. 07–036. Open Geospatial Consortium
34. Tele Atlas (2006) *Accurace in digital mapping*. White paper. www.teleatlas.com
35. Tse ROC, Gold C (2002) TIN meets CAD—Extending the TIN concept to GIS. In: Sloot P (ed) *Computational science-ICCS 2002, Proc. of the International conference International Conference on Computational Science*, Amsterdam, The Netherlands, April 21–24, 2002. Lecture notes in computer science, vol. 2329–2331. Springer, Berlin, New York. pp. 135–143
36. Ueda K (1985) Guarded horn clauses. In: *Logic Programming'85. Lecture notes in computer science*, vol. 221. Springer, Berlin Heidelberg, pp. 168–179

37. Vretanos PA (2004) Web feature service implementation specification, version: 1.1.0, Open Geospatial Consortium, OGC Doc.-No. 04–094
38. Widom J, Ceri S (1996) Active database systems. Morgan Kaufmann
39. Zender H, Martínez Mozos O, Jensfelt P, Kruijff G-M, Burgard W (2008) Conceptual spatial representations for indoor mobile robots. *Rob Autom Syst* 56(6):493–502
40. Zlatanova S (2000) 3D GIS for urban development. ITC dissertation; vol. 69, International Institute for Aerospace Survey and Earth Sciences; Institute for Computer Graphics and Vision, Graz University of Technology Enschede, The Netherlands; Graz, Austria



Gerhard Gröger is a senior researcher at the Institute for Geodesy and Geoinformation, University of Bonn, Germany. He studied computer science and obtained his Master's degree from the University of Bonn, his Ph.D. degree from the University of Vechta and his *venia legendi* from the University of Bonn. His current research interests are geographic information systems (GIS), spatial data models, integrity and data quality in GIS, standardization in GIS, 3-D models, urban GIS and multi-scale GIS. He is chair of the modeling subgroup of the Special Interest Group 3-D (SIG 3D) of the initiative 'Spatial Data Infrastructure Germany (GDI.DE)'.



Lutz Plümer is a full professor at the Institute for Geodesy and Geoinformation, University of Bonn, Germany. He initially studied computer science, received his Master's and Ph.D. degree from the University of Dortmund and his *venia legendi* from the University of Bonn. His current research focuses on geoinformation, geometric-topological integrity, 3-D models and urban GIS, spatio-temporal data mining and early detection of stress in plants.