Robust Representation and Analysis of Geo-information

Rod Thompson
Delft University of Technology and
Department of Natural Resources and Mines, Queensland Australia
Jafalaan 9,
2628BX Delft,
The Netherlands
rodnmaria@gil.com.au
tel.: +31 15 2785154
fax: +31 15 2782745

Peter van Oosterom
Delft University of Technology
Jafalaan 9,
2628BX Delft,
The Netherlands
oosterom@otb.tudelft.nl
tel.: +31 15 2786950
fax: +31 15 2782745

David Pullar
University of Queensland, School of Geography, Planning and Architecture
Chamberlain Bldg Room 035-507
University of Queensland
Brisbane Q 4072
Australia
D.Pullar@uq.edu.au
tel.: +61 7 3365-6522

Abstract

One of the perennial difficulties in the representation of spatial data in database structures is that of the finite nature of the number representations. The vast majority of the mathematical analyses of the issue are based on the theory of real numbers, and metric space topology. The realization of this theory within a finite precision computer representation has been largely overlooked. Implementations often ignore this issue, resulting in unexpected errors, according to real number mathematics. In exceptional cases these errors will invalidate analysis computations such as map overlay, or cause data to become unusable after a coordinate transformation or projection, because the algorithms assume 'correct' real-number behaviour (which is not the case in a finite digital computer). Therefore alternative computer representations are being investigated: the "Constraint Spatial Database" approach (Kanellakis, Kuper et al. 1995), the "Rational Polygon" (Lemon and Pratt 1998), the "Realms" approach (Guting and Schneider 1993), the "Dual Grid" (Lema and Guting 2002), and the "Regular Polytope" (Thompson 2005a).

These approaches share various difficulties which have, so far, inhibited their use in commercially available software, but have the major advantage that they provide support for a rigorous logic, with no complex "special case" programming being necessary to allow for finite arithmetic issues. In particular, the Regular Polytope approach implements the full Region Connection Calculus (RCC) (Randell, Cui et al. 1992) in a rigorous form such that the representation itself forms a topological space. Nevertheless, some practical issues remain with this approach. This paper will address the following issues with the Regular Polytope approach, and propose a possible alternative solution:

- The Regular Polytope storage mechanism differs from the more familiar point/line/polygon paradigm commonly used in GIS, and requires non-trivial conversion routines.
- The calculations require the use of very large precision integer arithmetic (as does the Dual Grid approach, and to an even larger extent, the Rational Polygon).
- The storage requirements are significantly larger than required for simple polygon encoding.
- It is not easy to map this storage form to/from the topological encoding form (Louwsma 2003).

This proposed solution, the "Approximated Polytope", while retaining the rigour of the Regular Polytope will address these issues, providing a mechanism which can use floating point arithmetic for the day-to-day calculations, uses a storage form more closely aligned to the point/line/polygon paradigm, and has space requirements somewhere between those of the Regular Polytope and those of polygon encoding. The Approximated Polytope is compatible with, and can utilise, the topological encoding method of storing spatial data. Therefore, the Approximated Polytope is potentially the first practical solution for robust representation and analysis of geo-information.

1. Outline of the Paper

Section 2 consists of a discussion of the more conventional vertex-based representations. This is followed by a brief description of the regular polytope representation, in sections 3 to 6. Sections 7 to 9 describe the proposed "Approximated Polytope" model, based on a simplified database schema, including a discussion of conversion to and from other representations. Section 10 introduces the possibility of topological encoding within the model, while Section 11 an 12 discuss practical implementation issues, and further research possibilities respectively.

2. Vertex-based Representations

In two-dimensional applications, the "Point/Line/Polygon" paradigm for the representation of spatial features is well entrenched, albeit with some significant variations (van Oosterom, Quak et al. 2003), and provides a degree of comfort in the user. This is spite of some serious difficulties in terms of rigorous definitions of concepts such as validity, and equality (Thompson 2005a). The available 3D structures take various forms (Arens, Stoter et al. 2005), with no one having proved to be the best in all circumstances (Zlatanova, Rahman et al. 2004).

In this paper, we use the term "vertex based" representation to cover all ways to model spatial data in 2 or more dimensions based on point coordinates of vertices as the major determinants of the shape and position of the objects. For example, in the 3D FDS (Formal Data Structure) (Molenaar 1990), and the "Simplified Spatial Model" (SSM) (Zlatanova 2000), the node is defined as a point with coordinates (x,y,z), while all other geometric objects are defined in terms of sets of nodes or higher order constructive objects. This is true of virtually all 2 and 3 dimensional spatial data models, regardless of the level of topological encoding supported (Ellul, Haklay et al. 2005).

One major challenge for 3D modelling is the fact that any definition of a face by more than three vertices runs the risk that that face may not be unambiguously planar. This could occur in two ways – the point values can be incorrectly calculated, or rounding errors can cause a small departure from planarity. Two different approaches may be taken 1) a tolerance value may be applied (provided that the departure of the face from planarity does not exceed a given tolerance, it is accepted); or 2) the faces may be triangulated (since any three points are always co-planar).

The first of these approaches adds a certain level of extra complexity, and like all approaches that use a tolerance, raises issues of non-transitivity of operations (e.g. where $A = B$, $B = C$, but $A \neq C$). The second strategy, of triangulation or tetrahedronisation of the objects, is quite acceptable for topographic applications, but in many applications, the loss of identity of the faces is significant.

3. The Regular Polytope

A regular polytope as described by Thompson (2005a) represents spatial objects as the union of a finite set of (possibly overlapping) convex polytopes, which are in turn defined as the intersection of a finite set of half spaces (in 3D, half planes in 2D). These half spaces (planes) are defined by finite precision integer representations (3 values in 2D, 4 in 3D etc.). The concept of a domain-restricted rational number x, has been useful in the definition of continuity of regular polytopes. A dr-rational number x is defined as a pair of integers (I,J) of restricted value (not potentially infinite) $-M_1 \leq I \leq M_1$, $0 < J \leq M_2$ (M_1 and M_2 being large positive integers), This is interpreted as $x = I/J$.

In the following discussion, capital letters (such as X) will be used for to represent computational integers, or the integer values they represent. Lower case letters (such as x) will be used for rational or domain-restricted rational numbers, but occasionally lower-case will be used for small integer values (e.g. i=1..n). No notational distinction is made in this paper between computational operations +,-,., =, etc, and the mathematical operations they implement, since the integer and rational number arithmetic available in computers is exact[1]. There is however, a distinction to be made. For example, it must be remembered that $A+B$ as a computational operation may result in overflow.

[1] By contrast, floating point is not exact, and it cannot be asserted that if $a := b*c$; (as a computation and assignment) then $a = bc$ (as a mathematical equation).

4. Half Space Definition

In 3D a half space $H(A,B,C,D)$ is defined as the set of all dr-rational points $P(x,y,z)$, $-M \leq x,y,z < M$ for which computational evaluation of the following inequalities yields these results:

$(A.x + B.y + C.z + D) > 0$ or

$[(A.x + B.y + C.z + D) = 0$ and $A > 0]$ or [2]

$[(B.y + C.z + D) = 0$ and $A=0$ and $B>0]$ or

$[(C.z + D) = 0$ and $A=0$, $B=0$ and $C>0]$

Where M is the range of integer values allowed for point representations.

The values of the integers A,B,C and D define the half space. In 3D applications, we place the restriction that $-M < A,B,C < M$, $-3M^2 < D < 3M^2$, $H(0,0,0,0)$ is not a permitted half space.

Two special half spaces are defined,

$H_\varphi = H(0,0,0,-1)$(‘empty’ i.e. points for which $-1 > 0$).

$H_\infty = H(0,0,0,1)$(‘everything’ i.e. points for which $1 > 0$).

The complement of a half space is defined as:

$\overline{H} = (-A,-B,-C,-D)$, where $H = (A,B,C,D)$.

Referring to the definition of a half space, it is readily apparent that: $p \in H \Leftrightarrow p \notin \overline{H}$, and that: $\overline{\overline{H}} = H$.

5. Convex Polytope Definition

A convex polytope is defined as the intersection of any finite number of half spaces[3]; see Figure 1 for a 2D and Figure 2 for a 3D example.

Convex polytope representation C is defined as:

$C = \{H_i, i = 1..n\}$ where H_i, $i=1..n$ is a set of half spaces. This is interpreted as the intersection of the half-planes, $C = \bigcap_{i=1..n} H_i$.

---

[2] This form of the definition with four parts, rather than just $(A.x + B.y + C.z + D) > 0$, is chosen so as to ensure a clean definition of complement. This results in the regular polytope being a boundary-free representation.

[3] In this paper, the term half space will be used generically to indicate half space or half plane depending on whether a 3D or 2D geometry is being considered. Most of the illustrations are in 2D for ease of visualisation.

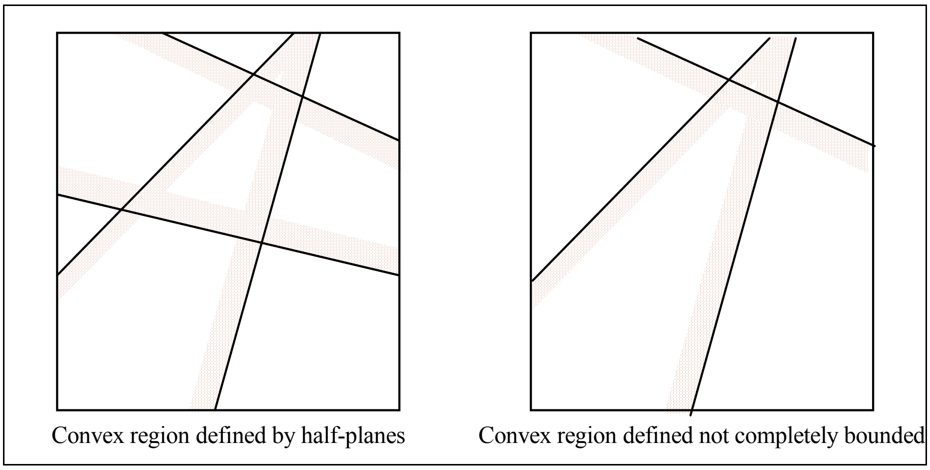

Figure 1 Convex polytopes defined by half planes.

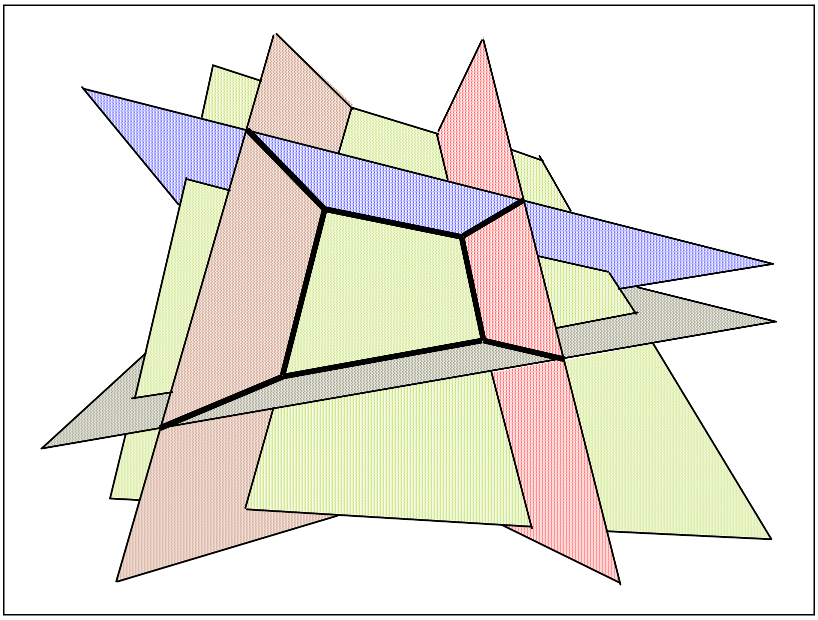

Figure 2 A convex polytope in 3D defined by half spaces

A special convex polytope is defined, $C_\infty = \{\}$ (no half spaces), with no constraints on allowed points, is used in the definition of O_∞ (the infinite polytope).

The intersection of two convex polytopes is defined as the intersection of the half planes that define each of them. It is clear that the intersection of two convex polytopes is itself a convex polytope.

6. Regular Polytope Definition

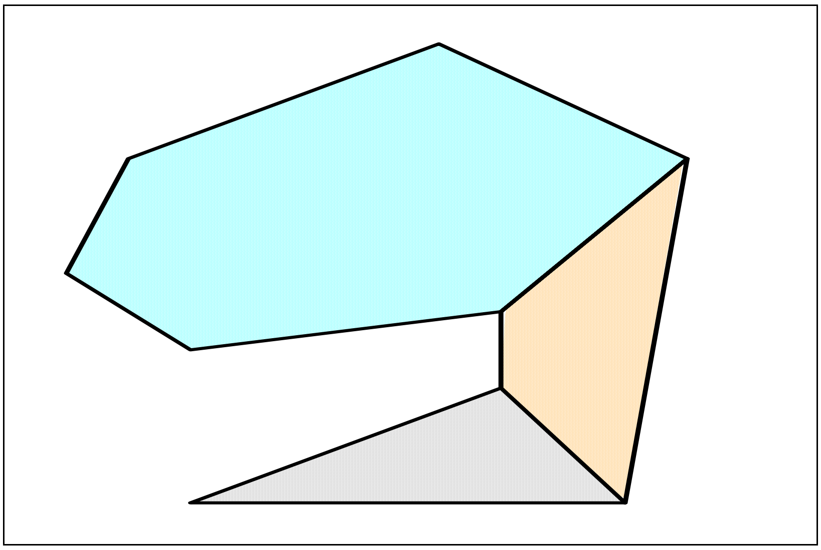

Figure 3 Definition of Regular Polytope from convex polytopes.

A regular polytope O is then defined as the union of a finite set of convex polytopes; see Figure 3 for example.

$$O = \bigcup_{i=1..m} Ci \text{ where } C_i,\ i=1..m \text{ are convex polytopes.}$$

Again, two special regular polytopes are defined,

$O_\Phi = \{\}$ (i.e. a set containing no convex polygons)

$O_\infty = C_\infty$.

These sets are the empty and infinite sets required for the definition of a topological space.

The union of a set of regular polytopes is simply the union of the sets of convex polytopes that define them: $\bigcup_{i=1,m} O_i = \bigcup_{i=1..m, j=1..n_i} C_{ij}$ where $O_i = \bigcup_{j=1..n_i} C_{ij}$. Note that this union is itself a regular polytope.

The intersection of two regular polytopes is the union of the pair-wise intersections of their component convex polytopes: $O \cap O' = \bigcup_{i=1..n} C_i \cap \bigcup_{j=1..m} C'_j = \bigcup_{i=1..n, j=1..m} (C_i \cap C'_j)$. Again, note that this defines a regular polytope, since the intersections of convex polytopes are themselves convex polytopes.

It is clear that the set of regular polytopes forms a topological space (Thompson 2005b).

Note - there are conceptual differences between convex polyhedra and polytopes. The latter may be unbounded, and part only of the boundaries belong to the object.

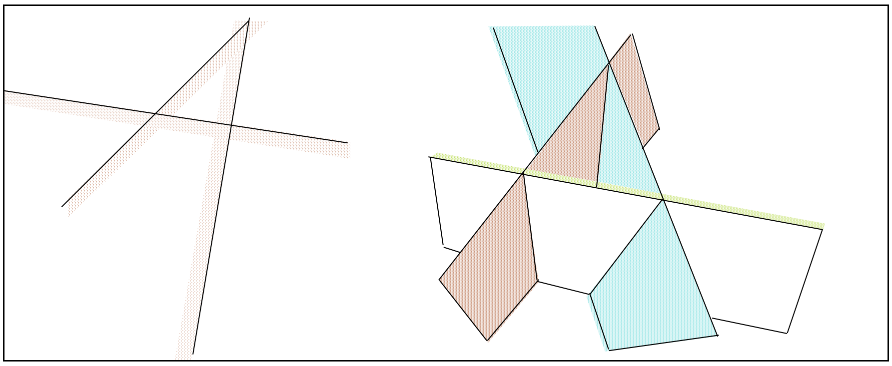

Figure 4 Convex polytopes, not fully bounded (left 2D, right 3D)

Two possible definition of connectivity have been proposed, with the preferred one being that regular polytopes O_1 and O_2 are considered to be C_b connected if there exists a convex polytope C such that $C \subseteq O_1 \cup O_2$, $C \cap O_1 \neq O_\Phi$ and $C \cap O_2 \neq O_\Phi$. This will be denoted as $C_b(O_1,O_2)$. Note that O_1 and O_2 do not need to overlap , but a convex polytope must fit within the union of O_1 and O_2 and must overlap both. The alternate, weaker, definition of connectivity C_a, (see Figure 5) requires the notion of overlapping pseudo-closures. It has been shown that the full set of relations in the Relation Connection Calculus (Randell, Cui et al. 1992) can be supported with full rigour by the regular polytope representation using either of these connectivity criteria (Thompson and van Oosterom 2006b).

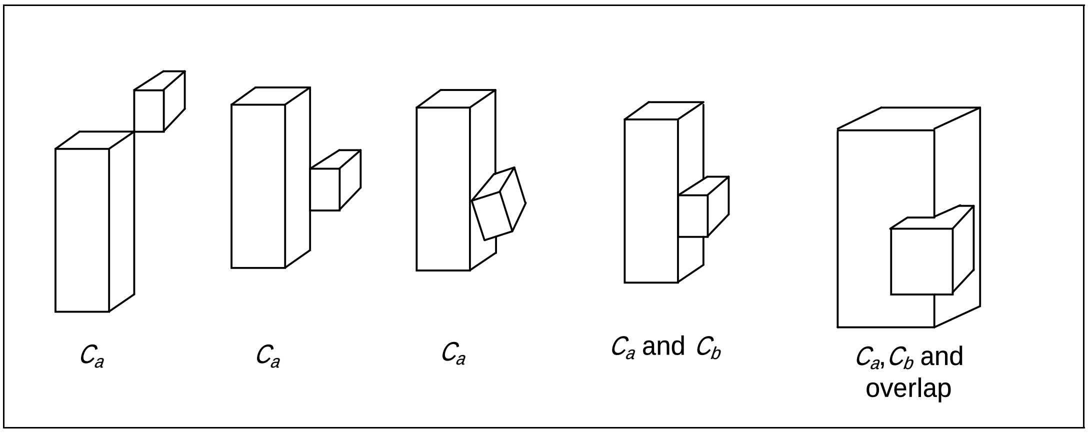

Figure 5 Types of connectivity. Overlap $\Rightarrow C_b \Rightarrow C_a$

7. The Approximated Polytope Model

In the description of the regular polytope, the suggestion was made that vertices be stored redundantly as a means of making processing of the objects more efficient.(Thompson 2005a) (see Figure 6). Also suggested was the possibility of omitting the actual polytope structure itself. This paper explores an approach where the regular polytope storage mechanism is replaced by one more like the traditional paradigm, while still retaining the same degree of rigour.

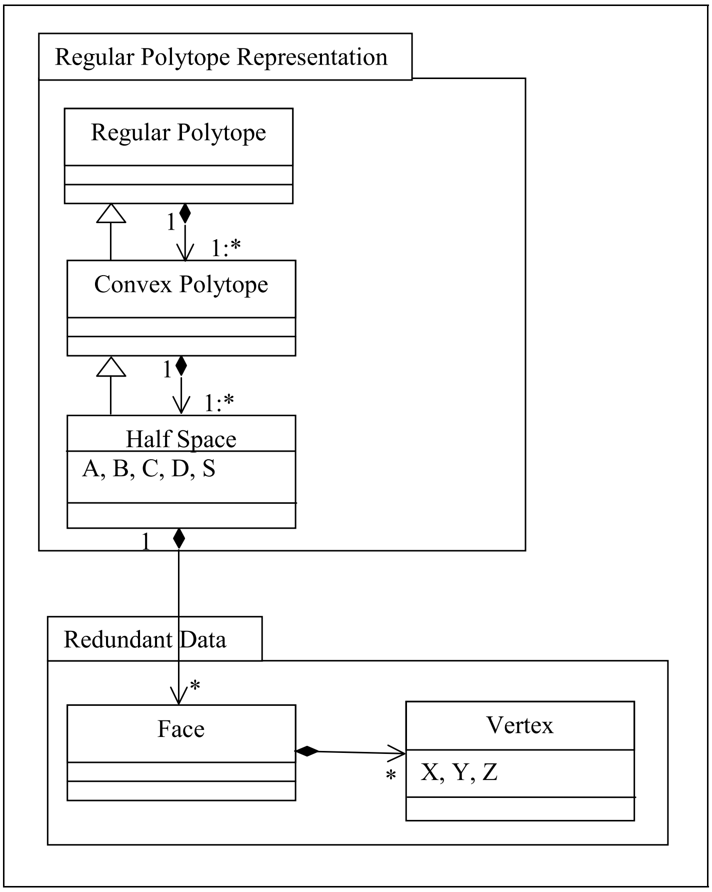

Figure 6 The original Regular Polytope model

As was indicated in section 2, there are many possible 3D representations with differing levels of topology, so a simplified structure will be considered here, where the topological encoding is omitted and each feature is encoded as a separate object (here referred to as a “body”). The extension of this structure to topological encoding is discussed in section 10. The model chosen for discussion is not particularly elegant, and contains redundant storage, but is fairly simple to describe and investigate.

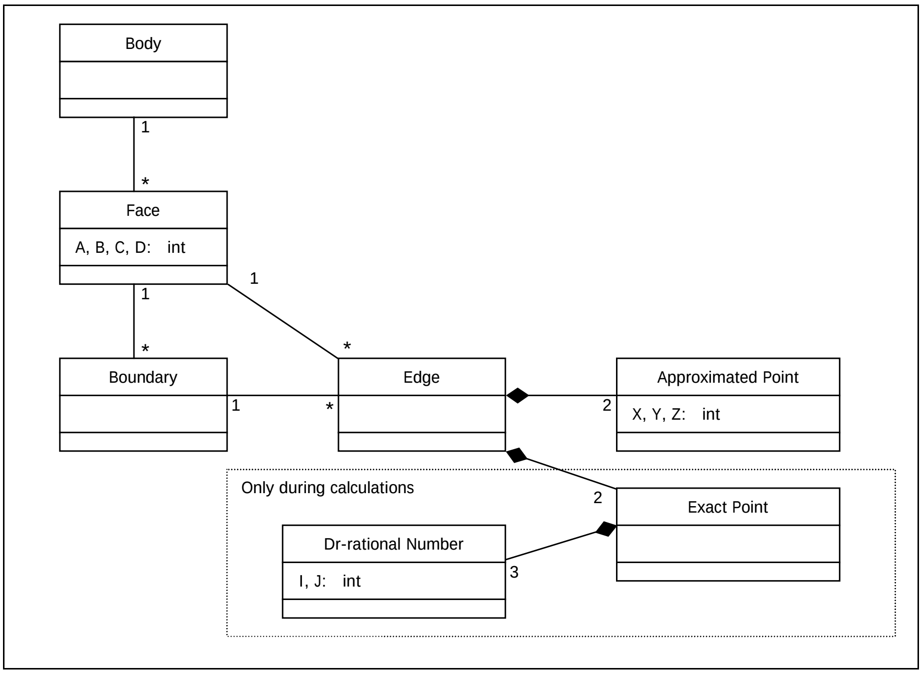

Figure 7 The simplified model

The classes are:

Body: is the volume of space which represents a feature.

Face: is a geometrically flat facet of the bounding surface(s) of a body. (Note – a body could have an internal surface – like the interior of a tennis ball).

Boundary: is a planar ring which defines the edges of a face (stored clockwise for an outer boundary, anticlockwise for inner - as viewed from outside the body).

Edge: Is a single line segment in a boundary – there will always be a pair of opposite edges for the junction of two faces.

Approximated Point: Is a representation of the points that define the edges.

Exact Point: is an exact representation of the approximated point, as a set of 3 domain-restricted rational numbers. Exact points would not be stored in the database, being only used when intersection or union operations are being calculated.

Dr-rational Number: is a representation of a domain-restricted rational number. It consists of two integers $-M_1 \le I \le M_1$, $0 < J \le M_2$, interpreted as I/J.

Figure 7 shows a simplified model for discussion here. A body is considered to be defined by a number of faces. Each face has attributes of A,B,C and D, with the same interpretation as half spaces (see section 4)and is bounded by one or more boundaries (with at least one of them being an outer boundary). An edge is the junction of exactly two faces, and defines the boundary of one of them. (Note that edges are thus stored twice – this will be discussed further in section 11).

Since the aim of this representation is to support the operations of the regular polytope, and the regular polytope is not necessarily fully bounded, it is necessary to define "faces at infinity". For example, the face (1,0,0,-M) defines the face X=M, where M is the maximum range of X,Y,Z and is interpreted as "infinity". Likewise, points with one or more of the x,y,z coordinates equal to $\pm M$ are considered to be "points at infinity". The universal regular polytope O_∞ can be represented as a body object (the universal box B_∞), with six faces. Each face has a single outer boundary consisting of four edges. Each edge starts at a point at exactly ($\pm M$, $\pm M$, $\pm M$).

The point-set definition of a body is simply the set of points which satisfy the "point in body" test. Briefly, for point $p = (x_p,y_p,z_p)$, this consists of running a ray in the –x direction, from the point and counting the faces it cuts. A face is deemed to be cut if the x intercept on this ray is $\leq x_p$. (note the equality is included). To cut a face, the point of intersection of the ray on the face must be within the boundaries of the face. This is tested by running a ray in the –y direction along the face, and counting the boundary edges it cuts. An edge is deemed to cut this ray if the y intercept is $\leq y_p$. The edge must also be such that $z_{\max} > z_p$ and $z_{\min} \leq x_p$ where $z_{\max}$ and $z_{\min}$ are the max and min z values of the edge. Note – the details of the use of > and ≤, and the direction of the rays is important in showing the equivalence of this approach with the regular polytope approach, but other strategies could be adopted – such as running in the –z direction first if this were not an issue.

In order to ensure the rigorous logic of the regular polytope can be transferred to this representation it is necessary to show either that:

1. This representation can be mapped reliable to and from the regular polytope representation, or
2. The operations Union, Intersection and Inverse can be implemented rigorously.

Clearly, 1 => 2 above, since if two-way mappings are available, then union, intersection and inversion can be implemented by mapping to the regular polytope representation, applying the operation, and then mapping back.

The other side of the equivalence (2 => 1) can be shown by considering the following:

By calculating where the half space $H = (A,B,C,D)$ intersects the universal box B_∞, a body can be generated B_H. Thus a half space can be represented as a body, as shown in Figure 8. The body that represents a half space can have from three to seven faces. It can readily be verified that if $p = (x,y,z) \in H$, ($-M \leq x,y,z < M$) $\Leftrightarrow p \in B_H$.

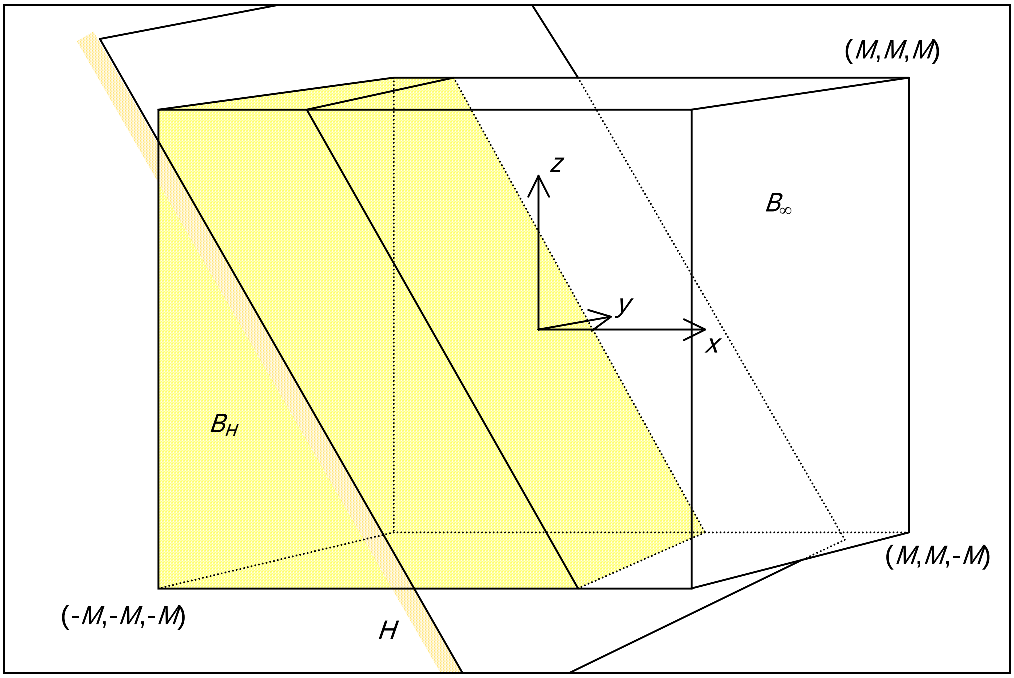

Figure 8 The universal box B_∞, and a half space H represented as a body B_H

If $C = \bigcap_{i=1..n} H_i$ is a convex polytope, and B_i is the body representation of H_i, then we can define $B_C = \bigcap_{i=1..n} B_i$ as the body representation of C. It can readily be verified that $p \in C \Leftrightarrow p \in B_C$.

By a similar argument, the union of any set of convex polytopes can be represented as a body. Therefore any regular polytope can be represented as a body using this structure.

In the reverse direction – If a body is convex – that is all faces meet at edges so that the dihedral angle of that meeting is less than 180°, then the body defines exactly that set of points which would be defined by a convex polytope defined by the faces

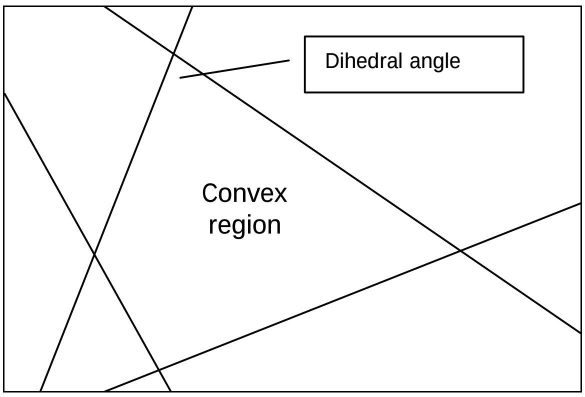

Figure 9 Convex body defining a convex polytope

For a non-convex body, if m pairs of faces meet at a dihedral angle of > 180°, then one of those faces can be converted into a half space and its inverse, and therefore into a complementary pair of bodies that covers the universal box. The original body is then replaced by two bodies – each being the intersection of the original body with on of the complementary pair. By this process, two bodies are created that each have at most m-1 pairs of faces with dihedral angle of > 180°. Continuing this process, we are left with a set of convex bodies whose union is the original body. This can then be expressed as a regular polytope (see Figure 10).

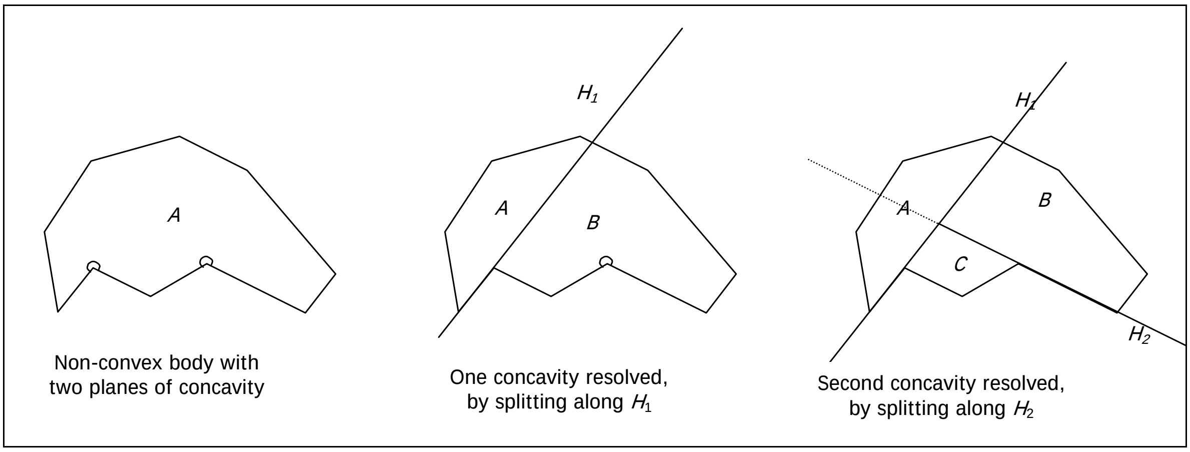

Figure 10 Cutting a non-convex body into convex subregions.

Using domain-restricted rational arithmetic, the operations of union, intersection and inverse can be defined, and by careful considerations of the rules of inclusion, it can be verified that the results of the operations are consistent with the results of those same operations on the regular polytope representation. Thus this representation is logically equivalent to the regular polytope. It can therefore be asserted that this approach can implement all the Region Connection Calculus (RCC) operations. The use of domain-restricted rational numbers requires some care in the specification of the algorithms, since it is necessary to verify that the results of any calculations do not violate the domain limits, but the algorithms themselves are significantly simpler than those often employed in floating point arithmetic, since no calculation or rounding errors need be accommodated.

8. The Approximation

The approach has been called "approximated polytope", but so far, we have been discussing exact operations. The approximation takes the form of an approximated point – stored instead of the dr-rational points. This allows a form of the body representation to be available for "everyday use". It is envisaged that these points would be used for such operations as visualization, geographic search and indexing, detection of possible overlap, etc. In fact, all operations except for those involving the exact calculation of intersection, union, inverse or combination of these. It is possible for these approximated points to be stored in integer or floating point form, at whatever accuracy is desired, and the calculation of them from dr-rational numbers can be highly accurate. If integers are used, the approximated point may be determined to within one unit of resolution of the exact point in x,y, and z.

Using the "point within body" test as described above, but using the approximated points for calculation (and not needing dr_rational arithmetic), it can be seen that the correct result will be obtained provided the point is not within one unit of resolution of the surface of the body. It can further be seen that it is possible to determine whether the point is within a specified distance from the surface. It is thus possible to convert from this representation to a more conventional (point location based) representation, simply by using these approximated points. Note however that while the faces themselves are flat by definition, the approximated points may lie up to one unit of resolution off the flat plane, so the usual issues of faces defined by more than three points not being planar will then apply (see Figure 11). It is also the case that two or more points may approximate to the same value, so the interpretation into conventional form must be carefully approached (see section 11).

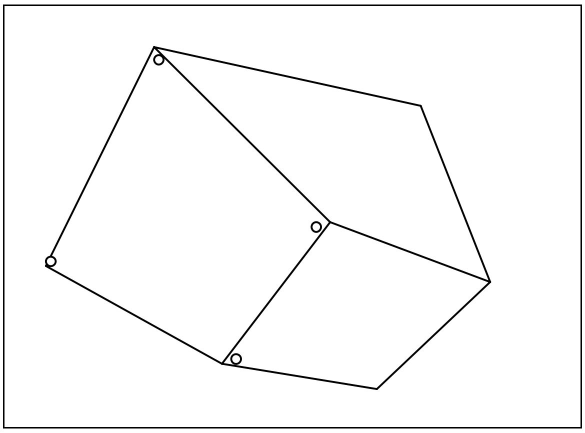

Figure 11 Approximated points used in place of dr-rational vertices

9. Converting vertex representations to Approximated Polytope.

There are some important issues to be considered in this conversion. It is assumed that the conventional form is defined by the point locations of vertices, with sufficient face definition to construct the bodies. It must be remembered that the basic primitive for this representation is the half space. A half space cannot in general be found to pass through any three points – the best that can be guaranteed is that a half space can be found that will pass within one unit of resolution of any three points (Thompson 2005b). Thus there is some approximation involved in the calculation of the approximated polytope[4]. An outcome of this is that the dr-rational points of intersection of two faces may differ significantly from the original vertices, and so the approximated points will also differ. This is not a serious issue, since the accuracy of the data is usually significantly lower than the resolution used to store that data, but has to be considered in algorithms, since it can cause points to merge.

[4] Also, of course, it must be verified that any face defined by four or more vertices is sufficiently close to being planar.

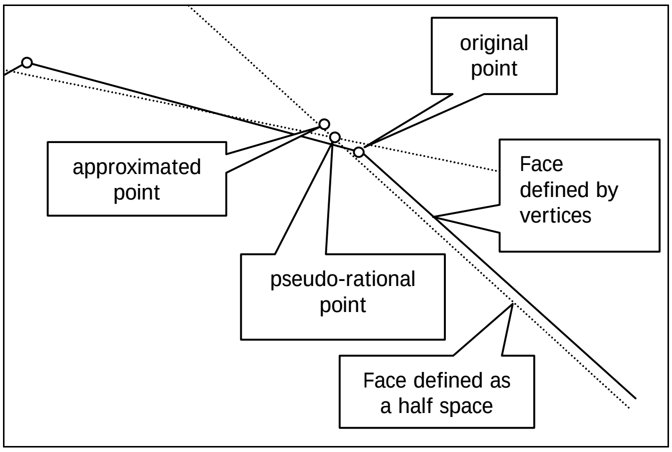

Figure 12 Movement of points in the approximation process

A further issue is that an edge is defined by the meeting of exactly two faces. It is not possible in general to generate another face that passes through the edge of intersection of two existing faces (See Figure 13). This does not significantly affect the model being discussed here, but will have an impact on topological encoding (see below).

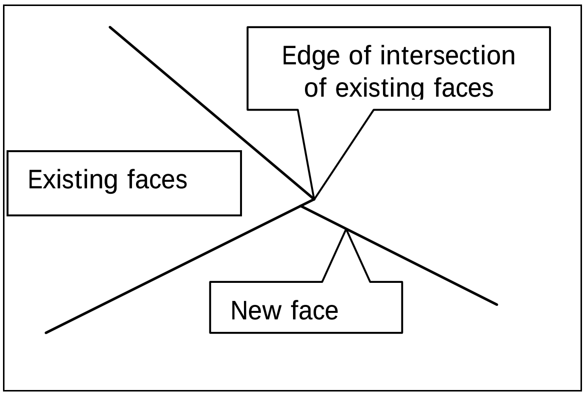

Figure 13 In general, a face cannot be guaranteed to pass through the edge of intersection of two other faces.

10. Extension to Topological Encoding

The extension to topological encoding requires that a set of faces be grouped into a compound surface, while the bodies are defined by the surfaces that surround and separate them. A surface is typically linked to the body (or bodies) on the left and the right of it. The other requirement is that of a a “nodal edge” which is usually defined as the meeting of three or more faces. It is the nodal edge which is the difficulty in the approximated polytope approach, since at most two faces can be guaranteed to meet at a single edge (see above).

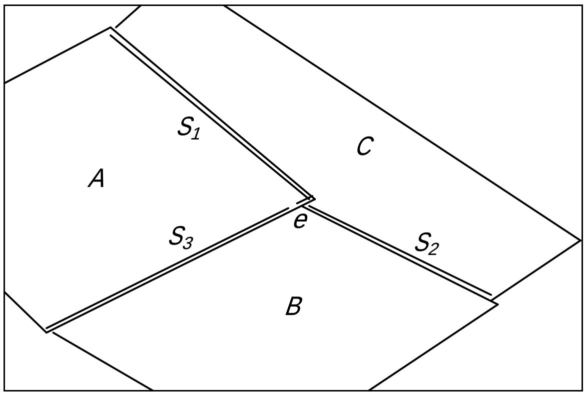

Figure 14 Topological encoding of bodies by surfaces.

For example, in Figure 14, surface S_1 has C to the left, A to the right. S_2 has B to the left and C to the right. S_3 has B to the left, and A to the right. But note that S_1 consists of two faces, one of which is very small, and is coplanar with S_3. Also note that edge e is defined by three faces, but two of them are coplanar. There is also a high probability that the approximate points that define small faces such as that at edge e will become degenerate, so that routines that generate conventional vertex based representations from this will need to accommodate degenerate faces.

In summary, there is nothing to prevent a data structure with topological encoding being developed, but some care is required in the handling of nodal edges, both on import from and on export to conventional vertex-based representations.

11. Practical Questions

It is considered unnecessary to store the exact dr-rational vertices in the database, since they can be recalculated as necessary in $O(v)$ time, where v is the number of vertices. Since the dr-rational numbers are of finite precision, the operations are of constant duration (unlike the true rational numbers, where the time of calculation of arithmetic operations depends on the magnitude of the numerators and denominators. Calculations using dr-rational numbers in demonstration classes implemented in Java are quite slow, since for convenience they have used the BigInteger class, which is potentially infinite. In a practical implementation a faster (finite) arithmetic could be used.

If this approach was being used for data interchange and validation, both the exact and the approximated vertices could be omitted, and re-generated on arrival as part of the validation operation. This validation could be completely and rigorously specified, so that issues of validation failure in transit (Thompson and van Oosterom 2006a) could be avoided.

In the schema described here (in Figure 7), the edge of intersection between two faces is recorded as a pair of edge objects. Each edge object contains a pair of points. This is obvious duplication, and would be addressed in any practical implementation. This issue would be addressed in conjunction with deciding the form of topological encoding to be used.

12. Further Research

It is intended that proof of concept software be developed in the simplified form discussed here. It is further hoped that this can be extended to a topological encoded form to ensure that the issues raised above can be satisfactorily handled.

13. References

Arens, C., J. Stoter, et al. (2005). "Modelling 3D spatial objects in a geo-DBMS using a 3D primitive." Computers & Geosciences Vol 31(2): 165-177.

Ellul, C., M. Haklay, et al. (2005). Deriving a generic Topological Data Structure for 3D Data. Topology and Spatial Databases Workshop, Glasgow.

Guting, R. H. and M. Schneider (1993). Realms: A foundation for spatial data types in database systems. 3rd International Symposium on Large Spatial Databases (SSD), Singapore.

Kanellakis, P. C., G. M. Kuper, et al. (1995). "Constraint query languages." Journal of Computer and System Sciences 51: 26-52.

Lema, J. A. C. and R. H. Guting (2002). "Dual grid: A new approach for robust spatial algebra implementation." Geoinformatica Vol 6(1): 57-76.

Lemon, O. and I. Pratt (1998). "Complete logics for QSR [Qualitative Spatial Reasoning]: A guide to plane mereotopology." Journal of Visual Languages and Computing 9: 5-21.

Louwsma, J. H. (2003). Topology versus non-topology storage structures. (case study report) http://www.geo.tudelft.nl/gist/

Molenaar, M. (1990). A formal data structure for three-dimensional vector maps. Fourth International Symposium on Spatial Data Handling, Zurich, Switzerland.

Randell, D. A., Z. Cui, et al. (1992). A spatial logic based on regions and connection. 3rd International Conference on Principles of Knowledge Representation and Reasoning, Cambridge MA, USA, Morgan Kaufmann.

Thompson, R. J. (2005a). 3D Framework for Robust Digital Spatial Models. Large-Scale 3D Data Integration. S. Zlatanova and D. Prosperi. Boca Raton, FL, Taylor & Francis.

Thompson, R. J. (2005b). Proofs of Assertions in the Investigation of the Regular Polytope, GDMC, Delft University of Technology. http://www.geo.tudelft.nl/gist/

Thompson, R. J. and P. van Oosterom (2006a). Interchange of Spatial Data – Inhibiting Factors. Agile 2006, Visegrad, Hungary.

Thompson, R. J. and P. van Oosterom (2006b). Implementation Issues In The Storage Of Spatial Data As Regular Polytopes. UDMS 2006, Aalborg Denmark.

van Oosterom, P., W. Quak, et al. (2003). Polygons: The unstable foundation of spatial modeling. International Society of Photogrammetry and Remote Sensing, Quebec.

Zlatanova, S. (2000). 3D GIS for Urban Development. Graz Austria, Graz University of Technology. PhD Thesis http://www.gdmc.nl/zlatanova/PhDthesis/PhDthesis.html.

Zlatanova, S., A. A. Rahman, et al. (2004). "Topological models and frameworks for 3D spatial objects." Computers and Geosciences.