Three Dimensional Spatial Object Model †

Sun Jun Lee*, Sang Ho Kim, Seong Ho Lee†, Jae Du Chung, Keun Ho Ryu

Database Laboratory, Chungbuk National University
Cheongju Chungbuk, 361-763, Korea,
e-mail: {sunjun, shkim, khryu}@dblab.chungbuk.ac.kr
Phone : 043-261-2254 Fax : 043-275-2254

† Electronics and Telecommunications Research Institute, Korea
e-mail: sholee@etri.re.kr

Abstract

As Geographic Information Systems represents three-dimensional topological Information, the systems provide accurate and delicate services for user. In order to execute three-dimensional topological operations, a dimensional transformation and heterogeneous spatial models should be used. However, the existing systems that use the dimensional transformation and the heterogeneous models, are not only difficult to operate the spatial operators, but also happened to support non-interoperability.

Therefore, in order to support the spatial operation as well as interoperability between dimensions, we propose three-dimensional spatial operators for the proposed models. We defined the three-dimensional spatial operators prior to designing the proposed model. We also implemented the operators of proposed model and evaluated the implemented model on the component environment. Finally, the proposed model is able to not only support interoperability among systems but also execute spatial queries efficiently on three-dimensional spatial objects.

Keyword: GIS, Spatial Database, Spatial Object Model

1. Introduction

In order to manage the systems which relates to rapidly growing spatial data, the systems has developed to relate the database. Until now the existing relational database satisfied for text base data management. However, because of supportive data type, query language and limit of the operator, high dimensional spatial operators could not be expressed correctly. Both databases and GIS have been researched to manage spatial data of complicated structure in a telecommunication line system, a territory management system, digital mapping, science database, and a medical care management system together.

Generally a spatial database, through abstraction of a point, a line and a plane, represent a spatial object[1]. However, the existing concepts such as a dimensional transformation have a difficulty to manage real three-dimensional objects. At the reasons, Three-dimensional spatial operators are required to solve this problem. And spatial databases use an international standard OGIS model, an OGIS service model and an information communities model to support the interoperability of spatial data[2]. However the two-dimensional spatial models which presented in an OGIS consortium were not compatible with three-dimensional spatial models.

Therefore, in order to solve the problems, we proposed three dimensional spatial object models and implemented them to show validity of the proposed models. When designing the three dimensional topological operators, we used 3DE-9IM which extended DE-9IM to support three dimensional concepts, and implemented operators on the component environment with object oriented concepts.

The remainder of this paper is organized as follows: we discuss existing two-dimensional spatial models in chapter 2, and propose a three-dimensional object model in chapter 3. We use a model described in chapter 3, define and design three-dimensional spatial operators and implementate them in chapter 4 and 5. In chapter 6, we summarize our contributions and mention some future directions for research.

2. Related Work

2.1 Two-dimensional spatial models

Because we implemented three-dimensional object models that proposed in this paper according to an international standard, we discuss two-dimensional spatial models and topological operators of an OGIS consortium in a related work.

Figure 1 is based on extending the geometry model specified in the OpenGIS abstract specification[3,4] with

† This work was supported by the Spatial information Technology Center of ETRI and University IT Research Center Project

specialized zero-dimensional, one-dimensional and two-dimensional collection classes named MultiPoint, MultiLineString and MultiPolygon for modeling geometries corresponding to collections of Points, LineStrings and Polygons respectively. MultiCurve and MultiSurface are introduced as abstract superclasses that generalize the collection interfaces to handle Curves and Surfaces. Figure 1 shows aggregation lines between the leaf collection classes and their element classes, the aggregation lines for non-leaf collection classes are described in the text.

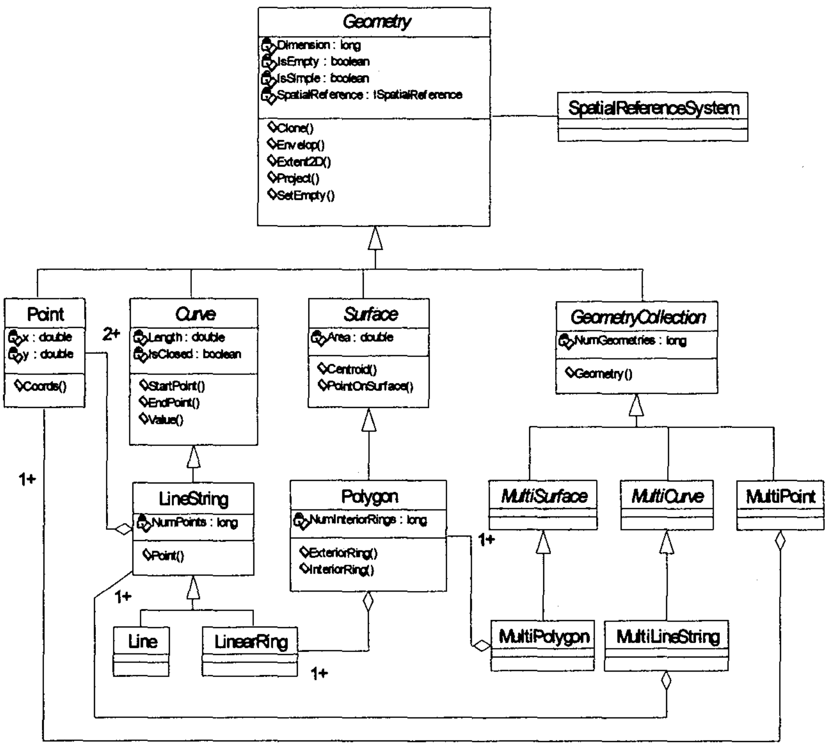

Figure 1. Two-dimensional spatial model of OpenGIS

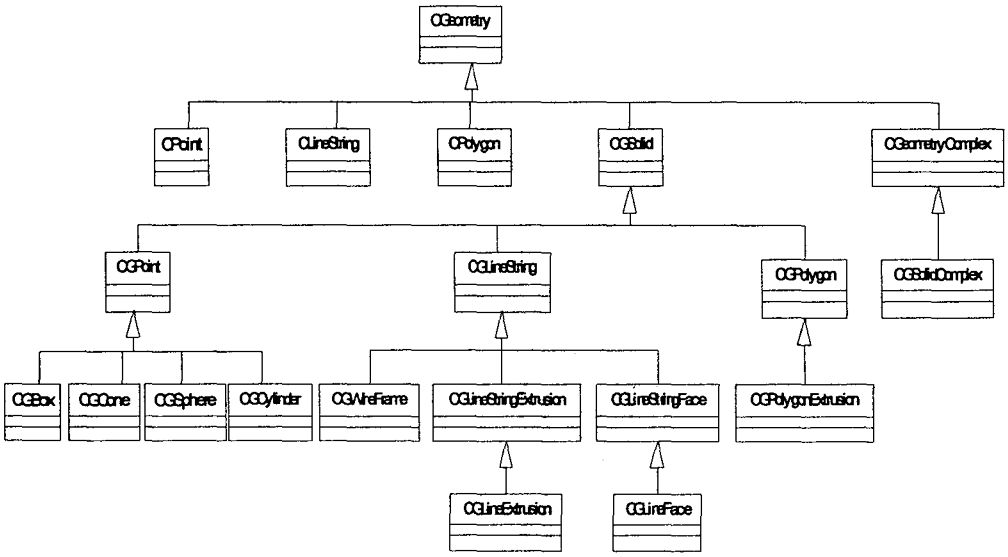

Figure 2. Three-dimensional spatial model class

3. Three-dimensional spatial object model design

3.1 Design principle

In this paper, we considered the following and defined three-dimensional object models.

- The models consider three-dimensional spatial objects.
- The models include object-oriented general concepts.
- The models consider interoperability and transformation of data.

3.2 Three-dimensional object model

Three-dimensional object models are chosen as a way to extend an opengis model in order to consider interoperability and transformation of data. However, the OGIS model[5,6] for an international standard is limited to two-dimensions, and an OGIS model is not extended to three-dimensions. Therefore, we extend two-dimensional spatial models in this paper and define a three-dimensional spatial model.

Figure 2 is three-dimensional spatial models. Abstract class GSolid have three-dimensional spatial attributes and methods. A sub class of GSolid has GPoint, GlineString and GPolygon. This class has relationship to be same as the structure that they proposed in two-dimensional models of [3], but it is expressed to three-dimensional coordinates. Each GBox, GCone, GSphere, GCylinder, GWireFrame, GLineStringExtrusion, GLineStringFace and GPolygonExtrusion is sub classes of three classes, and eight classes become three-dimensional spatial objects for instance.

4. Three-dimensional spatial operator

In this chapter, we propose a 3DE-9IM, which is extended by DE-9IM to support three-dimensional concepts. The 3DE-9IM can distinguish topological relationships between three-dimensional spatial objects. We design three-dimensional spatial objects in the following Section 4.1.

4.1 A design of three-dimensional spatial object

Three-dimensional spatial objects are simple objects having a point, a line, an area, and a solid. A point means one point of x, y, z, and a line consists of two points, and an area has a closed scope with lines and a solid is three-dimensional spatial objects which consisted of area.

P, L, A, S means each Point, Line, Area, Solid, and * means one during four space objects.

Table 1. Relationships between three-dimensional spatial geometry objects

	Interior	Boundary	Exterior
Interior	3	2	3
Boundary	2	1	2
Exterior	3	2	3

3DE-9IM has dimension price (-1, 0, 1, 2, 3) of intersection scope.

We define Disjoint3D, Touches3D, Crosses3D, Within3D, Overlaps3D Contains3D, and Intersects3D operators and design these operators. First of all, we extend DE-9IM to 3DE-9IM and define topological relationships between three-dimensional spatial objects.

(Definition 1) The pattern matrix consists of a set of 9 pattern-values, one for each cell in the matrix. The possible pattern values p are {T, F, *, 0, 1, 2,3} and their meanings for any cell where x is the intersection set for the cell are as follows:

$$p = T \Rightarrow \dim(x) \in \{0,1,2,3\} \quad \text{ie.} x \neq \phi$$
$$p = F \Rightarrow \dim(x) = -1 \quad \text{ie.} x = \phi$$
$$p = * \Rightarrow \dim(x) \in \{-1,0,1,2,3\} \quad \text{ie.Dont'tCare}$$
$$p = 0 \Rightarrow \dim(x) = 0$$
$$p = 1 \Rightarrow \dim(x) = 1$$
$$p = 2 \Rightarrow \dim(x) = 2$$
$$p = 3 \Rightarrow \dim(x) = 3$$

According to definition 1, we define an individual operator to design topological operators. Please refer to definitions which are from definiton 2 to definition 7.

(Definition 2) Disjoint3D Operator is defined as follows:
All case:

$$a.\text{Disjoint3D}(b) \Leftrightarrow (I(a) \cap I(b) = \phi) \wedge (I(a) \cap B(b) = \phi) \wedge (B(a) \cap I(b) = \phi) \wedge (B(a) \cap B(b) = \phi)$$
$$\Leftrightarrow a.\text{Relate3D}(b, "FF*FF****")$$

(Definition 3) Touches3D Operator is defined as follows:

All group (a P/P group exclusion) :

$$a.\text{Touches3D}(b) \Leftrightarrow (I(a) \cap I(b) = \phi) \wedge ((B(a) \cap I(b) \neq \phi) \vee (I(a) \cap B(b) \neq \phi) \vee (B(a) \cap B(b) \neq \phi))$$
$$\Leftrightarrow a.\text{Relate3D}(b, "FT*******") \vee a.\text{Relate3D}(b, "F**T*****") \vee a.\text{Relate3D}(b, "F***T****")$$

(Definition 4) Crosses3D Operator is defined as follows:

L/L group:

$$a.\text{Crosses3D}(b) \Leftrightarrow \dim(I(a) \cap I(b)) = 0$$
$$\Leftrightarrow a.\text{Relate3D}(b, "0********")$$

P/L, P/A, P/S, A/A, A/L, A/S group:

$$a.\text{Crosses3D}(b) \Leftrightarrow ((\dim(I(a)) \cap \dim(I(b))) < (\max(\dim(I(a)), \dim(I(b))))) \wedge (I(a) \cap I(b) \neq \phi) \wedge (I(a) \cap E(b) \neq \phi)$$
$$\Leftrightarrow a.\text{Relate3D}(b, "T*T******")$$

(Definition 5)Within3D Operator is defined as follows:

All group(dim(a) <= dim(b)):

$$a.\text{Within3D}(b) \Leftrightarrow (\dim(a) \leq \dim(b)) \wedge (I(a) \cap I(b) \neq \phi) \wedge (I(a) \cap E(b) = \phi) \wedge (B(a) \cap E(b) = \phi)$$
$$\Leftrightarrow a.\text{Relate3D}(b, "TF***F***")$$

(Definition 6) Overlaps3D Operator is defined as follows:

L/L group:

$$a.\text{Overlaps3D}(b) \Leftrightarrow (\dim(I(a) \cap I(b)) = 1) \wedge (I(a) \cap E(b) \neq \phi) \wedge (E(a) \cap I(b) \neq \phi)$$
$$\Leftrightarrow a.\text{Relate3D}(b, "1*T***T**")$$

A/A, S/S group:

$$a.\text{Overlaps3D}(b) \Leftrightarrow (I(a) \cap I(b) \neq \phi) \wedge (I(a) \cap E(b) \neq \phi) \wedge (E(a) \cap I(b) \neq \phi)$$
$$\Leftrightarrow a.\text{Relate3D}(b, "T*T***T**")$$

(Definition 7) Contains3D Operator is defined as follows:

$$a.\text{Contains3D}(b) \Leftrightarrow b.\text{Within3D}(a)$$

Figure 3 shows Overlaps3D relationship between three-dimensional objects.

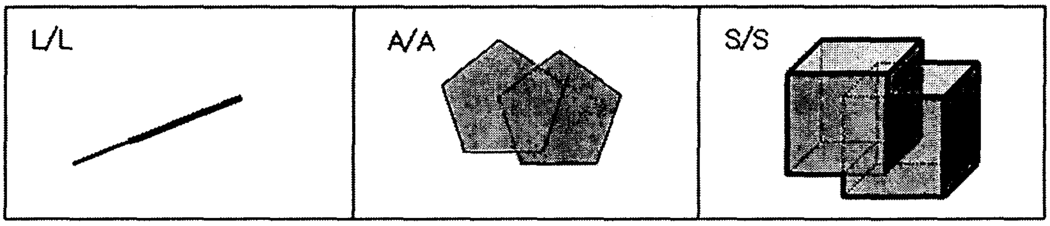

Figure 3. Overlaps3D

(Definition 8) Intersects3D Contains3D Operator is defined as follows:

$$a.\text{Intersects3D}(b) \Leftrightarrow ! a.\text{Disjoint3D}(b)$$

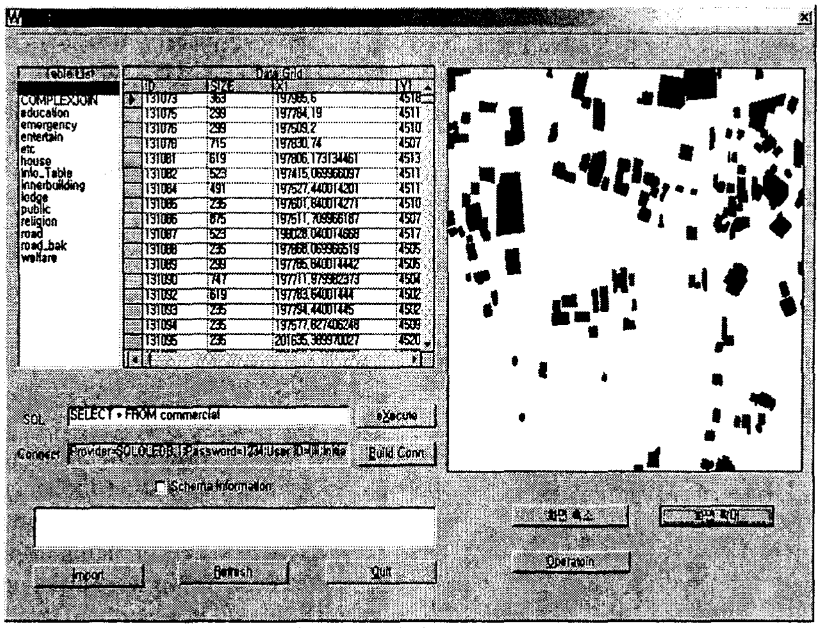

Figure 4. Original data

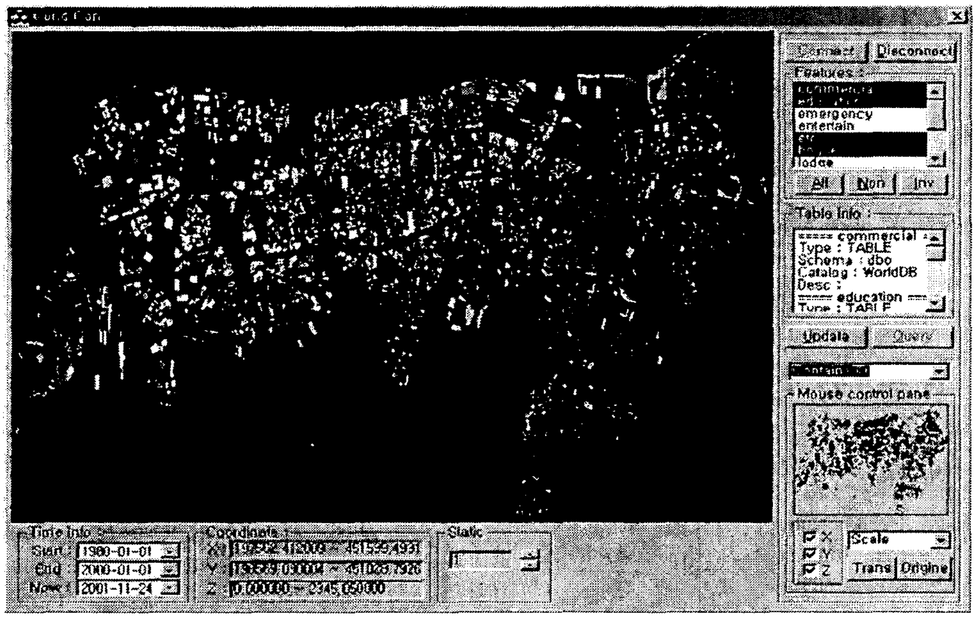

Figure 5. Result of Contains3D operation

5. Implementation and Evaluation

We implement the three-dimensional topological operators that designed in a chapter 4.

5.1 Implementation platform

Our implementation platform is follows, operating system is windows 2000 server, language is Visual C++6.0 of Microsoft and database server is Microsoft SQL Server 97.

A component GIS is composed of data provider, data consumer, data manager and data viewer. Three-dimensional data providers support datasource, session, command and rowset interface. They use the interface that OLEDB provider component is same. An OLEDB provider provides meta information, schema information and query results.

5.2 Implementation and operation

In order to perform topological operation, the provider must provide data table, meta data table and index table. The data table is a database table for real feature data to be saved, and the meta tables have information of spatial data structure. An index table has indexing information. A test data is building data of Seoul City. Each building corresponds to a layer.

Figure 5 shows the results, which applied a Contains3D topological operator. It is in building data of JungGu and, as for each tables displayed in a table list, a selected road table is managing road data of JungGu. We analyze WKB data and draw results.

5.3 Evaluation

We implemented the proposed three-dimensional operators. In order to compare the existing GIS with the proposed system, we use the following standards in Table 2.

<Table 2> Comparison of existing geographic information system and proposed models

	GOTHIC	GEUS	Small World	This paper
①	O	O	O	O
②	△	△	△	O
③	X	△	△	O
④	O	O	O	O
⑥	X	△	△	O
⑥	O	O	O	O
⑦	X	X	X	O

①Can it support the two-dimensional spatial objects? ② Can it support the three-dimensional spatial objects? ③ Can it support interoperability with OGIS model? ④ Can it support the two-dimensional spatial topological relations? ⑤ Can it support the three-dimensional spatial topological relations? ⑥ Can it support the two-dimensional spatial operators? ⑦ Can it support the two-dimensional spatial operator?

We choose SmallWorld[7], GEUS[8], GOTHIC[9,10] to compare proposed system and we did not include query processing time and system safety in the standards.

6. Conclusion

Generally a spatial database, through abstraction of a point, a line and a plane, represent a spatial object. However, the existing concepts, a dimensional transformation, have a difficulty to manage real three-dimensional objects. At the reasons, Three-dimensional spatial operators are required to solve this problem.

Therefore, in order to solve the problems, we proposed three dimensional spatial object models and implemented them to show validity of the proposed models. When implementing them, three-dimensional spatial objects, which are Gbox, Gcone, GSphere, GCylinder, GWireFrame, GLineStringExtrusion and GPolygonExtrusion, are defined. Also, Three-dimensional spatial operators, which are Contains3D, Within3D, Overlaps3D, Intersects3D, Touches3D, Disjoint3D, and Crosses3D, are defined. The defined operators are implemented on the component environment with object-oriented concepts.

Three-dimensional spatial topological relational operators implemented in this paper not only support the efficient spatial topological relational operation about three-dimensional spatial objects, but also support interoperability between data and each other data providers.

7. References

[1] M. Yuan, "Modeling Semantical, Temporal, and Spatial Information in Geographic information Systems," M. Craglia and H. Couclelis eds. Geographic Information Research: Bridging the Atlantic, Taylor & Francis, pp.334-347, 1996.

[2] Open GIS Consortium, Inc. OpenGIS, "The OpenGIS Guide: Introduction to Interoperable Geoprocessing," OpenGIS TC Document, 96-001, 1996.

[3] Open GIS Consortium, Inc. OpenGIS, "Simple Features Specification For OLE/COM Revision 1.1," OpenGIS Project Document, 99-050, 1999.

[4] Open GIS Consortium, Inc. OpenGIS, "Simple Features Specification For OLE/COM Revision 1.1," OpenGIS Project Document, 99-050, 1999.

[5] Open GIS Consortium, Inc. OpenGIS, "The OpenGIS Abstract Specification Topic 1:Fearture Geometry (ISO 19107 Spatial Schema)," version 5, Open GIS Project Document, 01-101, 2001.

[6] Open GIS Consortium, Inc. OpenGIS, "The OpenGIS Abstract Specification Topic 5:Fearture," version 4, Open GIS Project Document, 99-105, 1999.

[7] http://www.gesmallworld.com, 2002

[8] http://www.ktdata.co.kr, 2002

[9] Laser Scan Ltd., "Writing and developing applications using GOTHIC ADE," PRG_602_TRG Issue 2.0, 1995.

[10] Laser Scan Ltd.,"Training Course:Gothic Concepts," PRG_205_TRG issue 2.0, 1995.