

3D 공간 DBMS 개발에 관한 연구

민경욱*, 김주완*, 진성일**
*한국전자통신연구원 텔레매틱스연구부
**충남대학교 컴퓨터공학과
e-mail : kwmin92@etri.re.kr

A Study on 3D Spatial DBMS Development

Kyoung-Wook Min*, Ju-Wan Kim*, Seong-Il Jin **
*Telematics Research Department, Electronics and Telecommunications Research Institute
**Dept. of Computer Engineering, Chungnam University

요 약

지리정보시스템(GIS: Geographic Information System)에서는 1 차원이 아닌 2 차원 공간상의 공간 데이터 처리를 위한 많은 연구가 수행되어 왔으며 DBMS 에서 이러한 2 차원 공간 데이터의 처리를 위하여 많은 기술들이 연구 개발되어 왔다. 최근 GIS 응용에서는 2 차원이 아닌 3 차원 공간 데이터 처리에 대한 요구사항이 급증하고 있다. 이에 본 연구에서는 기존 2 차원 공간 DBMS 에서 3 차원 공간데이터 처리를 위하여 확장 개발해야 할 요소에 대하여 살펴보고자 한다.

1. 서론

GIS 응용은 도시정보관리, 시설물관리, 공간분석 등 다양한 분야에서 적용되어 왔다. 이러한 GIS 응용을 위하여 기존 2 차원(이하 2D) 공간 DBMS 가 널리 활용되어 왔다. DBMS 는 대용량 데이터의 중앙 집중적 관리가 가능하고 데이터의 일관성 유지 보장, 다수 사용자의 동시 접근을 가능하게 하는 등 효과적인 데이터의 저장 관리를 가능하게 해준다. GIS 를 위한 2D 공간 DBMS 는 공간데이터 자료형과 연산자를 제공하고 빠른 데이터 접근을 위한 공간 색인을 이용하여 기존 관계형 DBMS 를 확장하도록 기술이 개발 되어 왔다.

3 차원(이하 3D) GIS 응용은 실제 공간 데이터를 3D 로 구축함으로써 보다 높은 상황 인지를 이끌어 내기 위하여 사용되고 있다. 지금까지의 3D GIS 응용은 클라이언트 위주로 개발되어 왔다. 즉, DBMS 가 아닌 클라이언트에서의 단일 파일시스템을 이용하여 3D 공간데이터를 시각화하고 분석하는 등의 소프트웨어를 개발하여 응용에 활용하여 왔다. 또한 3D 가 아닌 2D 에 높이 정보만을 저장(2.5D)하고 시각화를 위해서만 3D 객체를 생성하는 등의 방법을 이용하여 왔다. 하지만 u-City 등이 활성화 되면서 대용량 3D 공간 데이터를 효율적으로 저장 관리하기 위한 요구사항이 대두되어 기존의 클라이언트 기반의 단일 파일시스템을 이용하여 데이터를 관리하기에는 부족한 점들이 많다. 대표적으로 클라이언트에서의 데이터 중복 관리, 표준 인터페이스를 이용하지 않음으로 해서 중복 개발, 데이터의 일관성 부재 등의 심각한 문제를 야기할 수 있어 기존 2D 공간 DBMS 를 3D 공간 데이터를 처리할 수 있도록 확장 개발의 필요성이 대두 되고 있다.

이에 본 연구에서는 기존 2D 공간 DBMS 를 3D 공간 DBMS 로 확장하기 위한 요소 기술들과 본 연구에서 진행하고 있는 3 차원 공간 DBMS 의 기술개발 내용에 대해서 설명하고자 한다.

2. 관련연구

OGC(Open GIS Consortium) 국제 산업계 표준 기구에서는 2006 년에 기존 2D 공간 데이터 모델을 3D 로 확장하였다[1]. 또한 국제 표준 기구인 ISO 19107 에서도 3D 를 포함하는 공간 데이터 모델을 정의하였다. 많은 산업계에서는 이러한 국제 표준기구의 데이터 모델을 이용하여 기술개발을 수행하고 있다. OGC 에서의 기존 2D 공간 데이터 모델의 주요 객체인 Point, LineString, Polygon 을 3D 좌표계인 x, y, z 를 확장하도록 정의하였으며 건물과 같은 시설물의 표면을 표현하기 위한 PolyhedralSurface, 지형을 표현하기 위한 TIN (Triangulated Irregular Network)에 대한 자료형을 추가로 정의하였다. ISO에서는 3D 시설물을 Solid 자료형과 Surface 자료형을 정의하고 있으며 추가적으로 3D 위상(Topology) 데이터 모델을 정의하고 있다. OGC에서는 이 이외에도 XML 기반의 교환 포맷인 GML[2]과 도시공간을 표현하기 위한 CityGML[3] 규격을 정의하고 있다. 또한 최근 구글어스(<http://earth.google.com>) 서비스의 활용이 증대되고 있으며 웹에서 구글어스 서비스와의 데이터 전송 및 교환을 위한 KML[4]에 대한 규격도 정의하고 있다.

전 세계적으로 DBMS 기술을 선도하고 있는 오라클(www.oracle.com)은 2007년에 Oracle 11g Database에서 3D 공간 데이터를 처리할 수 있도록 확장하였다. 지형을 위한 Point Cloud, TIN 을 제공하고 3D 시설물을 표현하기 위한 Solid 자료형과 공간 연산자를 추가

로 제공하고 있다[5].

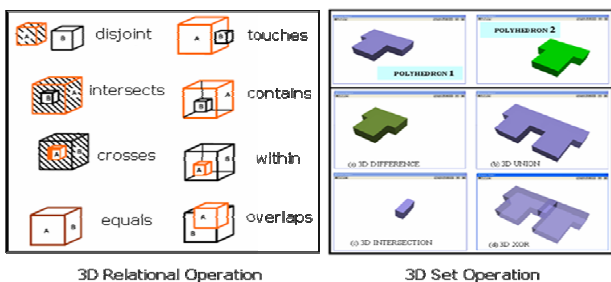
3. 3D 공간 DBMS 확장 개발을 위한 요소

기존 2D 공간 DBMS 에서 3D 공간 데이터를 수용할 수 있도록 확장 개발하기 위하여서는 여러 요소들을 고려해야 한다. 이에 해당되는 요소들로는 공간데이터 모델, 질의처리 방법, 저장 및 색인 그리고 인터페이스 등이 있다.

3.1 데이터 모델

3D 공간 데이터 모델에서 고려해야 할 사항은 3D 공간 객체를 표현하기 위한 어떤 자료형을 추가하고 연산자를 제공할 것인지를 정의하여야 한다. 그리고 공간 객체 특성상 위상 공간 데이터 모델을 제공할 것인지도 고려되어야 한다. OGC 단순 기하 데이터 모델에서는 2D Point, LineString, Polygon 의 3D 확장에 추가적으로 PolyhedralSurfaceZ, TINZ 등의 3D 공간 객체 자료형을 정의하였다. ISO 에서는 Surface, Solid 자료형을 정의하고 있으며 추가적으로 위상 공간 데이터 모델의 객체인 TP_Node, TP_Edge, TP_Face, TP_Solid 를 정의하였다. Oracle 에서는 Surface, Solid 자료형을 제공하고 있다. OGC 의 PolyhedralSurface 가 Closed 된 경우 ISO 의 Solid 와 동일하며 3D 건물이 예가 될 수 있다.

3D 공간 객체 자료형에 대한 주요 연산자는 공간 관계 연산과 공간 분석 연산이 있다. 공간 관계 연산은 공간객체의 위상관계를 정의하는 연산으로 2D 에 대하여 Egenhofer 의 9 Intersection Model 이 있다[6]. 분석 연산으로는 ConvexHull, Buffer, Intersection, Difference, Union, SymmetricDifference 등이 있다.



(그림 1) 3D 공간 관계 연산 및 집합 연산

[7] 에서는 [6]의 8 가지 2D 위상관계 연산을 3D 공간 객체에 적용하였으며, [8]에서는 공간 분석 연산 중 3D 공간 객체의 집합 연산을 구현하였다. (그림 1)에서는 3D 공간 관계 연산과 집합 연산을 나타내고 있다.

[1]에서는 공간객체의 표현을 위한 자료구조 WKS(Well Known Structure)를 정의하고 있으며 DBMS 에서 객체의 제공은 이러한 자료구조를 이용하여 클라이언트에 전달하게 된다. WKS 에는 텍스트 형태의 WKT(Well Known Text)와 바이너리 형태의 WKB(Well Known Binary)가 있으며 PolyhedralSurfaceZ 와 TINZ 에 대한 정의를 다음과 같이 하고 있다. PolyhedralSurfaceZ 의 경우 자료구조상에서는 3D Polygon 의 Set 으로 구성되어 있지만, 모든 Polygon 이

같은 방향(Orientation)으로 되어 있어야 하며, 모든 Polygon 이 연결(patch)되어 있어야 하는 제약조건을 가진다. TINZ 의 경우 마찬가지로 3D Polygon 의 Set 으로 되어 있지만 3 각형으로 구성되어 있는 Polygon 이어야 하며 제약 조건은 PolyhedralSurfaceZ 와 동일하다.

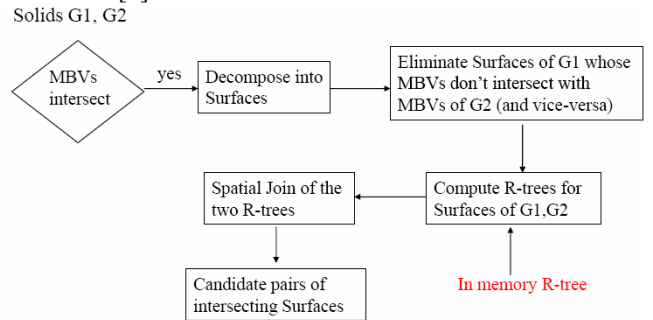
```

WKBPolyhedralSurfaceZ {
    byte                byteOrder;
    static uint32      wkbType=1015;
    uint32             numPolygons;
    WKBPolygonZ       polygons[numPolygons];
}

WKBTINZ {
    byte                byteOrder;
    static uint32      wkbType=1016;
    uint32             numPolygons;
    WKBPolygonZ       polygons[numPolygons];
}
    
```

3.2 질의처리

DBMS 에서 공간 데이터 질의 처리는 일반적으로 공간 색인에 의한 1 차 여과(Filtering) 단계, 공간 연산을 이용한 2 차 정제(Refinement) 단계를 거치게 된다. 일반적으로 주요한 공간 질의는 크게 Windows 질의 K-Nearest Neighbor 질의, 공간 Join 질의가 있다. 각각의 질의는 1 차 여과 단계를 거쳐 후보 객체 집합을 추출하고 2 차 정제 단계를 거쳐 최종 객체 집합을 반환한다. 3D 공간 질의 처리의 경우도 마찬가지로 여과-정제단계와 동일한 과정을 거쳐서 수행된다. 하지만, 3D 공간객체의 경우 2D 공간객체에 비해 객체의 구성 요소인 Surface 또는 Face 가 많기 때문에 별도 처리하는 방법이 필요할 수도 있다. Oracle 의 경우 3D 공간객체 질의처리를 위하여 (그림 2)와 같은 단계를 거친다[5].



(그림 2) Oracle 의 3D Window 질의처리

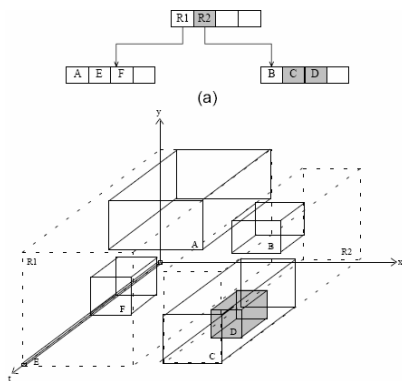
공간 색인에 의한 MBV(Minimum Bounding Volume) 여과 단계를 거친 후 후보 객체를 Surface 객체로 분할한다. 이 분할된 객체 중 MBV 와 intersect 하지 않는 surface 를 제외하고 나머지 surface 에 대해 메인메모리 공간색인(R-tree)을 실시간 구축하여 공간 조인을 수행함으로써 최종 객체를 찾게 된다. 3D Solid 를 구성하는 Surface 가 많을 경우 이 처리 방법은 효과적이다.

3.3 저장 및 색인

DBMS 에서 레코드는 여러 필드로 구성되며 객체 페이지의 슬롯(Slot)에 저장한다. 하나의 필드의 데이터가 큰 객체인 경우 페이지 오버플로우가 발생할 수 있어 체인으로 연결하여 관리한다. 3D 공간 객체인

Solid 또는 PolyhedralSurface 의 경우 하나의 필드, 즉 데이터 페이지의 하나의 슬롯내에 저장하는 방법을 적용할 수 있으나, 지형의 자료형인 TIN 또는 Point Cloud 는 이러한 방법을 이용하여 저장할 수가 없다. 이유는 지형의 경우 데이터의 크기가 아주 크기 때문이다. 따라서 별도 저장/관리하는 방법이 필요하다. Oracle 의 경우 이러한 지형 데이터를 3D 공간 영역을 3D 격자(Grid)로 분할하여 저장한다[5]. 분할된 데이터는 각각의 MBV 를 포함하고 있어 색인에 의한 여과를 할 수 있다. Point Cloud 의 경우 이 방법이 효과적일 수 있으나, TIN 의 경우에는 전체 연결성을 보장할 수 없는 문제가 발생한다.

공간 색인은 공간 객체의 빠른 검색을 위하여 사용한다. 가장 많이 사용되는 공간 색인 R*-Tree 가 있다. 이 색인은 다차원 공간 데이터의 색인으로 적용이 가능하며, 2D 의 경우 MBR(Minimum Bound Rectangle)이 트리 노드의 key 이며 3D 의 경우 MBV(Minimum Bounding Volume)가 key 이다. (그림 3)은 3D 공간색인(3DR-tree)를 보여주고 있다[9].



(그림 3) 3D R*-Tree

3DR-tree 는 height-balanced 트리이며 key 인 MBV 의 결합이 최소가 되도록 공간 클러스터링을 보장한다. 따라서 공간 질의 여과단계에 디스크 I/O 를 최소가 되도록 성능을 보장할 수 있다.

3.4 인터페이스

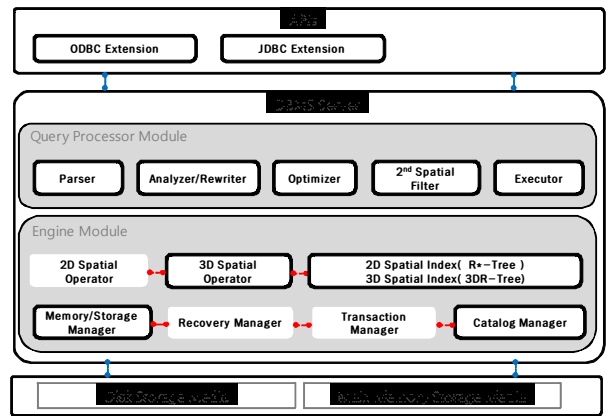
DBMS 의 인터페이스는 일반적으로 표준 인터페이스인 ODBC/JDBC 를 따른다. 3D 공간 SQL 을 확장하기 위해서는 기존 SQL 에 3.1 의 데이터 모델의 3D 공간 자료형 및 공간 연산자가 반영되도록 확장해야 한다. 최근 데이터의 교환 포맷으로 XML 을 이용한 다양한 규격이 정의되고 활용되고 있으며 산업계에서 상호 운용성을 위하여 많이 요구되고 있다. XML 기반의 공간 데이터의 교환 포맷은 GML, KML 그리고 CityGML 등이 있으며, DBMS 에서 웹 서비스를 이용하여 이러한 형태로 제공이 가능하다.

4. 3D 공간 DBMS 기술 개발

본 연구에서는 기존 2D 공간 DBMS 를 이용하여 3D 공간 DBMS 로 확장하는 연구를 수행하고 있다. 본 연구에서의 주요 접근 방법은 대용량 3D 공간데이터의 저장 관리를 위한 디스크 기반 3D 공간

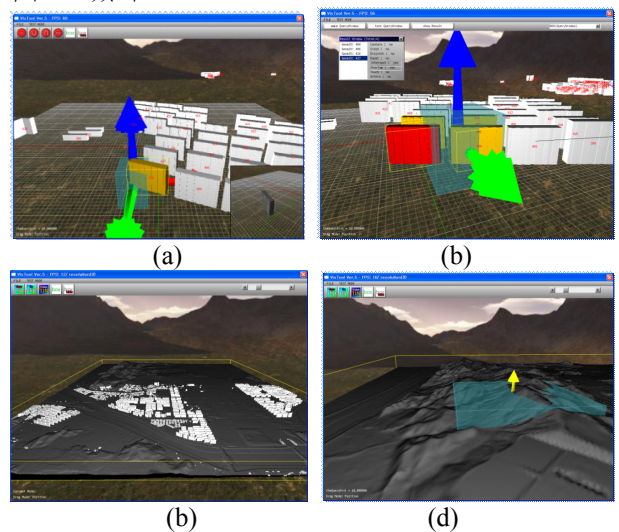
DBMS 와 실시간 3D 공간 데이터 처리를 위한 메인 메모리 기반 3D 공간 DBMS 를 연구 개발하고 있다. 디스크와 메인메모리는 저장 매체로서의 특성이 다르기 때문에 저장/색인/질의처리 등의 프로그래밍 기법이 다르다. 디스크의 경우 성능에 가장 큰 영향을 끼치는 Disk I/O 를 최소화하기 위한 접근 방법으로 개발해야 하며, 메인메모리의 경우 Disk I/O 가 없기 때문에 효과적인 질의처리 및 메모리 관리에 초점을 맞춰 개발해야 한다.

본 연구에서 수행하고 있는 DBMS 의 구조는 (그림 4)와 같다. 박스로 표현된 부분이 3D 공간 DBMS 로 확장하기 위한 부분이다. 크게 API 부분과 질의처리 모듈, 저장 엔진 모듈의 전반적인 부분을 3D 공간 객체를 수용하기 위한 확장이 필요하다.



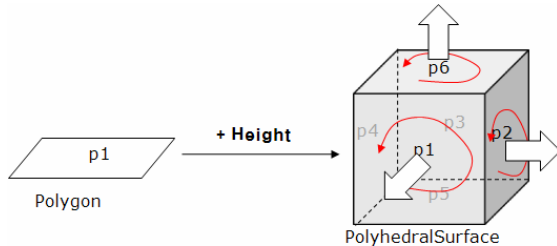
(그림 4) 3D 공간 DBMS 확장 구조

본 연구에서는 [1]의 공간 데이터 모델을 참조하고 있다. 지원되는 3D 객체 자료형은 Polyhedral Surface 와 TIN 이 있으며 공간 연산자로는 3.1 절의 8 가지 위상관계 연산과 4 가지 분석(Set) 연산을 지원한다. 그리고 TIN 의 경우 클리핑 연산을 지원한다. (그림 5)는 본 연구에서의 이러한 연산의 수행 결과를 보여주고 있다.



(그림 5) (a) 3D Set 연산, (b) 3D 공간관계 연산, (c) TIN 시각화, (d) TIN Clipping 연산 실행 화면

본 연구에서의 3D 데이터 생성 및 변환은 지형의 경우 LiDAR (Point Cloud, DEM) 데이터를 이용하여 DBMS 에서 TIN 을 생성한다. 주로 건물인 3D 시설물은 2D Polygon 객체에 높이정보를 이용하여 3D 객체로 extrusion 하여 PolyhedralSurface 객체를 생성한다(그림 6). 뿐만 아니라 3D 그래픽 포맷으로 가장 많이 사용하는 3DS 포맷에 대한 변환도 가능하다.



(그림 6) 2D Polygon Extrusion

현재 3D 공간 DBMS 기술 개발은 진행 중에 있으며, 개발이 완료될 경우 다양한 요소에 대한 성능 분석이 수행 될 예정이다.

5. 결론 및 향후 연구과제

본 연구에서는 기존 2D 공간 DBMS 를 3D 공간 DBMS 로 확장하기 위한 요소 기술들에 대해서 살펴 보았다. 고려해야 할 요소들로는 데이터 모델, 즉 어떤 3D 자료형과 연산을 지원할 것인지를 고려해야 하고 질의처리 방법, 저장 방법, 색인 방법, 인터페이스 방법이 고려되어야 한다. 그리고, 현재 개발 진행 중인 본 연구에서의 DBMS 구조와 자료형 및 연산자에 대하여 살펴보았다.

향후 연구과제로는 현재 연구개발이 진행되고 있는 단계이므로 본 연구개발이 종료되는 시점에 타 제품과 본 연구개발 결과물인 디스크 기반, 메인메모리 기반 3D 공간 DBMS 의 성능을 분석하여야 한다.

감사의 글(Acknowledge)

본 연구는 국토해양부 첨단도시기술개발사업 - 지능형국토정보기술혁신 사업과제의 연구비지원(07 국토정보 C05)에 의해 수행되었습니다.

참고문헌

- [1] OGC Implementation Specification for Geographic Information-Wimple feature access – Part1 Common Architecture v1.2.0, <http://www.opengeospatial.org>
- [2] OGC Geography Markup Language (GML) v3.2.1, <http://www.opengeospatial.org>
- [3] OGC City Geographic Markup Language(CityGML) Encoding Standard v1.0.0, <http://www.opengeospatial.org>
- [4] OGC KML(formerly Keyhole Markup Language) v2.2.0, <http://www.opengeospatial.org>
- [5] Siva Ravada, Baris M. Kazar, Ravi Kothuri, “Query Processing in 3-D Spatial Databases: Experiences with Oracle Spatial 11g”, 3rd International Workshop on 3D Geo-Information, November 13-14 2008, Seoul, South

Korea

- [6] Egenhofer, M.J., Herring, J.R., 1990, “A mathematical framework for the definition of topological relationships”, Proceedings of Fourth International Symposium on Spatial Data Handling, Zurich, Switzerland, pp.803-813
- [7] Siyka Zlatanova, Alias Abdul Rahman and Wenzhong Shi, “Topological models and frameworks for 3D spatial object”, Computers & Geosciences 30, pp.419-428, 2004
- [8] Chen Tet-Khuan, Alias Abdul-Rahman and Sisi Zlatanova, “3D Spatial Operations in Geo DBMS Environment for 3D GIS”, LNCS 4705, Part I, pp. 151-163, 2007.
- [9] Qing Zhu, Jun Gong, Yeting Zhang, “An efficient 3D R-tree spatial index method for virtual geographic environments”, ISPRS Journal of Photogrammetry & Remote Sensing, pp. 217-224, 2007.