Determining Topological Relationships Between 3D Legal Components using Conformal Geometric Algebra

Conference Paper · May 2020

2 authors:


Dillon Pullano
The University of Calgary


Michael Barry
The University of Calgary

Determining Topological Relationships Between 3D Legal Components using Conformal Geometric Algebra

Dillon PULLANO and Michael BARRY, Canada

Key words: 3-D Cadastre, CGA, Topology

SUMMARY

This paper reports on research that applies conformal geometric algebra (CGA) theory, objects and operations along with simple point-point distance checks to build a computational model for performing topological relationship analysis between two 3-D legal boundary components. The 3-D legal boundary components modelled are boundary points with euclidean [x, y, z] coordinates, straight boundary lines defined by a start and end point, and flat boundary planes closed and bound by at least three lines. Each boundary component is supplemented with a CGA projected object representation.

The topological relationships that this research identifies are if two 3-D boundary components touch, overlap (intersect), or are disjoint from each other. Relationships between boundary component pairs are initially sorted into 1 of 15 categories using the concepts of projected CGA object representations being parallel, collinear, coplanar, or intersecting at a point or line. CGA object intersections are generated and interpreted to determine which category of relationship is occurring. This reduces the complexity of the remaining problem in many cases to calculating various 3-D point-point distance checks and applying conditions to them.

In theory, the model can identify and classify 53 topological relationships that exist between 3-D legal boundary components being modelled in the context of a digital 3-D cadastre. Fifteen classifications describe disjoint relationships while 38 classifications describe various touch and overlap relationships that produce common point, line, or plane geometries. The model was implemented and tested against simulated datasets consisting of various boundary point, line, and plane component pairs where the topological classifications between simulated boundary components were known before testing. The results support that the model can correctly identify the sample of topological relationships presented for the datasets used.

Determining Topological Relationships Between 3D Legal Components using Conformal Geometric Algebra

Dillon PULLANO and Michael BARRY, Canada

1. INTRODUCTION

The paper reports on research that applies conformal geometric algebra (CGA) theory, objects and operations along with simple distance checks to build a computational model for performing topological relationship analysis between 3-D legal boundary components. The literature suggests that the theory and methodology have not yet been applied to the topological classification of 3-D boundaries in the way we apply it here. In theory, the model can identify and classify 53 topological relationships that exist between 3-D legal boundary components being modelled in the context of a digital 3-D cadastre.

1.1 Problem Context

As urban centers continue to grow and develop, there is an increasing need for institutions to be able to digitally model and perform legal boundary analysis on 3-D geospatial data. There is an increased need for mapping and managing complex legal structures that exist above and below the ground such as buildings, underground facilities, and utilities. In many jurisdictions, 3-D property can be registered to a title with a supplementary condominium or strata subdivision survey plan. Cadastral boundaries are recorded and registered as 2D drawings, vertical profiles, and 3-D isometric drawings. An example layout describing this can be seen in Figure 1. Boundary analysis can be performed by visual inspection and calculation of the survey plan drawings, but this often requires professional expertise such as that of a land surveyor. In more complex layouts, investigating these plans becomes more difficult and can be time consuming.

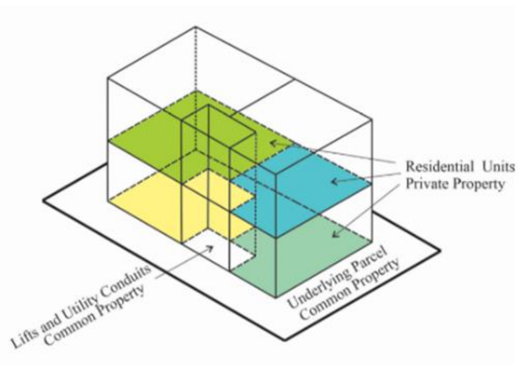

Figure 1: Common Strata Schema (Created by Barry, M. 2015)

1.2 Objectives

This research designed and tested a data model that can be used to perform topological boundary analysis between 3-D cadastral boundary components. Formal mathematical processes using CGA data types and operations, as well as 3-D point-point distance checks are proposed, implemented, and tested with the objective of classifying topological relationships that exist between 3-D cadastral object boundary components. The main objectives were as follows:

- Define 3-D data structures that are commonly used to represent 3-D property boundaries
- Identify topological relationships that can exist between these 3-D boundaries
- Propose theoretical computational procedures that use CGA object representations and operations, along with various point-point distance checks to identify and classify topological relationships existing between 3-D legal boundary components
- Test proposed computational procedures against simulated datasets that have known topological relationship classifications to show the extent that they can work under

1.3 Conformal Geometric Algebra Background

Conformal Geometric Algebra is a mathematical tool that can be used to model objects that exist in 3-D Euclidean space ($\mathbf{R}^3$) such as points, lines, and planes using concepts such as the outer product, geometric product, and vector subspaces (Dorst, 2007, p. 10). It has been proven to be useful for many applications in engineering and computer science. This research uses CGA objects and intersection operations to assist in classifying the topological relationship and identifying any common geometries that exist between two 3-D boundary components being modelled in the context of a 3-D cadastre. Details of CGA are discussed in section 2.2.

1.4 Scope and Application

This is the first part of a research project that tested and classified boundary relationships between a subset of various 3-D legal spaces that could be defined and registered by a strata survey plan. It tests the application of existing CGA theory to the problem context of classifying relationships and identifying common geometries between 3-D cadastral boundaries.

The experimental work here is limited to classifying relationships between 3-D points, lines, and planes (i.e. 3-D boundary components) only. We do not cover relationships between volumetric spaces. The discussion is limited to proposing and testing theoretical processes that can be used to classify relationships between individual 3-D boundary components which may define the boundary of a 3-D volumetric strata space (Figure 1). Results presented in section 4 show that relationships that can be classified between point-point, line-point, line-line, plane-point, plane-line, and plane-plane boundary component pairs. A sample of relationships between two 3-D boundary planes with geometries is shown in Figure 2 below.

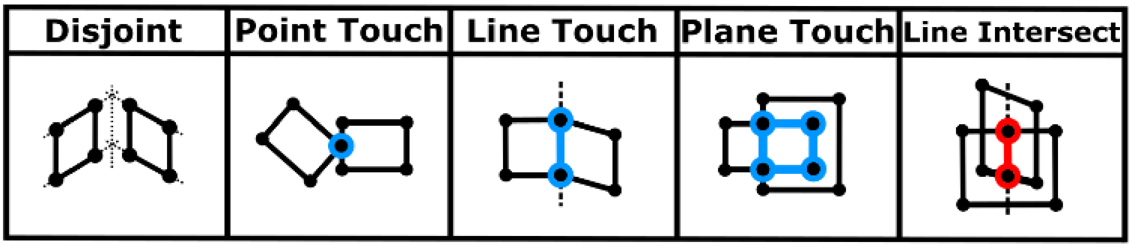

Figure 2: Sample of Topological Classifications Between Boundary Planes

2. Theory

Theory is presented in three sections. The first section presents existing theory related to topology frameworks that can be used to define and organize the relationships between objects in 3-D GIS. The second section covers mathematical theory related to the implementation of the CGA model. Main concepts and equations related to CGA space and objects of interest are presented here. The third section provides an overview of research that uses CGA theory to model and perform topological analysis on 3-D cadastral objects.

2.1 3-D Topology Frameworks and Characteristics

Ellel (2006) reviewed several research articles to identify a general list of common requirements for topology in 3-D GIS applications. Core standard analysis requirements for 3-D object relationships were grouped into terms of object adjacency, intersection, containment, and disconnectedness. Ellel (2006) notes that while topology frameworks need to be able to examine relationships between 3-D objects, to be comprehensive they also need to examine topological relationships between lower dimensional (0-D, 1-D, and 2-D) components that might be part of a 3-D object (i.e. points, lines, and planes). It is important that each topological element should have some form of geometrical representation so that analysis of results can be visualized.

2.1.1 The Dimensional Model Framework

The dimensional model (DM) is a topology framework introduced to provide insight on the spatial relationships that can exist between objects. The model is presented using the concepts of dimensional elements and dimensional relationships (Billen, 2002). Dimensional elements are organized as extensions and limits of dimensionally ordered boundary elements. Figure 3 shows common spatial elements that can exist in 3-D Euclidean $\mathbf{R}^3$ as a composition of their extension and boundary limit elements. Extensions are bolded while boundaries (limits) are not.

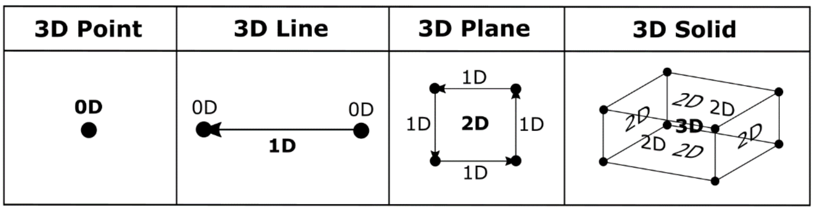

Figure 3: Dimensional Elements as Extensions and Limits - Modified from (Billen, 2002)

The DM classifies the relationship between two 3-D objects by combining the relationships that exist between the dimensional elements (0D, 1D, 2D, 3D) of both objects. It is suggested that elements and relationships could be computed using collinearity and coplanarity algorithms.

2.1.2 Approaches to Topological Relationship Definition

Fu (2018) organized topology between 3-D spatial objects using disjoint, touch, overlap, contain, or equal relationships (see Figure 4). This paper considers overlap, contain, and equal relationships as the same type and can be differentiated using their return geometries.

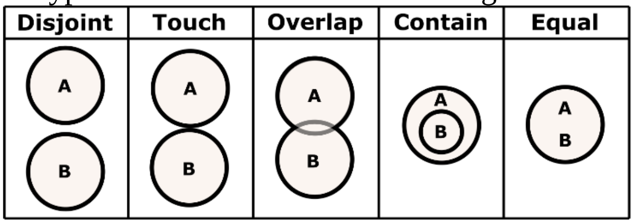

Figure 4: Classes of Topological Relationships (Derived from Fu, 2018)

Fu (2018) identified 30 types of disjoint and 30 types of touch relationships that can theoretically exist between 3-D cadastral objects using the concepts of collinear and coplanar lines and planes. For example, in the physical world a building may contain several strata units that share the same wall. In this scenario, these boundary planes between units should be coplanar. The majority, however, will not share a boundary. This research follows a similar approach in that relationships between objects are organized using collinearity and coplanarity.

2.2 Geometric Algebra and the Conformal Model

Geometric algebra combines the inner product and outer product to create the geometric product (see Equations 1, 2, & 3). The inner product can be applied to vectors to calculate metric information such as distances or angles (i.e. scalar values). The outer product can be used to span two vectors to create higher dimensional objects such as projected lines and projected planes (see Figure 6). The geometric product can be used to apply orthogonal transformations and rotations on objects. Its product produces a multivector that represents the symmetric and antisymmetric components between 2 vectors, **a** and **b** (Dorst, 2007, p. 143).

Inner Product: $$\boldsymbol{a} \cdot \boldsymbol{b} = \frac{1}{2}(\boldsymbol{ab} + \boldsymbol{ba}) \quad [1]$$

Outer Product: $$\boldsymbol{a} \wedge \boldsymbol{b} = \frac{1}{2}(\boldsymbol{ab} - \boldsymbol{ba}) \quad [2]$$

Geometric Product: $$\boldsymbol{ab} = \boldsymbol{a} \cdot \boldsymbol{b} + \boldsymbol{a} \wedge \boldsymbol{b} \quad [3]$$

Multivectors consist of blade parameters that Hildenbrand (2015) defines as the basic algebraic element of GA. Dorst (2007, p. 44) defines them as k-blades that can be used to represent a k-dimensional homogeneous subspace in a vector space model. The value k is referred to as the grade of the k-blade and represents the dimensionality of the k-blade subspace. The maximum value for k is the dimension of the vector space being worked in.

Vector Subspace Blade Parameters of CGA (R4,1)					
Scalar (0-blades)	Vector (1-blades)	Bivector (2-blades)	Trivector (3-blades)	Quadvector (4-blades)	Pseudoscalar (5-blade)
1	no	no^e1	no^e1^e2	no^e1^e2^e3	no^e1^e2^e3^ni
	e1	no^e2	no^e1^e3	no^e1^e2^ni	
	e2	no^e3	no^e1^ni	no^e1^e3^ni	
	e3	no^ni	no^e2^e3	no^e2^e3^ni	
	ni	e1^e2	no^e2^ni	e1^e2^e3^ni	
		e1^e3	no^e3^ni		
		e1^ni	e1^e2^e3		
		e2^e3	e1^e2^ni		
		e2^ni	e1^e3^ni		
		e3^ni	e2^e3^ni		

Figure 5: 32 Subspace Blade Parameters of CGA R4,1

The conformal model of geometric algebra embeds 3-D Euclidean (x, y, z) space $\mathbf{R^3}$ into an orthogonal 5-D conformal vector space (no, e1, e2, e3, ni) using an $\mathbf{R^{4,1}}$ metric. Note that the two added orthogonal dimensions are the point at the origin (no) and the point at infinity (ni) (Dorst, 2007, p. 9). The 32 subspace blades of this model can be seen in Figure 5.

A variety of geometric objects can be represented using k-blade multivectors. Yu (2016) makes a distinction between CGA objects as being rounds, flats, Euclidean blades, free blades, and tangent blades. Rounds can have finite length, area, or volume while flats include an (ni) component and can stretch outwards to infinity. Flat objects will be referred to as 'projected objects' for the rest of this paper. The CGA elements of interest are null vectors (points), as well as projected lines and projected planes (see Figure 6).

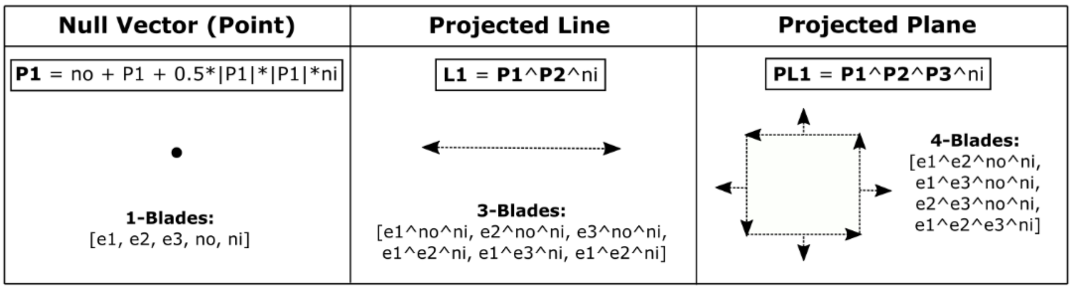

Figure 6:CGA Objects of Interest (with Blade Parameters)

Null Vector (Point): $$\boldsymbol{P} = n_0 + P + \frac{1}{2}|P|^2 * n_i \quad [4]$$

$$\rightarrow \quad \boldsymbol{P} = n_0 + (x * e_1) + (y * e_2) + (z * e_3) + (\frac{1}{2}|P|^2 * n_i)$$

Equation 4 can be used to embed a 3-D point (P = [x, y, z]) of $\mathbf{R^3}$ into the 5-D conformal space $\mathbf{R^{4,1}}$ as a null vector having 5 1-blade vector parameters.

Projected Line: $$\boldsymbol{L} = \boldsymbol{P_1} \wedge \boldsymbol{P_2} \wedge n_i \quad [5]$$

Equation 5 can be used to generate a projected line **L** that connects 2 null vector points. Using the outer product of these points with the point at infinity allows this line to continue through the space in both directions indefinitely. Hildenbrand (2015) conceptually explains these lines as circles with an infinite radius. There are six 3-blade parameters that are used to represent a projected line **L** in $\mathbf{R}^{4,1}$. Knowing these parameters allows CGA intersections to be interpreted into the correct geometry type in section 3.2.

Projected Plane: $$\boldsymbol{PL} = \boldsymbol{P_1} \wedge \boldsymbol{P_2} \wedge \boldsymbol{P_3} \wedge n_i \qquad [6]$$

Like Equation 5, Equation 6 uses the outer product to generate a projected plane **PL** that divides space into 2 half spaces. The plane can be defined by any 3 null vector points that lie on its surface and represents the projection of a flat surface outwards to infinity. Like Hildenbrand's explanation for projected lines, projected planes can be conceptualized as spheres with an infinite radius. There are four 4-blade parameters that can be used to represent a projected plane **PL** in $\mathbf{R}^{4,1}$. These allow for intersections involving projected planes to be performed.

Line & Point Distance	$\boldsymbol{D_{L\text{-}P}} = norm(\boldsymbol{L} \cdot \boldsymbol{P})$		[7]
Dualization:	$\mathbf{M}^* = \mathbf{M} \rfloor I_n^{-1}$	(Dorst, 2007, p. 80)	[8]
Intersect/Meet:	$\mathbf{Meet(A, B)} = (\mathbf{B} \rfloor \mathbf{I}_n^{-1}) \rfloor \mathbf{A}$ $\rightarrow = \mathbf{B}^* \rfloor \mathbf{A}$	(Dorst, 2007, p. 131)	[9]

The last few equations in this section can be used to derive topological information between CGA objects. Equation 7 is used to calculate the distance $\mathbf{D_{L\text{-}P}}$ between a projected line **L** and a null vector point **P**. The norm of a multivector is defined as the magnitude of its 0-blade scalar components. Equations 8 and 9 use the left contraction (⌋) which is a generalization of the inner product for multivectors. The dual **M*** of a multivector **M** (Equation 8) can be thought of as the orthogonal component of a multivector with respect to the space it resides in (Dorst, 2007). This concept is used to generate planeline-point distances in section 3.2. Equation 9 is used as a general intersection operation between multivectors **A** and **B**. The meet operation is applied to projected CGA boundary component representations in section 3.2 to initially categorize the type of relationship that is occurring between 2 3-D boundary components by interpreting the return geometry type that the meet produces.

2.3 Modelling Geometry and Topology using Conformal Geometric Algebra

2.3.1 Implementations using the CAUSTA Environment

There have been multiple studies in recent years that use CGA to model and perform analysis on cadastral objects using the CAUSTA geometric algebra environment introduced by Yuan (2010). Yuan (2011) created a 3-D GIS spatial data model in CAUSTA that was used to represent volumetric objects and their boundaries in CGA space. They show how distances and angles between objects can be derived and how 3-D topological analysis is performed on individual volumetric objects by intersecting its planes and lines to analyze structure. Yuan (2012) then shows how multidimensional objects can be organized in the same space with CGA. Yu (2016) created a framework to compute geometry oriented topological relations between 3-D objects being rendered using this model. They used the concepts of an object being inside, on, or outside another object to create a general topology operator for determining 8 relationships between multidimensional round and flat objects. Results were presented between sets of disjoint 3-D objects in the CAUSTA environment. Zhang, J. (2016) proposed a 3-D cadastral data model based on CGA. In this model, a 3-D parcel is stored as a multivector consisting of a summation of its points, lines and planes. Boundary lines and planes are represented as a combination of the objects projected (flat) component and the set of lines/points that define its boundary.

2.3.2 Rule Based Relationship Classification

Zhang, F. (2016) used the intersection (meet) results of CGA projected lines and planes along with various other CGA operation checks to create specific judgement rules for testing if 2 3-D boundary components intersect or not. The boundary objects considered were 3-D points, straight line segments, and closed polygons.
Six sets of judgement rules were determined for the point-point, line-point, polygon-point, line-line, polygon-line, and polygon-polygon boundary object pairs. By first performing intersections between CGA projected objects, the complexity of the 3-D object intersection problem was reduced to several smaller problems. These problems were approached differently based on the return geometry type of the projected CGA intersections. Their model was able to distinguish between 22 different types of intersections between coplanar boundary components.

This research uses methods proposed by Zhang, F. (2016) to categorize and sort boundary component pair relationships into several smaller problem groups which will all be approached differently. Figure 9 presents CGA object geometries that can exist for the intersection result between point, projected line, and projected plane boundary component pairs in 3-D space.

3. Methods

The methodology is presented in four sections. Section 3.1 defines the geometric data structures for the 3-D strata boundary components considered. Point set topology is presented here for each 3-D boundary component with respect to the dimensional model (see section 2.1.1). The processes that were followed to generate and store 3-D boundary components with their projected CGA counterparts using point set operations are presented here as well. Section 3.2 categorizes different types of relationships that can exist between 3-D boundary component types using the concepts of projected CGA components being parallel, collinear, coplanar, or intersecting. CGA intersections are generated between projected boundary component pairs. Return geometry types are interpreted to identify which CGA scenario is occurring. Using these return geometry interpretations, more refined relationships are derived by determining the extent that the point set representations of each object overlap the projected CGA object intersection geometry. Section 3.3 presents the computational testing processes that are followed to classify the topological relationship type and to collect the return geometries between two boundary components if any exists. Processes for the point-point, line-point, line-line, plane-point, plane-line, and plane-plane boundary component pairs are derived.

3.1 3-D Strata Boundary Component Definitions

The 3-D legal boundary components modelled in this paper (Figure 7) are boundary points (BP) with Euclidean [x, y, z] coordinates, straight boundary lines (BL) defined by a start and end point, and flat boundary planes (BPL) closed and bound by at least three lines. Each boundary component type is defined with respect to their point set topology having extensions and limits. Lines and planes have an additional projected CGA representation (Figure 6) that is used in section 3.2 to initially sort topological relationships into different categories.

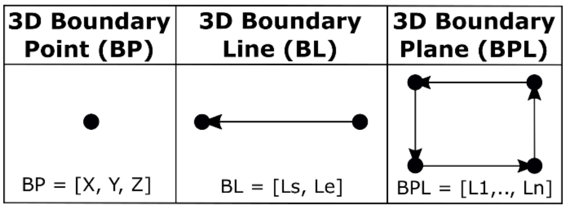

Figure 7: 3-D Boundary Component Object Types

3.1.1 Cadastral Boundary Component Point Sets

Most boundary points in a 3-D survey plan are often defined relative to governing boundary markers nearby that localize the survey site into a coordinate network. Therefore, when generating digital versions of 3-D boundaries, point location estimates need to be calculated using metric information that is included in the survey plan drawings. Referring to section 2.2.1 and Figure 3, boundary points have a 0D extension with no limits.

The information available for straight boundary lines in a survey plan are their drawing representations and the included angles and distances between points. Therefore, once estimated coordinates have been calculated for vertex points, boundary lines can be represented by the coordinates of the start and end points that bound them. Each boundary line has a 1D extension with two 0D boundary limit points. Each boundary line is situated on a projected CGA line that has a 1D extension through space.

Boundary planes are defined by a set of coplanar, straight lines that start and end at the same point. There must be at least 3 lines defining a plane and they must follow each other. Boundary planes do not have to be convex (i.e. interior angles between lines can be more than 180 degrees). Every boundary plane has a 2D extension with 1D line limits and 0D point limits. Each is situated on a projected CGA plane that has a 2D extension through space.

3.1.2 Generating Digital Boundary Objects

The following processes were used to generate digital representations for the 3-D boundary components defined above. These processes were followed to create the simulated datasets presented in the experimentation section of this study. They can be explained in three steps.

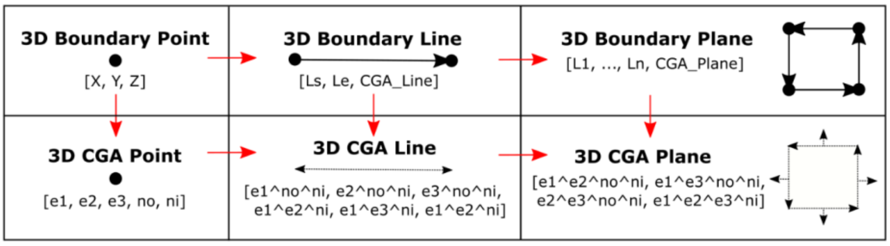

Figure 8: Digital 3-D Boundary Component Object Creation Process Flow

The first step is to calculate the locations of all 3-D boundary points defining a volumetric object registered by a strata survey plan. This can be done through direct (polar) calculations using known control point locations, recorded distances, and recorded bearing measurements and is common practice in surveying operations.

The second step is to define 3-D boundary line point set topology order by identifying all boundary plane surfaces that bound and close the volumetric object. This order enforces the concept of clockwise rotation when looking at the plane from inside the object. This will result in a list of lines that are bounded and closed by a topologically ordered start and end point, as well as a set of planes that are bounded and closed by a series of lines.
The third step converts all 3-D Euclidean boundary points into their respective 5D CGA representations by adding an 'n_o' and 'n_i' coordinate derived through their x, y, and z coordinate values using Equation 4. This conversion process was implemented in MATLAB. CGA representations for projected lines can then be created by using boundary line start and end points and Equation 5. CGA representations for projected planes can then be created by using any 3 topologically ordered boundary points defining each plane and Equation 6.

All CGA operations were performed using an open source geometric algebra program GAViewer (Dorst, 2007). 3-D boundary components were stored in EXCEL tables using point set orders. These point sets for boundary component pairs were loaded into MATLAB, where projected CGA object creation operations (Equations 4-6) and CGA object intersections (Equations 7-9) were written to a file. This file was loaded into GAViewer for implementation. Results were exported as a text file and loaded back into MATLAB to update the blade parameters for each boundary component and to store intersection results between CGA objects. Final interpretations were performed using code the first author wrote in MATLAB.

3.2 Topological Relationship Classification Processes

The topological relationships that this research identifies are if two 3-D boundary components 'Touch', 'Overlap/Intersect', or are 'Disjoint' from each other (see Figure 4). Like the approach of Zhang, F. (2016), relationships between boundary components are initially categorized using the concepts of projected CGA objects being parallel, collinear, or coplanar (see Figure 9). Category 1 (Disjoint or equal points) is determined through a 3-D Euclidean point-point distance check. Categories 2-15 are determined by interpreting projected CGA boundary component intersections. Each example represents a disjoint relationship for each category. The abbreviations BP, BL, and BPL are explained in Figure 7 above.

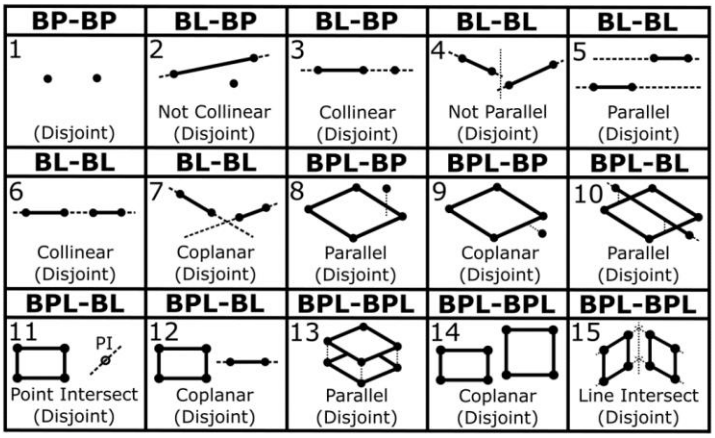

Figure 9: 3-D Boundary Component Relationship Categories (Disjoint Examples)

CGA intersection operators are used to determine which category of relationship is occurring. The intersections between the dimensional elements of each boundary component were generated in GAViewer for each boundary component pair and exported as a text file. These intersections were loaded into MATLAB and categorized by what type of CGA object they produced. Table 1 shows how CGA object intersections were interpreted.

Table 1: CGA Object Intersections, Return Geometries, and Relationship Categories

CGA Object Intersections and Return Geometries							
Boundary Component Pair	**CGA Intersection operator**	**Not Parallel**	**Parallel**	**Collinear**	**Coplanar**	**Intersection Point**	**Intersection Line**
Line-Point	Equation 7	N/A	Scalar (+)	Scalar (0)	N/A	N/A	N/A
Plane-Point	Equation 9	N/A	Scalar (+/-)	N/A	Point	N/A	N/A
Line-Line	Equation 9	Free Scalar	Free Vector	Line	Flat Point	N/A	N/A
Plane-Line	Equation 9	N/A	Free Vector	N/A	Line	Flat Point	N/A
Plane-Plane	Equation 9	N/A	Free Bivector	N/A	Plane	N/A	Line

The approach for this research was to first classify the topological relationship between lower dimensional boundary component pairs that exist between two boundary objects (point-point, line-point, and line-line). These were then combined with additional checks to help derive relationships between higher dimensional pairs (plane-point, plane-line, and plane-plane).

Figures 10-15 describe the mathematical processes that were followed to classify relationships between 3-D boundary component pairs. 3-D boundary components were stored in EXCEL tables using point set orders. These point sets for boundary component pairs were loaded into MATLAB, where projected CGA object creation operators (Equations 4-6) and CGA object intersection operators (Equations 7-9) were written to a file. This was loaded into GAViewer for implementation. Results were exported as a text file and loaded back into MATLAB to update the blade parameters for each boundary component and to store the intersections. Final interpretations of CGA intersections and distance checks were done in MATLAB

[1] Calculate the point-point distance between 3-D boundary points **BP₁** = [x₁, y₁, z₁] and **BP₂** = [x₂, y₂, z₂] using:

BP1→BP2 = $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$

[2-a] If **BP1→BP2** = (0) → Equal points → 0D-0D PT

[2-b] If **BP1→BP2** = (+) → Disjoint

Figure 10: BP-BP Topological Relationship Classification Processes

[1] Calculate the line-point distance between **BL**'s projected line **L** and **BP**'s null vector point **P** using: $\boldsymbol{D_{L-P}} = norm(\boldsymbol{L} \cdot \boldsymbol{P})$

[2-a] If $\boldsymbol{D_{L-P}} = (+)$ → Parallel → Disjoint

[2-b] If $\boldsymbol{D_{L-P}} = (0)$ → Collinear

→ Calculate point-point distances: [**Ls→Le**, **Ls→P**, **Le→P**]

If (**Ls→P** || **Le→P**) = 0 → 0D-0D PT

If (**Ls→P** & **Le→P**) < **Ls→Le** → 1D-0D PT

Figure 11: BL-BP Topological Relationship Classification Processes

Figures 10 and 11 show processes to determine the relationship between two 3-D boundary points (BP-BP) and between a boundary line and boundary point (BL-BP), respectively. This follows the same approach as Zhang, F. (2016) but uses 3-D Euclidean point checks instead of CGA null vector point intersections for the final relationship classification. It should be noted that in Figure 12 there are multiple component pair configurations resulting in the '1D-1D LT' classification between 2 collinear lines. These configurations represent varying amounts of overlap between lines but are treated as the same relationship for this paper. In future research, these could be further interpreted to derive more detailed overlap relationships.

[1] Calculate the meet intersection between projected lines $\mathbf{L_1}$ and $\mathbf{L_2}$ using: $\mathbf{Meet(L_1, L_2)} = (\mathbf{L_2} \rfloor \mathbf{I_n}^{-1}) \rfloor \mathbf{L_1}$

[2-a] If $\mathbf{Meet(L_1, L_2)}$ = Free Scalar → Not Parallel → Disjoint

[2-b] If $\mathbf{Meet(L_1, L_2)}$ = Free Vector → Parallel → Disjoint

[2-c] If $\mathbf{Meet(L_1, L_2)}$ = Projected Line (+) → Collinear (Common Orientation)

→ Calculate point-point distances: [**L1s→L1e**, **L1s→L2s**, **L1s→L2e**, **L1e→L2s**, **L1e→L2e**, **L2s→L2e**]

If (**L1s→L2e** = Largest) || (**L1e→L2s** = Largest) → Disjoint
If (**L1s→L2e** = 0) || (**L1e→L2s** = 0) → 0D-0D PT
If (**L1s→L2e** = Largest) & (**L1s→L2s** < **L1s→L1e**) → 1D-1D LT
If (**L1e→L2s** = Largest) & (**L1s→L2s** < **L2s→L2e**) → 1D-1D LT
If (**L1s→L2s** = 0) & [(**L1s→L2e** & **L2s→L2e**) = Largest || (**L1e→L2s** & **L1s→L1e**) = Largest] → 1D-1D LT
If (**L1e→L2e** = 0) & [(**L1e→L2s** & **L2s→L2e**) = Largest || (**L1s→L2e** & **L1s→L1e**) = Largest] → 1D-1D LT
If (**L1s→L1e** = Largest) || (**L2s→L2e** = Largest) → 1D-1D LT
If (**L1s→L2s** = 0) & (**L1e→L2e** = 0) → 1D-1D LT

[2-d] If $\mathbf{Meet(L_1, L_2)}$ = Projected Line (-) → Collinear (Reverse Orientation)

→ Calculate point-point distances: [**L1s→L1e**, **L1s→L2s**, **L1s→L2e**, **L1e→L2s**, **L1e→L2e**, **L2s→L2e**]

If (**L1s→L2s** = Largest) & (**L1s→L2e** > **L1s→L1e**) → Disjoint
If (**L1e→L2e** = Largest) & (**L1s→L2e** > **L2s→L2e**) → Disjoint
If (**L1s→L2s** = 0) || (**L1e→L2e** = 0) → 0D-0D PT
If (**L1s→L2s** = Largest) & (**L1e→L2s** < **L2s→L2e**) → 1D-1D LT
If (**L1e→L2e** = Largest) & (**L1s→L2s** < **L2s→L2e**) → 1D-1D LT
If (**L1s→L2e** = 0) & [(**L1s→L2s** & **L2s→L2e**) = Largest || (**L1e→L2e** & **L1s→L1e**) = Largest] → 1D-1D LT
If (**L1e→L2s** = 0) & [(**L1e→L2e** & **L2s→L2e**) = Largest || (**L1s→L2s** & **L1s→L1e**) = Largest] → 1D-1D LT
If (**L1s→L1e** = Largest) || (**L2s→L2e** = Largest) → 1D-1D LT
If (**L1s→L2e** = 0) & (**L1e→L2s** = 0) → 1D-1D LT

[2-e] If $\mathbf{Meet(L_1, L_2)}$ = Flat Intersection Point (IP) → Coplanar

→ Calculate collinear line-point relationships for:

$\mathbf{Set_1}$ = [**L1s→L1e**, **L1s→IP**, **L1e→IP**], $\mathbf{Set_2}$ = [**L2s→L2e**, **L2s→IP**, **L2e→IP**]

If ($\mathbf{Set_1}$ relationship = Disjoint) || ($\mathbf{Set_2}$ relationship = Disjoint) → Disjoint
If ($\mathbf{Set_1}$ relationship = 0D-0D PT) & ($\mathbf{Set_2}$ relationship = 0D-0D PT) → 0D-0D PT
If ($\mathbf{Set_1}$ relationship = 0D-0D PT) & ($\mathbf{Set_2}$ relationship = 1D-0D PT) → 0D-1D PT
If ($\mathbf{Set_1}$ relationship = 1D-0D PT) & ($\mathbf{Set_2}$ relationship = 0D-0D PT) → 1D-0D PT
If ($\mathbf{Set_1}$ relationship = 1D-0D PT) & ($\mathbf{Set_2}$ relationship = 1D-0D PT) → 1D-1D PI

Figure 12: BL-BL Topological Relationship Classification Processes

[1] Check topological relationship classification results between **BP** and all lines defining **BPL** ($\mathbf{BL_{1\text{-}n}}$)

If any result = 0D-0D PT → Carry relationship forwards and stop → 0D-0D PT
If any result = 1D-0D PT → Carry relationship forwards and stop → 1D-0D PT

[2] If no result, calculate meet between **BPL**'s projected plane **PL** and **BP**'s null vector **P** using: $\mathbf{Meet(PL, P)} = (\mathbf{PL} \rfloor \mathbf{I_n}^{-1}) \rfloor \mathbf{P}$

[3-a] If $\mathbf{Meet(PL, P)}$ = Scalar (+/-) → Parallel → Disjoint

[3-b] If $\mathbf{Meet(PL, P)}$ = Flat Intersection Point → Coplanar

→ Generate projected lines $\mathbf{ProjL_{P\rightarrow P1\text{-}n}}$ between **P** and all points defining **BPL** ($\mathbf{P_{1\text{-}n}}$) using: $L = P \wedge P_n \wedge n_i$

→ Perform line-line relationship classification for all $\mathbf{ProjL_{P\rightarrow P1\text{-}n}}$ against each line defining **BPL** ($\mathbf{BL_{1\text{-}n}}$)

If any classification results between $\mathbf{ProjL_{P\rightarrow Pn}}$ and $\mathbf{BL_{1\text{-}n}}$ is [1D-1D LT] or [1D-1D PI]

→ Flag **BPL** boundary point $\mathbf{BP_n}$
→ Flag any **BPL** boundary plane lines that use $\mathbf{BP_n}$ as an end point

→ Calculate distance between **P** and projected lines ($\mathbf{L_{1\text{-}n}}$) that are not flagged using: $\mathbf{D_{PL\text{-}L\text{-}P}} = -[((\mathbf{L}^*) \cdot (\mathbf{PL}^*))^* \,\&\, \mathbf{P}]$

If all $\mathbf{D_{PL\text{-}L\text{-}P}}$ = (+) → Disjoint
If all $\mathbf{D_{PL\text{-}L\text{-}P}}$ = (-) → 2D-0D PT

Figure 13: BPL-BP Topological Relationship Classification Processes

[1] Calculate the meet between **BPL**'s projected plane **PL** and **BL**'s projected line **L** using: $\mathbf{Meet(PL, L)} = (\mathbf{PL} \rfloor \mathbf{I_n}^{-1}) \rfloor \mathbf{L}$

[2-a] If **Meet(PL, L)** = Free Vector → Parallel → Disjoint

[2-b] If **Meet(PL, L)** = Projected Line → Coplanar

→ Collect topological relationship classification results between **BL** and all lines defining **BPL** ($\mathbf{BL_{1\text{-}n}}$)

If classification result has point geometry → Add point to list

If classification result has line geometry → Add start point and end point to list

→ Add **BLs** point and **BLe** point to list and order points

→ Remove duplicate points and calculate the distances between $\mathbf{BL_s}$ and all other points in list

→ Sort points into sequential order using their increasing distances away from $\mathbf{BL_s}$

→ Split **BL** into line segments $\mathbf{LS_{1\text{-}m}}$ with start **LSs** and end **LSe** points using sorted points as line breaks

→ Generate a mid point so that each $\mathbf{LS_{1\text{-}m}}$ is defined by [**LSs**, $\mathbf{LS_{mid}}$, **LSe**]

→ Perform plane-point relationship classifications $\mathbf{BPL\text{-}LS_s}$, $\mathbf{BPL\text{-}LS_{mid}}$, and $\mathbf{BPL\text{-}LS_e}$ between all $\mathbf{LS_{1\text{-}m}}$ and **BPL**

If $\mathbf{BPL\text{-}LS_{mid}}$ = Disjoint → 1D line extension = Disjoint

If $\mathbf{BPL\text{-}LS_{mid}}$ = 1D-0D PT → 1D line extension = 1D-1D LT

If $\mathbf{BPL\text{-}LS_{mid}}$ = 2D-0D PT → 1D line extension = 2D-1D LT

If ($\mathbf{BPL\text{-}LS_s}$ || $\mathbf{BPL\text{-}LS_e}$) = 0D-0D PT → 0D point boundary = 0D-0D PT

If ($\mathbf{BPL\text{-}LS_s}$ || $\mathbf{BPL\text{-}LS_e}$) = 1D-0D PT → 0D point boundary = 1D-0D PT

If ($\mathbf{BPL\text{-}LS_s}$ || $\mathbf{BPL\text{-}LS_e}$) = 2D-0D PT → 0D point boundary = 2D-0D PT

[2-c] If **Meet(PL, L)** = Flat Intersection Point → Coplanar

→ Calculate plane-point relationship classification **BPL-IP** between boundary plane **BPL** and intersection point **IP**

→ Calculate line-point relationship classification **BL-IP** between boundary line **BL** and intersection point **IP**

If (**BPL-IP** = Disjoint) || (**L-IP** = Disjoint) → Disjoint

If (**BPL-IP** = 0D-0D PT) & (**L-IP** = 0D-0D PT) → 0D-0D PT

If (**BPL-IP** = 0D-0D PT) & (**L-IP** = 1D-0D PT) → 0D-1D PT

If (**BPL-IP** = 1D-0D PT) & (**L-IP** = 0D-0D PT) → 1D-0D PT

If (**BPL-IP** = 1D-0D PT) & (**L-IP** = 1D-0D PT) → 1D-1D PT

If (**BPL-IP** = 2D-0D PT) & (**L-IP** = 0D-0D PT) → 2D-0D PT

If (**BPL-IP** = 2D-0D PT) & (**L-IP** = 1D-0D PT) → 2D-1D PI

Figure 14: BPL-BL Topological Relationship Classification Processes

[1] Calculate the meet between $\mathbf{BPL_1}$ and $\mathbf{BPL_2}$'s projected planes, $\mathbf{PL_1}$ and $\mathbf{PL_2}$ using: $\mathbf{Meet(PL_1, PL_2)} = (\mathbf{PL_2} \rfloor \mathbf{I_n}^{-1}) \rfloor \mathbf{PL_1}$

[2-a] If $\mathbf{Meet(PL_1, PL_2)}$ = Free Bivector → Parallel → Disjoint

[2-b] If $\mathbf{Meet(PL_1, PL_2)}$ = Projected Plane → Coplanar

→ Collect all non-disjoint topological classification results between $\mathbf{BPL_1}$ and all lines defining $\mathbf{BPL_2}$ and add to a list

→ Collect all non-disjoint topological classification results between $\mathbf{BPL_2}$ and all lines defining $\mathbf{BPL_1}$ and add to a list

→ Remove duplicate points and lines having the same geometry

→ Combine line segments having common start or end points to make closed polygons if any exist

If closed polygon geometry exists → 2D-2D FT

If 1D-1D LT exists → 1D-1D LT

If 0D-0D PT exists → 0D-0D PT

If 0D-1D / 1D-0D exists → 0D-1D / 1D-0D PT

[2-c] If $\mathbf{Meet(PL_1, PL_2)}$ = Projected Line → Intersection Line

→ Collect all topological classification results between $\mathbf{BPL_1}$ and all lines defining $\mathbf{BPL_2}$ and add points to list

→ Collect all topological classification results between $\mathbf{BPL_2}$ and all lines defining $\mathbf{BPL_1}$ and add points to list

→ Sort points into ordered list and create line segments $\mathbf{LS_{1\text{-}n}}$ where each is defined by [$\mathbf{LS_s}$, $\mathbf{LS_{mid}}$, $\mathbf{LS_e}$]

→ Perform plane-point relationship classifications $\mathbf{BPL_1\text{-}LS_s}$, $\mathbf{BPL_1\text{-}LS_{mid}}$, and $\mathbf{BPL_1\text{-}LS_e}$ between all $\mathbf{LS_{1\text{-}n}}$ and $\mathbf{BPL_1}$

→ Perform plane-point relationship classifications $\mathbf{BPL_2\text{-}LS_s}$, $\mathbf{BPL_2\text{-}LS_{mid}}$, and $\mathbf{BPL_2\text{-}LS_e}$ between all $\mathbf{LS_{1\text{-}n}}$ and $\mathbf{BPL_2}$

If ($\mathbf{BPL_1\text{-}LS_{mid}}$ = Disjoint) || ($\mathbf{BPL_2\text{-}LS_{mid}}$ = Disjoint) → 1D line extension = Disjoint

If ($\mathbf{BPL_1\text{-}LS_{mid}}$ = 1D-0D PT) & ($\mathbf{BPL_2\text{-}LS_{mid}}$ = 1D-0D PT) → 1D line extension = 1D-1D LT

If ($\mathbf{BPL_1\text{-}LS_{mid}}$ = 1D-0D PT) & ($\mathbf{BPL_2\text{-}LS_{mid}}$ = 2D-0D PT) → 1D line extension = 1D-2D LT

If ($\mathbf{BPL_1\text{-}LS_{mid}}$ = 2D-0D PT) & ($\mathbf{BPL_2\text{-}LS_{mid}}$ = 1D-0D PT) → 1D line extension = 2D-1D LT

If ($\mathbf{BPL_1\text{-}LS_{mid}}$ = 2D-0D PT) & ($\mathbf{BPL_2\text{-}LS_{mid}}$ = 2D-0D PT) → 1D line extension = 2D-2D LI

If ($\mathbf{BPL_1\text{-}LS_s}$ || $\mathbf{BPL_1\text{-}LS_e}$ || $\mathbf{BPL_2\text{-}LS_s}$ || $\mathbf{BPL_2\text{-}LS_e}$) = 0D-0D/1D-1D PT → 0D point boundary = 0D-0D/1D-1D PT

If ($\mathbf{BPL_1\text{-}LS_s}$ || $\mathbf{BPL_1\text{-}LS_e}$ || $\mathbf{BPL_2\text{-}LS_s}$ || $\mathbf{BPL_2\text{-}LS_e}$) = 1D-0D/0D-1D PT → 0D point boundary = 1D-0D/0D-1D PT

If ($\mathbf{BPL_1\text{-}LS_s}$ || $\mathbf{BPL_1\text{-}LS_e}$ || $\mathbf{BPL_2\text{-}LS_s}$ || $\mathbf{BPL_2\text{-}LS_e}$) = 2D-0D/0D-2D PT → 0D point boundary = 2D-0D/0D-2D PT

Figure 15: BPL-BPL Topological Relationship Classification Processes

4. Results and Discussion

The 3-D simulated datasets and topological classification testing results between boundary components are limited to those between point-point (BP-BP), line-point (BL-BP), line-line (BL-BL), plane-point (BPL-BP), plane-line (BPL-BL), and plane-plane (BPL-BPL) boundary component pairs. Datasets were generated using point geometry and processes described in section 3.1.2. The topological relationship classifications and common return geometries between each simulated component pair were known *a priori* to testing.

Datasets for all results were initially created by inserting 3-D point coordinates, line point sets, and plane point sets into excel sheets. Additional multivector blade parameters for CGA null vector points, projected lines, and projected planes were then generated and added into these tables using Equations 4-6 and the processes described in section 3.1.2. Intersections between the two objects' CGA representations were then generated and stored to a text file using GAViewer. This intersection result was loaded into MATLAB where multivector blade parameters were interpreted using Table 1 to sort the current relationship into one of the 15 categories described in Figure 9. The additional steps followed to determine final relationship classifications between individual boundary component pairs were implemented in MATLAB and are described in Figures 10-15.

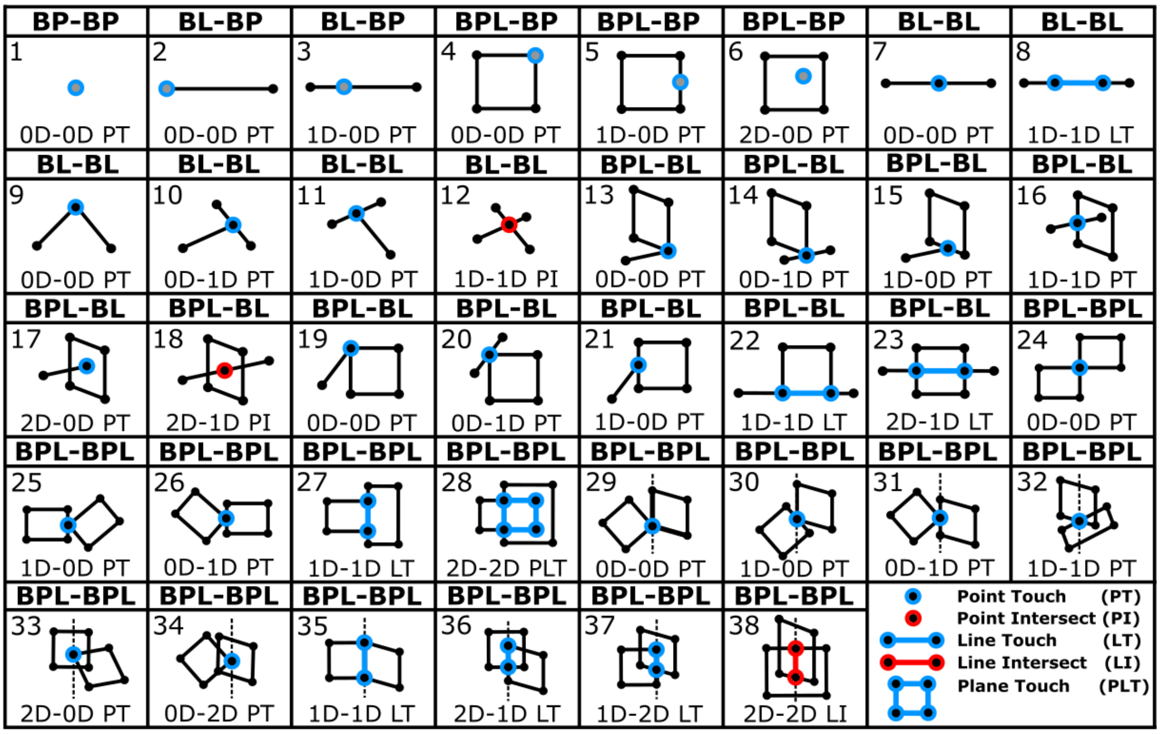

Figure 16: Experimental Topological Relationship Classifications Between 3-D Boundary Component Pairs

Figure 16 shows the touch and intersect results from different component pair datasets that were simulated in 38 different possible cases using the processes defined in section 3.1.2. Each square in Figure 16 has a case number. These generated figures are to be used as visual aids to present the simulated data inputs and output classifications.

Classification 1 (BP-BP) is the only relationship that is solved for without first computing projected CGA intersections and can be performed with a simple 3-D distance check (Figure 10). In the BL-BP and BL-BL cases (2-3 and 7-12 respectively), the complexity of the topology classification process can be reduced to point-point distance checks after being categorized as being parallel, collinear or coplanar using the CGA intersection result (see Figures 11 and 12).

Regarding the BPL-BP results, if it has not already been determined that a BP touches the edge of the BPL boundary at a line or point, an additional CGA check was introduced to determine if a coplanar point is outside or inside the boundary of a closed boundary plane. Projected lines were generated between the BP and all points defining BPL using Equation 5. By determining the intersection of each of these with all lines defining BPL, a list of 'fully visible' lines from BP can be generated. The planeline-point distances from these visible lines can then be used to determine if BP is inside or outside BF (see Figure 13).

Regarding the BPL-BL results, any line crossing the edge of the boundary or intersecting the interior of a boundary plane is split into line segments using the touch or intersect points between the two boundary objects as limits between each segment (see Figure 14). Each segment has a start point, end point, and generated middle point. By running these 3 points through the additional coplanar BPL-BP check, topological classifications can be returned for each of the end point limits and the line segment extension with respect to the boundary plane.

Regarding the BPL-BPL results, classifications are collected between plane 1 and all lines in plane 2, and vice versa (see Figure 15). For coplanar planes, non-disjoint point and line segment results were added to a list. Line segments were combined into closed polygons when possible using common start/end point coordinates (Figure 16, result 28). For projected planes that meet at an intersection line, all point results are sorted to create ordered line segments. The start, middle, and end points that define these line segments are classified with respect to both planes to reach final topology classifications.

Table 2: Topological Relationships Between 3-D Boundary Component Pairs

Topological Relationships Between 3-D Boundary Component Pairs									
Return Geometry	Relationship Type	Dimension	3-D Boundary Component Pair Type						
			BP-BP (2)	BL-BP (4)	BL-BL (10)	BF-BP (5)	BF-BL (14)	BF-BF (18)	Total (53)
--	Disjoint	--	1	2	4	2	3	3	15
Point	Touch	0D-0D (PT)	1	1	2	1	2	2	9
Point	Touch	0D-1D (PT)	--	--	1	--	2	2	5
Point	Touch	1D-0D (PT)	--	1	1	1	2	2	7
Line	Touch	1D-1D (LT)	--	--	1	--	1	2	4
Point	Intersect	1D-1D (PI)	--	--	1	--	--	--	1
Point	Touch	1D-1D (PT)	--	--	--	--	1	1	2
Point	Touch	2D-0D (PT)	--	--	--	1	1	1	3
Point	Intersect	2D-1D (PI)	--	--	--	--	1	--	1
Line	Touch	2D-1D (LT)	--	--	--	--	1	1	2
Point	Touch	0D-2D (PT)	--	--	--	--	--	1	1
Line	Touch	1D-2D (LT)	--	--	--	--	--	1	1
Line	Intersect	2D-2D (LI)	--	--	--	--	--	1	1
Face	Touch	2D-2D (FT)	--	--	--	--	--	1	1

Relationships are organized in Table 2 using the type of return geometry, topological classification, and dimension of the touch or intersect with respect to the dimensional model. There are 53 different types of topological relationship classifications that can be identified using the data structure and processes described in section 3 of this study. This includes the 15 disjoint relationships shown in Figure 9, as well as 38 other relationships that produce point, line, or plane geometries (Figure 16). There are 2 relationships between BP-BP, 4 between BL-BP, 10 between BL-BL, 5 between BF-BP, 14 between BF-BL, and 18 between BF-BF.

5. Conclusion

Being able to digitally model and perform topological analysis on 3-D legal object boundaries would increase the access to spatial information regarding current and future land and property developments and appraisals. This paper covers the first phase of a research project. Initial classification results from simulated datasets consisting of various boundary points, lines, and planes support that the mathematical testing procedures proposed in section 3 to classify topological relationships between different 3-D boundary component pairs. 15 classifications describe disjoint relationships while 38 classifications describe various touch and overlap relationships that produce common point, line, or plane geometries.

While the relationships presented here were generalized to disjoint, touch, and overlap/intersect classifications, processes could be expanded to specify if containment and equal relationships exist (see Figure 4). The different variations of 1D-1D line results in Figure 12 suggest this could be done. Since simulated datasets only included simple boundary planes, more testing will be done involving complex planes where interior angles can be greater than 180 degrees. The model will be expanded to include classifying the topological relationships between the 3-D boundary components discussed here and a 3-D closed solid object defined by four or more boundary planes. It will then be applied to 3-D property units defined by a strata survey plan to explore its application regarding 3-D cadastral object analysis and management.

REFERENCES

Billen, R. et al. (2002). The Dimensional Model: A Framework to Distinguish Spatial Relationships. *Advances in Spatial Data Handling.*

Clementini, E. (2019). A Conceptual Framework for Modelling Spatial Relations. *Journal of Information Technology and Control, 48(1), p. 5-17.*

Dorst, L. et al. (2007) Geometric Algebra for Computer Science (Revised Edition): An Object-Oriented Approach to Geometry. *The Morgan Kaufmann Series in Computer Graphics, Elsevier Science.*

Ellul, C. & Haklay, M. (2006). Requirements for Topology in 3D GIS. *Transactions in GIS, 10(2), p. 157-175.*

Fu, L. et al. (2018). Characteristics and Classification of Topological Spatial Relations in 3-D Cadasters. *Information, 9,71.*

Hildenbrand, D. (2015). Geometric Algebra: A Foundation of Elementary Geometry with Possible Applications in Computer Algebra based Dynamic Geometry Systems. *The Electronic Journal of Mathematics and Technology, 1(1).*

Yu, Z. et al. (2016). Geometric Algebra Model for Geometry-Oriented Topological Relation Computation. *Transactions in GIS, 20(2), p. 259-279*

Yuan, L W. et al. (2014). Multidimensional-unified Topological Relations Computation: A Hierarchal Geometric Algebra-based Approach. *International Journal of Geographical Information Science.*

Yuan L W. et al. (2012). Geometric Algebra Method for Multdimensionally-unified GIS Computation. *Chin Sci Bull, 57, p. 802-811.*

Yuan, L W. et al. (2011). A 3D GIS Spatial Data Model based on Conformal Geometric Algebra. *Sci China Earth Sci, 54: p. 101-112.*

Yuan, L W. et al. (2010). CAUSTA: Clifford Algebra-based Unified Spatio-Temporal Analysis. *Transactions in GIS, 14(s1), p. 59-83.*

Zhang, F. et al. (2016). Unified Spatial Intersection Algorithms Based on Conformal Geometric Algebra. *Mathematical Problems in Engineering.*

Zhang, J. et al. (2016). 3D Cadastral Data Model Based on Conformal Geometry Algebra. *International Journal of Geo-Information, 5, 20.*

CONTACTS

Dillon Pullano
University of Calgary
2500 University Drive NW
Calgary, Alberta
CANADA
Email: dgpullan@ucalgary.ca
Web site: www.ucalgary.ca/mikebarry/team