What is a valid *n*D GIS object? Extending the validity notions embedded in the geoinformation standards for 2D and 3D

Ken Arroyo Ohori Hugo Ledoux Jantien Stoter

This is a paper that was originally accepted to the ISPRS Workshop on Multi-dimensional & Multi-scale Spatial Data Modeling in Istanbul in October 2016. It was withdrawn when I couldn't present it due to illness.

An alternative to the traditional approaches to model separately 2D/3D space, time and scale in GIS lies in the higher-dimensional modelling of geographic information, in which a chosen set of non-spatial characteristics, e.g. time and scale, are modelled as extra geometric dimensions perpendicular to the spatial ones, creating a higher-dimensional model. While the notions of valid objects and space partitions in 2D and 3D can be deduced from existing geoinformation standards, their extension to a dimension-independent formulation has been hardly studied, and thus it remains an open question whether a given higher-dimensional model is valid or not. This paper presents a concrete interpretation of the notions of validity as embedded in the most important 2D and 3D geoinfomation standards, and then uses these to create a dimension-independent definition of a valid *n*D object and an *n*D space partition that follows in the spirit of these standards.

1 Introduction

The traditional approaches to model 2D/3D space, time and scale in GIS are mostly based on adaptations to well-known 2D data structures, such as the DCEL (Muller and Preparata, 1978) and the quad-edge (Guibas and Stolfi, 1985). An alternative to this approach lies in the higher-dimensional modelling of geographic information (Arroyo Ohori, 2016), in which a chosen set of non-spatial characteristics, e.g. time and scale (van Oosterom and Stoter, 2010), are modelled as extra geometric dimensions perpendicular to the spatial ones, creating a higher-dimensional model. Such higher-dimensional models can then be populated by sets of multi-dimensional objects, which in an n-dimensional model can be of varied dimensions from zero (i.e. points) up to n (i.e. n-cells).

This means that much like in 2D and 3D GIS, and as discussed in Section 2, validity constraints for higher-dimensional objects are a very relevant subject of study, as invalid objects lead to software crashes and erroneous calculations. Unfortunately, the existing geoinformation standards are already rather vague about what exactly is a valid object or set of objects and—most relevantly for this paper—their definitions are by design limited to 2D and 3D.

This paper attempts to remedy these omissions in two steps. Based on the most important 2D and 3D geoinformation standards (ISO, 2005, 2007, 2006; OGC, 2007), we first give a more concrete definition of validity for 2D objects and planar partitions in Section 3 and a similar one for 3D objects and 3D space subdivisions in Section 4. Afterwards, we use these in order to propose a concrete validity of nD objects and nD space subdivisions that follows the spirit of the aforementioned standards in Section 5, as well as a few alternatives. Finally, we finish the paper with a short discussion and some ideas for future work in Section 6.

2 Motivation

Invalid datasets are prevalent in 2D/3D GIS (Panigrahi, 2014, Ch. 7) and a major source of problems for those who work with them. In fact, according to McKenney (1998), users of 3D CAD models for finite element analysis—which has similar requirements as certain computations in GIS, such as well-shaped and non-overlapping mesh elements—spend up to 70% of their time fixing the input CAD models. While similar figures for GIS are to the best of our knowledge not available, it is worth noting that CAD software tends to produce better quality models than GIS software[1].

Among other problems, invalid datasets can be interpreted inconsistently in different software, leading to silent errors that cause inconsistent or erroneous results when using in different ways or in different environments. They can also make it impossible to perform a certain operation, either due to a failing precondition check or due to software crashes. All of these situations cause major problems for users of GIS software.

In our view, the problems related to invalid GIS datasets are being tackled on three incremental steps, which tend to build on each other: (i) the definition of clear validity criteria, as is currently provided by various geoinformation standards in 2D and 3D; (ii) the use of validation methods and tools that verify if a dataset complies with such standards (Plümer and Gröger, 1997; Ledoux, 2013); and (iii) the creation of repair tools that fix the errors in invalid datasets either automatically or semi-automatically (Arroyo Ohori et al., 2012; Ledoux et al., 2014; Zhao et al., 2013). As the GIS community pushes towards reusable GIS datasets that are not application-specific—an effort spearheaded by CityGML in 3D (Gröger et al., 2012)—as

---

[1] There are many reasons for this. For instance, CAD software makes wider use of topological data structures, and also has topology-aware and smart interactive editing tools (e.g. snapping to guide lines and nearby objects), which help to avoid problems where objects seem to be valid but have small errors, such as sliver polygons and shells that have tiny gaps and are thus not properly closed.

well as the integration of non-spatial characteristics into GIS datasets, it only makes sense to extend these efforts to higher dimensions. This paper is a first effort focusing on the first of these steps only: the definition of clear validity for both nD objects and nD space partitions.

3 Validity in 2D

In most GIS file formats and the software that reads and writes them, polygons and multipolygons are defined in a manner that is consistent with the definitions in the Simple Features Specification (OGC, 2011; ISO, 2006)—an implementation of the ISO 19107 standard (ISO, 2005). The specification states that: '*A `Polygon` is a planar `Surface` defined by 1 exterior boundary and 0 or more interior boundaries. Each interior boundary defines a hole in the `Polygon`*'. Each of these boundaries is described as a `LinearRing` (Figure 1). According to the specification, an outer ring should be oriented *anticlockwise* when viewed from a predefined *top* direction, which is generally (but not necessarily) the viewing direction in 2D or *outwards* when the polygon specifies part of the boundary of a polyhedron. Inner rings should be oppositely oriented, i.e. generally *clockwise* when viewed from the top direction.

The Simple Features Specification provides several **validity rules for polygons**, which Ledoux et al. (2014) summarise as follows:

- each ring defining the exterior and interior boundaries is *simple*, i.e. non-self-intersecting;
- each ring is closed, i.e. its first and its last points should be the same;
- the rings of a polygon do not cross, but they may intersect at one tangent point;
- a polygon does not have cut lines, spikes or punctures;
- the interior of every polygon is a connected point set;
- each interior ring creates a new area that is disconnected from the exterior.

Similarly, the specification provides a definition and some **validity rules for multipolygons**. A `MultiPolygon` is defined as a `MultiSurface` forming an aggregation of `Polygons`, which also follows certain validity criteria, which we summarise as follows:

- the interiors of its polygons do not overlap, i.e. their point set intersection should be empty;
- the boundaries of its polygons may only touch at a finite number of points;
- a multipolygon does not have cut lines, spikes or punctures;
- the interior of a multipolygon with more than one polygon is *not* a connected point set.

Intuitively, a *planar partition* is a set of polygons that form a subdivision of a region of the plane. Planar partitions are thus commonly used to model concepts where objects are expected not to overlap, such as land cover, cadastral parcels, or the administrative boundaries of a given country. Despite being a very frequently used representation in GIS, planar partitions are not explicitly defined in the main GIS standards.

Within the classes in the ISO 19107 standard (ISO, 2005, §6.6), a planar partition could be considered as a `GM_CompositeSurface`, defined in the standard as '*a collection of oriented surfaces that join in pairs on common boundary curves and which, when considered as a whole, form a single surface*'. By following this definition, overlaps between polygons are explicitly forbidden, as a `GM_Complex` (a parent of `GM_CompositeSurface`) is defined as '*a set of primitive geometric objects (in a common coordinate system) whose interiors are disjoint*'. However, a `GM_CompositeSurface` explicitly allows gaps between the surfaces, as these would simply result in inner rings within the overarching single surface.

An alternative definition could be created based on the ISO 19123 standard (ISO, 2007, §6.8)—a standard focusing on coverages of various types. According to the standard, a planar partition can be considered

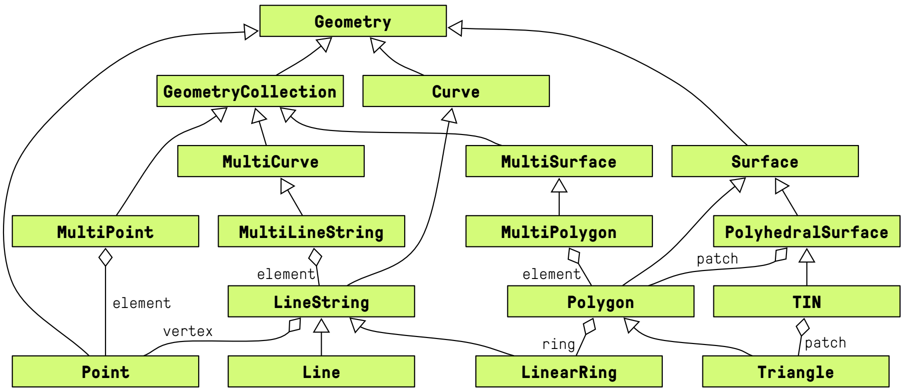


Figure 1: The geometry class hierarchy defined in the Simple Features Specification (OGC, 2011).

as a type of `CV_DiscreteSurfaceCoverage` where '*the surfaces that constitute the domain of a coverage are mutually exclusive and exhaustively partition the extent of the coverage*'. Overlapping polygons are disallowed by them being '*mutually exclusive*' and gaps are disallowed by the surfaces '*exhaustively partitioning*' the extent. However, the standard states these conditions as something that occurs '*in most cases*', whereas in a planar partition it should be considered as a strict prerequisite.

In a **valid planar partition**, there should thus be no overlapping polygons, and no gaps between them either unless these gaps are considered to be outside of the region. These two conditions are covered by the ISO 19107 standard in a different context, when it lists some possible inconsistencies of 'spaghetti' datasets represented as a `GM_Complex`, stating that '*slivers and gaps are multiple lines that should represent the same geometry, but do not coincide, leaving areas of overlap between two surface boundaries (slivers), and gaps between them*' (ISO, 2005, §6.2.2.6).

4 Validity in 3D

The ISO 19107 standard (ISO, 2005, §6.3.18) defines 3D objects with 3D holes that are known as *solids*, which are specified based on a boundary representation scheme. As shown in Figure 2, the standard thus defines a `GM_Solid` with a *boundary* operation returning a `GM_SolidBoundary`, which is a '*sequence of sets of `GM_Surfaces` that limit the extent of [the] `GM_Solid`*'. Each of these sets of surfaces describes one of the boundaries of the `GM_Solid` as a `GM_Shell`, corresponding to either the outer boundary for the solid[2] or one of its holes.

A `GM_Shell` (ISO, 2005, §6.3.8) thus represents '*a single connected component of a `GM_SolidBoundary`*'. It is known to be *simple*, and consists of a set of oriented instances of `GM_Surface` composed of instances of `GM_SurfacePatch`, which intuitively form a cellular subdivision of the surface and themselves have a `GM_SurfaceBoundary`. A `GM_SurfaceBoundary` represents an area potentially with any number of holes, each of which is stored as a reference to a `GM_Ring`. A `GM_Ring` (ISO, 2005, §6.3.6) is additionally defined as being *simple*.

`GM_Object` (ISO, 2005, §6.2.2), a parent class to all the classes previously mentioned, defines every object as a point set and provides the definition of simple as a '*GM_Object [that] has no interior point of self-intersection or self-tangency. In mathematical formalisms, this means that every point in the interior of the object must have a metric neighborhood*

---

[2] In some cases, there might not be an outer boundary of a solid, such as in non-Euclidean spaces or in the representation of unbounded solids. However, there is nearly always an outer boundary in the context of geographic information.

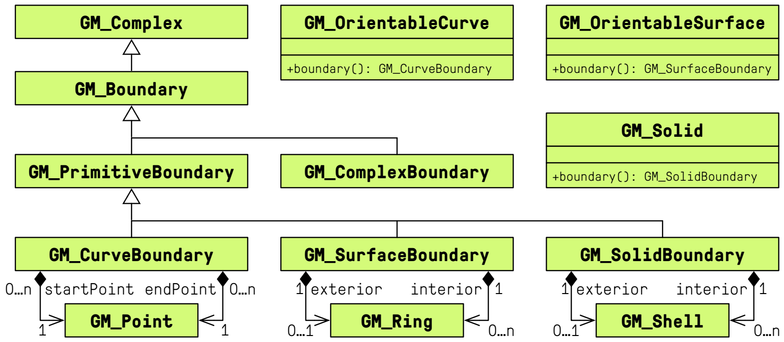


Figure 2: The ISO 19107 standard (ISO, 2005, §6.3.2) is able to specify the boundaries of `GM_Curve`, `GM_Surface` and `GM_Solid` as subclasses of `GM_Boundary`, respectively a `GM_CurveBoundary` linked to a pair of `GM_Point` (the end-points of a line segment), `GM_SurfaceBoundary` linked to a set of instances of `GM_Ring`, and a `GM_SolidBoundary` linked to set of instances of `GM_Shell`.

whose intersection with the object is isomorphic to an n-sphere, where n is the dimension of this `GM_Object`'. As discussed by Ledoux (2013), this implies that shells are effectively 2-manifolds. Rings are similarly 1-manifolds.

It is important to note that even though each `GM_Ring` and `GM_Shell` is individually simple, the boundary of the `GM_Surface` or `GM_Solid` that they together describe does not need to be simple. A common example would involve an inner ring/shell tangent to the outer ring/shell containing it. Arguably, the standard does appear to explicitly forbid intersections between the interior of rings or shells as `GM_Complex` is a parent class of `GM_SurfaceBoundary` and `GM_SolidBoundary` and this class requires its composing primitives to be '*geometrically disjoint*'. However, this interpretation is problematic as it would arguably also forbid inner rings being inside their containing outer ring.

Alternatively, it is possible to consider that the standard does not specify *any* restrictions regarding the interactions between rings of a surface or between shells of a solid. As the standard explicitly states that '*implementations may enforce stronger restrictions on the interaction of boundary elements*', it might be the responsibility of other implementing standards to place appropriate restrictions.

Although the GML standard (OGC, 2007) implementing ISO 19107 does not specify such restrictions, it is possible to use those defined in the Simple Features Specification in 2D (Section 3) and define analogous ones in 3D (Ledoux, 2013). One possible formulation of these could be as follows:

- the shells of a solid do not cross, but the shells on the boundary of a solid may intersect only at a vertex or edge;
- the interior of every solid is a connected point set;
- each interior shell creates a new volume that is disconnected from the exterior.

Intuitively, a 3D space partition is a subdivision of a region of 3D space into non-overlapping solids. However, just as with planar partitions, 3D space partitions are usually not strictly defined. Following the same logic as with planar partitions in Section 3, a 3D space partition can be considered as an ISO 19107 `GM_CompositeSolid` (ISO, 2005, §6.6.13), which is defined in the standard as a '*a set of solids that join in pairs on common boundary surfaces to form a single solid*'. While overlapping solids are explicitly forbidden by a `GM_CompositeSolid` inheriting from `GM_Complex` in which '*[primitive] interiors are disjoint*', gaps between the solids are explicitly allowed.

An alternative definition could also be created based on the ISO 19123 standard by considering a 3D space partition as a type of `CV_DiscreteSolidCoverage` (ISO, 2007, §6.10), which states that '*generally, the solids that constitute the domain of a coverage are mutually exclusive and exhaustively partition the extent of the coverage*'. While overlaps and gaps are respectively eliminated by the '*mutually exclusive*' and '*exhaustively partition*' conditions, the word '*generally*' implies that these are not always enforced.

5 Validity in nD

The standards for geographic information in 2D and 3D described previously (Simple Features (OGC, 2011), GML (OGC, 2012) and ISO 19107 (ISO, 2005)) are in theory limited to 2D and 3D. Concretely, the ISO 19107 standard explicitly states that '*this International Standard is restricted to at most three dimensions*'. However, as the mathematics of point set topology behind the standard are dimension-independent (Poincaré, 1895), the definitions given in the standards extend naturally to higher dimensions. This effort would mostly involve the addition of new classes and corresponding definitions. However, it is important to note that the standards do contain minor hard-coded assumptions that are only valid for the 2D and 3D cases, such as how ISO 19107 and GML consider orientable curves and surfaces, but not orientable solids (Figure 3).

This section therefore defines higher-dimensional objects in a manner that is (mostly) harmonious with the standards used in the GIS world. An n-cell can be thus represented by the set of $(n-1)$-cells in its (outer) boundary, using a similar mechanism as how other boundaries are represented in the ISO 19107 standard, which was shown previously in Figure 2.

Following the terminology used in the standard and as shown in Figure 4, such an extension of the would mainly entail a `GM_OrientableGeometricPrimitive` with a `dimension` attribute, which would set to n. This class would be analogous to `GM_OrientableCurve` for `dimension` 1, `GM_OrientableSurface` for `dimension` 2 and a newly created `GM_OrientableSolid` for `dimension` 3, which would be a subclass of `GM_OrientablePrimitive`. The `GM_OrientableGeometricPrimitive` would be bounded by a `GM_GeometricPrimitiveBoundary`, which would be linked to aggregations of $(n-1)$-dimensional instances of a newly created `GM_Cell` (with their `dimension` attribute set to $n-1$). This `GM_Cell` class would be analogous to `GM_Point` for `dimension` 0, `GM_Curve` for `dimension` 1, `GM_Ring` for `dimension` 2, and `GM_Shell` for `dimension` 3. Each of the instances of `GM_Cell` bounding a `GM_OrientablePrimitive` would represent either the outer boundary of the geometric primitive (if any), or one of any number of inner boundaries representing n-dimensional holes. This extension of the standard would seem to follow most in the spirit of ISO 19107.

However, other alternative extensions could be considered. As `GM_Curve`, `GM_Surface`, and `GM_Solid` would essentially be special cases of `GM_Cell`, all of the former could be seen as redundant and eliminated. However, the standard already contains many specialisations that are somewhat redundant but that cover common use cases in geographic information, such as `GM_Triangle` and `GM_Tin`. Another possibility would be considering `GM_Curve`, `GM_Surface`, and `GM_Solid` as subclasses of `GM_Cell` or substituting the abstract `GM_Primitive` for a non-abstract `GM_Cell`, but this would involve a major change in the standard and seems to run counter to the preferred use of abstract top classes in the standard.

The definition of an `GM_OrientableGeometricPrimitive` as explained above also lends itself to the definition of sets of disjoint cells (akin to the `Multi`… classes in the standard) and cell complexes (akin to the `Composite`… classes in the standard), which could also be handled in the same manner as in the ISO 19107 standard. As shown in Figure 5, the standard already defines composite curves, surfaces and solids, which are equivalent to 1-, 2- and 3-dimensional cell complexes. A `GM_CompositeCurve` is '*a list of orientable curves (`GM_OrientableCurve`)*

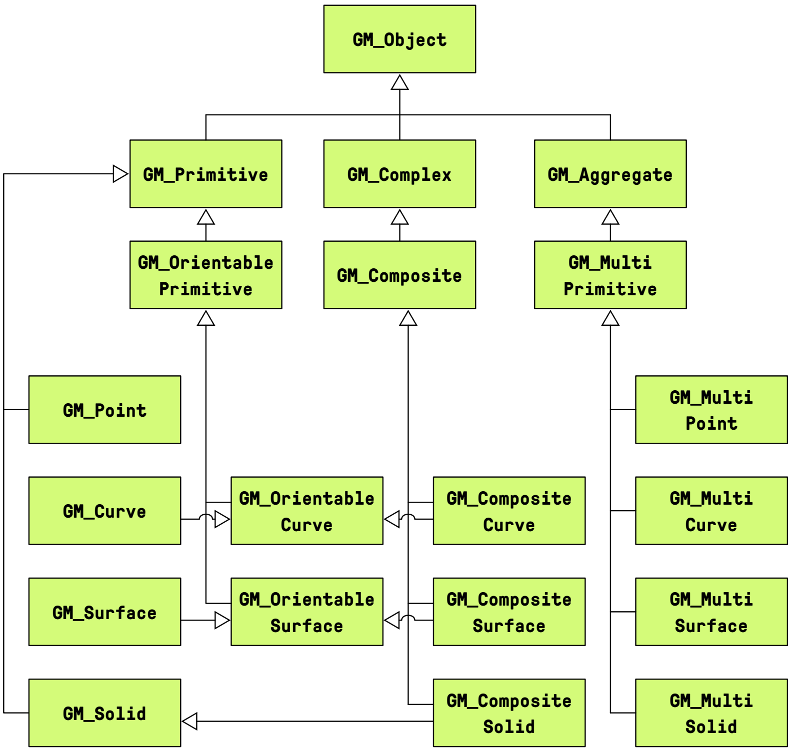


Figure 3: The geometric classes from the ISO 19107 standard (ISO, 2005) that are implemented in the GML standard (OGC, 2007).

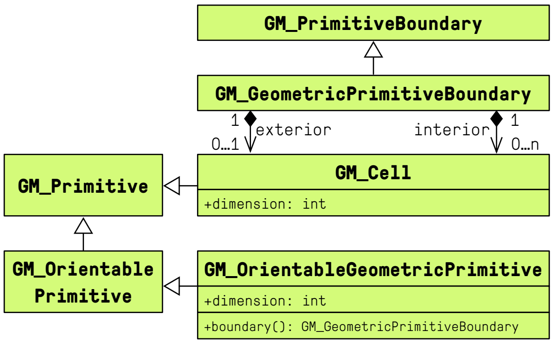


Figure 4: A dimension-independent definition of a cell in a harmonised manner with other classes in the ISO 19107 standard (ISO, 2005).

agreeing in orientation in a manner such that each curve (except the first) begins where the previous one ends.', a `GM_CompositeSurface` is '*a collection of oriented surfaces that join in pairs on common boundary curves*', and a `GM_CompositeSolid` is '*a set of solids that join in pairs on common boundary surfaces*'.

A similarly defined `GM_CompositeGeometricPrimitive`, shown in Figure 6 and which should contain the `dimension` as a parameter, would thus be equivalent to a representation of a space partition of any dimension that allows objects with holes. It could be defined as 'a set of n-dimensional orientable geometric primitives (`GM_OrientableGeometricPrimitive`) that join in pairs on common $(n-1)$-dimensional boundary geometric primitives'. Note that this implies that the primitives combinatorially form an n-quasi-manifold[3] (Figure 7), although

---

[3] A combinatorial interpretation of the point set topology concept of an n-manifold.

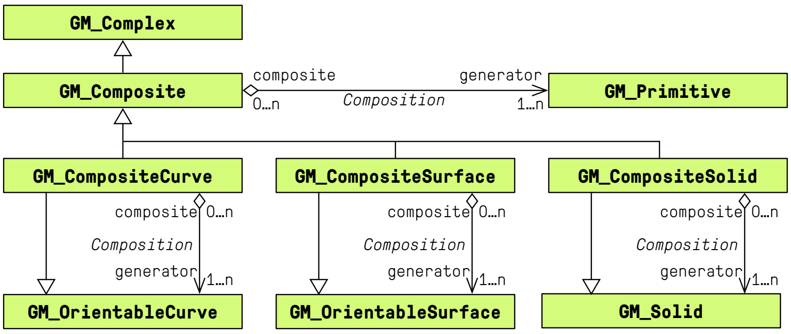


Figure 5: The cell complexes of dimension 1, 2 and 3 are respectively defined in the ISO 19107 standard (ISO, 2005, §6.6.3) as the classes `GM_CompositeCurve`, `GM_CompositeSurface` and `GM_CompositeSolid`.

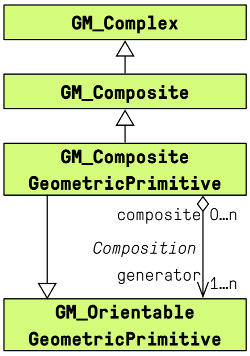


Figure 6: A definition of an n-dimensional space subdivision in a harmonised manner with other classes in the ISO 19107 standard (ISO, 2005).

geometrically they might not do so due to the presence of holes.

Following the validity criteria previously described in Section 3 and Section 4, it is possible to define additional validity criteria for an n-dimensional geometric primitive, which would serve to specify the conditions upon which its bounding cells may interact. These would be as follows:

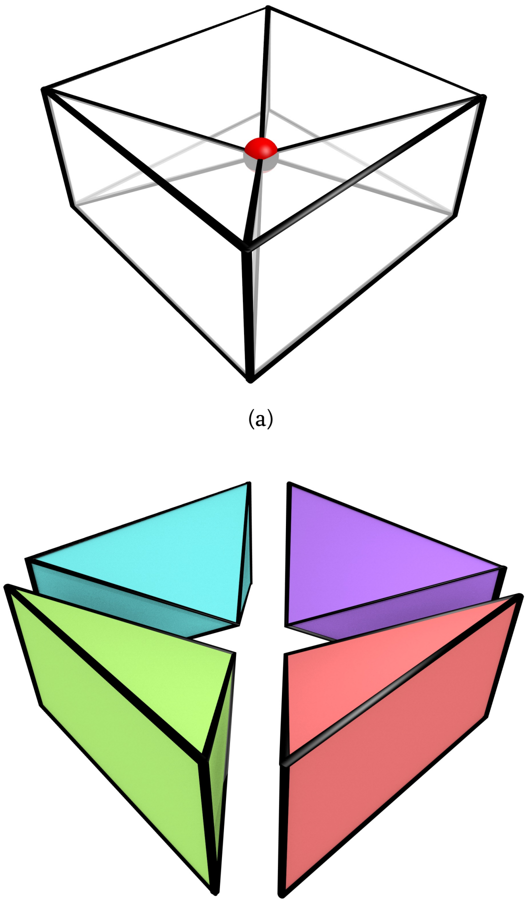
- the bounding $(n-1)$-cells of an n-dimensional geometric primitive do not cross, but they might intersect only at a cell of dimension $n-2$ or lower;
- the interior of every geometric primitive is a connected point set;
- each interior n-cell creates a new point set in $\mathbb{R}^n$ that is disconnected from the exterior.

Meanwhile, an n-dimensional space subdivision should consist of a set of n-dimensional geometric primitives that are mutually exclusive and exhaustively partition an extent, itself a well-defined subset of $\mathbb{R}^n$. As with the definitions of a planar partition and 3D space subdivision, this implies that there should be no overlapping primitives, and no gaps between them unless these gaps are considered to be outside the extent.

6 Conclusions and future work

While a clear definition of validity for 2D and 3D objects and space partitions is not currently provided explicitly in the most important 2D and 3D geoinformation standards, there are clear validity *notions* that are embedded in them. As this paper has shown, the notions behind these standards extend well to higher dimensions, including the modelling of multi-dimensional objects of any dimension, as well as the possibility to constrain them to form manifolds, be orientable or form space partitions.

Only minor modifications to the main geoinformation standards seem to be required in order to make them dimension-independent. Within this paper, we have shown a few of the possible ways to do these



(a)

(b)

Figure 7: (a) A ‘pinched cake’ configuration of a 3D cell complex formed by (b) joining four rectangular pyramids together is an example of a non-manifold that is nevertheless a quasi-manifold. The neighbourhood of the vertex highlighted as a red sphere in (a) is not homeomorphic to a ball, but it remains representable in the data structures that are based on joining pairs of combinatorial elements at the common boundary of two objects, such as the DCEL (Muller and Preparata, 1978) in 2D and generalised/combinatorial maps (Lienhardt, 1994; Damiand and Lienhardt, 2014) in *n*D. When such configurations are present in higher-dimensions, they can cause a mismatch between the point set topology manifold definitions used in the ISO standards with their explanations based on common pairs.

modifications, mainly concerning the addition of a dimension-independent `GM_Cell` class in ISO (2005), the related classes used to describe its boundary, and the addition of the subclasses of `GM_Complex` for dimension-independent cell complexes and of `GM_MultiPrimitive` for sets of disjoint cells. Some small redefinitions of the existing classes are however also necessary, such as changing the parent class of `GM_CompositeSolid` from `GM_Solid` to a new `GM_OrientableSolid` (from which from `GM_Solid` also should inherit).

In addition to the validity constraints enforced by the definition of the aforementioned classes, a few additional conditions should be explicitly added, which involve the relationships between cells and their boundaries (Section 5). We envision that these conditions can serve as a base for the future validation of *n*-dimensional objects as well as the automatic repair of at least some invalid configurations, such as non-manifolds through the use of duplicated cells.

Acknowledgements

This research is supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by the Ministry of Economic Affairs (Project code: 11300).

References

Arroyo Ohori, K., 2016. Higher-dimensional modelling of geographic information. PhD thesis, Delft University of Technology.

Arroyo Ohori, K., Ledoux, H. and Meijers, M., 2012. Validation and automatic repair of planar partitions using a constrained triangulation. Photogrammetrie, Fernerkundung, Geoinformation 5, pp. 613–630.

Damiand, G. and Lienhardt, P., 2014. Combinatorial Maps: Efficient Data Structures for Computer Graphics and Image Processing. CRC Press.

Gröger, G., Kolbe, T. H., Nagel, C. and Häfele, K.-H., 2012. OGC City Geography Markup Language (CityGML) Encoding Standard. Version 2.0.0. Open Geospatial Consortium.

Guibas, L. J. and Stolfi, J., 1985. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. ACM Transactions on Graphics 4(2), pp. 74–123.

ISO, 2005. Geographic Information — Spatial Schema. International Organization for Standardization.

ISO, 2006. Geographic information — Simple feature access — Part 1: Common architecture. International Organization for Standardization.

ISO, 2007. Geographic information — Schema for coverage geometry and functions. International Organization for Standardization.

Ledoux, H., 2013. On the validation of solids represented with the international standards for geographic information. Computer-Aided Civil and Infrastructure Engineering 28(9), pp. 693–706.

Ledoux, H., Arroyo Ohori, K. and Meijers, M., 2014. A triangulation-based approach to automatically repair GIS polygons. Computers & Geosciences 66, pp. 121–131.

Lienhardt, P., 1994. *n*-dimensional generalized combinatorial maps and cellular quasi-manifolds. International Journal of Computational Geometry and Applications 4(3), pp. 275–324.

McKenney, D., 1998. Model quality: The key to CAD/CAM/CAE interoperability. Technical report, International TechneGroup Incorporated.

Muller, D. E. and Preparata, F. P., 1978. Finding the intersection of two convex polyhedra. Theoretical Computer Science 7(2), pp. 217–236.

OGC, 2007. OpenGIS Geography Markup Language (GML) Encoding Standard. Version 3.2.1. Open Geospatial Consortium.

OGC, 2011. OpenGIS Implementation Specification for Geographic Information - Simple Feature Access - Part 1: Common Architecture. Version 1.2.1. Open Geospatial Consortium.

OGC, 2012. OGC Geography Markup Language — Extended schemas and encoding rules. Version 3.3.0. Open Geospatial Consortium.

Panigrahi, N., 2014. Computing in Geographic Information Systems. CRC Press.

Plümer, L. and Gröger, G., 1997. Achieving integrity in geographic information systems—maps and nested maps. GeoInformatica 1(4), pp. 345–367.

Poincaré, M., 1895. Analysis situs. Journal de l'École polytechnique 2(1), pp. 1–123.

van Oosterom, P. and Stoter, J., 2010. 5D data modelling: Full integration of 2D/3D space, time and scale dimensions. In: S. I. Fabrikant, T. Reichenbacher, M. van Kreveld and C. Schlieder (eds), Geographic Information Science: 6th International Conference, GIScience 2010, Zurich, Switzerland, September 14-17, 2010. Proceedings, Springer Berlin Heidelberg, pp. 311–324.

Zhao, J., Ledoux, H. and Stoter, J., 2013. Automatic repair of CityGML LoD2 buildings using shrink-wrapping. In: ISPRS 8th 3DGeoInfo Conference & WG II/2 Workshop, Vol. II-2/W1, ISPRS Archives, pp. 309–317.