Validity of Mixed 2D and 3D Cadastral Parcels in the Land Administration Domain Model

Rodney James THOMPSON, Australia and Peter VAN OOSTEROM, The Netherlands

Key words: Integrating 2D and 3D, Parcels, Validation, LADM

SUMMARY

The Land Administration Domain Model (ISO 19152) (ISO-TC211 2012) defines a specific geometry form which allows the clean and simple mixing of 3D cadastral objects such as volumetric parcels with the more common 2D parcels. It does this by introducing a concept of boundary surfaces, known as “face strings”. The unbounded nature in the vertical direction of the “face strings” corresponds to the unbounded nature of 2D parcels, that is, they comprise the boundaries of infinite 3D columns of space (see ISO-TC211 2012 Figure B.2 Page 48). This, in combination with the more usual finite “face” used to define a completely bounded 3D parcel, allows a highly efficient storage structure, without sacrificing the rigour of the definitions (while remaining compatible with the vast amount of existing 2D cadastral data).

A set of axioms has been developed that can be used for formalizing the validation of individual completely bounded 3D cadastral parcels in general, or for formalizing the validation of a set forming a complete coverage of completely bounded 3D parcels (Thompson and van Oosterom 2011; Thompson and van Oosterom 2012). This paper extends this set of geometric validity axioms specifically to the LADM, ensuring that the individual parcels can be correctly validated, whether they be: 1. defined by 2D primitives (the “face strings”, representing unbounded 3D columns), 2. defined by 3D primitives (“faces”, representing bounded 3D spaces), or 3. defined by a combination of the two (see ISO-TC211 2012 Figure B.4 Page 49). It will also be ensured that the “outside of the world” is correctly handled. That is to say, where cadastral parcels abut the outer limit of the jurisdictional region. Further, the validity of the remaining top and bottom parts of the columns of the volume of interest (after “extracting” completely bounded 3D parcels) is explored. These parts also correspond to partially bounded 3D parcels. Finally, the validity of “liminal” parcels (2D parcels which abut 3D parcels) is addressed.

The other major aspect considered is that the LADM defines several levels of encoding – from the purely textural definition of parcels through point-based parcels, line based parcels and polygon based parcels to the full topological encoding of a parcel coverage. Not all of these are equally suitable for geometric validation, but those that are are identified, and the appropriate axioms and necessary extensions are identified. The axioms have to be ‘translated’ and applied at the appropriate encoding level. It should be noted that axioms use concepts that typically belong to (or best fit) a certain encoding level; e.g. using topology concepts such as node and edge.

Validity of Mixed 2D and 3D Cadastral Parcels in the Land Administration Domain Model

Rodney James THOMPSON, Australia and Peter VAN OOSTEROM, The Netherlands

1. INTRODUCTION

In the move towards a 3D Cadastre, many jurisdictions are considering a hybrid 2D/3D database as either a stage of development or as a target in itself (van Oosterom, Stoter, Ploeger, Thompson and Karki 2011). The Land Administration Domain Model (LADM), which is the underlying model for the ISO 19152 standard assists in this approach by defining a data structure which allows a relatively seamless mixing of 2D and 3D "spatial units" (Lemmen, Van Oosterom, Thompson, Hespanha and Uitermark 2010). This standard is in its final stages of approval (Lemmen, Uitermark and van Oosterom 2012).

For the purpose of this paper, there is no important distinction to be made between "parcels" "lots" or "spatial units", since it is only the spatial aspects (topology and geometry) that are being considered, so the term "parcel" will be used to mean the extent of land or space being considered.

1.1 The Validation of Plans of Survey

It is or should be common practice that data to be included in a database be validated. Typically this is to ensure that the incoming data is compatible with the database design assumptions. For example, if an attribute of "length" is being entered, it may well be validated to ensure that it is numeric, and within a specified value range (e.g. value > 0). This does not, of course, ensure that the value is correct, but may catch some errors.

By contrast, many jurisdictions specify a document (paper or computer file) which is used to define the spatial extents of a cadastral object. For example, a traditional "survey plan" may show the extents of a property, along with enough supporting information to be sure that the property can be legally identified. This frequently is separate from the "Title" or other document that defines the rights that certain parties have to the property. The same happens in the 3D world, where a property's geometry may be defined in terms of its "metes and bounds". It is important that a survey plan document can be validated to ensure it can define the extents of a cadastral object in an unambiguous way that can stand up to legal scrutiny. For example, it must be possible to determine rigorously what subset of space is "within" the cadastral parcel – and therefore who has rights to it.

The significant difference in this case is that there may not even be a database that is being loaded with the parcel definitions. For example, in Australia cadastral databases (DCDBs) are limited to 2 spatial dimensions, while many cadastral parcels and plans are 3D.

Historically, the paper plans of survey were inspected by officers in the registering authority, but with the rising complexity of subdivision, and in particular with the definition of 3D parcels, the manual "validation" proved to be too time consuming and error prone (especially for more complex configurations). Thus it is necessary to develop an automatic validation suite to ensure that the standard of survey information is sufficient to the task. It could be argued that the fact that there is no 3D cadastral database behind the validation actually makes the task more difficult. Checking an individual 3D object may be feasible, but it is may be more difficult to check a new 3D object in relation to existing 2D and 3D objects for potential unwanted overlaps or gaps.

1.2 3D Geometric Validation

Much work has been done on the validation of 3D objects, but it has almost invariably been aligned towards the entry of data to a database. For example Kazar, Kothuri, van Oosterom and Ravada (2008) is designed to control the entry of objects to an Oracle database. While it may be acceptable as a database geometry structure, it would not be reasonable to insist that a survey plan have “construction lines” included to ensure that faces can be defined as “simple polygons” just because Oracle allows no inner rings in faces. Many approaches to 3D geometry also draw on the concept that the boundary of a 3D region should be a 2D manifold (Gröger and Plümer 2011), but while this makes the definition and processing (validation) simpler, it is also a rather arbitrary rule to enforce on a cadastral plan (Thompson and van Oosterom 2012).

There are many structures proposed for the storage, manipulation and retrieval of 3D geometric objects in terms of boundary representations (b-rep) (Ledoux and Gold 2007; Boguslawski and Gold 2009), including non-manifold objects (De Floriani and Hui 2003).

The concept of “2.8D” geometry was introduced by Gröger and Plümer (2005) as an extension of the planar polygon coverage, to include a third dimension, and to allow overhangs, and finally “handles” (such as bridges and tunnels). This resulted in a rigorous set of axioms (Gröger and Plümer 2011) that allow a well defined set of validation rules to be written, and finally a set of transactions that allow update while ensuring validity is maintained (Gröger and Plümer 2012); see Figure 1B. This work has formed a starting point for the set of axioms presented here, with restatement as needed to accommodate:

- Non-manifold boundaries of cadastral parcels
- The mixture of 2D and 3D parcels found in a cadastre (*viz* LADM).
- The requirement to validate 2D and 3D parcels together.
- The fact of non-exact mathematical representation.

Belussi, Liguori, Marca, Migliorini, Negri, Pelagatti and Visentini (2011) take a different, but compatible approach to the validation of (possibly 3D) datasets, using a schema defined in UML (Unified Modelling Language) and OCL (Object Constraint Language), and known as GeoUML. They approach the issue of non-exact number representation and arithmetic using a requirement similar to that of Milenkovic (1988), that a buffer of at least ten times the “internal resolution” must be provided between distinct geometric objects. Parcel definitions could be extended to a full 3D coverage of the region, defining arbitrary symbolic “Top” and “Bottom” planes to encompass all the 3D parcels, and redefining 2D parcels to extend vertically to these planes (Figure 1C), but this is wasteful of storage.

1.3 Plans in 2 and 3 Dimensions

It might be possible to cover the mixture of 2D and 3D parcels commonly found in a practical cadastre using a non-manifold representation such as that of De Floriani and Hui (2003) (see Figure 1A) which allows a mixture of 3D, 2D, 1D and 0D primitives to be used to define geometric objects. It might be thought that a subset of this (3D and 2D only) could be used, and this would be the case if the aim were just to develop a database structure to store cadastre. The problem domain we are considering here is the validation of plan information (with description of extend of spatial objects), which requires more rigour in some respects (Belussi et al. 2011), but conversely cannot impose arbitrary restrictions.

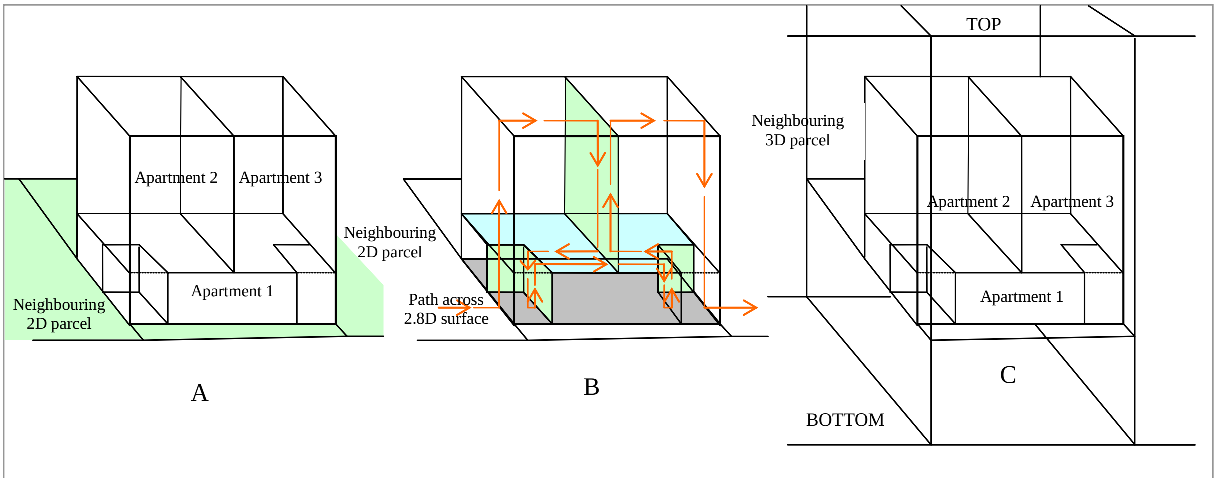


Figure 1. Three possible representations of a complex with three apartments (based on Stoter and van Oosterom 2006 Page 41), however in this example it is assumed that the neighbouring parcels are 2D: A. represented as a non-manifold complex (with the 2D parts in colour). B. Represented as a 2.8D map. (The grey surface is not stored, while the coloured surfaces are used twice and the white surfaces are used once. A path across the surfaces is shown in coloured arrows). C. Represented as a full 3D coverage, with arbitrary symbolic top and bottom surfaces. Neighbouring parcels are converted to 3D parcels.

The technique suggested by the LADM is ideally suited to this mixture of 2D and 3D representation of parcels. 2D parcels typically account for the vast majority of the land surface, and can be individually quite large and complex, or in many cases may be simple rectangles. 3D parcels on the other hand are usually much fewer in number, are relatively simple in terms of number of vertices (compared to the total number of vertices in the DCDB), but high in monetary value. The LADM approach allows the 2D parcels to be represented as simple "GM_Curve" objects (Lemmen et al. 2010), but to be thought of as 3D prisms of space with no defined top or bottom: open columns. This makes for a smooth transition between 2D and 3D areas of subdivision. For example, using the polygon encoded form, the apartments and their 2D neighbour would be encoded as in Figure 2. All that needs to be stored for the 2D parcel is a single 2D gm-curve. A point $p = (x,y,z)$ in space is to be considered to be within the 2D parcel if the point $p' = (x,y,0)$ is within the gm-curve. That is to say, even though a 2D object is stored, it represents a 3D region of space. 3D parcels are represented using a fairly conventional b-rep, but, as described below, with an extension to allow spaces to be partially unbounded above and/or below. Note that in Figure 2, the apartments are fully bounded, but the common property is unbounded above and below.

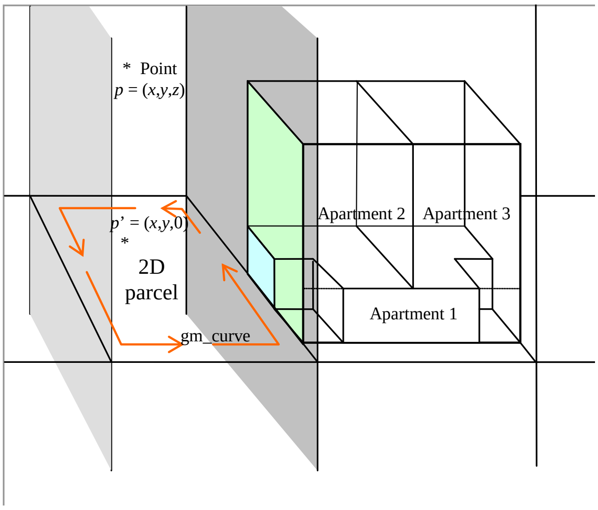


Figure 2. The same three apartments represented in LADM polygon encoded form. The faces that adjoin the 2D neighbouring parcel are highlighted (blue for apartment 1, green for apartment 2, grey for the common property). The light grey surface represents a face string between two 2D parcels.

There is an additional complexity where topological encoding is being used (as allowed by the LADM). In this case, a face of a 3D parcel must always be matched with an anti-equal face of an adjoining parcel. In Figure 2, this means that the 2D neighbour must be defined with faces to match apartments 1 and 2, and therefore cannot be built from GM_curves alone. This case of a 2D parcel that must be defined by a mixture of faces and face strings is known as a "liminal" parcel representation. This will be discussed in detail later.

This paper further defines the mathematical consequences of this mixture of representations. First in section 2 the relevant concepts are defined. Next a set of geometric validation axioms are presented in section 3, that take the consequences of mixing 2D and 3D representations into account. Section 4 treats in more detail the concept of connectivity, while section 5 addresses the concept of a space partition in the context of topology based spatial units. Finally, conclusions and future work are described in section 6.

2. DEFINITIONS

Because the LADM defines the concept of parcels which are unbounded above or below (or both), we need to extend the concept of "point" to include points of unspecified Z value – points at "infinity".

2.1 Points and Nodes

We define the concept of infinity in this approach as a particular bit pattern which behaves in a way analogous the mathematical concept of infinity. It may already be available in a particular computer language or number encoding – e.g. the IEEE floating point infinity (Goldberg 1991), or the implementer may choose a particular number (e.g. 2^{31}-1 = 2147483647). For real points with coordinates $x, y, z \in \mathbf{R}$ The critical behavior is:

$$\forall x \in \mathbf{R} \Rightarrow x < \infty.$$
$$\forall x \in \mathbf{R} \Rightarrow x > -\infty.$$

We now define $\mathbf{P}$ to be $\mathbf{R} \cup \{-\infty, \infty\}$, and $\mathbf{P}^3$ as $\{(x,y,z): x,y \in \mathbf{R}, z \in \mathbf{P}\}$.

2.2 Distances

For $p_1 = (x_1,y_1,z_1)$, $p_2 = (x_2,y_2,z_2) \in \mathbf{P^3}$,

if $z_1 \neq \pm\infty$ and $z_2 \neq \pm\infty$

$$D(p_1, p_2) =_{def} \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \ . \qquad (D1)$$

if $z_1 = \pm\infty$,

$$D(p_1, p_2) =_{def} \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \text{ if } z_2 = z_1 \qquad (D2)$$

$D(p_1, p_2) = \infty$ otherwise.

if $z_2 = \pm\infty$,

$$D(p_1, p_2) =_{def} \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \text{ if } z_2 = z_1 \qquad (D3)$$

$D(p_1, p_2) = \infty$ otherwise.

Corollary $D(p_1, p_2) = D(p_2, p_1)$ (C1)

Point equality:

$$p_1 = p_2 =_{def} x_1{=}x_2 \wedge y_1{=}y_2 \wedge z_1{=}z_2 \qquad (D4)$$

Corollary $p_1 = p_2 \Rightarrow D(p_1, p_2) = 0$; (C2)

In what follows, where the context is clear, the definitions of variables are omitted. For example, if p_1 represents a point, the definition $p_1 \in \mathbf{P^3}$ will be omitted.

A **node** $n \in \mathbf{P^3}$ is a special case of point, which can be represented in the number system of the computer (for example as a tuple of floating point numbers).

A **node at infinity** is a node with coordinates (x,y,z) where $z = +\infty$ or $z = -\infty$.

Let N be the set of all possible nodes, $N \subset \mathbf{P^3}$. Note that N is a finite set, $\mathbf{P^3}$ is uncountably infinite.

Directed Edges

A **directed-edge** e is a straight line segment between a pair of nodes: $e = (n_1, n_2)$:

Let E be the set of all possible directed-edges.

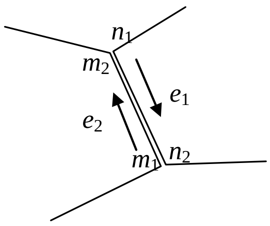


Figure 3 Anti-equal directed edges $e_1 \Lsh\!\Rsh e_2$

For directed edges $e_1 = (n_1, n_2)$, $e_2 = (m_1, m_2) \in E$,

$$e_1 \parallel e_2 =_{def} n_1 = m_1 \wedge n_2 = m_2 \text{ (equal edges)} \qquad (D5)$$

$$e_1 \upharpoonleft\!\downharpoonright e_2 =_{def} n_1 = m_2 \wedge n_2 = m_1 \text{ (anti-equal - see Figure 3)} \qquad (D6)$$

The notation is used that $n \in e$ means if $e = (n_1, n_2)$ then $n = n_1$ or $n = n_2$.

A directed edge at + infinity is a directed edge

$$e = (n_1, n_2), n_1 = (x_1,y_1,z_1), n_2 = (x_2,y_2,z_2), z_1 = z_2 = +\infty.$$

A directed edge at - infinity is a directed edge

$e = (n_1, n_2)$, $n_1 = (x_1,y_1,z_1)$, $n_2 = (x_2,y_2,z_2)$, $z_1 = z_2 = -\infty$.

A directed edge at infinity is a directed edge at + infinity or at – infinity.

A directed edge extending to + infinity is a directed edge

$e = ((x_1,y_1,z_1), (x_2,y_2,\infty))$, $z_1 \neq \infty$. (Note – this definition does not specify that $x_1 = y_1$, $x_2 = y_2$. That requirement will introduced as Axiom AE1 below).

A directed edge extending from + infinity is a directed edge

$e = ((x_1,y_1, \infty), (x_2,y_2,z_2))$, $z_2 \neq \infty$.

A directed edge extending to and from - infinity are similarly defined.

An open directed edge is a directed edge running to or from $\pm\infty$.

A finite edge is one whose end points are finite (i.e. it is not open or at infinity).

A tall directed edge is a directed edge extending between $+\infty$ and $-\infty$.

Axiom AE1

If $e = ((x_1,y_1,z_1), (x_2,y_2,z_2))$ is an open directed edge, then $x_1 = x_2$ and $y_1 = y_2$.

(That is to say, any open edge must be vertical. This is necessary because otherwise it is not possible to assign a meaningful value to the slope of the edge).

Definition of Point on Directed Edge

For a finite directed edge, $e = (n_1,n_2)$, $n_1 = (x_1,y_1,z_1)$, $n_2 = (x_2,y_2,z_2)$:

$\mathrm{On}(p, e) =_{\mathrm{def}} \exists\, t \in \mathbf{R}: 0 \leq t \leq 1, x = x_2 + t(x_1-x_2), y = y_2 + t(y_1-y_2), z = z_2 + t(z_1-z_2)$. (D7)

For a directed edge extending to $+\infty$ $e = (n_1,n_2)$, $n_1 = (x_1,y_1,z_1)$, $n_2 = (x_2,y_2, \infty)$:

$\mathrm{On}(p, e) =_{\mathrm{def}} x = x_1, y = y_1, z \geq z_1$ (note $x_2=x_1$ and $y_2=y_1$ by (AE1)). (D8)

The definitions for other open directed edges are equivalent.

For a directed edge at $+\infty$ $e = ((x_1, y_1, \infty), (x_2, y_2, \infty))$

$\mathrm{On}(p, e) =_{\mathrm{def}} \exists\, t \in \mathbf{R}: 0 \leq t \leq 1, x = x_2 + t(x_1-x_2), y = y_2 + t(y_1-y_2), z = \infty$. (D9)

For a directed edge at $-\infty$ the definition is equivalent.

Distance from Directed Edges

For directed-edge e, point p,

$\mathrm{D}(p, e) =_{\mathrm{def}} \min_{On(p_1,e)} \mathrm{D}(p, p_1)$. (D10)

For directed-edges e_1, e_2,

$\mathrm{D}(e_1,e_2) =_{\mathrm{def}} \min_{On(p_1,e_1)} \mathrm{D}(p_1,e_2)$. (D11)

(These apply equally for edges at or extending to $\pm\infty$).

Collary

For an edge at infinity, the distance to any finite point or edge is ∞. (C3)

Definition of Face

A **face** f is defined as a set of at least 3 nodes f_n, a set of directed-edges f_e and a tuple of numbers $f_p = (a,b,c,d)$: $a,b,c,d \in \mathbf{R}$ restricted as follows:

$\forall e = (n_1,n_2) \in f_e: n_1, n_2 \in f_n$. (D12)

$\forall n \in f_n: \{e_1=(n_1,n_2): n_1 = n\}$ and $\{e_2=(n_1,n_2): n_2 = n\}$ are of same cardinality. (D13)

$a^2+b^2+c^2 = 1$; (D14)

$f =_{\mathrm{def}} (f_n, f_e, f_p)$. (D15)

Where the context is clear, f will be used to mean f_e, f_n or f_p. e.g. $n \in f$.

The plane for face $f_p = (a,b,c,d)$ is

$f_p = \{p = (x,y,z) \in \mathbf{P^3}: ax+by+cz+d = 0\}$. (D16)

That is to say, a face is defined as a set of nodes, directed edges and a normalised plane definition, such that the nodes for every directed edge are included, and every directed edge running into a node can be paired with a directed edge running from that node. Note that this definition does not require the nodes or edges to lie on the plane. This requirement will be introduced later as Axiom A8.

Let F the set of all possible faces.

Inverse of a Face

The inverse of a face $f = f(f_n, f_e, f_p)$ is defined as: $-f =_{def} f(f_n, -f_e, -f_p)$. where $-f_e = \{e(n_1, n_2): e(n_2, n_1) \in f_e\}$, and $-f_p = (-a,-b,-c,-d)$ – that is the same face but with the reversed sense.

Axiom A0

For any faces defined on the same set of nodes, the plane parameters must agree.

For $f = f(f_n, f_e, f_p), f' = f'(f'_n, f'_e, f'_p), f_n = f'_n \Rightarrow f_p = f'_p \vee f_p = -f'_p$.

This is potentially a difficult issue for the implementor, because if the planar parameters are not supplied, it is up to the receiving program to determine them. This means that the algorithm must be exactly repeatable, or that the equality of the node sets must be detected, and the calculation carried out once only.

If this constraint is not respected, many of the tests related to dihedral angles between faces will be complicated and difficult to make consistent.

Angle between Directed Edges at shared node

For $e_1, e_2, n \in f, n \in e_1, n \in e_2$, let $a(e_1, e_2, n)$ be the angle between e_1 and e_2 at n measured anticlockwise around n, as viewed from outside the face (i.e. from the side of the face for which $ax+by+cz+d > 0$) (see Figure 4 left). (D17)

In calculating angles where one or other of the directed edges is open or at infinity, the z value of ∞ can be replaced by a large number (larger than any other z value, but consistent for all nodes). Likewise -∞ can be replaced by a large negative number. The axiom (AE1) will ensure that edges running to/from infinity are vertical, and edges at infinity are calculated as horizontal.

Angle between Faces (at an edge)

For $e_1 \in f_1, e_2 \in f_2, e_1 = e_2 \vee e_1 \rightleftharpoons e_2$, let $A(f_1, f_2, e_1)$ be the dihedral angle between f_1 and f_2 at e_1 measured anticlockwise around the directed-edge looking in the direction of the edge – so that $A(f_1, f_2, e_1) = -A(f_1, f_2, e_2) = -A(f_2, f_1, e_1) = A(f_2, f_1, e_2)$. (see Figure 4 right). (D18)

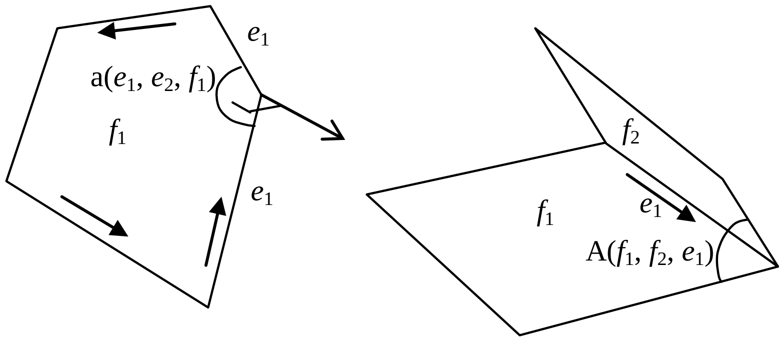


Figure 4. Left: angle between edges a(e_1, e_2, n), and Right: angle between faces A(f_1, f_2, e_1)

Distance from a Face

For plane f_p and point $p = (x,y,z)$,

$D(p, f_p) = |ax+by+cz+d|$. (D19)

$\forall\ n_i \in f$, let n_i' be the point at the base of the normal from n_i to f_p. The points n_i' form a planar multi-polygon.

Let On(p, f) be $D(p, f_p) = 0 \wedge$ (p is inside the closure of the planar polygon). (D20)

For face f, point p, $D(p, f) = \min_{On(p_1, f)} D(p, p_1)$. (D21)

Definition of a Shell

A **shell** $s = (s_f, s_e, s_n)$ is a set of faces s_f and their associated directed edges s_e and nodes s_n.

$s_f \subseteq F$ (D22)

$s_e = \{e: \exists f: e \in f \wedge f \in s_f\}$ (D23)

$s_n = \{n: \exists f: n \in f \wedge f \in s_f\}$ (D24)

shell $s =_{\text{def}} (s_f, s_e, s_n)$ (D25)

Note that this definition does not require a shell to close. Later the concept of a “cycle shell” will be introduced as a partially bounded (“closed”) shell.
Where the context is clear, s will be used to mean s_f, s_e or s_n, e.g. $e \in s_e$.

Definition of an Undirected Edge

An edge is a collection of the directed edges of a shell that define the junction between faces.
An **edge** u (s, n_1, n_2) within the shell s is defined as:

$u(s, n_1, n_2) = \{e = (m_1,m_2): e \in s \wedge m_1 = n_1 \wedge m_2 = n_2\}$
$\quad \cup \{e = (m_1,m_2): e \in s \wedge m_1 = n_2 \wedge m_2 = n_1\}$. (D26)

Corollary $u(s, n_1, n_2) = u(s, n_2, n_1)$, hence u is undirected. (C4)
Corollary if $e_1, e_2 \in u$, $e_1 \parallel e_2$ or $e_1 \upharpoonleft\!\downharpoonright e_2$. (C5)
Let the set of all undirected edges be U.

Definition of Corner and Fold

A **corner** $v_2(f, e_1, n, e_2)$ within face f is the meeting at node n of two directed edges e_1 and e_2 such that

$\exists\ n \in N: (e_1, e_2, n \in f, n \in e_1, n \in e_2)$ and
$\quad e \in f, n \in e \Rightarrow a(e_1, e_2, n) < a(e_1, e, n)$ (D27)

(Descriptively, a corner is a pair of edges meeting at a node, with no intervening edges in the same face. Note - if the edges do not have a node in common, a corner is not defined – see Figure 5A).

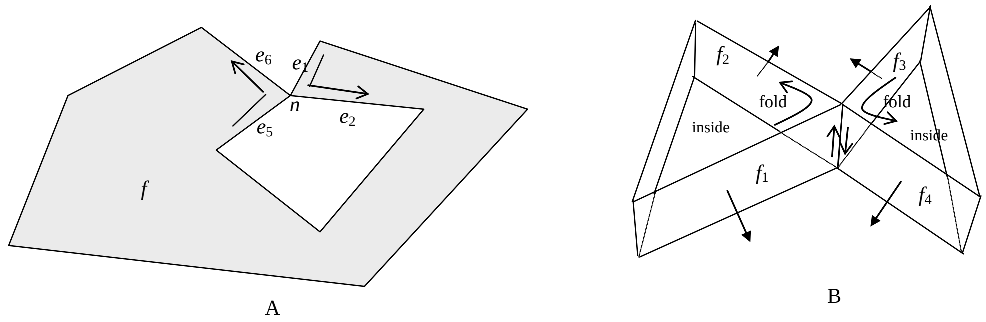


Figure 5. A. a single face with two corners marked. Note that (f, e_1, n, e_2), (f, e_2, n, e_3),…, (f, e_5, n, e_6),… are all corners, but that (f, e_1, n, e_6) is not a corner, because e_2 intervenes. B. An edge which is the meeting of 4 faces, 2 folds and 4 directed edges

Let V_2 be the set of all corners.

Notation: $e \in v_2$ where $v_2 = v_2\,(f, e_1, n, e_2)$, ($v_2 \in V_2$) is taken to mean $e = e_1$ or $e = e_2$.

A **fold** $v_3(s, f_1, e_1, f_2)$ within shell s is the meeting of two faces f_1 and f_2 at directed edges e_1 and e_2 such that $f_1 \in s$, $f_2 \in s$, $e_1 \in f_1$, $e_2 \in f_2$, $e_1 \rightleftharpoons e_2$ and:

$$(f \in s, e \in f, e \upharpoonleft\!\downharpoonright e_1) \Rightarrow \mathrm{A}(f_1, f_2, e_1) < \mathrm{A}(f_1, f, e_1) \text{ and}$$
$$(f \in s, e \in f, e \parallel e_1) \Rightarrow \mathrm{A}(f_1, f_2, e_1) \leq \mathrm{A}(f_1, -f, e_1). \qquad \text{(D28)}$$

(Descriptively, a fold is a pair of faces that meet at an edge, with no intervening faces at the same edge between them - see Figure 5B).

Let V_3 be the set of all folds.

Notation: $f \in v_3$ where $v_3 = v_3(s, f_1, e_1, f_2)$, ($v_3 \in V_3$) is taken to mean $f = f_1$ or $f = f_2$, while $e \in v_3$ is taken to mean $e \parallel e_1$ or $e \upharpoonleft\!\downharpoonright e_1 \wedge e \in f_2$.

Corollary: a fold is a subset of an undirected edge. (C6)

3. THE AXIOMS

These are reprinted from Thompson and Van Oosterom (2011) as a convenience, however axioms A2 and A6 have been modified to exclude their application at $\pm\infty$.

Axiom A1: $n_1, n_2 \in s$: $(n_1 \neq n_2) \Rightarrow \mathrm{D}(n_1, n_2) > \varepsilon$. No two nodes are closer than ε apart.

Axiom A2: $\forall n = (x, y, z) \in s, -\infty < z < \infty$: $\exists f_1, f_2, f_3 \in s$: $n \in f_1, n \in f_2, n \in f_3$ $f_1 \neq f_2 \wedge f_2 \neq f_3 \wedge f_1 \neq f_3$. Each finite node has at least 3 incident faces. (Optional axiom).

Axiom A3: $\forall n \in s$: $\forall f_1, f_2 \in s, n \in f_1, n \in f_2$; $\mathrm{On}(p, f_1), \mathrm{On}(p, f_2), \mathrm{D}(p, n) > \varepsilon$ $\Rightarrow \exists\, e_1 \in f_1 \wedge e_2 \in f_2 \wedge (e_1 \upharpoonleft\!\downharpoonright e_2 \vee e_1 \parallel e_2)$. The faces incident at a node do not intersect one another except at a common edge.

Axiom A5[1]: $\forall e_1, e_2 \in s$, $D(e_1, e_2) < \varepsilon \Rightarrow \exists\, n: n \in e_1, n \in e_2$. Non-intersecting edges must not be within a distance ε of each other

Axiom A6: $\forall e \in s$, e not at infinity, $\exists\, v \in s: e \in v$. Every directed-edge of a face in the shell except those at infinity, belongs to a fold. (Note that edges at infinity have no adjoining edges – see (C12) below. Edges running to or from ±∞ do follow this axiom).

Axiom A7: $\forall n \in s, \exists f_1, f_2 \in s: n \in f_1, n \in f_2, f_1 \neq f_2$. The semi-edges that delineate a hole in a face must be part of the outer boundary of other faces. (Optional axiom)

Axiom A8: $\forall\, f \in F, \forall\, n \in f, D(n, f_p) \leq \varepsilon'$. Bounded faces are planar to a tolerance of ε'.

Axiom A9: $\forall f, n \in s, n \notin f \Rightarrow D(n, f) > \varepsilon$. No node is within ε of a face unless it is part of the definition of that face.

Axiom A10: $\forall e \in s, \forall f \in s: e \notin f, \;\; (\exists\, p: D(p, e) = 0 \wedge D(p, f) = 0) \Rightarrow n_1 \in f_2 \vee n_2 \in f_2$, where $e = (n_1, n_2)$. No directed-edge intersects a face except at a node of that edge.

Axiom S1: $f_1 = (f_{1n}, f_{1e}, f_{1p}) \in s \wedge f_2 = (f_{2n}, f_{2e}, f_{2p}) \in s \;\Rightarrow f_{1n} \neq f_{2n}$. No face may be paired with an anti-equal face in the same shell.

Types of Faces

It is useful to assign names to certain types of faces (see Figure 6), especially those that are associated with the meeting of 2D and 3D parcels.

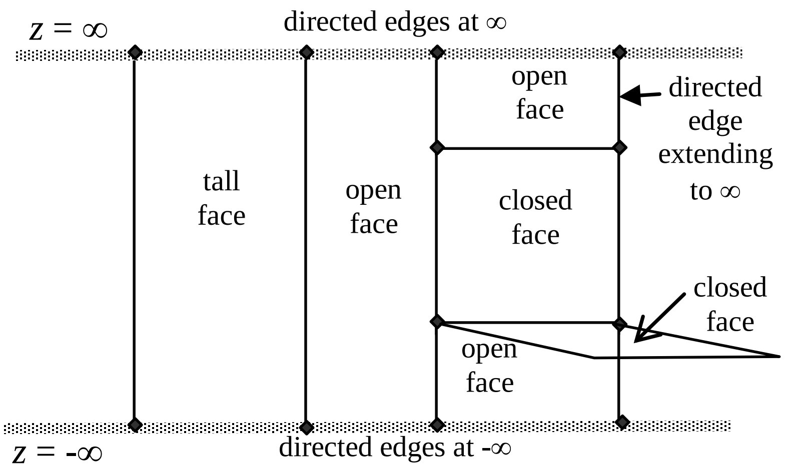


Figure 6. Types of faces and edges

Corollary: In any face, any directed edge extending to infinity must connect to a directed edge at infinity. (Assuming it does not, it must connect to an edge which also has the same x and y coordinates on both nodes – violating A1 or A5). (C7)

An open face is one which has at least one edge at infinity. (D29)

Corollary: an open face has at least two open directed edges which meet the directed edge(s) at infinity. (C8)

A tall face is an open face with exactly 4 edges, one of which is at +∞, another at -∞ (i.e. it is open at the top and bottom). (D30)

[1] Note Axiom A4 was defined in an earlier paper, but proved to be redundant. The numbering system has been retained to assist with comparison with earlier papers on this subject.

Corollary the other two edges are vertical and run between the two infinities. (C9)
A closed face is one which is not open. (D31)
Corollary a closed face has no vertices at infinity. (C10)
Corollary any face must have at least two edges which are not at infinity. (C11)
Corollary: if *u* is an undirected edge at infinity of spatial unit *s*, it consists of exactly one directed edge *e* of *s*). (C12)
Note that the definition of a face $f_p = (a,b,c,d)$: $a,b,c,d \in \mathbf{R}$ implies that a face cannot be defined "at infinity" – that is a face where each z value is ∞ (or -∞). This leads to C12.

Definition of LA_FaceString

LA_FaceString is defined in ISO1952 (ISO-TC211 2009), but this ties that definition in with the axioms of this paper. By this approach, an LA_FaceString is a shell composed of *n* tall faces $f_1 \ldots f_n$, (with edge sets $fe_1..fe_n$) (see Figure 7).

$fe_i = \{e_i, g_i, h_i, j_i\}$ $e_i, g_i, h_i, j_i \in E$ where
$e_i = ((x_i,y_i, \infty), (x_i,y_i, -\infty))$
$g_i = ((x_i,y_i, -\infty), (x'_i,y'_i, -\infty))$
$h_i = ((x'_i,y'_i, -\infty), (x'_i,y'_i, \infty))$
$j_i = ((x'_i,y'_i, \infty), (x_i,y_i, \infty))$
with $x'_i = x_{i+1}$ and $y'_i = y_{i+1}$ (i+1 cyclic if LA_FaceString is a polygon)
and for $i = 1..n-1$, $h_i \upharpoonleft\!\downharpoonright e_{i+1}$.

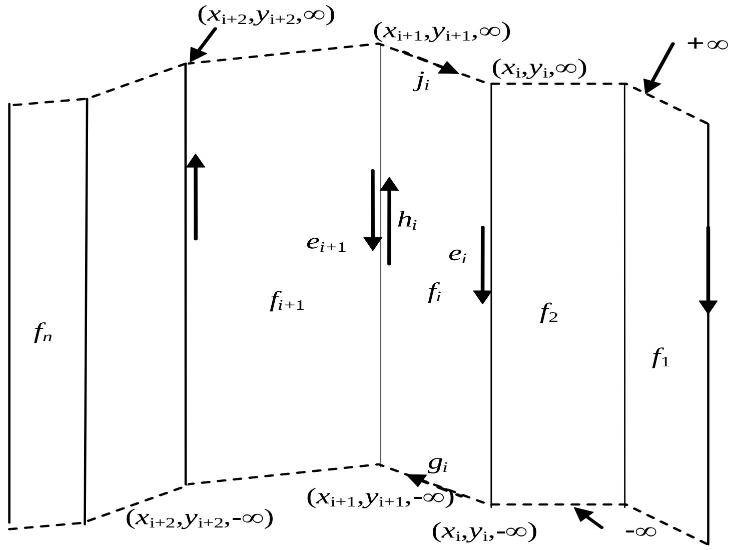


Figure 7. A face string (LA_FaceString)

Definition of a Cycle Shell

Define a cycle shell as a shell *s* that satisfies axioms A0 to A10 apart from A2 and A7.

Note that the terms "open" and "closed" are rather overloaded in meaning, so the terms "cycle" and "interior defining" are used here to indicate that it is possible to determine for any point in $\mathbf{P}^3$ whether it is inside the shell or not. The term "closed" would not be appropriate to an "interior defining shell" that has point(s) at infinity within its interior. Note that the point in cycle shell test (Thompson & van Oosterom 2011) can return a result of +1 or -1, and so a cycle shell can be defined as "interior-defining" or "negative interior defining". In any case a cycle shell divides the space into inside and outside. (Normally for a polyhedron there is one outer cycle shell and 0 or more inner cycle shells).

4. DEFINITION OF CONNECTIVITY

Point-wise connectivity: A $\mathbf{C_0}$ **face** $f' = f'(e)$ in shell s is a subset of a face $f \in s$ such that:

$e_1 \in f' \Rightarrow (\forall n \in e_1 \Rightarrow n \in f')$

$n \in f' \Rightarrow (\forall e \in f\!: n \in e \Rightarrow e \in f')$ (D32)

(Descriptively, for any edge in f' the nodes that define it are in f', for every node, all directed edges that meet at that node are in f').

$C_0(f) =_{def} \exists\, e \in f \wedge f = f'(e)$ (D33)

(Descriptively, a face is point-wise connected if it has at least one directed edge, and all other edges/nodes are point-wise connected to it – see Figure 8)

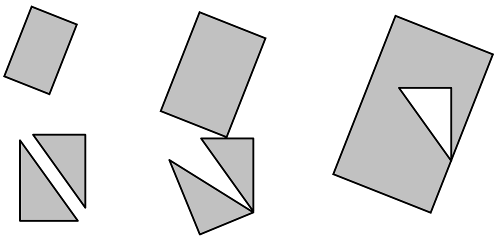
Strong connectivity: A $\mathbf{C_1}$ **face** $f'' = f''(e)$ in shell s is a subset of a face $f \in s$ such that:

$e_1 \in f'' \Rightarrow (\forall n \in e_1 \Rightarrow n \in f')$

$e_1 = (n_1, n_2) \in f'' \wedge e_2 = (n_2, n_3) \in f \wedge v_2(f, e_1, n_2, e_2) \in V_2 \Rightarrow e_2 \in f''$ (D34)

(Descriptively, for any edge in f'' the nodes that define it are in f'', and all directed edges that meet at that node in a corner are in f'').

$C_1(f) =_{def} \exists\, e \in f \wedge f = f''(e)$ (D35)



face C

face C_0 face C_1 face

Figure 8. Faces, with weak and strong connectivity depicted (to the left the more generic concept and to the right the more specific concept; i.e. a C_1 face *is a* C_0 face *is a* face, but the reverse is not per se true)

In common parlance, a face and a C_0 face would be referred to as multi-polygons, with a C_1 face being a simple polygon – see Figure 8.

A $\mathbf{C_0}$ **shell** $s' = s'(f)$ in cycle shell s a subset of s such that

$C_0(f_1) \wedge f_1 \in s' \wedge n \in f_1 \wedge f_2 \in s \wedge C_0(f_2) \wedge n \in f_2 \Rightarrow f_2 \in s'$ (D36)

(For every face in s' all C_0 connected faces that meet it at a node are also in s').

Pointwise connectivity: $C_0(s) =_{def} \exists\, f \in s \wedge s = s'(f)$ (D37)

A $\mathbf{C_1}$ **shell** $s'' = s''(f)$ in cycle shell s a subset of s such that

$C_1(f_1) \wedge f_1 \in s'' \wedge e_1 \in f_1 \wedge f_2 \in s \wedge C_1(f_2) \wedge e_2 \in f_2 \wedge (e_1 \upharpoonleft\!\downharpoonright e_2) \Rightarrow f_2 \in s''$ (D38)

(For every face in s'' all C_1 connected faces that meet it at any edge are also in s'').

1D Connectivity: $C_1(s) =_{def} \exists\, f \in s \wedge s = s''(f)$ (D39)

A $\mathbf{C_2}$ **shell** $s^\circ = s^\circ(f)$ in cycle shell s a subset of s such that

$C_1(f_1) \wedge f_1 \in s^\circ \wedge e_1 \in f_1 \wedge f_2 \in s \wedge C_1(f_2) \wedge \exists v_3\,(s, f_1, e_1, f_2) \Rightarrow f_2 \in s^\circ$ (D40)

(For every face in $s°$ all C_1 connected faces that meet it at a fold are also in $s°$).
Strong Connectivity: $C_2(s) =_{def} \exists f \in s \wedge s = s° (f)$ (D41)

The cycle shell, C_0 shell and C_1 shell could be referred to as multi-spatial units, while the C_2 shell can be termed a simple spatial unit (see Figure 9). Note that any spatial unit could, by this definition, have holes (to define dents in the surface of the outer and/or inner shells), which may "tunnel through" the body (note no inner shell), or be total inclusions with no connection to the outer boundary (a true inner shell within the outer shell).

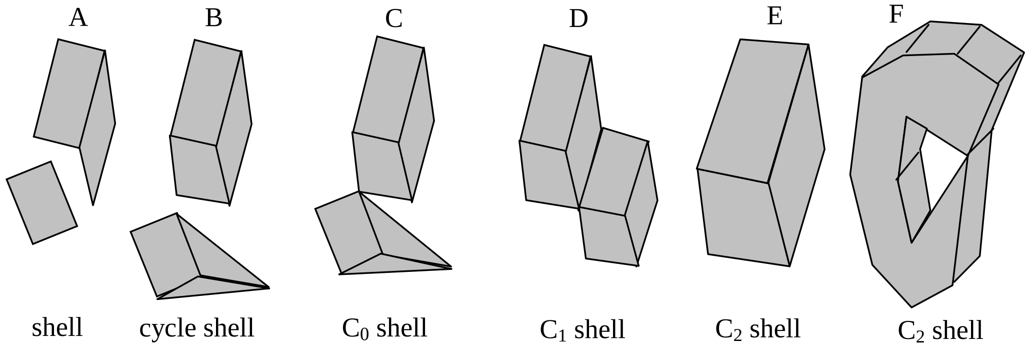


Figure 9. Shell, cycle shell, two types of weak connectivity and strong connectivity. Note that A is not an "interior-defining" shell, and that F is C_2, although it has a weak connection between some of its faces.

5. FORMING A SPACE PARTITION USING TOPOLOGY BASED SPATIAL UNITS

As was briefly discussed in Thompson & van Oosterom (2011), we define a normal (LADM based) spatial unit as an interior-defining C_2 shell (alternatives – C_1 shell, C_0-shell or even cycle shell not requiring connectivity) with zero or more negative-interior-defining shells inside.

The LADM also supports multi-polygon spatial units, although this is not the preferred option as they should better be modeled as separate spatial units belonging to the same LA_BAUnit. In any case, 'multi-polygon' spatial units will not be discussed further in this paper.

We can define a 2D spatial unit as a spatial unit where all faces are tall.

We can define a liminal face as a set of faces within a single spatial unit which can be dissolved into a tall face. That is, by removing all anti-equal edges from the set of edges in the faces, and removing all nodes along straight lines, we are left with 4 edges that define a tall face (see Appendix 1). For example, in Figure 2 the boundary between the apartment block and the 2D adjoining parcel would be represented as a liminal face.

We can define a liminal spatial unit as a spatial unit where all faces are tall or liminal.

We define a 3D spatial unit as one which has at least one face which cannot be made part of a liminal face.

Within an individual spatial unit, require Axiom S1 (no anti-equal pairs of faces).

We can define the outer boundary of all parcels as a single negative-interior-defining C_2 cycle shell. (Alternatives – C_1 shell, C_0-shell or even cycle shell not requiring connectivity).

A space partition consists of *m* spatial units $s_1 \ldots s_m$ and one "rest of the world" s_0 such that for $0 \le i \le m, f \in s_i \Rightarrow \exists f' \in s_j: 0 \le j \le m: f \Downarrow\!\Uparrow f'$. That is: for every face in the partition, there is an anti-equal face. Axiom S1 ensures that $s_i \neq s_j$.

Notes on the "rest of the world" parcel.

Using the terminology of Gröger and Plümer (2005), we could define a parcel called OUT, which is not bounded in any direction, and consists of the remainder of the space when all parcels are removed. i.e. for point $p \in \mathbf{R}^3$, $p \in$ P (for some parcel P) $\Leftrightarrow p \notin$ OUT. OUT has no outer boundary, but it has a cycle shell that is an inner boundary (s_0 above).

1) We can require OUT to be a 2D spatial unit. This means that if we are using topology based spatial units, there can be no 3D spatial units directly adjoining OUT. This should not be a problem, and it has some advantages. The alternative is that OUT could be defined as a liminal parcel.

2) We can define OUT to be C_2, or any other degree of connectivity desired. This will prevent any voids within the real coverage of 2D and 3D spatial units.

3) We do not need any boundary at a z value of plus or minus infinity, since the individual parcels have no boundary at $z = \pm\infty$.

6. CONCLUSIONS

This paper has addressed the issue of geometric validity of cadastral information in relation to the LADM and ISO19152. This includes the issues of the mixture of 2D and 3D parcels that exist in virtually all cadastres worldwide, and shows that a rigorous meaning can be attached to the concepts of "face string", and "liminal" parcels. It has further shown that the outer boundary of the universe of discourse can be correctly handled.

6.1 Future Research

A restatement of the validation axioms should be made using GeoUML (Belussi et al. 2011) as a cross-check on both approaches.

The question of closure of the algebra should be researched – perhaps in relation to the "Dual Grid" (Lema and Güting 2002) and the "Regular Polytope" (Thompson 2007) approaches

In loading the existing paper plans, some have validity failures in the actual measurements. In particular, many have non-planar surface. This is a problem because, even though they are slightly ambiguous, they are the legal definition of the parcels, so cannot be easily changed.

In case of partially unbounded parcels (to below or above or both directions), what would be appropriate size measures for area and volume?

6.2 Formalizing LADM Levels

The LADM also allows the use of levels to organise the parcels (spatial units). One of the options is having: a. "level one" being a complete 2D coverage, implying a collection of unbounded columns covering the complete 3D space (within the domain) and b. another level, being completely bounded 3D parcels, which have a higher priority (i.e. from the conceptual unbounded 3D columns the explicit 3D spaces should be subtracted). Also, for this possibility within the LADM a set of axioms defines valid configurations. Note that the second level with completely bound 3D parcels may contain objects that cross many 2D parcels (when considered as vertically unbounded 3D columns), for example, when modelling the legal space related to a tunnel. Also note that for level one a 2D topology encoding is the natural approach (but not per se needed) and that for the second level a 3D geometry (simple features) encoding is the natural approach. The concept of LADM levels have to be included in the (extended) axioms for valid representation.

6.3 Finite Precision

The axioms do not in themselves require infinite precision mathematics, and note that a node is defined based on computer representable points (on $\mathbf{P}^3$), while a point is defined in $\mathbf{R}^3$. Provided the accuracy of calculation is considerably finer than ε and ε' it is believed that they can be tested successfully. What is not assured is that the same answers will be obtained on different hardware platforms. For example, a point that is very close to being a distance of ε' from a face may be accepted on one machine but rejected on another.

Provided these are seen as "nominal" limits, where the data should always be considerably better that the minimum quality requirement, this should not be a problem.

6.4 Algebraic Closure

What cannot be promised by this approach is a closed algebra. It is easy to generate counter-examples, where for example, the intersection of two valid parcels is not a valid parcel. As a result, this approach must be seen as a means of exchanging cadastral information, not a way of storing and manipulating it.

6.5 Horizontal and Vertical Faces

In the definitions and axioms, there was heavy use of the concepts of horizontal and vertical faces. This may be obvious when using a (local) large scale coordinate reference system. When switching to another coordinate reference system, it may very well be that the horizontal (or vertical) faces are not longer horizontal (or vertical) anymore. Also, when taking an earth science view, the intended faces are not horizontal planes, but rather concentric sphere patches. Similary, the vertical faces are not defined by vertical (and parallel) planes, but rather by planes converging in the center of the earth. How to address this issue in our formalization of valid representations (as valid in one coord reference system, may not be valid in another coordinate reference system) is not trivial. In the meantime, our guideline is that we intend horizontal and vertical in the sense of a large scale representation.

ACKNOWLEDGEMENTS

This research is supported in part by the Dutch Technology Foundation STW (project numbers 11300 and 11185), which is part of the Netherlands Organisation for Scientific Research (NWO) and partly funded by the Ministry of Economic Affairs, Agriculture and Innovation.

REFERENCES

Belussi, A., F. Liguori, J. Marca, S. Migliorini, M. Negri, G. Pelagatti and P. Visentini (2011). Validation of Geographical Datasets against Spatial Constraints at Conceptual Level. The Urban Data Management Symposium (UDMS 2011). Delft, The Netherlands.

Boguslawski, P. and C. Gold (2009). Construction Operators for Modelling 3D Objects and Dual Navigation Structures. Lecture notes in goinformation and cartography: 3d Geoinformation Sciences, Part II. S. Zlatanova and J. Lee, Springer**:** 47-59.

De Floriani, L. and A. Hui (2003). A scalable data structure for three-dimensional non-manifold objects. In: Proceedings 1st Eurographics Symposium on Geometry Processing, Aachen, Germany.

Goldberg, D. (1991). What Every Computer Scientist Should Know About Floating-Point Arithmetic. ACM Computing Surveys, March 1991.

Gröger, G. and L. Plümer (2005). How to Get 3D for the Price of 2-D - Topology and Consistency of 3D Urban GIS. In: Geoinformatica, 9.2: 139-158.

Gröger, G. and L. Plümer (2011). How to achieve consistency for 3D city models. In: Geoinformatica, 15: 137-165.

Gröger, G. and L. Plümer (2012). Transaction Rules for Updating Surfaces in 3D GIS. In: ISPRS Journal of Photogrammetry and Remote Sensing 69 (1).

ISO-TC211 (2009). Geographic Information - Land Administration Domain Model (LADM). ISO/CD 19152.

Kazar, B. M., R. Kothuri, P. van Oosterom and S. Ravada (2008). On Valid and Invalid Three-Dimensional Geometries. Advances in 3D Geoinformation Systems. P. Van Oosterom, F. Penninga, S. Zlatanova and E. Fendel. Berlin, Springer.

Ledoux, H. and C. Gold (2007). Simultaneous Storage of Primal and Dual Three-Dimensional Subdivisions. Computers, Environment and Urban Systems ,31: 393-408.

Lema, J. A. C. and R. H. Güting (2002). Dual grid: A new Approach for Robust Spatial Algebra Implementation. In: GeoInformatica, 6(1): 57-76.

Lemmen, C., H. T. Uitermark and P. van Oosterom (2012). The Final Steps Towards an International Standard for Land Administration. FIG Working Week 2012, Rome.

Lemmen, C., P. van Oosterom, R. J. Thompson, J. Hespanha and H. Uitermark (2010). The Modelling of Spatial Units (Parcels) in the Land Administration Domain Model (LADM). FIG Congress 2010, Sydney, Australia.

Milenkovic, V. J. (1988). Verifiable implementations of geometric algorithms using finite precision arithmetic. In: Artificial Intelligence, 37: 377-401.

Stoter, J. and P. van Oosterom (2006). 3D Cadastre in an International Context. Boca Raton FL, Taylor & Francis.

Thompson, R. and P. van Oosterom (2011). Axiomatic Definition of Valid 3D Parcels, potentially in a Space Partition. 2nd International Workshop on 3D Cadastres. Delft, The Netherlands, International Federation of Surveyors (FIG).

Thompson, R. and P. van Oosterom (2012). Modelling and validation of 3D cadastral objects. UDMS 2011. Delft, The Netherlands.

Thompson, R. J. (2007). Towards a Rigorous Logic for Spatial Data Representation. PhD thesis, Delft University of Technology, Delft, The Netherlands, Netherlands Geodetic Commission.

van Oosterom, P., J. Stoter, H. Ploeger, R. Thompson and S. Karki (2011). World-wide Inventory of the Status of 3D Cadastres in 2010 and Expectations for 2014. FIG Working Week, Marrakech, Morocco.

APPENDIX I Testing for a liminal set of faces.

Take all the faces of s that are candidates to form a tall face. All must have the same plane definition f_p (see Figure 10).

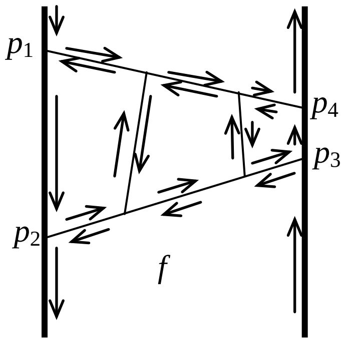


Figure 10 Testing for a liminal face. All the anti-equal pairs of directed edges are removed, and then unnecessary points removed from remaining edges ($p_1 - p_4$).

Form a set of directed edges from all these faces.

For every pair of edges e_1, e_2 such that $e_1 \upharpoonleft\downharpoonright e_2$, remove e_1 and e_2 from the set.(note – these must be exactly anti-equal)

For every pair of remaining edges in the set $e_1 = (p_1, p_2)$, $e_2 = (p_3, p_4)$ such that p_2 is not a node at infinity, and $p_2 = p_3$ (exact equality), remove e_1 and e_2, and replace them with the directed edge (p_1, p_4).

If we are left with exactly 4 directed edges, one at $+\infty$, one at $-\infty$, and the others being tall directed edges, the set of faces is liminal.

BIOGRAPHICAL NOTES

Rod Thompson has been working in the spatial information field since 1985. He designed and led the implementation of the Queensland Digital Cadastral Data Base, and is now principal advisor in spatial databases. He obtained a PhD at the Delft University of Technology in December 2007.

Peter van Oosterom obtained an MSc in Technical Computer Science in 1985 from Delft University of Technology, The Netherlands. In 1990 he received a PhD from Leiden University for this thesis 'Reactive Data Structures for GIS'. From 1985 until 1995 he worked at the TNO-FEL laboratory in The Hague, The Netherlands as a computer scientist. From 1995 until 2000 he was senior information manager at the Dutch Cadastre, were he was involved in the renewal of the Cadastral (Geographic) database. Since 2000, he is professor at the Delft University of Technology (OTB Research Institute for the Built Environment) and head of the GIS Technology Section. He is the current chair of the FIG joint commission 3 and 7 working group on '3D Cadastres' (2010-2014).

CONTACTS

Rod Thompson
Queensland Government, Department of Natural Resources and Mines
Landcentre,
Main and Vulture Streets,
Woolloongabba
Queensland 4102
AUSTRALIA
Tel.: +61 7 38963286
E-mail: Rod.Thompson@qld.gov.au
Web site: http://www.dnrm.qld.gov.au/

Peter van Oosterom
Delft University of Technology
OTB Reserch Institute for the Built Environment
GIS Technology Section
P.O. Box 5030
2600 GA Delft
THE NETHERLANDS
Tel. +31 15 2786950
Fax +31 15 2784422
E-mail: P.J.M.vanOosterom@tudelft.nl
website http://www.gdmc.nl