Geometric Validation of GML Solids with the Constrained Delaunay Tetrahedralization*

Hugo Ledoux[1], Edward Verbree[1] and Hang Si[2]
[1]Delft University of Technology
[2]WIAS, Berlin

1 Introduction

To facilitate and encourage the exchange and interoperability of geographical information, the ISO[1] and the OGC[2] have developed in recent years standards that define what the basic geographical primitives are (ISO, TC211), and also how they can be implemented (OGC, 2007, 2006). While the definitions for the primitives are not restricted to 2D, most of the efforts for the implementation of these types have been done only in 2D. There exist indeed several tools to ensure that a line or a polygon in 2D is *valid*, i.e. it respects the standardised definitions: the JTS Topology Suite[3] and GEOS[4] are the most well-known and used tools, and are being used by different software packages. Although the topic might appear simplistic—"a polygon is simply a polygon, no?"—it is in practice a problem and is a topic of research, see van Oosterom et al. (2004) for a critic of the current ISO/OGC standards and an overview of the difficulties in 2D. It should be noticed here that validation is a necessary tool to guarantee the output of *processing* or *manipulation* GIS operations such as: calculation of the area of polygons; creation of buffers; conversion to other formats; boolean operations such as intersection, touch, contain, etc.

The efforts to implement in 3D a polyhedron type—also called a solid or a volume—and to be able to validate instances of the type are recent. Oracle Spatial has in their latest version implemented their own 3D type and are currently working on the geometric validation (Kazar et al., 2008). Interestingly, the Oracle type does not follow that of ISO/GML exactly, but Kazar et al. (2008) claim that this does not restrict the users and that it is simpler to implement. The validation of 3D polyhedron is also a topic that is being tackled in other disciplines (e.g. engineering where laser-scanned objects have to be modelled), but the definition of a polyhedron is often simpler and more restrictive than that of the ISO/GML used in GIS-related applications. Indeed, a polyhedron is composed only of an exterior boundary, and holes are not permitted. For examples of applications where polyhedra are validated and repaired see, among other, Guéziec et al. (2001), Attene and Falcidieno (2006) and Liepa (2003).

Two of the authors of this abstract have presented in a previous paper (Verbree and Si, 2008) an approach to the validation of one polyhedron: the *constrained Delaunay tetrahedralization* (CDT) was used as a supporting data structure. The CDT permits us to circumvent and simplify the difficulties arising with the method of Kazar et al. (2008), who uses an "edge-face" approach. The original paper

*A version of this published in the Proceedings of the 4th International Workshop on 3D Geo-Information, 2009, p. 143–148. Ghent, Belgium
[1]International Organization for Standardization
[2]Open Geospatial Consortium
[3]`http://www.vividsolutions.com/jts`
[4]Geometry Engine, Open Source: `http://trac.osgeo.org/geos`

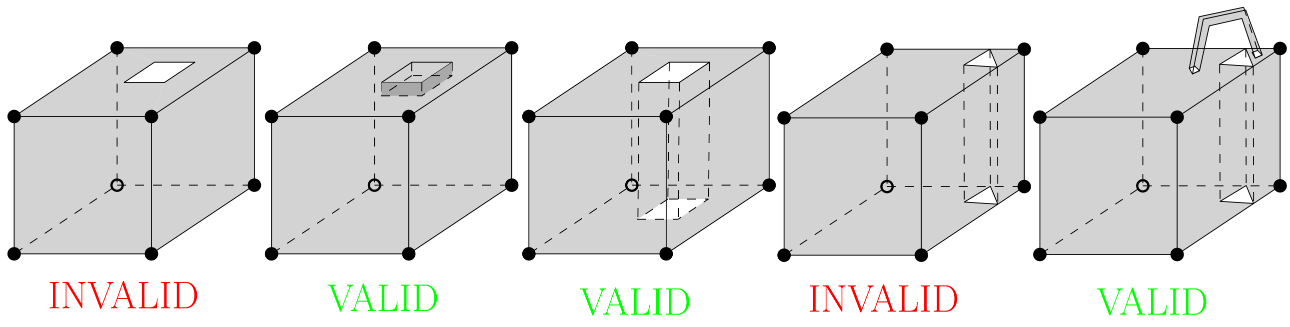


Figure 1: Five different polyhedra. (Figure after Kazar et al. (2008))

was restricted to simple polyhedra, and did not consider explicitly holes in polyhedra. We are currently working on the extension of this work so that polyhedra stored in GML (which can have holes) can be validated, and potentially also automatically repaired. The presence of holes significantly increases the complexity of the problem, as holes are themselves also polyhedra. We present briefly in this abstract the ISO/GML's definition of a polyhedron, the CDT and describe how it can help us to test if one polyhedron is valid or not. We believe our method to be significantly better at catching special geometric configurations that often arise with real-world data.

2 ISO Solids and their Validation

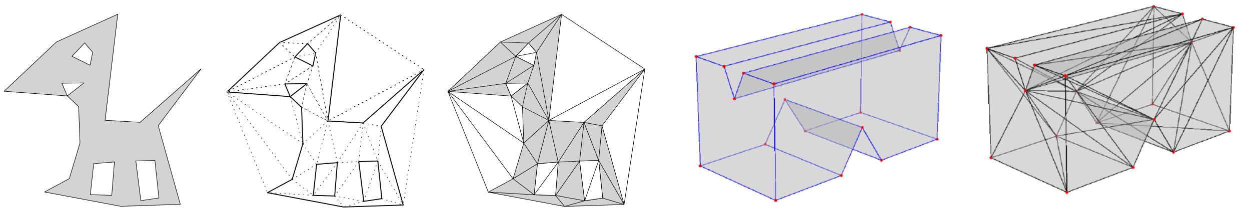
While a polyhedron/solid has different definitions in different disciplines, we focus here on the definition given in the ISO standards (ISO, TC211) and implemented with GML (OGC, 2007). A GML `Solid` "is the basis for 3-dimensional geometry. The extent of a solid is defined by the boundary surfaces as specified in ISO 19107:2003. `gml:exterior` specifies the outer boundary, `gml:interior` the inner boundary of the solid" (OGC, 2007). Without going into all the details, we can state that a solid is represented by its boundaries (surfaces), and that like its counterpart in 2D (the polygon), a solid can have "holes" (inner shells) that are allowed to touch each others, or the outer boundary, under certain circumstances.

To be considered a *valid* solid, a solid must be fulfil several properties. The most important are: (i) it must be simple (no self-intersection of its boundary); (ii) it must be closed, or 'watertight'; (iii) its interior must be connected; (iv) its boundary surfaces must be properly oriented; (v) its surfaces are not allowed to overlap each others. Figure 1 shows five different polyhedra, some of them valid, some not. The first is invalid because it is not watertight; the fourth one is invalid because the hole separate the interior of the polyhedron into 2 non-connected parts; the fifth one is valid since, as Kazar et al. (2008) points out, the "handle" now creates a connected interior. It should also be noticed that since a solid is formed of 2D primitives (embedded in 3D space), these also have to be valid. For instance, if a surface has a hole (an inner ring), than this ring is not allowed to to overlap with the outer boundary of the surface.

For more details about validation rules in 3D, the reader is directed towards ISO (TC211) and Kazar et al. (2008).

3 Constrained Delaunay tetrahedralization

The constrained Delaunay tetrahedralization is the 3D counterpart of a constrained Delaunay triangulation. Just like in 2D, the CDT permits us to decompose an object (a polyhedron in our case) into non-overlapping tetrahedra. This is shown in Figure 2 for the 2D and the 3D cases. Observe here that in 2D any polygon can be triangulated, but in 3D this is more complex as new vertices often



(a) (b)

Figure 2: **(a)** Left: a 2D polygon containing 4 holes. Middle and right: the constrained Delaunay triangulation of the polygon. **(b)** A polyhedron and its CDT.

have to be inserted. It is nevertheless known that it is always possible to decompose a polyhedron into tetrahedra if new vertices are allowed (Cohen-Steiner et al., 2004). These extra vertices do not modify the shape of the polyhedron. For more information about the CDT see for instance Shewchuk (1997) and Si (2008).

4 Validation with the CDT

To our knowledge the only other work related to the geometric validation of 3D polyhedra with holes is that of Kazar et al. (2008). They take a different approach: they build the graph (nodes and vertices) of the polyhedron and performs graph-traversal algorithms to validate. While this method works for the simpler cases, some configurations such as the one with the handle in Figure 1 are extremely difficult to catch. We believe that instead of building a edge-based data structure to validate, a space-filling data structure would give much better results. Nooruddin and Turk (2003) also tried space-filling methods, but used voxels (which could introduce errors depending on the resolution used). Our approach to geometric validation consists of buildings the CDT of the input polyhedron, and use the properties of the CDT to verify if it fulfils the definition of a valid polyhedron. It is also a space-filling data structure, but does not have the shortcomings of raster.

We have identified 11 tests that need to be performed in order to ensure that one polyhedron is geometrically valid. Some of them are simple and do not require the auxiliary data structure. For instance, verifying that a surface is planar simply implies checking that all its points are on a plane; and verifying that the boundary of a surface is closed implies that the first and last points are the same. Observe here that the former implies the use of a *tolerance* for validation, as in van Oosterom et al. (2004). For the verification tests that require the CDT we use TetGen (Si, 2004). Our method is conceptually simple: we tetrahedralize a polyhedron, and then assign to each tetrahedron a IN or OUT flag; this can be done automatically by "walking" in the CDT from the exterior and stopping at constraints. Difficult tasks like verifying the connectedness of a polyhedron are immensely simple once the CDT is built: we start at a point inside the polyhedron, and perform a depth-first search on the dual graph of the CDT, counting the tetrahedra visited on the way (constraints obviously stop you). Then the connectedness test boils down to comparing the number of visited tetrahedra with the total number of tetrahedra flagged as IN. Another example is the test to verify if a polyhedron is watertight. With graph-based methods this is not trivial, but if the method of walking from the outside just described is used then again the result is automatically found: all the tetrahedra are flagged as OUT. Lack of space prevents us from presenting all the 11 verification tests that we designed, but other tests are also simplified if a CDT is first built.

At the conference, we will discuss these tests and show our latest developments.

References

M. Attene and B. Falcidieno. ReMESH: An interactive environment to edit and repair triangle meshes. In *Proceedings IEEE International Conference on Shape Modeling and Applications*, page 41, Matsushima, Japan, 2006.

D. Cohen-Steiner, E. Colin de Verdière, and M. Yvinec. Conforming Delaunay triangulations in 3d. *Computational Geometry—Theory and Applications*, 28:217–233, 2004.

A. Guéziec, G. Taubin, F. Lazarus, and B. Horn. Cutting and stitching: Converting sets of polygons to manifold surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 07(2):136–151, 2001. ISSN 1077-2626.

ISO(TC211). ISO 19107:2003: Geographic information—Spatial schema. International Organization for Standardization, 2003.

B. M. Kazar, R. Kothuri, P. van Oosterom, and S. Ravada. On valid and invalid three-dimensional geometries. In P. van Oosterom, S. Zlatanova, F. Penninga, and E. Fendel, editors, *Advances in 3D Geoinformation Systems*, Lectures Notes in Geoinformation and Cartography, chapter 2, pages 19–46. Spinger, 2008.

P. Liepa. Filling holes in meshes. In *Proceedings 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 200–205, 2003.

F. Nooruddin and G. Turk. Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):191–205, 2003.

OGC. Geography markup language (GML) encoding standard. Open Geospatial Consortium inc., 2007. Document 07-036, version 3.2.1.

OGC. OpenGIS implementation specification for geographic information—simple feature access. Open Geospatial Consortium inc., 2006. Document 06-103r3.

J. R. Shewchuk. *Delaunay Refinement Mesh Generation.* PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburg, USA, 1997.

H. Si. Tetgen: A quality tetrahedral mesh generator and three-dimensional Delaunay triangulator. User's manual v1.3 9, WIAS, Berlin, Germany, 2004.

H. Si. *Three Dimensional Boundary Conforming Delaunay Mesh Generation.* PhD thesis, Berlin Institute of Technology, Berlin, Germany, 2008.

P. van Oosterom, W. Quak, and T. Tijssen. About invalid, valid and clean polygons. In P. F. Fisher, editor, *Developments in Spatial Data Handling—11th International Symposium on Spatial Data Handling*, pages 1–16. Springer, 2004.

E. Verbree and H. Si. Validation and storage of polyhedra through constrained Delaunay tetrahedralization. In T. J. Cova, H. J. Miller, K. Beard, A. U. Frank, and M. F. Goodchild, editors, *Proceedings 5th international conference on Geographic Information Science*, volume 5266 of *Lecture Notes in Computer Science*, pages 354–369. Springer, 2008.