

# 3차원 공간 위상 관계 연산자의 설계

김 상 호<sup>†</sup> · 강 구<sup>\*\*</sup> · 류 근 호<sup>\*\*\*</sup>

## 요 약

지리정보시스템은 3차원 위상정보를 표현함으로써 사용자에게 정확하고 정교한 서비스를 제공한다. 이 때에 3차원 위상정보연산을 위해 차원변경방식과 서로 이질적인 공간모델을 사용해야 하는데, 이 방식을 사용할 때는 공간 연산이 어려울 뿐 아니라 서로 호환성이 부족한 문제가 발생한다. 따라서 이 논문에서는 이 문제를 해결하기 위하여 2차원 공간 객체 모델을 수용하는 3차원 공간 객체 모델을 제안하였고, 타당성을 보이기 위하여 구현하여, 그 실행을 보였다. 3차원 공간 위상 관계 연산자를 위하여 DE-9IM 방식을 3차원 개념으로 정의한 DE-9IM을 이용해서 설계하였고, 객체지향 개념을 지원하는 컴포넌트 환경에서 3차원 공간 위상 연산자를 구현하였다. 이 논문에서 제안된 3차원 공간 위상 연산자는 타 시스템과의 상호 운용성을 보장하며, 구현된 공간 위상 관계 연산을 이용하여, 3차원 공간 객체에 대한 효율적인 공간 질의를 수행할 수 있다.

## Design of Three Dimensional Spatial Topological Relational Operators

Sang Ho Kim<sup>†</sup> · Goo Kang<sup>\*\*</sup> · Keun Ho Ryu<sup>\*\*\*</sup>

## ABSTRACT

As Geographic Information Systems represent three dimensional topological information, The Systems provide accurate and delicate services for users. 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, is not only difficult to operate the spatial operators, but also happened to support non-interoperability. Therefore, in order to solve the problems, we proposed three dimensional spatial object models that supported two dimensional 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 proposed three dimensional spatial object models and topological operators can support interoperability between systems, and execute spatial queries efficiently on three dimensional spatial objects.

**키워드 :** 3차원 공간 객체 모델링(three dimensional spatial object modeling), 공간 위상 관계 연산자(spatial topological relational operator)

### 1. 서 론

실세계의 다양한 공간 개체를 대상으로 하는 시스템들은 점차 대량화 되어가는 공간 데이터를 효율적으로 관리하기 위해 데이터베이스와 연계하여 개발되고 있다. 기존의 관계형 데이터베이스는 지금까지 텍스트 기반의 데이터 관리에 대한 요구 사항들을 충족시켜주었으나, 시스템에서 지원되는 데이터 타입, 질의어 그리고, 연산자의 한계 때문에 고차원의 공간 데이터를 제대로 표현할 수가 없었다. 이런 한계를 극복하고 고차원 공간 데이터를 표현하기 위해, GIS 기술

그룹과 데이터베이스 그룹간의 상호 연관된 연구를 통한 공간 데이터의 관리를 위해 최적화된 시스템을 구축하고 있다.

GIS와 함께 통신 선로 시스템, 국토 관리 시스템, 디지털 지도 제작, 과학 데이터베이스, 의료 관리 시스템 등에서 복잡한 구조의 공간 데이터를 관리하기 위해 공간 데이터베이스가 연구되어 왔다. 공간 데이터베이스에서는 일반적으로 공간 객체를 점, 선, 면의 추상화를 통해 데이터베이스의 공간적 표현을 확장하였다[1]. 그러나 여러 분야의 응용 업무들이 3차원 공간의 개념을 지원하는 데이터를 대상으로 하고 있기 때문에 3차원 공간을 2차원 좌표계에 투영시켜서 연구되어왔던 데이터베이스의 개념을 3차원 공간 데이터에 그대로 적용하기는 어렵다. 이 문제를 해결하기 위하여 3차원 공간 연산이 필요한데, 이 3차원 공간 데이터 모델링에서 우선적으로 고려해야 할 점은 무엇보다도 데이터 상호 운용성과 표현의 다양성이다. 그러나, 이 데이터

\* 이 연구는 한국전자통신연구원의 공간정보기술센터와 한국과학재단 RRC (청주대 정보통신 연구센터) 및 대학 IT연구센터 육성, 지원사업의 연구비 지원으로 수행되었음.

† 준 회 원 : 충북대학교 대학원 전자계산학과

\*\* 정 회 원 : 한국통신데이터

\*\*\* 총신회원 : 충북대학교 전기전자 컴퓨터공학부 교수

논문접수 : 2002년 7월 12일, 심사완료 : 2002년 8월 1일

상호 운영성의 부재와 데이터의 다양성 때문에 데이터의 공유 및 호환이 불가능하고 최신 데이터의 유지가 힘들다 [2]. 이 문제를 해결하기 위해 국제 표준화 기구로써 조직된 개방형 GIS 컨소시엄에서는 개방형 지리 자료 모델, OGIS (OpenGIS) 서비스 모델 그리고, 정보통신 모델[3]을 제안하였지만 개방형 GIS 컨소시엄에서 제시한 지리 자료 모델은 2차원 공간 모델이며, 3차원 공간 모델의 상호 운용성은 사용하기가 어렵다. 따라서 이 논문에서는 2차원 모델의 정의를 수용하면서 차원을 3차원으로 확장하고, 이 확장된 3차원 모델의 위상 관계 연산은 3DE-9IM(3 Dimension Extended 9 Intersection Method)을 이용하여 설계한다. 그리고 제안한 모델의 타당성을 보이기 위하여 2차원 공간객체 모델을 수용하는 3차원 모델을 3DE-9IM 방식을 이용하여 구현한다. 이 논문의 전개를 위해, 제 2장에서는 기존의 2차원 공간 모델과 위상 관계 연산자에 대해 기술하고, 제 3장에서는 3차원 객체 모델을 제안한다. 그리고 제 4장에서는 제 3장에서 제안된 모델을 이용하여 3차원 공간 연산자를 정의하고 설계한다. 제 5장에서는 3차원 객체 컴포넌트와 연산자를 시스템에 적용한다. 마지막으로 제 6장에서는 이 논문을 요약하며 결론을 맺는다.

## 2. 관련 연구

### 2.1 2차원 공간 모델

이 논문에서 제안한 3차원 객체모델을 국제 표준에 따라 구현하였으므로 관련연구에서 OGIS 컨소시엄의 2차원 공

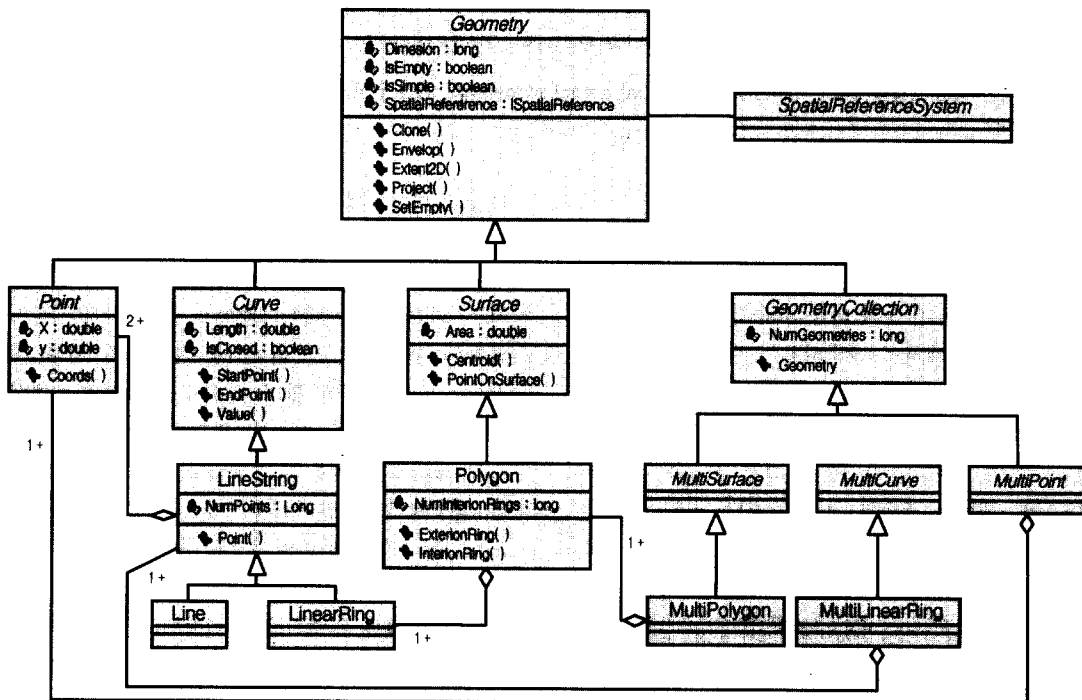
간 모델과 위상연산 그리고 2차원 DE-9IM을 설명한다.

OpenGIS 명세서[4, 5]는 Geometry라는 추상 클래스로부터 Point, Curve, LineString, Polygon등의 단일 2차원 공간 객체를 정의하고, 단일 객체의 집합(collection) 형태로 Multi-point, MultiLineString 그리고 MultiPolygon을 정의한다. 이러한 공간 객체들 간의 공간 관계 연산자들은 ISpatialRelation 인터페이스를, 공간 객체들 간의 공간 분석 연산자들은 ISpatialOperator 인터페이스를 통하여 Geometry 객체에 제공된다.

(그림 1)은 OpenGIS 2차원 공간 모델이다. 이탤릭 체로 표기된 클래스들은 추상 클래스이고, 삼각형(△)은 상속 관계를, 마름모꼴(◇)은 집합 관계를 나타내며, 숫자는 객체의 집합의 최소 개수를 표현한다.

개방형 GIS 컨소시엄에서 제안하고 있는 2차원 공간 모델을 요약하면 다음과 같다.

- 공간 객체는 점들의 집합으로 구성된다.
- 지리 객체를 위한 추상 클래스인 Geometry를 정의하고 Geometry에서 모든 공간 연산자를 지원한다.
- Geometry의 하위 객체는 Point, Curve, Surface이며, 이 중 Curve와 Surface는 각각 1차원 객체와 2차원 객체를 위한 추상 클래스이다.
- 인스턴스화 될 수 있는 객체는 Point, LineString, Polygon이며 각 객체는 내부, 경계, 외부로 구성된다.
- 점은 경계가 없고 선은 양 끝점을 경계로 하며, 면은 시작점과 끝점이 동일한 선을 경계로 한다.



(그림 1) OpenGIS의 2차원 공간 모델

- 모델의 구성은 잘 알려진 타입들, 즉 int, double과 같은 타입의 리스트로 이루어진다.

최상위 계층인 Geometry는 DE-9IM을 기반으로 정의된 공간 관계 분석 연산과 함께 위상 연산, 기타 공간 관련 연산을 정의하고 상속을 통해 공간 연산들을 하위 클래스에 제공한다.

그러나 이 모델은 2차원 공간으로 한정되어 있으므로 3차원 객체를 설계하기 위해서는 2차원 모델의 정의를 유지하는 3차원 공간 모델의 정의가 필요하다.

### 2.2 위상 관계 연산자

위상 연산자에서 가장 중요한 점은 실 세계에 존재하는 모든 공간 관계의 표현이다. 위상연산에서 위상 관계를 표현하기 위하여 Spatial SQL[6]에서는 공간 객체를 내부와 경계로 나누는 방법을 사용하여 disjoint, meet, equal, covers, contain, overlap 관계를 표현하였다. Clementini[7]는 차원으로 확장된 방법과 계산에 기반 한 방법을 이용하여 공간상의 위상 관계를 표현하였다. 차원으로 확장된 방법은 많은 관계가 발생하며, 이들 각각에 의미를 부여할 수 없고, 계산에 기반 한 방법은 touch, in, cross, overlap, disjoint의 다섯 가지 관계와 b(boundary), f(from), t(to) 연산자를 사용하여 객체들간의 모든 이진 위상 관계를 표현 할 수 있음을 보였다. Geo-SAL[8]에서는 공간상의 위상 관계를 이항 위상 연산자로 분류하며 disjoint, meets, equals, contains, overlaps, covers, intersects의 일곱 가지로 표현하였다. 기하 연산자는 공간 객체간의 거리, 둘레, 각도, 방향, 면적 등을 구하는 연산으로서 공간 연산자 중에서도 기본적인 연산자이다. 기타 공간 관련 연산자는 기하 연산자와 위상 연산자를 구현하기 위해 기초가 되는 간단한 연산자를 의미한다. 기타 연산자의 경우는 응용 프로그램과 직접 관련이 깊은 연산자이므로 많은 연산자가 추가 되어야 한다.

### 2.3 2차원 DE-9IM

공간 객체의 위상 관계를 표현하는 방법은 4IM(4 Intersection Method)[9], 9IM(9 Intersection Method)[11], DEM(Dimension Extended Method)[7] 그리고 DE-9IM 방법[11]이 있다. 각 방법들 중에서 두 객체간의 연산을 가장 잘 표현 할 수 있는 것은 DE-9IM 방법이다. <표 1>은 공간 객체를 표현할 수 있는 경우의 수를 나타낸다. 여기서 A는 Area를 의미하고, L은 Line을 의미하며, P는 Point를 의미한다. 그리고 Area와 Area(A/A), Line과 Area(L/A), Point와 Area(P/A), Line과 Line(L/L), Point와 Line(P/L) 그리고 Point와 Point(P/P)들의 관계를 의미한다.

우리는 객체간의 관계를 가장 잘 표현할 수 있는 DE-9IM 방법을 사용하여 제안모델을 표현한다. DE-9IM 방법은 주어진 기하를 내부, 경계, 외부로 각각 I(a), B(a) 그리고 E(a)

라고 표현한다. I(a), B(a), E(a)에 대한 임의의 두 기하의 교집합은 혼합된 차원의 기하 집합 x를 가져올 수 있다. 예를 들면, 두 Polygon의 경계의 교집합은 점과 선으로 구성될 수 있다. dim(x)는 x 기하의 최대 차수(-1, 0, 1 또는 2)를 반환한다. -1의 수치 값은 dim(∅)과 같다. 차원적으로 확장된 9개의 교집합 행렬 (DE-9IM)은 다음의 형태를 갖는다.

<표 1> 공간 객체 표현 방법의 비교

method	A/A	L/A	P/A	L/L	P/L	P/P	Total
4IM	6	11	3	12	3	2	37
9IM	6	19	3	23	3	2	56
DEM	9	17	3	18	3	2	52
DE-9IM	9	31	3	33	3	2	81

내부와 경계, 외부의 교집합의 차원을 계산하는 위상적으로 닫혀진 입력 기하는 이러한 집합의 명확한 계산과 표현을 전제조건으로 갖지는 않는다[13]. 두 개의 닫혀진 Polygon 내부의 교집합 연산이나, 이 교집합의 차원을 확인하는 연산은 분리된 기하의 두 Polygon의 내부를 명백하게 표현할 필요는 없다. 대부분의 경우, 셀(cell)에서의 교집합 값의 차원은 주어진 두 기하의 타입에 의존하게 된다. 예를 들면, 선과 면의 경우 내부 셀들에 대해 가능한 유일한 값은 {-1, 1}로부터 나오고, 면과 면의 경우 내부의 셀들에 대한 유일하게 가능한 값은 {-1, 2}로부터 나온다. 이러한 경우에는 교집합을 찾는 작업을 제외한 어떠한 작업도 요구되지 않는다.

<표 2> DE-9IM

	내부	경계	외부
내부	dim(I(a) ∩ I(b))	dim(I(a) ∩ B(b))	dim(I(a) ∩ E(b))
경계	dim(B(a) ∩ I(b))	dim(B(a) ∩ B(b))	dim(B(a) ∩ E(b))
외부	dim(E(a) ∩ I(b))	dim(E(a) ∩ B(b))	dim(E(a) ∩ E(b))

## 3. 삼차원 공간 객체 모델 설계

### 3.1 설계 원칙

이 논문에서는 다음과 같은 점을 고려하여 3차원 객체 모델을 정의하였다.

- 3차원 공간 객체를 대상으로 한다.

객체는 3차원 유클리안 공간에서 표현된다. 여기서 객체는 지리 객체뿐만 아니라 지리에 관련된 모든 범위의 데이터를 말한다.

- 객체 지향 개념을 도입한다.

객체 지향 데이터 모델은 객체, 클래스, 캡슐화, 상속 그리고 다형성 등의 객체 지향 패러다임에 기반 한다. 각 추

상 클래스의 속성과 메소드는 인스턴스화 될수 있는 하위 클래스로 상속된다.

- 데이터의 호환성과 이식성을 고려한다.

3차원 객체 모델은 표준 모델을 기반으로 사용함으로써 각 시스템간 데이터의 확장성과 이식성을 보장한다. 그리고 컴포넌트 형태로 구현함으로써 시스템의 유지 보수 및 구축 비용의 절감의 이점을 얻을 수 있다. 데이터의 논리적인 이식성 뿐만 아니라 물리적인 이식성도 함께 고려되어야 한다. 데이터의 저장 형태는 잘 정의된 이진(WKB : Well-Known Binary)구조를 사용한다. WKB의 구조는 개방형 모델에서 제안한 WKB를 3차원 객체 모델의 인스턴스 객체에 맞도록 확장하여 설계한다.

3.2 삼차원 객체 모델

3차원 객체 모델은 데이터의 호환성 및 이식성을 고려하기 위해 개방형 모델을 기반으로 확장하는 방식을 택하고 있다. 그러나 국제 표준화 기구인 개방형 GIS 컨소시엄[14, 15]에서 제시한 공간 모델은 2차원이고, 아직 3차원 공간에 대한 개방형 모델은 제시되어 있지 않다. 따라서 이 논문에서는 2차원 공간 모델을 확장하여 3차원 공간 모델을 정의한다.

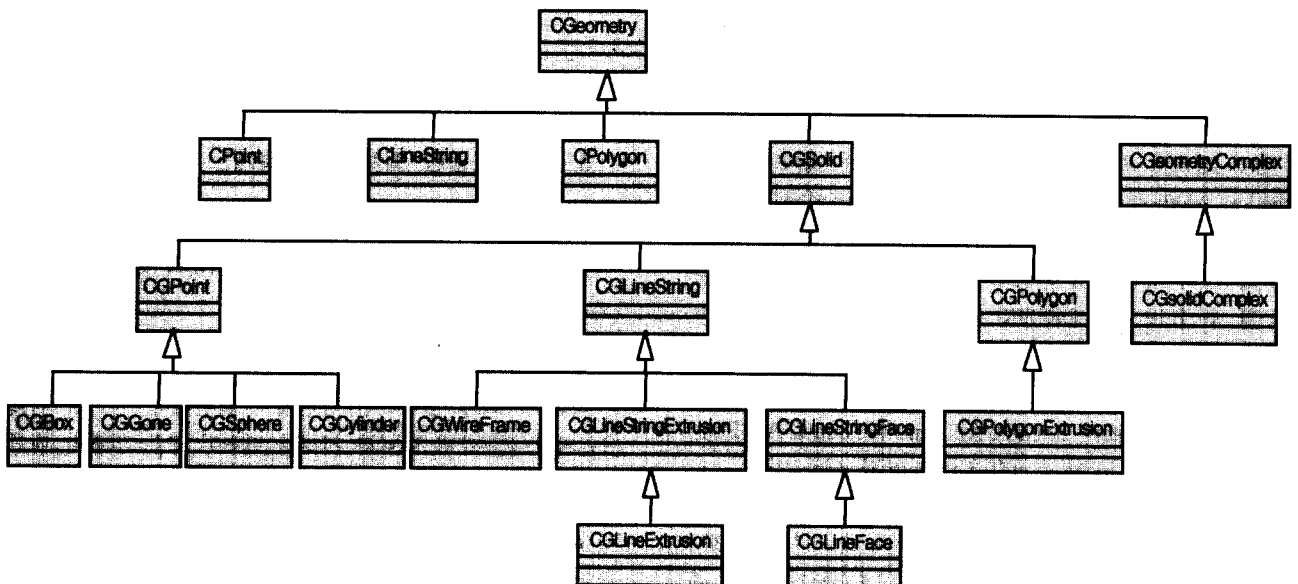
(그림 2)는 확장된 3차원 공간 모델이다. 일반적인 공간 속성과 메소드를 가지고 있는 Geometry와 3차원 공간 속성과 관련 메소드를 가지고 있는 GSolid가 추상 클래스로 정의된다. GSolid의 하위 클래스는 GPoint, GLineString, GPolygon이 있다. 이 세 클래스는 [4]의 2차원의 모델에서 제안한 구조와 동일한 관계를 갖지만 3차원 좌표로 표현된다. 이 세가지 클래스의 하위에는 각각 GBox, GCone, GSphere,

GCylinder와 GWireFrame, GLineStringExtrusion, GLineStringFace 그리고 GPolygonExtrusion이 존재하며 이 8개의 클래스가 인스턴스화 될수 있는 3차원 공간 객체가 된다.

3차원 공간 객체의 경계는 다각형들의 집합으로 구성된다. 그러나 하나의 3차원 공간 객체를 저장할때 경계를 구성하는 모든 점들을 저장하는 것은 저장 비용 및 데이터의 단순화 측면에서 비효율적이다. 따라서 각 공간 객체에 대한 기본 공간 요소들만 저장하게 된다.

GPoint의 하위 객체들은 하나의 GPoint를 갖고, 그 점을 기점으로 하는 공간 요소로 객체를 구성하게 된다. GBox 형태는 직육면체이고, 중심점과 함께 저장되는 공간 정보는 중심점을 기점으로 하는 가로, 세로, 높이다. GCone은 원뿔형이며, 중심점과 반지름, 높이를 갖는다. GSphere는 구의 형태로 중심점과 반지름을, GCylinder는 중심점과 반지름, 높이를 갖는 원통형 객체이다.

GLineString의 하위 객체들은 중심선이 되는 GLineString과, 선을 기점으로 하는 공간 요소로 객체를 구성한다. GLineStringExtrusion은 중심선과 중심선을 구성하는 점들이 중심이 되는 원의 반지름 값으로 그려지는 판의 형태이고 중심선이 정확히 두 점만으로 이루어질 경우 GLineExtrusion 이라고 정의한다. GLineStringFace는 중심선을 구성하는 점들로부터 일정한 넓이의 폭을 갖는 형태이다. GLineFace는 중심선이 정확히 두 점만으로 이루어지는 객체이며, 이러한 GLineExtrusion과 GLineFace를 이용하여 다양한 굵기로 연결되는 여러 전선이나 상(하수도관의 망을 나타내거나, 다양한 넓이를 갖는 도로망에 대한 정보를 저장할 수 있다. GWireFrame은 실세계 객체의 투시도를 그리기 위한 객체이다. 다른 인스턴스 객체들은 중심 객체와 기타 중요 공간 요소만으로 구성되지만, 상세한 건물의 투시도는 대상



(그림 2) 3차원 공간 모델 계층 클래스도

객체의 주요점 데이터를 모두 가지고 있어야 한다. 따라서 GWireFrame으로 정의된 객체는 다른 객체에 비해 많은점 데이터를 포함하게 된다.

GPolygon의 하위 객체는 GPolygonExtrusion 하나뿐이다. GPolygonExtrusion의 형태는 밀면으로 하나의 GPolygon이고, 밀면을 구성하는 각 점들로부터 일정한 높이 값을 갖는 객체이다. 이때 밀면인 GPolygon은 구멍이 없다고 가정한다.

#### 4. 삼차원 공간 연산자

이 장에서는 공간 연산을 설계하기 위하여 3차원으로 확장된 9가지 교차 방법을 확장한다. 우리는 이 확장된 공간 위상 연산 방법을 3DE-9IM이라고 하겠다. 3DE-9IM은 DE-9IM을 3차원 지리 객체에 대한 정의로 확장한 것으로, 3차원 공간 객체들간의 위상 관계를 식별할 수 있다.

아래의 4.1절에서는 3차원 공간객체를 DE-9IM 방식으로 설계한다.

##### 4.1 삼차원 공간 객체의 설계

3차원 공간 객체는 점, 선, 면, 부피를 갖는 간단한 객체들이다. 점은 x, y, z의 한 점을 말하고, 선은 두 개의 점으로 이루어져 있고, 면은 여러 개의 선으로 닫혀진 영역을 말하고, 부피는 면들로 이루어진 3차원 공간 객체이다. 공간 객체들과의 관계성을 다루기 위하여 먼저, 각각의 공간 객체들은 3차원( $R^3$ )이라고 가정한다. 그리고 P, L, A, S는 각각 Point, Line, Area, Solid를 의미하고,  $\lambda$ 는 4가지 공간 객체들 중 하나를 의미한다.

공간 객체의 경계, 내부, 그리고 외부를 정의 하면 다음과 같다.

- $\partial P$ : 점 객체의 경계는 없다.
- $\partial L$ : 라인 객체의 경계는 두 개의 끝 점을 포함하는 집합이다.
- $\partial A$ : 영역 객체의 경계는 닫혀진 커브이다.
- $\partial S$ : 부피 객체의 경계는 닫혀진 면들의 집합이다.

3DE-9IM은 교차 영역의 차원 값(-1, 0, 1, 2, 3)을 갖는다. 이것은 이론적으로  $5^9 = 1953125$  가지의 경우의 수를 갖는다. 위상적으로 닫혀진 3차원 객체간의 내부, 경계, 외부간의 관계는 <표 3>과 같이 나타낼 수 있다.

<표 3> 3차원 공간 기하 객체 간의 차원 관계

	내부	경계	외부
내부	3	2	3
경계	2	1	2
외부	3	2	3

이 논문에서 3차원 위상관계 연산자로 Disjoint3D, Touches3D, Crosses3D, Within3D, Overlaps3D, Contains3D, Intersects3D, 그리고 Equals3D 연산들로 정의하고 이 연산자를 설계한다. 3차원 공간 객체간에 Disjoint, Touch, Cross, Within, Overlap, Contain, Intersect, 그리고 Equal의 관계가 있는 지를 검사한다.

위상 공간 연산자를 정의하기 전에 먼저 DE-9IM 방법을 3차원으로 확장하여, 3차원 공간 객체간의 위상 관계를 정의한다.

(정의 1) 각각의 객체 간에 관계에서 가질 수 있는 값들 즉, 가능한 패턴 값 p는 다음과 같다.

$$P = \{T, F, *, 0, 1, 2, 3\}$$

이때 P는,

- $p = T \Rightarrow \dim(x) \in \{0, 1, 2, 3\}$  ie.  $x \neq \phi$
- $p = F \Rightarrow \dim(x) \in -1$  ie.  $x = \phi$
- $p = * \Rightarrow \dim(x) \in \{-1, 0, 1, 2, 3\}$  ie. Don't Care
- $p = 0 \Rightarrow \dim(x) \in 0$
- $p = 1 \Rightarrow \dim(x) \in 1$
- $p = 2 \Rightarrow \dim(x) \in 2$
- $p = 3 \Rightarrow \dim(x) \in 3$

이다. 그리고 dim은 차원을 의미한다.

(정의 1)에 따라 우리는 공간 위상 연산자를 설계하기 위한 각각의 연산자를 (정의 2)~(정의 7)과 같이 정의한다. 연산자 Disjoint3D는 위상적으로 닫혀진 두 기하 3차원 공간 객체 a, b가 주어졌을때 (정의 2)와 같이 표현한다.

(정의 2) 연산자 Disjoints3D는 아래 식으로 정의된다. 모든 경우 :

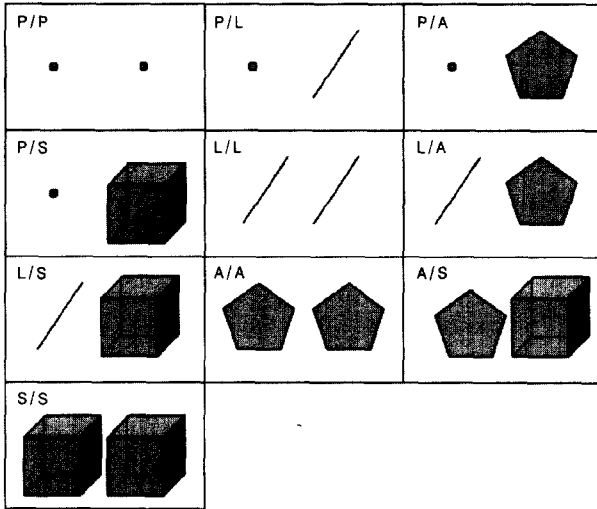
$$\begin{aligned}
 a.Disjoints3D(b) &\Leftrightarrow (I(a) \cap I(b) = \phi) \wedge \\
 &\quad (I(a) \cap B(b) = \phi) \wedge \\
 &\quad (B(a) \cap I(b) = \phi) \wedge \\
 &\quad (B(a) \cap B(b) = \phi) \\
 &\Leftrightarrow a.Relate3D(b, "FF*FF****")
 \end{aligned}$$

Disjoints3D 연산은 주어진 두 개의 3차원 공간 객체가 위상적으로 떨어져있는가를 식별한다. 이것은 두 공간 객체의 내부와 내부간에, 내부와 경계간에, 그리고 경계와 경계간에 서로 인접하지 않아야 위상적으로 떨어져있다고 할 수 있다.

Disjoints3D는 모든 차원의 객체에 대하여 동일한 조건을 갖는다. 모든 그룹간에 동일한 조건을 갖는다. (그림 3)은 Disjoint3D 관계이고, <표 4>는 패턴 행렬 값이다.

3차원 공간 객체 a, b간의 연산자 Touches3D는 정의 3과 같이 표현한다.

두 기하 a, b 사이의 Touch관련성은 P/P 그룹을 제외한 그룹에서만 적용될 수 있다.



(그림 3) Disjoints3D

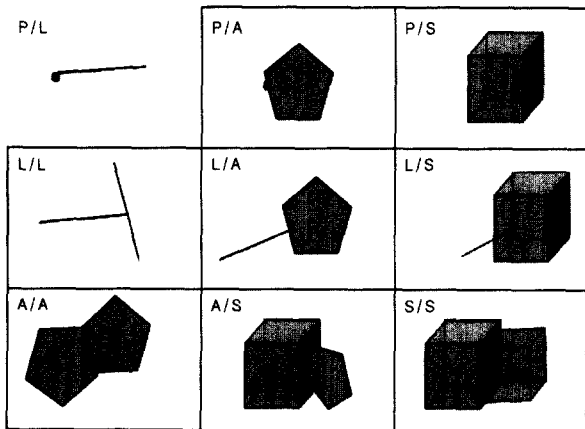
<표 4> Disjoints3D의 패턴 행렬

	내부	경계	외부
내부	F	F	*
경계	F	F	*
외부	*	*	*

(정의 3) 연산자 Touches3D는 아래 식으로 정의 된다.

모든 경우(P/P 그룹 제외) :

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



(그림 4) 공간 객체간의 Touches3D 연산

Touches3D 연산은 점과 점간의 관계를 제외하고 나머지

모든 경우에 대해서 다음 조건을 만족한다. 점에 대한 경계가 존재하지 않기 때문에, 점과 점과의 관계에서 Touches는 발생할 수 없다. (그림 4)는 각 그룹에 대한 Touches3D 연산을 보여준다.

(정의 4) 연산자 Crosses3D는 아래 식으로 정의된다  
(단, 다음 두 조건을 만족해야 한다.

- ① 닫힌 a, b에서  $(a \cap b \neq a) \wedge (a \cap b \neq b)$
- ② 두 객체 교차하는 공간의 내부공간의 차원이 최대 차원보다 작아야 한다.

L/L 그룹일 경우 :

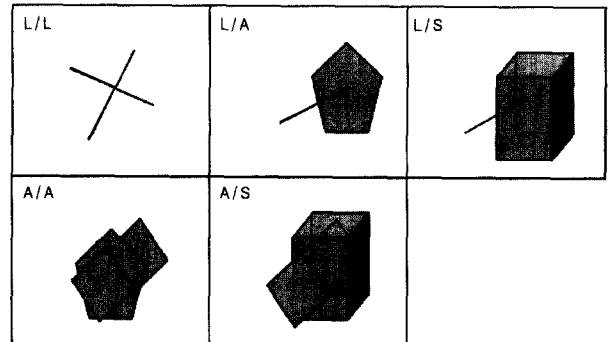
$$\begin{aligned}
 a.Crosses3D(b) &\Leftrightarrow \dim(I(a) \cap I(b)) = 0 \\
 &\Leftrightarrow a.Relate3D(b, "0*****")
 \end{aligned}$$

P/L, P/A, P/S, A/A, A/L, A/S 그룹일 경우 :

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

주어진 두 개의 3차원 공간 객체가 위상적으로 가로지르는가를 식별한다. 이것은 두 공간 객체의 내부와 내부간의 차원이 두 객체의 내부의 차원들의 최대 차원보다 작아야 하고, a객체와 b객체의 교집합이 a나 b가 아니어야 한다.

점과 점의 교차 관계는 "교차하는 영역이 a나 b가 아니다"라는 조건에 위배되기 때문에 교차관계를 나타낼 수 없다. 그리고 부피와 부피의 교차 관계는 "두 객체 교차하는 공간의 내부공간의 차원이 최대 차원보다 작아야 한다"는 조건에 위배되기 때문에 교차관계를 나타낼 수 없다. (그림 5)는 위상적으로 닫힌 두 객체간에 발생 가능한 Crosses3D 연산이다.



(그림 5) 공간 객체간의 Crosses3D 연산

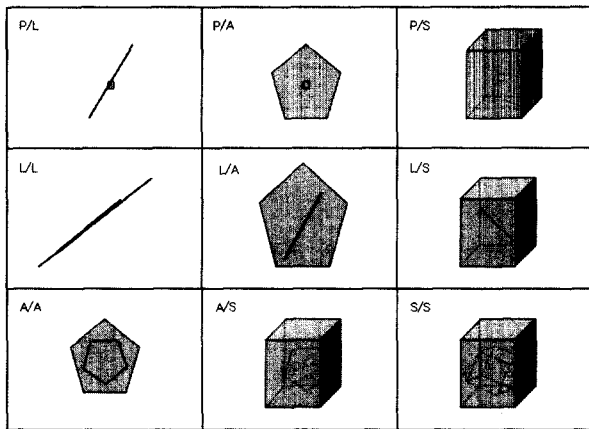
(정의 5) 연산자 Within3D는 아래식으로 정의된다.

모든 경우(단 객체 a가 객체 b보다 차원이 작거나 같아야 한다).

$$\begin{aligned}
 a.\text{Within } 3D(b) &\Leftrightarrow (\dim(a) \leq \dim(b)) \wedge \\
 &(I(a) \cap I(b) \neq \phi) \wedge \\
 &(I(a) \cap E(b) = \phi) \wedge \\
 &(B(a) \cap E(b) = \phi) \\
 &\Leftrightarrow a.\text{Relate } 3D(b, "TF***F***")
 \end{aligned}$$

주어진 두 개의 3차원 공간 객체가 위상적으로 포함되는가를 식별한다. 이것은 두 공간 객체의 내부와 내부간에 교차가 있고, a의 내부와 b의 외부간의 교차가 있고, 그리고 a의 경계와 b의 외부간의 교차가 있을때, 위상적으로 포함된다고 할 수 있다.

(그림 6)의 예는 위상적으로 닫힌 객체간의 Within3D 관계이다.



(그림 6) Within3D

(정의 6) 연산자 Overlaps3D는 아래 식으로 정의한다.

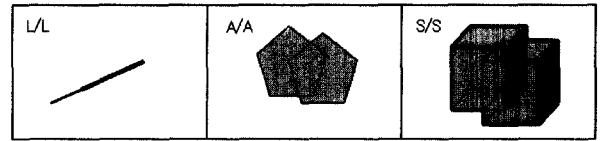
L/L 그룹일 경우 :

$$\begin{aligned}
 a.\text{Overlaps } 3D(b) &\Leftrightarrow (\dim(I(a) \cap I(b)) = 1) \wedge \\
 &(I(a) \cap E(b) \neq \phi) \wedge \\
 &(E(a) \cap I(b) \neq \phi) \\
 &\Leftrightarrow a.\text{Relate } 3D(b, "1*T***T**")
 \end{aligned}$$

A/A, S/S 그룹일 경우 :

$$\begin{aligned}
 a.\text{Overlaps } 3D(b) &\Leftrightarrow (I(a) \cap I(b) \neq \phi) \wedge \\
 &(I(a) \cap E(b) \neq \phi) \wedge \\
 &(E(a) \cap I(b) \neq \phi) \\
 &\Leftrightarrow a.\text{Relate } 3D(b, "T*T***T**")
 \end{aligned}$$

주어진 두 개의 3차원 공간 객체가 위상적으로 겹치는가를 식별한다. Overlap 연산은 같은 유형의 공간 객체에 대해서 평가될 수 있다. 이것은 각각의 공간 객체의 내부 차원과 교차된 공간의 차원이 같아야 하고, 또한 a객체와 b객체의 교집합이 a나 b가 아니어야 한다. Overlaps3D 연산은 점과 점, 선과 선, 면과 면, 그리고 부피와 부피간에 관계를 갖을 수 있다. (그림 7)은 3차원 객체들간의 Overlaps3D 관계를 나타낸다.



(그림 7) Overlaps3D

(정의 7) 연산자 Contains3D는 아래 식으로 정의된다.

$$a.\text{Contains } 3D(b) \Leftrightarrow b.\text{Within } 3D(a)$$

주어진 두 개의 3차원 공간 객체가 위상적으로 포함하는가를 식별한다. 이것은 두 공간 객체의 내부와 내부간에 교차가 있고, a의 외부와 b의 내부간의 교차가 있고, 그리고 a의 외부와 b의 경계간의 교차가 있을때, 위상적으로 포함된다고 할 수 있다.

Contains3D 연산은 모든 차원의 객체에 대하여 동일한 조건을 갖는다. 점과 점, 점과 선, 점과 면, 점과 부피 등의 모든 객체 타입간에 동일한 조건을 갖는다. Within3D 연산과 반대의 의미를 갖는다.

(정의 8) Intersects3D는 아래 식으로 정의된다.

$$a.\text{Intersects } 3D(b) \Leftrightarrow !a.\text{Disjoint } 3D(b)$$

Intersects3D 연산은 모든 차원의 객체에 대하여 동일한 조건을 갖는다. 주어진 두 개의 3차원 공간 객체가 위상적으로 교차하는가를 식별한다. 이것은 Disjoint3D와 반대로, 떨어져 있지 않은 모든 객체들은 교차한다고 할 수 있다.

## 5. 구현 및 평가

이 장에서는 제 4장에서 설계한 3차원 공간 위상 관계 연산자를 구현한다. 공간 위상 관계 연산자를 위한 데이터 제공자 컴포넌트 구현 플랫폼 및 수행 결과와 평가에 대하여 기술한다.

### 5.1 구현 플랫폼

데이터 제공자 컴포넌트에 사용된 운영체제는 Microsoft의 Windows 2000 server, 개발 언어로는 Microsoft의 Visual C++ 6.0, 그리고 데이터베이스 서버는 Microsoft SQL Server 97을 사용하여 개발하였다.

컴포넌트 기반의 지리정보 시스템은 데이터 제공자, 데이터 접근자, 데이터 관리자, 데이터 뷰어로 구성된다. 데이터 제공자는 3차원 공간 데이터를 제공하는 컴포넌트로 Data Source, Session, Command, Rowset 인터페이스를 지원한다.

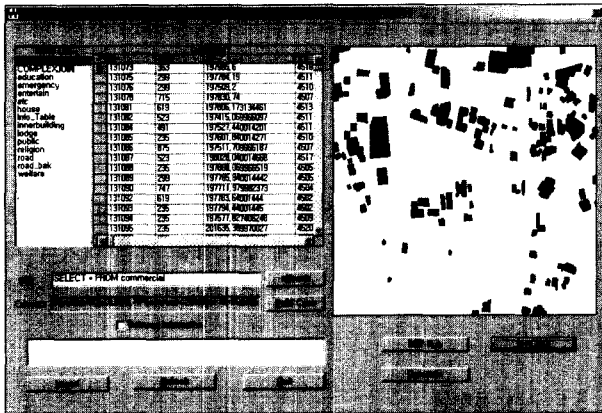
표준 OLEDB 컴포넌트는 Microsoft에서 제공하는 인터페이스를 이용하여 데이터를 제공하는 방법이다. OLEDB 제공자 컴포넌트 또한 같은 인터페이스를 이용하여 3차원 공간 데이터를 제공한다. 3차원 공간 데이터가 저장되어있는

데이터 소스는 사용자에게 투명성을 제공한다. 이러한 투명성을 제공하기 위해서 각각의 데이터 소스에 종속적인 OLEDB 컴포넌트를 통해 데이터를 제공 받으며, 제공 받은 데이터를 처리하여 OLEDB 인터페이스를 통하여 또다시 데이터를 제공한다. OLEDB 제공자는 사용자의 질의뿐만 아니라 메타 정보, 스키마 정보 등을 제공한다. 각각의 저장장치에 따른 OLEDB 제공자를 이용하여 엔진 레벨에서 이들 제공자를 이용하여 데이터에 접근하고, 접근된 데이터를 제공자 레벨에서 응용 프로그램에 제공한다.

5.2 구현과 연산의 수행

공간 위상 연산을 수행하기 위해 공간 데이터 제공자가 제공해야 하는 테이블에는 크게 데이터 테이블과 메타데이터 테이블이 있다. 데이터 테이블은 실제 Feature 데이터들이 저장되기 위한 데이터베이스 테이블이고, 메타데이터 테이블은 공간 데이터 제공자가 제공하는 정보들을 갖는 테이블들이다. 또한, 인덱싱 정보를 위한 인덱스 테이블이 있다. 테스트 데이터는 서울시의 공간 데이터들로서, 건물들을 나타내고 있다. 각각의 건물 유형은 레이어에 해당한다.

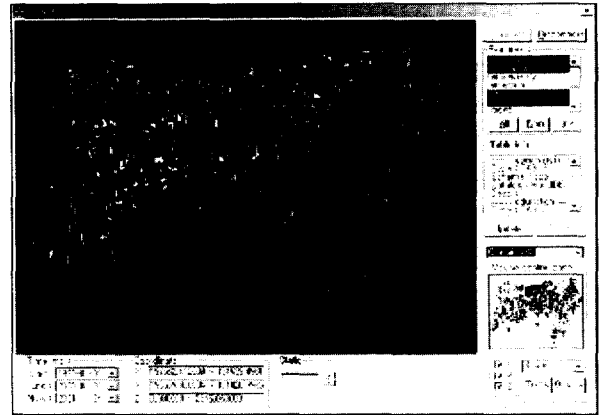
이와 같은 유형의 원본 데이터는 (그림 8)에서 보여주고 있다. (그림 8)은 데이터베이스 내의 테이블과 각 테이블의 레코드들을 나타내며, 각 레코드를 3차원 공간객체로 표현한 그림이다.



(그림 8) 수행될 원본 데이터

(그림 8)의 원본데이터를 기본으로 하여 제 4장에서 설계된 연산자를 이용하여 공간 연산을 수행한다. 설계된 Contains3D, Within3D, Overlap3D, Intersects3D, Touches3D, Equals3D, Disjoint3D 그리고 Crosses3D 연산들 중에서 Contains3D 위상관계 연산자를 적용시킨 결과화면을 (그림 9)에서 보여주고 있다. 테이블 리스트에 출력된 각 테이블들은 서울특별시 중구의 건물 데이터이며 (그림 9)에서 선택된 road 테이블은 서울특별시 중구의 도로 데이터와 상업적 건물에 대한 데이터를 관리하고 있다. 이때 그림을 그리기 위해 사용된 좌표들은 WKB 데이터를 분석한 데이터

를 사용하기 때문에 그래픽 뷰어의 출력은 WKB 데이터 분석이 성공적으로 수행되었음을 알려준다. 이렇게 나타난 데이터의 사용자 인터페이스를 통해 해당하는 위상관계 연산자를 나타내었다.



(그림 9) Contains3D 연산 적용 결과

5.3 평가

앞 절에서는 제안된 3차원 공간 연산자를 구현하여 수행하였다. 제안된 모델을 다른 모델들과 비교하기 위하여, 기존의 지리정보시스템과 제안된 시스템을 다음과 같은 7가지 기준을 사용하여 평가 하였다.

7가지 기준은 ① 2차원 공간객체의 지원 ② 3차원 공간객체의 지원 ③ 개방형 모델과의 호환성 ④ 2차원 공간 위상 관계의 표현 ⑤ 3차원 공간 위상 관계의 표현 ⑥ 2차원 위상 관계 연산자의 지원 ⑦ 3차원 위상 관계 연산자의 지원으로 구성하였다.

<표 5> 제안된 모델과 기존의 지리정보시스템의 비교

	GOTHIC	GEUS	SmallWorld	제안된 모델
기준 1	○	○	○	○
기준 2	△	△	△	○
기준 3	×	△	△	○
기준 4	○	○	○	○
기준 5	×	△	△	○
기준 6	○	○	○	○
기준 7	×	×	×	○

<표 5>는 이 논문에서 제안된 모델과 기존의 지리정보시스템 모델에 대한 비교이다. 이때 비교된 기존의 지리정보시스템은 SmallWorld[16], GEUS[17], GOTHIC[18,19]이다. 기존의 지리정보시스템은 3차원 공간객체를 주로 TIN 형태의 데이터로 제공하고 있으나, 완벽하게 3차원 데이터를 제공하지는 못하고 있다. 또한 이러한 형태의 데이터를 사용하기 때문에 3차원 공간 객체들간의 위상관계들을 왜곡되지 않고 정확하게 표현할 수 없었다. 그러나 이 논문에



서 제시한 모델은 3차원 공간 객체를 지원하고, 그 3차원 공간 객체들간의 위상 관계가 표현 가능하다. 그리고 기존 7처럼 3차원 위상 관계 연산자를 지원하는 시스템은 아직 없으나, 이 논문에서 제시한 모델은 3차원 위상 관계 연산자를 지원한다.

여기서의 기준 7가지는 개방형 모델을 지원하는 여부와, 2차원과 3차원 공간의 위상관계, 위상연산자와 공간 객체 모델들로 구성하였고, 성능평가 요소인 질의 처리 속도, 시스템 안정성, 대용량 데이터 지원 여부 등의 기준은 포함하지 않았다.

## 6. 결 론

현존하는 지리정보시스템들은 3차원 공간 객체를 표현하기 위한 각기 다른 방법을 사용하고 있고, 3차원 공간객체를 2차원 좌표계에 투영시켜 관리 하는 방법을 사용하여 데이터의 위상 관계를 정확하게 표현하지 못하는 문제점을 갖고 있다. 또한, 지리정보 시스템간의 호환성을 위한 OpenGIS의 공간데이터 모델은 2차원에 한정되어 있었다. 따라서 이 논문에서는 OpenGIS의 2차원 공간 데이터 모델의 정의를 수용하면서 차원적으로 확장하여 3차원 데이터 모델을 제안하였다. 제안된 모델을 사용하여 공간 위상 연산을 정의하였다. 즉, 공간데이터 모델들은 GBox, GCone, GSphere, GCylinder와 GWireFrame, GLineStringExtrusion, GLineStringFace 그리고 GPolygonExtrusion등으로 분류할 수 있는데, 이 정의를 바탕으로 3차원 모델에 대한 Contains3D, Within3D, Overlaps3D, Intersects3D, Touches3D, Equals3D, Disjoint3D, 그리고 Crosses3D 연산을 정의하였다. 공간 객체를 표현하기 위하여, 공간 연산을 정의할 때 객체들의 관계를 가장 잘 표현할 수 있는 3차원의 DE-9IM을 통하여 정의하고 설계하였다. 그리고 타당성을 보이기 위하여 컴포넌트 기반의 공간 데이터 제공자를 통해 설계된 연산자를 구현하였다.

이 논문에서 제시된 3차원 공간 위상 관계 연산자는 지리정보 시스템에서 3차원 공간 객체에 대한 효율적인 공간 위상 관계 연산을 가능하게 하고, 데이터들간의 상호 호환성을 지원하여 서로 다른 데이터 제공자들간의 공유를 가능하게 한다.

## 참 고 문 헌

- [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, pp.96-001, 1996.
- [3] <http://www.opengis.org/>, 2002.
- [4] Open GIS Consortium, Inc. OpenGIS, "Simple Features Specification For OLE/COM Revision 1.1," OpenGIS Project Document, pp.99-050, 1999.
- [5] Open GIS Consortium, Inc. OpenGIS, "Simple Features Specification For SQL Revision 1.1," OpenGIS Project Document, pp.99-049, 1999.
- [6] M. Egenhofer, "Spatial SQL : A query and presentation language," IEEE Transactions on Knowledge and Data Engineering, Vol.6, No.1, pp.86-95, 1994.
- [7] E. Clementini, Di Felice P. and P. van Oosterom, "A Small Set of Formal Topological Relationships Suitable for End-User Interaction," Symposium on Large Spatial Databases, pp.277-295, 1993.
- [8] P. Svensson and Z. Huang, "Geo-SAL : A Query Language for Spatial Data Analysis," In Proceedings of the 2nd International Symposium on Spatial Databases, Vol.525, pp.119-140, 1991.
- [9] M. Egenhofer and Robert D. Franzosa, "Point Set Topological Relations," International Journal of Geographical Information Systems, Vol.5, pp.161-174, 1991.
- [10] E. Clementini and J. Herring, "Categorizing binary topological relationships between regions, lines, and points in geographic databases," Technical report, Department of Surveying Engineering, University of Maine, Orono, ME, 1991.
- [11] E. Clementini and Di Felice P., A Comparison of Methods for Representing Topological Relationships, Information Sciences 80, pp.1-34, 1994.
- [12] Stephan Winter, Andrew U. Frank, "Topology in Raster and Vector Representation," GeoInformatica, Vol.4, No.1, pp.35-65, 2000.
- [13] A. J. Roy and J. G. Stell, "Spatial Relations between Indeterminate Regions," GeoInformatica, Vol.5, No.3, pp.261-289, 2000.
- [14] Open GIS Consortium, Inc. OpenGIS, "The OpenGIS Abstract Specification Topic 1 : Feature Geometry (ISO 19107 Spatial Schema)," version 5, Open GIS Project Document, pp.01-101, 2001.
- [15] Open GIS Consortium, Inc. OpenGIS, "The OpenGIS Abstract Specification Topic 5 : Feature," version 4, Open GIS Project Document, pp.99-105, 1999.
- [16] <http://www.gesmallworld.com>, 2002.
- [17] <http://www.ktdata.co.kr>, 2002.
- [18] Laser Scan Ltd., "Writing and developing applications using GOTHIC ADE," PRG\_602\_TRG Issue 2.0, 1995.
- [19] Laser Scan Ltd., "Training Course : Gothic Concepts," PRG\_205\_TRG issue 2.0, 1995.



**김 상 호**

e-mail : shkim@dblab.chungbuk.ac.kr  
1997년 충북대학교 컴퓨터과학과 졸업  
1999년 충북대학교 대학원 전산학과 석사  
1999년~현재 충북대학교 대학원 전산학과 박사과정  
관심분야 : 시공간 데이터베이스, 시공간 뷰, Web Visualization, 시공간 연산자, Component GIS 등



**강 구**

e-mail : kangih@ktdata.co.kr  
2000년 건양대학교 정보전산학과 졸업  
2002년 충북대학교 대학원 전산학과 석사  
2002년~현재 한국통신데이터 근무  
관심분야 : 시공간 데이터베이스, GIS, Web GIS, GCRM, machine control



**류 근 호**

e-mail : khryu@dblab.chungbuk.ac.kr  
1976년 숭실대 전산과 졸업  
1980년 연세대학교 공학대학원 전산전공 (공학석사)  
1988년 연세대 대학원 전산전공(공학박사)  
1976년~1986년 육군군수지원사전산실(ROTC 장교), 한국전자통신연구소(연구원), 한국방송통신대, 전산학과(조교수) 근무  
1989년~1991년 Univ. of Arizona 연구원(TempIS Project)  
1986년~현재 충북대학교 전기전자 및 컴퓨터공학부 교수  
관심분야 : 시간 데이터베이스, 시공간 데이터베이스, Temporal GIS, 객체 및 지식베이스 시스템, 지식기반 정보검색 시스템, 데이터 마이닝, 데이터베이스 보안 및 Bio-Informatics