geometries (Penninga and Van Oosterom 2008; Zawadzki et al. 2012; Robles-Ortega et al. 2012; Kumar 2013, 2014). However, topological relation expression and computation are mostly based on pure algebra (e.g. set algebra with binary operators) (Duckham et al. 2011; Formica et al. 2013; Zhou et al. 2013). The results during the computational processes are purely algebraic relations that have no explicit correlation with the geometric properties (e.g. shapes, distances, directions) of original objects (Chen et al. 2011; Stoter et al. 2011; Gröger and Plümer 2011). Since most geometric properties are abstracted to achieve high-level invariance to certain topological transformations for algebraic expression and computation (Li et al. 2013; Wang et al. 2014), additional components (e.g. boundary, interior, and exterior) are required and the geometric properties of the original geographical objects (e.g. dimension, distance and directions) are lost. When the dimension of the objects arises, it could produce numerous different topological relations that exceed human cognition or even produce meaningless topological relations (Bhatt et al. 2011). All of these problems not only increase the complexity of GIS software, but also lead to separation of topological representation and calculation. These further affect the performance of data organization, dynamical data update, as well as spatial operations and analysis.

Recent studies have paid more attention to the integration of topology and geometric information (Clementini and Di Felice 1996; Egenhofer and Shariff 1998; Li et al. 2013). However, due to a lack of tools that can link geometric representation with algebraic computation, none of these works succeed in integrating the topological and geometric information seamlessly. In the topology-predominant solutions, geometric characteristics, measures and language terms are used to restrict the topological relations (Shariff et al. 1998). However, without a symbolic integration of the geometric characteristics, these geometric constraints can only be pre-computed or used externally to restrict the computational results but not make sense during the computation. In other geometric-predominant solutions, topological relations are used as supporting information. For example, in some CAD applications, the topological relations are integrated in the geometric construction (Peachavanish et al. 2006). Most of these solutions use parametric geometric curves for representation and computation. Although these solutions can increase the accuracy of specific geometric operations (e.g. overlap), the performance of a discrete spatial query and complex spatial analysis is unsatisfactory.

The integration of topological and geometric information can be performed from two aspects: representation and computation. From the representation perspective, a symbolic representation framework that can represent relations between objects at different dimensions will be useful for computing topological relations seamlessly and consistently. From the computation perspective, geometry-oriented and multidimensional operators are useful to fulfill the powerful computation. If the operators can reveal the geometric properties and relations during the entire computing process (e.g. the intersection operator applied to two objects cannot only judge whether they are intersected but also provide the information of what the resulting shapes of their intersection are), it can keep the consistency between the geometric information and topological information. Since geometric objects with different dimensions should be considered simultaneously, a unified dimensional-hierarchy-based computation model for topological relations representation and computation will also be very helpful.

Conformal geometric algebra (CGA), which integrates the geometric representation and algebraic computation (Parra Serra 2009; De Bie 2012; Hitzer et al. 2013), provides an ideal tool for geometry-oriented topological relation (GOTR) computation. The geometries are represented as subspaces in the CGA framework, which makes coordinate-free and adaptive expression and computation possible. The algebraic properties of subspaces also interlink the geometric representation and algebraic computation. GA operators (e.g. *meet*) can be directly performed on geometries with different dimensions in a coordinate-free way (Hestenes 2003). These operators not only are geometry-oriented (i.e. they represent geometries or geometric measures/transformations) but also have powerful algebraic computing abilities. The CGA-based GIS data model (Yuan et al 2011) and the computational framework (Yuan et al. 2012) can support the multi-dimensional unified object representation and operator-based computation (Yuan et al. 2013). Most recently, a hierarchical framework that can support computation for topological relations of simple geometries (e.g. triangles) in a simple way without referring to dimension, has been developed on the foundation of the *meet* product in CGA (Yuan et al. 2014). However, the generalization of the topological relation computation on complex objects and extraction of the power of the geometric expressions of the CGA approaches are still lacking.

In this article, we introduce CGA into the construction of the GOTR representation and computation model. The original CGA representation is first extended for GOTR calculation. Then, the unified *Topo* operator as well as a GOTR calculation algorithm is developed. Finally, we provide a case study to demonstrate how our method can be used to compute the topological relations between objects in the 3D scenes symbolical and dynamically.

2 Theoretical Foundation and Basic Ideas

2.1 Geometric Products and Subspace Representation

The geometric product, the fundamental operator in geometric algebra, is defined in the following form:

Definition 2.1: Let V^n be an n-dimensional linear space. a and b are within V^n. Then, the geometric product of a and b, denoted by ab, is defined as:

$$ab = a \wedge b + a \cdot b = < ab >_{grade(a)+grade(b)} + < ab >_{grade(a)-grade(b)} \tag{1}$$

where $\wedge$ and $\cdot$ represent the *outer* and *inner* products, respectively. *Grade* is the dimension of the subspace. Both the *outer* and *inner* products are the natural correspondence between geometric entities and elements, indicating the orthogonal and co-linear relations, respectively. The geometric product is a linear combination of the *inner* and *outer* products. It can be used to construct and deconstruct objects, and to reveal various kinds of Euclidean relations in a parametric way. By introducing two additional coordinates e_0 and e_∞, the Grassmann structure of geometries is consistent with the *outer* product, while the *inner* product inherits the Euclidean distance (Dorst et al. 2007).

In CGA, every geometry object is represented by subspace elements, while the geometric transformation can be calculated with the transformation products such as *rotor* and *translator* algebraically (Dorst et al. 2007). Any subspace can be expressed and defined by the blade in the following form:

Definition 2.2: A non-zero blade B of grade k, which is the *outer* product of k vector bases taken from a linearly independent set $\{v_i\}_{i=1}^k$, can be written as:

$$B = v_1 \wedge v_2 \wedge \cdots \wedge v_k \tag{2}$$

If given any basis $\{b_i\}_{i=1}^k$ other than $\{v_i\}_{i=1}^k$ for the vector sub-space represented by B, there exists a scalar $\lambda \in R$, such that:

$$B = \lambda b_1 \wedge b_2 \wedge \cdots \wedge b_k \tag{3}$$

© 2015 John Wiley & Sons Ltd

This means it is free to represent the subspace with any chosen factorization, which will be the foundation of the coordinate-free expression and convenient for solving problems algebraically and geometrically.

In CGA, the *intersection* defines the largest common subspace between two subspaces (Hitzer 2005). Assuming two objects W_r and W_s are r-blade and s-blade, respectively, the intersection part of the two objects meets the condition that $X \wedge W_r = 0$ and $X \wedge W_s = 0$. In other words, $X \cdot \langle W_r W_s \rangle_{2n-r-s} = 0 \Longleftrightarrow X \wedge \{[\langle W_r W_s \rangle_{2n-r-s}] I_n\} = 0$, where n is the minimal subspace that contains both objects. Therefore, the *meet* operator can be defined with the *dual* operator as follows:

Definition 2.3: Given two blades W_r and W_s, their *meet* operation can be defined as:

$$M = M_r \cap W_s = [\langle W_r W_s \rangle_{2n-r-s}]^* \tag{4}$$

It is clear that the result of *meet* product is also a multivector. To expand the entire multivector with the coordinate and coefficients representation of the vector form, a linear function can be used to interpret the *meet* operation. In $G(4,1)$ space, which directly corresponds to the three-dimensional Euclidean space R^3 (Yuan et al 2011), the *meet* product has a general form of:

$$M = \beta_0 + \beta_{e_1} e_1 + \cdots \beta_{e_1 e_2} e_1 e_2 + \cdots \beta_{e_1 e_2 e_3 e_0 e_\infty} e_1 e_2 e_3 e_0 e_\infty \tag{5}$$

where β_i is the coefficient of the result blade. Since the *meet* operator performs directly on subspaces, it is independent of objects as well as the dimension of the computation space. Therefore, it can be used to generalize the intersection between different objects. There is ample evidence from prior studies that the *meet* product is also geometrically adaptive, i.e. the result geometries depend only on the original geometries (Hitzer 2005; Yuan et al. 2014).

2.2 Basic Ideas

The goal of GOTR computation is to integrate the natural characteristics of hierarchy in geometric cognition and abstraction, geometric properties of the objects, and the abstract algebraic computations in a unified topological relations computation model. Since there are so many different topological relations, we first hope to find a minimal set of operators that can be used to deal with these relations (Agarwal 2005; Ligozat and Condotta 2005).

The conceptual neighborhood graph (CNG), a hierarchically layered structure of the topological relations (Egenhofer and Mark 1995; Dube and Egenhofer 2012), is an ideal tool to express the hierarchical structure of topological relations. A CNG has one node for each relation and an edge between two nodes if the corresponding relations are conceptual neighbors. Figure 1 illustrates the CNG structure of RCC-8 and the 9-Intersection model (Dube and Egenhofer 2012). The CNG structure of the topological relations can be separated neatly into two parts. In the left part, the *Disjoint*, *Meet* and *Overlaps* express the intersection relations. The only difference among the three relations is the amount they are intersected. In the right part, the relations, including *CoveredBy*, *Equal*, *Covers*, *Contained By* and *Contains*, are based on the *inside/outside* relations. The differences can be identified through the oriented distance between the boundaries of the two objects. Since the intersection and the *inside/outside* relations can be solved with the *meet* product and the *inner* and *outer* product, it is possible to define a unified topological relation between different elements.

© 2015 John Wiley & Sons Ltd

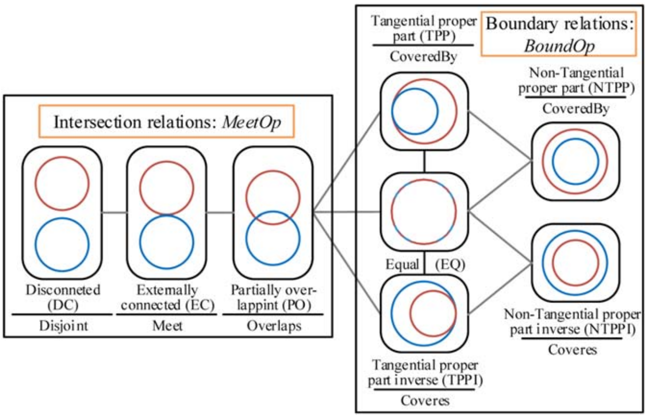

Intersection relations: MeetOp
Disconneted (DC)
Disjoint
Externally connected (EC)
Meet
Partially over-lappint (PO)
Overlaps
Boundary relations: BoundOp
Tangential proper part (TPP)
CoveredBy
Non-Tangential proper part (NTPP)
CoveredBy
Equal (EQ)
Non-Tangential proper part inverse (NTPPI)
Coveres
Tangential proper part inverse (TPPI)
Coveres

Figure 1 CNG structure of RCC-8 and the 9-Intersection model (Dube and Egenhofer 2012)

With the CNG structure of the topological relations, the overall framework of the GOTR calculation model can be constructed (Figure 2). Firstly, the CGA data model can be extended with new objects, definitions, and more powerful object expressions. By defining a geometric characteristic-preserved object abstraction structure, multidimensional objects can then be represented with the multivector structure hierarchically. Two topological operators, the *MeetOp* and the *BoundOp,* are developed on the foundation of the *meet* and *inner/outer* products to compute the *intersection* and *inner/outer* relations, respectively. Then a generalized operator *TopoOp* is defined by integrating the *MeetOp* and the *BoundOp*. With the *TopoOp*, a GOTR computation algorithm can be developed. The key procedures of the model are: (1) multivector-based multidimensional-unified definitions and abstractions for real geometric object representation with subspaces; (2) the intersection and *inside/outside* relations that are computed and unified with the *meet* operator and the *inner/outer* products, respectively; and (3) the *Topo* operator that computes basic topological relations directly from the intersection and *inside/outside* relations.

3 Multi-dimensional Representation Framework for GOTR Computation

3.1 Conceptual Modeling and Definitions

The representation framework for GOTR is different from existing solutions in the following ways: (1) the representation framework of objects should be abstracted correctly to support both complex object representation and topological relations computation; (2) a flexible structure of the object representation and organization should be constructed to support both the representation and computation; (3) the geometric semantics should be contained and preserved in the representation. The CGA-based multidimensional-unified data model by Yuan et al (2011) integrates the dimensional construction structures (Grassmann structure) with a single algebraic structure (Multivector). This data model can be extended to meet the above requirements. To achieve the hierarchal and semantic object representation, the following concepts are first defined:

3.1.1 GeoBase

The *GeoBase* represents the fundamental atom that can be expressed directly in CGA. To be compatible with existing GIS representation, it is important to maintain the dimensional

© 2015 John Wiley & Sons Ltd

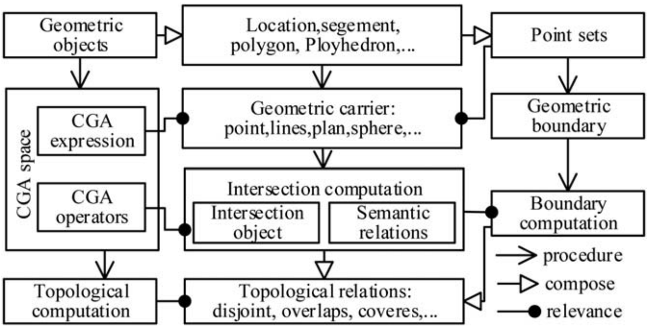


Figure 2 The framework of the GOTR computation model

structure (i.e. the Grassmann structure). Therefore, we use the outer product to express the *GeoBase*. The formal definition of *GeoBase* is:

Definition 3.1: A *GeoBase* is defined as the basic geometric object that can be directly expressed by the *outer* product of k null vectors (always representing conformal points) in the CGA space $Cl_{4,1}$:

$$GeoBase(k) = p_1 \wedge p_2 \wedge \cdots \wedge p_k \tag{6}$$

where k is the dimension and p_i is the i-th point (vector) components of $GeoBase(k)$. A k-dimension *GeoBase* can also be seen as a k-blade, which is the basic expression and computation subspace in geometric algebra.

3.1.2 GeoObj

For objects such as segments, planes and volumes, the traditional CGA representation has no clear boundaries. For complete and accurate representation, the boundary information with point coordinate constraints should be added to the CGA data model (Yuan et al. 2011).

Definition 3.2: *GeoObj* is defined as the geometric object with a boundary that should add an additional point set to restrict its extent, and can be written as:

$$GeoObj_k = \overset{k}{\underset{i=1}{\wedge}} (b_1, b_2, \cdots, b_{k+1})\{p_1, p_1, \cdots, p_n\} \tag{7}$$

where k is the dimension, $\wedge_{i=1}^{k+1}(b_1, b_2, \cdots, b_{k+1})$ is the CGA representation of the *GeoObj*, and $\{p_1, p_1, \cdots, p_n\}$ is the point set that constructs the object boundary.

3.1.3 GeoCarrier

Real geometries are irregular and have finite boundaries. Thus, it is not easy to directly construct the CGA representation with the *outer* or *inner* product. Since the partial surface of an infinite plane or sphere can be used to represent the boundaries of the objects, we can use it to form the *GeoCarrier*, i.e. the objects that contain or carry the other objects.

Definition 3.3: *GeoCarrier* is defined as the container or carrier of a *GeoObj*, which has no fixed boundary. So it cannot be directly expressed by blade. In CGA space, it can be written as:

$$GeoCarrier_k = CGA\{p_1, p_1, \cdots, p_n\} \tag{8}$$

where k is the dimension and $p_1, p_2, \cdots, p_{k+1}$ are the boundary feature points. $CGA\{...\}$ means the geometric representation function in CGA. In CGA, according to whether the infinity point is included, the *GeoCarrier* can be divided into *round GeoCarrier* and *flat GeoCarrier*. The *round GeoCarrier* has no e_∞ component, which suggests the object has finite area/volume/hypervolume, while the *flat GeoCarrier* can express lines and planes. The inclusion of e_∞ means it can stretch to infinity. There exists:

$$GeoCarrier_k = \begin{cases} p_1 \wedge p_2 \wedge \cdots \wedge p_{k+1}, & round\ GeoCarrier \\ p_1 \wedge p_2 \wedge \cdots \wedge p_{k+1} \wedge e_\infty, & flat\ GeoCarrier \end{cases} \tag{9}$$

In our data model, the *GeoCarrier* includes point, circle, sphere, line and surface. The integration of the finite and infinite expressions provides a solid background for unified expression and computation.

With Definition 3.3, *GeoObj* can be rewritten as:

$$GeoObj_k = \{GeoCarrier_k, \{p_1, p_1, \cdots, p_n\}\} \tag{10}$$

where $\{p_1, p_1, \cdots, p_n\}$ is the boundary points set, and $GeoCarrier_k$ can be constructed by the linearly independent feature points extracted from the boundary point set. Based on the modified expression, geometric objects can be commutated as CGA objects and the boundary characteristic is concurrently maintained.

3.1.4 GeoSema

Since semantics are clear and important in the CGA data model, we also define the sematic components for expressing rich information and accelerating calculation of topological relations. Semantic information is rich, although only a little of it is used here for the most necessary semantic components. The formal definition of the GeoSema is:

Definition 3.4: A *GeoSema* is defined as the CGA object, which serves as characters of objects, and may be written as:

$$GeoSema = \{n, blades\} \tag{11}$$

where n are the number of elements of semantics, *GeoCarriers* are the geometric elements and containers, and *blades* are the geometric algebra representation of sematic relations. Besides *GeoCarriers*, *GeoSema* is another form of *Geobase*, which is the computation result of CGA operators. The *GeoSema* also supports CGA calculation.

3.2 Basic Geometric Elements for the Geometric-oriented Representation

In 3D CGA space, there are total 32 subspace elements (Table 1). These blades contain the representation of geometries, metrics, normals and tangents, in both direct and dual

Table 1 Fundamental 32 blades in CGA $Cl(3,2)$

Objects	Subspace	Objects	Subspace	Objects	Subspace
Scalar	1.0, e_∞	Flat point	$e_{0\infty}$	Trivector	e_{123}
Vector	e_1, e_2, e_3	Bivector	e_{12}, e_{13}, e_{23}	Tangent trivector	e_{1230}
Point	e_0	Tangent bivector	e_{120}, e_{130}, e_{230}	Free trivector	$e_{123\infty}$
Tangent vector	e_{10}, e_{20}, e_{30}	Free bivector	$e_{12\infty}$, $e_{13\infty}$, $e_{23\infty}$	Plane	$e_{120\infty}$, $e_{130\infty}$, $e_{230\infty}$
Free vector	$e_{1\infty}$, $e_{2\infty}$, $e_{3\infty}$	Line	$e_{10\infty}$, $e_{20\infty}$, $e_{30\infty}$	Pseudoscalar	$e_{1230\infty}$

representation. There is more information that can be revealed from the CGA space. In addition, with the powerful CGA operators, objects can be constructed and represented from different perspectives. For example, the *outer* product of two points according to the Grassmann structure, a dual to its normal vector, and an intersection of two planes, etc., can represent a straight line. All these expressions are parametrical and can be transformed from one to another. It may be a little more complex, yet it provides a more flexible representational method than traditional Euclidean geometry for geometric objects.

As is shown in Table 1, six types of components that simplify the expression are selected as follows:

1. *Rounds*: rounds suggest that the objects have finite areas/volumes/hypervolumes. In CGA, they can be constructed by the outer product. Points, within the 32 blades, are the most basic round objects. Higher dimensional round objects can be obtained by the outer product of points (e.g. point pair, circle, sphere, hypersphere, etc.). The expression of a k-dimensional *round* is:

$$round : r_k = p_1 \wedge p_2 \wedge \cdots \wedge p_k \tag{12}$$

2. *Flats*: flats are the rounds containing the point at infinity e_∞ in its formula. The inclusion of e_∞ means it can stretch to infinity. There exist flat points, lines and planes. Flats can be extended to higher dimensions, where we have hyperplanes. The expression of *a* k-dimensional *flat* is:

$$flat : f_k = p_1 \wedge p_2 \wedge \cdots \wedge p_k \wedge e_\infty \tag{13}$$

3. *Euclidean blades*: In CGA there still exists *Euclidean blades* (e.g. vectors, bivectors, trivectors), which can be used to represent metric characteristics of geometrical objects (e.g. area, volume). The expression of k-dimensional *Euclidean blades* is:

$$Euclidean\ blades : v_k = ke_1 \wedge e_2 \wedge \cdots \wedge e_k \tag{14}$$

4. *Free Blades*: These are elements without position but indicating the direction (e.g. free vectors, free bivectors, free trivectors). From the definition of CGA, we know the blades

© 2015 John Wiley & Sons Ltd

have no position, which means there is no e_0 component in their formula. So, the expression is:

$$free\ \ blades : fb_k = v_k \wedge e_\infty \tag{15}$$

5. *Tangent Blades*: compared with the *Free Blades, Tangent Blades* have a clear point of tangent, including tangent vectors, tangent bivectors, tangent trivectors. So, these blades have the e_0 component. The expression of k-dimensional *Tangent Blades* is:

$$tangent\ \ blades : tb_k = v_k \wedge e_0 \tag{16}$$

6. *Others:* The scalar and pseudoscalar are not used directly in CGA representation, because the above blades, which are used to represent geometry in CGA, are all null-blades (i.e. their norms are zero), and the weight-independent characteristic just make no sense to representation. The pseudoscalars indicate the whole space of CGA, which can be used in the computation of *dual operator*.

Six types of basic geometries (*point*, *point pair,*[1] *line, plane, circle, and sphere*) are used for object expression in our previous multidimensional-unified data model (Yuan et al. 2011). Although the six basic geometric objects are suitable for expressing complex objects, more information should be extracted for topological relation computation. Since the dual representations have more abundant semantic meanings (e.g. orthogonality between the original space and the dual space), we further integrate their dual representations (Table 2). The other types (*Euclidean blades*, *Free Blades* and *Tangent Blades*) are integrated as components of *GeoSema*.

With the introduction of the null vector e_0, e_∞ in CGA, the expression of the round objects is generalized. In addition, the directions and center locations of objects can be directly reproduced with the cross ratio. The *dual* of a round object R encodes the center c and radius r, while the dual of a flat object is $F=-p \cdot ne_\infty$, where p is a certain point in the flat object F, and n is the normal blade. $e_\infty \wedge F=0$ also exists, as does $F^2 \neq 0$. Importantly, these characteristics are generalized for object relation calculation (Dorst and Mann 2002). Other elements, such as the free blades and tangent blades, have similar properties: $\delta^2=0$ is for free blades and $\eta^2=0$ for tangent blades. These properties will be helpful in the topological relation computation.

3.3 The Overall Representation Structure of GIS Objects

As the foundation of the geometric representation, the data model needs to represent the objects in CGA space and compute the geometric relations with CGA operators. These clear and meaningful geometric relations are then abstracted and summarized to produce the real topological relations between these objects. The overall structure of the data model is depicted in Figure 3. First, all geographic elements, including points, lines and polygons, are abstracted as point sets. Next, the geometric objects and boundaries are defined to express both the blade covering the objects and the real boundary. Four fundamental elements/blades (*GASphere*, *GAPoint*, *GALine* and *GAPlane*) are then introduced in this data model. In the infinity boundary model, we use these objects to fully express the sphere and point object. With the restrictions of boundary points, we can express the lines and segments, the flat plane objects, the triangle objects and polyhedron objects in three dimensions.

Table 2 Representation of elements in dual space

Type	Object	Dual expression	Geometric semantics
Rounds	Round point	$P{=}e_0{+}p{+}0.5p^2e_\infty$	$\left\{\begin{array}{l} \text{Direction} \;:\; -(e_\infty \cdot R)\wedge e_\infty \\ \text{Location} \;:\; A_k/e_\infty \cdot A_k \\ \text{Square} \;: \\ \left.\begin{array}{ll} \text{Imaginary rounds} & R^2<0 \\ \text{Point} & R^2{=}0 \\ \text{Real rounds} & R^2>0 \end{array}\right\} \end{array}\right.$
	Point pair	$T{=}(c{-}\frac{1}{2}\rho^2e_\infty)\wedge\prod\wedge\prod$	
	Circle	$K{=}(c{-}\frac{1}{2}\rho^2e_\infty)\wedge\prod$	
	Sphere	$\sum{=}c{+}\frac{1}{2}\rho^2e_\infty$	
Flats	Flat point	$\Psi{=}\lambda_1\wedge\lambda_2$	$\left\{\begin{array}{l} \text{Direction} \;:\; e_\infty \cdot R \\ \text{Location} \;:\; (q\cdot R)/R \\ \text{Square} \;:\; F^2\neq 0 \end{array}\right.$
	Line	$\Lambda{=}(p{+}\alpha e_\infty)\wedge(p{+}\beta e_\infty)$	
	Plane	$\prod{=}n{+}\delta e_\infty$	

p is the Euclidean vector, and A_k is the k blade in CGA space.

4 The Topological Relation Computation Framework

In this section, the hierarchical computation framework of the topological relations is developed according to the CNG structure. Firstly, two operators, the *MeetOp* and *BoundOp* are defined to compute the intersection relation and the *inside/outside* relations. Then the *TopoOp* is defined to compute the combination of the topological relations in a unified way. With these operators, both the geometric properties and the topological relations are iteratively computed according to the multidimensional hierarchy of objects representation.

4.1 The Unified Computation of the Intersection Relations

We first apply the *meet* product to compute the intersection relations between basic elements. From Equations (12) and (13), we know that *meet* can be unified to compute the intersection of the elements in G. However, it is not enough to compute the topological relations with the *meet* product directly. The reasons are: (1) the result of the *meet* product is a real geometric element; however, binary operations, which can determine whether the two objects are *intersected/touched/disjoined*, are much more convenient for topological computation; (2) even if the two objects are not intersected in R, their meet also exists, which will confuse the topological relation inference; and (3) the "∩" is computationally expensive, thus more efficient methods should be provided for computation of topological relations. Therefore, we define the *MeetOp* operator of *GeoObjs* by applying the *meet* product to *GeoCarrier* to work out the intersection relations of *GeoObjs*:

Definition 4.1: *MeetOp* is defined with the *meet* operator, which is used to calculate the intersection relations of *GeoObjs*, and can be written as:

$$\begin{aligned} MeetOp(GeoObjA, GeoObjB) &= GeoCarrierA \cap GeoCarrierB \\ &= \{RMeet, Tangent, IMeet\} \end{aligned} \tag{17}$$

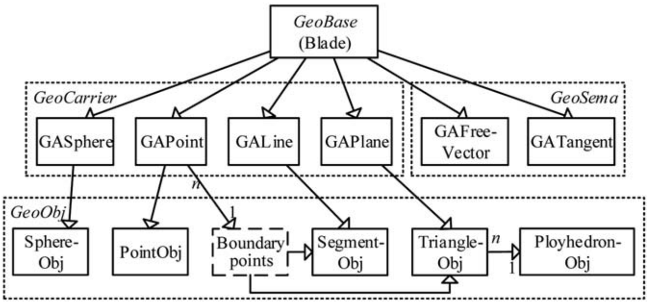


Figure 3 Representing geometry model based on CGA

Where $RMeet, Tangent$ and $IMeet$ mean the real intersection (i.e. the two objects intersect at a certain geometry in the Euclidean space), touch intersection (i.e. the two objects are tangent to each other) and imaginary intersection respectively (i.e. the two objects are disjoint in the Euclidean space but interlinked in the CGA space. e.g. the meet of two disjointed spheres results in an image circle; see Dorst (2007) for further explanations.) Since we have defined the imaginary intersection, the *MeetOp* between two *GeoObj*s will always be meaningful, and the result of *MeetOp* only depends on the relative location relations between the two *GeoObj*s. We then have the following theorem:

Theorem 4.1: For any two *GeoCarriers* $A, B \in G$, the *meet* operator B is closed and the result must be *GeoCarrier*. Assuming $A \neq B$, we have:

$$A \cap B = \{GeoCarrier, GeoSema\} \tag{18}$$

Proof: Since the set of all blades in G is closed under the *outer* product of geometric algebra, it can be easily proved that the *meet* always exists and is meaningful in G. In fact, for any vector $v \in G$, if $A \wedge B \neq 0$, then $v \cdot A = 0$ and $v \cdot B = 0$ if and only if $v \cdot (A \wedge B) = 0$, and if $v \wedge A = 0$ or $v \wedge B = 0$, then $v \wedge (A \wedge B) = 0$. So, we have:

$$A \cap B = A^* \wedge B^* \tag{19}$$

Clearly, the *outer* product can produce the dual representation of the intersection between the two geometries. Since both the *dual* and *outer* products in G are closed, the result of the *meet* product between blades always exists. Meanwhile, since the CGA system also includes a representation of projective space (connected with the role of e_0), every two elements are intersected with each other. The elements that are not intersected in the Euclidean space will produce an imaginary element, with the *meet* product applied in G. For example, two disjointed spheres intersect at an imaginary circle in G, will keep the interrelationship between the two spheres. This is important for computing the intersection relation in a more unified, general and geometry-oriented way.

The *meet* operator between basic elements and the dimensional computation results are listed in Table 3. Since in CGA, even two disjoint *GeoCarriers* can intersect at the infinite point to form the imaginary objects, it is possible to determine the intersection relationship using algebraic analysis (Roa et al. 2011). Here we have the following theorem:

© 2015 John Wiley & Sons Ltd

Theorem 4.2: Given two *GeoCarrier* W and S, their meet $B = W \cap S$, the W and S have no empty intersection in R^3 if and only if $B^2 \geq 0$, and there are empty intersections if and only if $B^2 < 0$.

Proof: According to Table 2, the geometric properties can be easily derived from the CGA expression of *GeoCarrier* and *GeoSema*, which has great significance for spatial relation calculations. The square properties, such as the module of blades, can be used to distinguish the type of *GeoBase*. When $B^2 \geq 0$, the result must be a real round, point or flat object, otherwise that must be imaginary round or *GeoSema* objects that indicate the same directions or tangent points and directions of the meet operands. As a result of the attributes described above, we discuss each relation between different *GeoSema* objects separately (Table 3).

So the *MeetOp* operation has three possible results that are partly consistent with B^2, which:

$$MeetOp(W,S) = W \cap S = \begin{cases} RMeet & \mathrm{B}^2 > 0 \\ Tangent & \mathrm{B}^2 = 0 \\ IMeet & \mathrm{B}^2 < 0 \end{cases} \tag{20}$$

The exceptional case is that when the intersecting objects are two *Flat GeoCarrier*, there is no *Tangent* result but an *IMeet* result for their parallel relation.

4.2 The Unified Computation of the Insider/outside Relations

Another important topological relation is whether an object is *inside* or *outside* another. Since the conformal transformation projects the Euclidean space into the Riemann sphere, some Euclidean objects, such as segments/lines and planes, have an infinite component, while spherical objects, such as points (a sphere with zero radius in CGA), circles, and spheres, do not have an infinite component. In other words, these objects should limit their real boundaries with boundary objects that can be constructed by the coordinate constraints through a series of points. However, spherical objects have native boundaries that are constrained by the distance of their center points and radii. Therefore, we separate these two sets of objects and discuss the classification of the inside/outside relations with different operators.

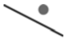
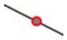
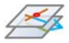
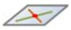
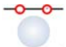
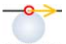
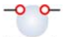
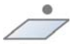
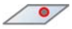
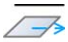
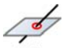
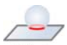
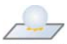
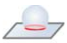
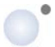
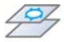
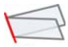
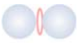
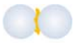
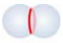
Definition 4.2: *BoundOp* is defined to work out the inclusion relation of point sets and *GeoCarrier*, and can be written as:

$$BoundOp(Pt, GeoObj) = \{Inside, Outside\} \tag{21}$$

The results of *BoundOp* operation are *inside* or *outside* relations, and Equation (21) can be generalized to *GeoObj*:

$$\begin{aligned} &BoundOp(GeoObjA, GeoObjB) \\ &= BoundOp(ptsA, GeoObjB) \& BoundOp(ptsB, GeoObjA) \\ &= (nInsideB, nInsideA) \\ &= (ma, mb), (na, nb), (1,1), (1, nb), (na, 1), (0,0), (0, nb), (na, 0) \end{aligned} \tag{22}$$

Table 3 The intersection table of basic elements

	Points meet (point with *k*-blade)		Flats meet (*m*-flat with *n*-flat)		Rounds meet (sphere with *k*-blade)		
Meet operands	$B^2>0$	$B^2=0$	$B^2=0$	$B^2>0$	$B^2<0$	$B^2=0$	$B^2>0$
$k=3$; $m=3$; $n=3$	 Scalar	 Point	 Fscalar/Fvector	 Flat point	 IPoint pair	 TVector	 Point pair
$k=4$; $m=3$; $n=4$	 Scalar	 Point	 FVector	 Point	 ICircle	 TBivector	 Circle
$k=4$; $m=4$; $n=4$	 Scalar	Point	 FBivector	 Line	 ICircle	 TBivector	 Circle

IPoint pair and ICircle indicate the image point pair and image circle; TVector and TBivector indicate the angent vector and tangent bivector; FVector and FBivector indicate the free vector and free bivector

where n_a and n_b are the number of *GeoObjs* boundary points, *nInsideA*, *nInsideB* are the number of boundary points located in the interior of *GeoObjs*, m_a, m_b meet $0 \le m_a \le n_a$, $0 \le m_b \le n_b$.

For the inside boundary points, two-tuples are used to represent the boundary relationship of *GeoObjs*. Since the *GeoCarrier* of a *GeoObj* can be a *round* or *flat* object and the geometric features of the round and the flat are much different from each other, the solution of *BoundOP* should be considered in two different conditions:

1. *BoundOp*(*Pt*, *round GeoObj*): An important advantage of CGA is the identical meaning of the *inner* product, which embeds the Euclidean distance metrics between two points/circles/spheres (i.e. *roundGeoCarrier*) in the CGA space. Given two points A and B, we have $d=-2A\bullet B$. If the two points are the same, then $d=0$, i.e. $A\bullet B=0$. So, if the inner product of the two points equals zero, the two points coincide. Conversely, the two points do not coincide if $A\bullet B \neq 0$. Since the points are spheres with radii of zero, we also have $-2(S_1\bullet S_2)=r_1^2-r_2^2$ (Hildenbrand 2013). Therefore, to degenerate the sphere S_1 to a point P_1 with $r_1=0$, we have:

$$\begin{cases} P_1 \ \textit{is inside}\ S_2 & P_1\bullet S_2 > 0 \\ P_1 \ \textit{is on}\ S_2 & P_1\bullet S_2 = 0 \\ P_1 \ \textit{is outside}\ S_2 & P_1\bullet S_2 < 0 \end{cases} \tag{23}$$

2. *BoundOp*(*Pt*, *flat GeoObj*): Since the *GeoCarrier* object is directly represented by the outer product from point sets, the anti-symmetric characteristic allows the constructed objects to have certain directions (e.g. the line $L_1=A\wedge B\wedge e_\infty$ and $L_2=B\wedge A\wedge e_\infty$ are in opposite directions). Since any arbitrary object can be seen as an orderly construction of a point series and any point p which meets the condition $p\wedge A=0$ on the object A, the wedge between the uncertain point and the point series that has already been on the object can be distinguished by the sign of the outer product. For a given point p and a *flat GeoCarrier* A that has the expression of $A=A_1\wedge A_2\wedge\cdots\wedge A_n\wedge e_\infty$, where

$n < 5$, we can replace p with the $i-th$ point A_i to form an outer product computation formula as:

$$t_i=(A_1\wedge\cdots\wedge A_{i-1}\wedge P\wedge\cdots\wedge A_n\wedge e_\infty) \tag{24}$$

So the *BoundOp* can be commutated by the following equations:

$$\begin{cases} P \text{ is inside } A & t_i < 0 \\ P \text{ is outside } A & t_i \geq 0 \end{cases} \tag{25}$$

4.3 *Composition of the Overall Topological Relations*

Based on the *MeetOp* and *BoundOp*, we can define the final topological operator, *TopoOp*, according to the CNG:

Definition 4.3: *TopoOp* is the combination of Definitions 4.1 and 4.2, which can classify and identify the specific topological relations. It can be defined as:

$$\begin{aligned} &TopoOp(GeoObiA, GeoObiB) \\ &=MeetOp(GeoObiA, GeoObiB)\&BoundOp(GeoObiA, GeoObiB) \\ &=\{PO, EQ, EC, TPPI, TPP, DC, NTPPI, NTPP\} \end{aligned} \tag{26}$$

According to the GNC structure of the topological relations, we can define eight basic topological relations: *PO*, *EQ*, *EC*, *TPPI*, *TPP*, *DC*, *NTPPI* and *NTPP*. These relations can be obtained by combining the *MeetOp* and *BoundOp* (Table 4).

The blade classification operators, such as *.isGaFlat*() and *.GaRound*(), and topological relation computation algorithm (Algorithm 1) are developed to obtain the topological relations. In Algorithm 1, the *GeoCarriers* of two objects *A* and *B* are first constructed by the boundary points *A.pts* and *B.pts*. The intersection relations are first computed by the square of the *meet* product of *A* and *B*. The real and imaginary intersection relations are determined by the sign of the square of *meet*. Then the inside and outside relations are computed with the inner and outer product. Finally, the topological relation between *A* and *B* is computed with the *TopoOp*. The structure of our algorithm is simpler and geometrically clearer than most of the current topological computation algorithms. In addition, since all the operators used in the algorithm are multi-dimensional unified, the computation of topological relations between objects with different dimensions can be performed with the same algorithm structures and same operators. Thus, the algorithm is also multidimensional-unified.

5 Case Study

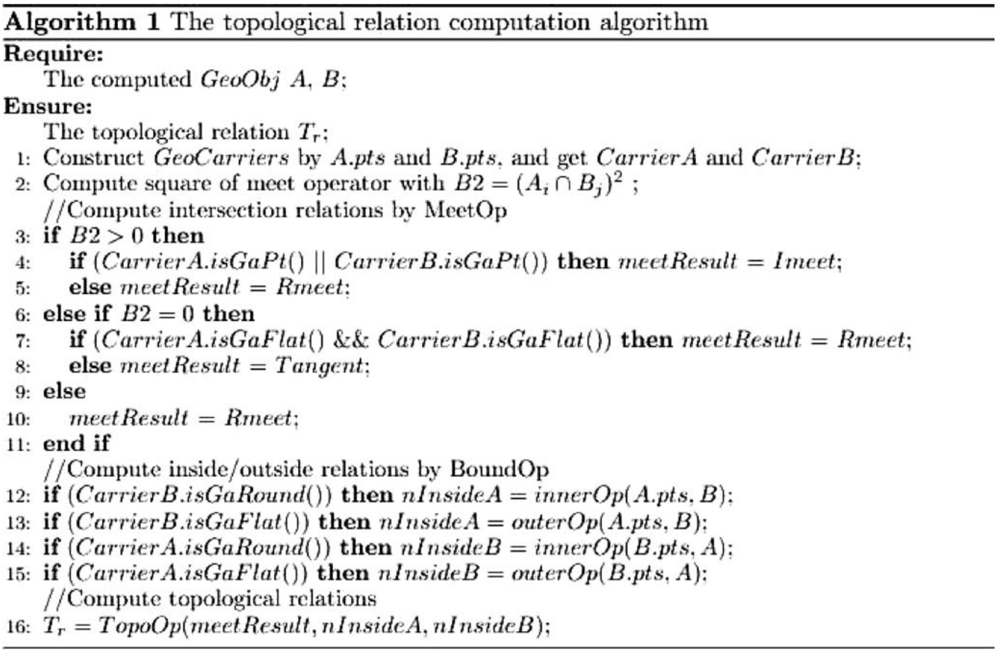
We implemented the geometry-oriented multidimensional topology representation and computation model in the Clifford algebra-based unified spatial-temporal analytic environment (CAUSTA) (Yuan et al. 2010). The data we used to demonstrate the topology computation is a 3D residential district (Figure 4). The buildings are first imported from the CityGML

Table 4 The *TopoOp* operator

MeetOp(*A*,*B*)	*BoundOp*(*A*,*B*)	*TopoOp*(*A*,*B*)	Topological Relations
Real meet	(m_a,m_b)	PO	Overlaps
	$(1, n_b)$	TPPI	Covers
	$(0, n_b)$	NTPPI	Contains
	$(n_a,1)$	TPP	CoveredBy
	$(n_a,0)$	NTPP	ContainedBy
	(n_a,n_b)	EQ	Equal
Tangent meet	(1,1)	EC	Meet
Image meet	(0,0)	DC	Disjoin

data and then represented by the CGA data model (Yuan et al. 2011). The buildings are represented object-oriented in the unified multivector structure. Unlike classical 3D GIS, which separates different dimensional objects into several different layers, the dimension hierarchy of the buildings is also represented by the CGA subspace representation with a multivector structure.

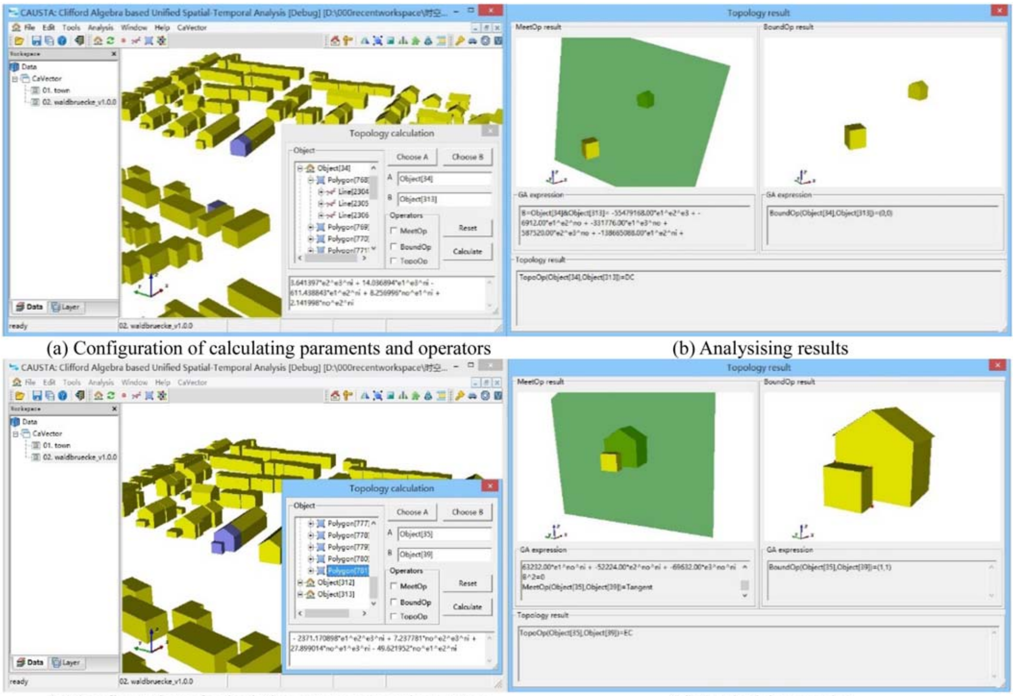
Two different buildings were selected from the scene to illustrate the topological representation and computation. The symbolic representation of the geometries with the CGA outer product representation, which can be adjusted and modified with a hierarchical tree structure (Figure 4a), clearly distinguishes our approach from existing representation models. Based on the symbolic representation, the computation parameters can be provided interactively before the computation and dynamically updated during the computation. This provides a flexible and convincing way for continuous or dynamic topological relations



Algorithm 1 The topological relation computation algorithm

```
Require:
    The computed $GeoObj\ A,\ B$;
Ensure:
    The topological relation $T_r$;
Construct $GeoCarriers$ by $A.pts$ and $B.pts$, and get $CarrierA$ and $CarrierB$;
Compute square of meet operator with $B2 = (A_i \cap B_j)^2$ ;
//Compute intersection relations by MeetOp
if $B2 > 0$ then
    if ($CarrierA.isGaPt()$ || $CarrierB.isGaPt()$) then $meetResult = Imeet$;
    else $meetResult = Rmeet$;
else if $B2 = 0$ then
    if ($CarrierA.isGaFlat()$ && $CarrierB.isGaFlat()$) then $meetResult = Rmeet$;
    else $meetResult = Tangent$;
else
    $meetResult = Rmeet$;
end if
//Compute inside/outside relations by BoundOp
if ($CarrierB.isGaRound()$) then $nInsideA = innerOp(A.pts, B)$;
if ($CarrierB.isGaFlat()$) then $nInsideA = outerOp(A.pts, B)$;
if ($CarrierA.isGaRound()$) then $nInsideB = innerOp(B.pts, A)$;
if ($CarrierA.isGaFlat()$) then $nInsideB = outerOp(B.pts, A)$;
//Compute topological relations
$T_r = TopoOp(meetResult, nInsideA, nInsideB)$;
```

© 2015 John Wiley & Sons Ltd



```
3.641397*e2^e3^ni + 14.036894*e1^e3^ni -
611.438843*e1^e2^ni + 8.256996*no^e1^ni +
2.141998*no^e2^ni
```

```
B=Object[34]&Object[313]= -55479168.00*e1^e2^e3 + -
6912.00*e1^e2^no + -331776.00*e1^e3^no +
587520.00*e2^e3^no + -138665088.00*e1^e2^ni +
```

```
BoundOp(Object[34],Object[313])=(0,0)
```

```
TopoOp(Object[34],Object[313])=DC
```

(a) Configuration of calculating paraments and operators (b) Analysising results

```
- 2371.170898*e1^e2^e3^ni + 7.237781*no^e2^e3^ni +
27.899014*no^e1^e3^ni - 49.621952*no^e1^e2^ni
```

```
63232.00*e1^no^ni + -52224.00*e2^no^ni + -69632.00*e3^no^ni
B^2=0
MeetOp(Object[35],Object[39])=Tangent
```

```
BoundOp(Object[35],Object[39])=(1,1)
```

```
TopoOp(Object[35],Object[39])=EC
```

(c) Configuration of calculating paraments and operators (d) Analysising results

Figure 4 Case study of topology analysis. The left side figures show the configuration of calculating paraments and operators; the right side figures are the analysis results based on the methods proposed by this article. To enumerate the sequences of operations detail, the CGA expressions and visualizations of *MeetOp*() and *BoundOp*() are presented in the result windows.

computation. By changing the shape and location of a building, the topological relations changes are logged and represented. Figures 4b and d represent how the shape changes of one building were reflected in the multivector representation and finally caused the topological relations changes simultaneously.

Compared with existing topological relations computation methods, the geometrically integrated topological computations are more compact and geometrically clearer. In addition, the hierarchical tree structure not only clearly represents the geometric structures and the dimensional hierarchy of the objects, but also provides a convincing way for multidimensional-unified topological relations computation. The *MeetOp*, *BoundOp* and *TopoOp* can be directly and recursively applied to the objects with different dimensions according to their dimension hierarchy (Figures 4a and c). Since both representation and computation of objects in the CGA model are based on the *MVTree* structure (Yuan et al. 2014), the representation and computation model of multidimensional objects are simplified and unified. Since the computation of the topological relations in our solution has algebraic and geometric representations, the *MeetOp*, *BoundOp* and *TopoOp* can be interactively applied to the whole multidimensional objects or an individual object in the multivector tree (Figures 4a, c). Both the algebraic and geometric representations are provided in the console box and scene, respectively. As each component of the algebraic multivector representation is a real geometry, the symbolic representation can also be visualized in the result windows (Figures 4b, d). This

© 2015 John Wiley & Sons Ltd

provides a direct way for the user to interactively explore and verify the computation procedure.

One of the key advantages of CGA representation is that each object in the scene is symbolically represented using the geometric algebra expressions. When the symbolic expression of the objects in the tree is modified or operated with certain operators, the geometric representation, as well as the relations between different dimensions, is changed adaptively according to the CGA construction rules. Both the coefficients of the object expression multivectors and the topological computation operators can be dynamically set and changed during the computation (Figures 4a, c). Since all the computations are dimensional and shape independent, our model will be helpful for interactive and dynamical topological relation computation and simulation. This also provides a flexible and effective structure for developing complex multidimensional spatial analysis algorithms.

6 Discussion

The traditional topology calculations are mainly based on the point-set topology theory in which all of the geometries, such as GSG, B-rep and GTP, are first abstracted into the topological representations (Wu 2004; Lee and Kwan 2005; Cheng et al. 2008). Several complex algorithms are developed to calculate the boundaries, the interior and exterior relations (Theobald 2001; Nedas et al. 2007; Du et al. 2008). Using the binary operators in topological computation models, the topological relation then can be calculated. However, calculating topological relations in the CGA framework is simple, unified, and direct. The geometries, such as points, lines and volumes, could be represented by subspaces and multivectors directly and integrated in the CGA data model (Yuan et al. 2011, 2014). The geometries can also be directly computed with the multidimensional-unified and coordinate-free CGA operators. These geometry operators (*projection*, *intersection*, etc.) can produce rich attribute and relation information and also contribute to direct and meaningful representation. Therefore, this computing model in CGA not only provides more meaningful topological relations, but also greatly reduces the complexity of current GIS. The property comparisons of our solution and existing methods are shown in Table 5.

GOTR computations can achieve the topological consistency with geometric transformations in a parametric way. The CGA-based topological relation computation can extract the boundaries, interior, and exterior parts of multidimensional objects in a geometry-oriented way with very few operators. The representation and computation in CGA space, which integrates the Euclidean and Projective geometry in a coordinate-free way, can very easily extract the topological equivalence relation (e.g. homeomorphism or diffiomorphism).

Although we have only defined eight types of topological relations that are compatible with the OGC standard, it can be easily extended to much more complex computations. The *M eetOp* and *BoundOp* can be easily used in the computation of the traditional 9-IM model, the CBM model, and in many other models. This characteristic will greatly improve the capability of the multidimensionality unified GIS. In addition, because the operators are reduced to only three basic fundamental products, the complexity of the algorithm construction is significantly reduced. The symbolic and direct geometric computation also provides convenience for algorithm optimization. Therefore, the theoretical analysis of topological relation changes will be possible.

© 2015 John Wiley & Sons Ltd

Table 5 The property comparison between different topological relation models

Model	Direct Expression	Formulized Expression	Immanence	Multidimen-sional-unified	Adaptive
Point-Set	-	+++	-	-	+
RCC Model	+	++	-	-	-
CBM Model	+	+++	+	-	-
Semantic Model	+	-	-	-	-
Our Model	++	+++	++	++	+++

The "+" means advantage in this property, and the more "+" means more advantage. The "-" means weakness in this property.

Since the topological relation computation under the CGA framework is based on the CNG, geometry-oriented topology is closer to human cognition. In our GOTR computations, the intersection, insider/outside relations can be not only separated but also combined together. There are clear and direct geometric representations during each computing step. All these computations are geometric and cognizable. In our CAUSTA system, we visualized the internal objects/relations and their geometric semantics interactively. In this way, users can not only recognize how the coefficients and characteristics of the geometries influence their topological relations, but can also dynamically change the coefficients of the representations and support the computation of the spatial-temporal topologies.

Powerful CGA computation tools can be introduced to further extend the border of topological relation computation. The extension can be provided in three ways: firstly, more powerful CGA operators and complex relations can be integrated in the CGA framework. In CGA, most of metrics and operators can be represented as certain products of geometries (e.g. a Euclidean distance can be represented as the inner product of two points/spheres). Relations such as distances, directions and even motions can be represented as subspaces. They can be integrated into the GIS spatial relation in a similar way. There exists great potential to build a unified model for geometry-oriented spatial relationship representation and computation. Secondly, the computation space can be extended. In CGA planes and spheres are isomorphic, so the Euclidean space and spherical space can be unified. Therefore, the topological relation computation model can easily be extended into the spherical space. This will be helpful in global scale analysis. Thirdly, CGA has powerful tools for representation of motions and transformations. It is possible to extend current spatial analysis tools to spatial-temporal dimensions.

Due to the simplicity and directness of the model, more geometry-oriented operators, metric relations, etc. can be integrated in a similar way. Thus, it provides the potential to be extended to a unified framework for the geometry-oriented spatial relation representation and computation. It also provides a possible way for adaptive and parametric temporal-spatial topology computation. Suggestions for further research include: (1) to develop more generalized and complete topological relation computation algorithms for complex objects; (2) to infer the minimal number of operators and meaningful topological relations that exist in certain subspaces; (3) to develop efficient parallel algorithms and integrate a lattice-theoretic approach for large-scale topological relation computing and updating; and (4) to study the dynamically adaptive topological relation computation based on the symbolized CGA representation.

7 Conclusions

In this article, an extended CGA data model and unified topological computation model are proposed. We defined the unified *MeetOp*, *BoundOp* and *TopoOp* to extract the intersection, inside/outside relations and other topological relations, respectively in a united way. Case studies proved that our method is not only geometrically meaningful, computationally direct, and unified at different dimensions, but can also compute the topological relations in a simple and compact way. The advantages of our model include: (1) linking the geometric properties with the abstract algebraic relation computation; (2) providing a set of GA operators for topological representation and computation; (3) supporting unified computation for objects of different dimensions and types. All the above advantages suggest that our method provides a new foundation for efficient and geometrically meaningful topological relation computation.

Note

[1] In CGA, point pair can be used to express a line segment, but it can also represent two points that are linked with certain relations or an intersection between two circles (Dorst et al. 2007).

References

Agarwal P 2005 Topological and geometric operators for ontological classification of vague geographic concepts. *Spatial Cognition and Computation* 5: 69–88

Agugiaro G and Kolbe T H 2012 A deterministic method to integrate triangular meshes of different resolution. *ISPRS Journal of Photogrammetry and Remote Sensing* 71: 96–109

Bhatt M, Guesgen H, Wölfl S, and Hazarika S 2011 Qualitative spatial and temporal reasoning: Emerging applications, trends, and directions. *Spatial Cognition and Computation* 11: 1–14

Chen S, Tan J, Claramunt C, and Ray C 2011 Multi-scale and multi-modal GIS-T data model. *Journal of Transport Geography* 19: 147–61

Cheng C, Niu F, Cai J, and Zhu Y 2008 Extensions of GAP-tree and its implementation based on a non-topological data model. *International Journal of Geographical Information Science* 22: 657–73

Clark O G, Kok R, and Champigny P 1997 Generation of a virtual terrain. *Environmental Modelling Software* 12: 143–9

Clementini E and Di Felice P 1996 A model for representing topological relations between complex geometric features in spatial databases. *Information Sciences* 90: 121–36

Clementini E, Sharma J, and Egenhofer M J 1994 Modelling topological spatial relations: Strategies for query processing. *Computers and Graphics* 18: 815–22

De Bie H 2012 Clifford algebras, Fourier transforms, and quantum mechanics. *Mathematical Methods in the Applied Sciences* 35: 2198–228

Dorst L and Mann S 2002 Geometric algebra: A computational framework for geometrical applications. *IEEE Computer Graphics and Applications* 22: 24–31

Du S, Guo L, Wang Q, and Qin Q 2008 Efficiently computing and deriving topological relation matrices between complex regions with broad boundaries *ISPRS Journal of Photogrammetry and Remote Sensing* 63: 593–609

Dube M P and Egenhofer M J 2012 An ordering of convex topological relations. In Xiao N, Kwan M-P, Goodchild M F, and Shekhar S (eds) *Geographic Information Science*. Berlin, Springer: 72–86

Duckham M, Nussbaum D, Sack J R, and Santoro N 2011 Efficient, decentralized computation of the topology of spatial regions. *IEEE Transactions in Computing* 60: 1100–13

Egenhofer M J and Mark D M 1995 Modelling conceptual neighbourhoods of topological line-region relations. *International Journal of Geographical Information Systems* 9: 555–65

Egenhofer M J and Shariff A R B M 1998 Metric details for natural-language spatial relations. *ACM Transactions in Information Systems* 16: 295–321

Formica A, Pourabbas E, and Rafanelli M 2013 Constraints relaxation of the polygon-polyline topological relation for geographic pictorial query languages. *Computer Science and Information Systems* 10: 1053–75

© 2015 John Wiley & Sons Ltd

Gröger G and Plümer L 2011 Topology of surfaces modelling bridges and tunnels in 3D-GIS. *Computers, Environment and Urban Systems* 35: 208–16

Gröger G and Plümer L 2012 Provably correct and complete transaction rules for updating 3D city models. *Geoinformatica* 16: 131–64

Hestenes D 2003 Oersted Medal Lecture 2002: Reforming the mathematical language of physics. *American Journal of Physics* 71: 104–14

Hildenbrand D 2013 *Foundations of Geometric Algebra Computing*. Berlin, Springer-Verlag

Hitzer E 2005 Conic sections and meet intersections in geometric algebra. In Li H, Olver P J, and Sommer G (eds) *Computer Algebra and Geometric Algebra with Applications*. Berlin, Springer-Verlag Lecture Notes in Computer Science Vol. 3519: 350–62

Hitzer E, Nitta T, and Kuroe Y 2013 Applications of Clifford's geometric algebra. *Advances in Applied Clifford Algebras* 23: 377–404

Hu Y, Ravada S, Anderson R, and Bamba B 2013 Supporting topological relations queries for complex line and collection geometries in Oracle Spatial. In *Proceedings of the Twenty-first ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Orlando, Florida: 94–103

Klippel A 2012 Spatial information theory meets spatial thinking: Is topology the Rosetta Stone of spatio-temporal cognition? *Annals of the Association of American Geographers* 102: 1310–28

Kumar N P 2013 A short note on the theory of perspective topology in GIS. *Annals of GIS* 19: 123–8

Kumar N P 2014 On the topological situations in geographic spaces. *Annals of GIS* 20: 131–7

Ledoux H and Meijers M 2011 Topologically consistent 3D city models obtained by extrusion. *International Journal of Geographical Information Science* 25: 557–74

Lee J and Kwan M P 2005 A combinatorial data model for representing topological relations among 3D geographical features in micro-spatial environments *International Journal of Geographical Information Science* 19: 1039–56

Li J, Ouyang J, and Liao M 2013 Representation and reasoning for topological relations between a region with broad boundaries and a simple region on the basis of RCC-8. *Journal of Computational Information Systems* 9: 5395–402

Ligozat G and Condotta J-F 2005 On the relevance of conceptual spaces for spatial and temporal reasoning. *Spatial Cognition and Computation* 5: 1–27

Mark D M and Egenhofer M J 1994 Modeling spatial relations between lines and regions combining formal mathematical models and human subjects testing. *Cartography and Geographic Information Systems* 21: 195–212

Nedas K A, Egenhofer M J, and Wilmsen D 2007 Metric details of topological line–line relations. *International Journal of Geographical Information Science* 21: 21–48

Parra Serra J M 2009 Clifford algebra and the didactics of mathematics. *Advances in Applied Clifford Algebras* 19: 819–84

Peachavanish R, Karimi H A, Akinci B, and Boukamp F 2006 An ontological engineering approach for integrating CAD and GIS in support of infrastructure management. *Advanced Engineering Informatics* 20: 71–88

Penninga F and Van Oosterom P J M 2008 A simplicial complex-based DBMS approach to 3D topographic data modelling. *International Journal of Geographical Information Science* 22: 751–79

Roa E, Theoktisto V, Fairén M, and Navazo I 2011 GPU collision detection in conformal geometric space. In *Proceedings of the Ibero-American Symposium in Computer Graphics*, Faro, Portugal: 153–7

Robles-Ortega M D, Ortega L, and Feito F R 2012 Design of topologically structured geo-database for interactive navigation and exploration in 3D web-based urban information systems. *Journal of Environmental Informatics* 19: 79–92

Shariff A R B, Egenhofer M J, and Mark D M 1998 Natural-language spatial relations between linear and areal objects: The topology and metric of English-language terms. *International Journal of Geographical Information Science* 12: 215–45

Shiode N and Shiode S 2011 Street-level spatial interpolation using network-based IDW and ordinary kriging. *Transactions in GIS* 15: 457–77

Siejka M, Slusarski M, and Zygmunt M 2014 Verification technology for topological errors in official databases with case study in Poland. *Survey Review* 46: 50–7

Stoter J, Visser T, van Oosterom P, Quak W, and Bakker N 2011 A semantic-rich multi-scale information model for topography. *International Journal of Geographical Information Science* 25: 739–63

Theobald D M 2001 Topology revisited: representing spatial relations. *International Journal of Geographical Information Science* 15: 689–705

Wang J, Cui C, Rui Y, Cheng L, Pu Y, Wu W, and Yuan Z 2014 A parallel algorithm for constructing Voronoi diagrams based on point-set adaptive grouping. *Concurrency and Computation: Practice and Experience* 26: 434–46

Wu L 2004 Topological relations embodied in a generalized tri-prism (GTP) model for a 3D geoscience modeling system. *Computers and Geosciences* 30: 405–18

Yuan L, Lü G, Luo W, Yu Z, Yi L and Sheng Y 2012 Geometric algebra method for multidimensionally-unified GIS computation. *Chinese Science Bulletin* 57: 802–11

Yuan L, Yu Z, Chen S, Luo W, Wang Y, and Lü G 2010 CAUSTA: Clifford algebra-based unified spatio-temporal analysis. *Transactions in GIS* 14: 59–83

Yuan L, Yu Z, Luo W, Yi L, and Lü G 2013 Geometric algebra for multidimension-unified geographical information system. *Advances in Applied Clifford Algebras* 23: 497–518

Yuan L, Yu Z, Luo W, Yi L, and Lu G 2014 Multidimensional-unified topological relations computation: A hierarchical geometric algebra-based approach. *International Journal of Geographical Information Science* 28: 2435–55

Yuan L, Yu Z, Luo W, Zhou L, and Lü G 2011 A 3D GIS spatial data model based on conformal geometric algebra. *Science China Earth Sciences* 54: 101–12

Zawadzki T, Nikiel S, and Warszawski K 2012 Hybrid of shape grammar and morphing for procedural modeling of 3D caves *Transactions in GIS* 16: 619–33

Zhou X G, Chen J, Zhan F B, Li ZL, Madden M, Zhao R L, and Liu W Z 2013 A Euler number-based topological computation model for land parcel database updating *International Journal of Geographical Information Science* 27: 1983–2005

© 2015 John Wiley & Sons Ltd