

레이저레이더 시뮬레이션을 위한 3차원 공간DB 설계

3D Spatial Database Design for Laser Radar Simulation

김근한*, 김혜영, 전철민

Geun-han Kim*, Hye-young Kim, Chul-min Jun

서울시립대학교 공간정보공학과

{nani0809*, mhw3n, cmjun}@uos.ac.kr

요약

3차원 오브젝트의 위치 및 정보 획득을 위한 레이저레이더 시뮬레이션의 성능을 향상시키기 위해서는 시뮬레이션의 결과로 획득되는 공간의 범위와 해당 사물의 정보를 정확하고 빠르게 획득해야 한다. 본 연구에서는 이러한 레이더 시뮬레이션의 성능을 향상시키기 위해서 3차원 공간 데이터를 공간DB에 저장하고, 질의를 수행하여 해당 3차원 오브젝트의 정보를 효과적으로 추출해내는 방법론을 제시하였다. 이를 위해 본 연구에서는 시뮬레이션에서 사용되는 3차원 지형지물(지형, 건물, 사물 등) 모델의 정보를 데이터모델링을 통해 토폴로지 형태를 갖도록 하였으며, 이를 공간DB에 저장하고, 레이저 신호와의 연산 쿼리를 시행하는 과정을 예시하였다. 이러한 과정을 구현하기 위하여 OGC 기반의 공간 데이터 타입, 함수, 인덱스들을 제공하는 PostgreSQL과 PostGIS를 사용하였다. 지형, 건물, 탱크 등 이렇게 세 가지의 범주의 사물로 나누어 각각을 공간DB로 구현, 쿼리를 실시하였다. 지형정보는 TIN을 사용하였고, 건물의 좌표 값들은 도화원도에서 추출하였으며, 탱크와 같은 사물은 VRML 모델의 좌표 값을 사용하였다.

1. 서론

선진국에서는 기존의 일반 탐색기와 달리 3차원 정보를 제공해 주는 Ladar 탐색기를 차세대 정밀타격 유도무기의 센서로 이용하는 관련 개발, 연구가 활발하게 진행되고 있다. 이러한 Ladar 탐색기를 장착한 유도무기의 표적인식 알고리즘을 개발하고 실험하기 위하여 실제로 Ladar 탐색기가 장착된 유도무기를 사용한다면 천문학적인 비용을 감수해야 하고 다양한 표적에 대하여 시험하기는 매우 어렵다. 따라서, Ladar탐색기를 사용하여 표적을 촬영한 것처럼 시뮬레이션 할 수 있는 컴퓨터 환경을 구축한다면, 매우 다양한 환경, 표적들에 대하여 시험이 가능하며, 유도무기, Ladar탐색기, 표적인식 알고리즘 등의 개발과 시험을 용이하게 할 수 있다. 따라서 본 연구에서는 표적인식 시뮬레이

션의 한 부분인 3D 지형 및 표적 모델링의 기능을 제공하는데 중점을 두었다. 시뮬레이션의 성능을 향상시키기 위해서 3차원 지형지물(지형, 건물, 사물 등)에 대한 데이터 모델링을 통해 토폴로지 형태를 갖도록 구조화 하였다. 이를 공간 DBMS를 이용하여 구축하고, 공간 객체의 위치와 속성정보를 공간DB에 저장, 질의를 수행하여 해당 3차원 오브젝트의 정보를 효과적으로 전달해주는 방법론을 제시하였다. 이러한 과정을 구현하기 위하여 공간 데이터 타입, 함수, 인덱스들을 제공하는 PostgreSQL과 PostGIS를 사용하였다. 본 연구는 시뮬레이션을 위해 제작한 3차원 지형지물 모델 정보를 위한 DB를 설계하고, PostgreSQL, PostGIS와 file stream을 사용하여 모델의 정보를 DB에 저장하고, DB에 저장된 정보를 레이저 신호와의 연산 쿼리를 시행하는 과정을 예

시하였다. 마지막으로 요약, 한계, 향후 연구 등을 언급하면서 마무리 하였다.

2. 연구 실험 데이터 획득

본 연구에서는 지형, 건물, 사물(예: 탱크) 이렇게 크게 3가지 범주의 데이터로 나누어 실험하였다. 지형과 건물은 영등포구의 일부 지역을 테스트 공간으로 선택하였다. 지형정보는 수치지도의 등고선과 표고점 레이어를 선택하여 지면의 고도값을 추출한 후, TIN을 생성하였으며, 건물의 좌표 정보는 도화원도에 포함된 지면 및 건물의 지붕면 모델 및 벽면 모델의 좌표를 자동으로 추출 생성하였다. 탱크와 같은 사물의 정보는 VRML의 모델의 좌표값을 사용하였다. 추출된 좌표 정보들은 텍스트 파일에 따로 저장하여 DB에 정보들을 삽입할 때 사용하였다.

3. DB 설계

Oracle, IBM DB2, Informix, Ingres와 비 상업화 DBMS인 PostgreSQL과 MySQL과 같은 거의 대부분의 DBMS는 공간 데이터 타입과 공간 함수들을 OGC의 SQL을 위한 Simple Features Specification을 기반으로 더 확장하거나, 축소하여 구현하고 있다. 그리고 포인트, 라인, 폴리곤과 같은 공간형상의 저장, 복원, 질의, 업데이트와 같은 기능들을 제공한다. 현재 오픈소스로 제공되는 공간 DBMS 중 PostgreSQL/PostGIS는 비교적 안정적인 성능 및 기능을 지원하는 것으로 평가되고 있다. 따라서 본 연구에서는 사물의 3차원 위치 정보 및 속성정보를 저장, 쿼리하는데 PostgreSQL/PostGIS를 사용하였다.

본 연구에서 시뮬레이터에 전달해야 할 정보는 면 또는 다면체의 위치 정보와 해당 객체의 속성 정보이다. 따라서 각 다면체와 이를 이루는 면들 간의 관계가 토폴로지를 형성되도록 데이터 모델링을 하였다. 건물과 탱크는 하나의 다면체와 다면체를 이루고 있는 면으로 구성하였고, 지형 정보는 별도로 TIN을 기반으로 지형 표면이 면들로 구성되도록 하였다. 그림1은 하나의 빌딩과 빌딩을 이루고 있는 면들 사이의 관계를 나타낸 UML class diagram이다. 면 객체는 building의 foreign key를 가지고 있다.

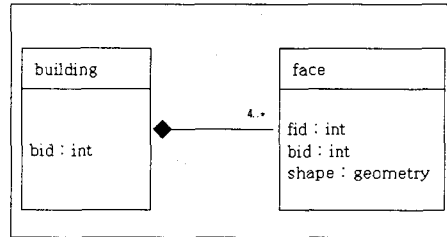


그림1. 건물의 UML class diagram

```

Create table Building(
    bid int PRIMARY KEY );

Create table face (
    fid int PRIMARY KEY,
    bid int,
    shape geometry);
  
```

그림2. 건물 생성 SQL 정의

표 1. 건물이 3D cube 일 때 face 테이블

FACE table		
FID	BID	SHAPE
1 (lower face)	1	polygon((x4,y4,z4, x3,y3,z3, x2,y2,z2, x1,y1,z1, x4,y4,z4))
2 (side 1)	1	polygon((x3,y3,z3, x4,y4,z4, x8,y8,z8, x7,y7,z7, x3,y3,z3))
3 (side 2)	1	polygon((x4,y4,z4, x1,y1,z1, x5,y5,z5, x8,y8,z8, x4,y4,z4))
4 (side 3)	1	polygon((x1,y1,z1, x2,y2,z2, x6,y6,z6, z5,y5,z5, x1,y1,z1))
5 (side 4)	1	polygon((x3,y3,z3, x2,y2,z2, x6,y6,z6, z7,y7,z7, x3,y3,z3))
6 (upper face)	1	polygon((x5,y5,z5, ,x6,y6,z6, x7,y7,z7, z8,y8,z8, x5,y5,z5))

4. 공간 객체의 DB 저장

본 연구에서 사용된 지형, 건물, 탱크와 같은 사물들은 수천에서 수 만개의 면들로 이루어진다. 이 정보들을 직접 SQL 명령어만을 이용하여 DB에 입력하기에는 수작업으로 거의 불가능하다. 따라서 본 연구에서는 텍스트 파일과 VRML 파일에 저장된 텍스트 정보를 자동으로 읽고, DB에 저장하는 방법을 선택하였다. 텍스트 파일과 VRML의 포맷에 맞도록 각각 변환하여 DB테이블에 저장하였다.

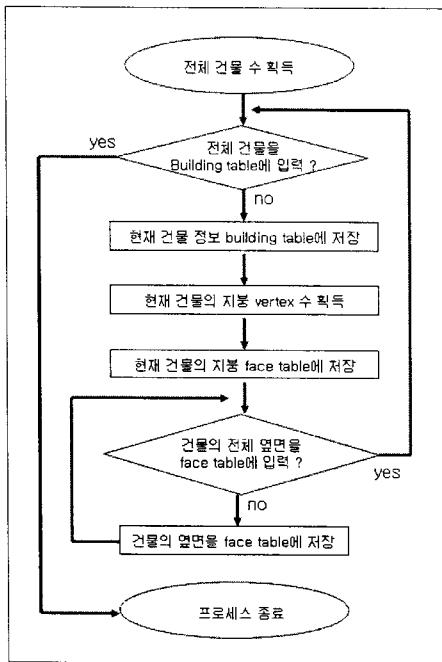


그림3. DB의 building 및 face 테이블의 text 파일 정보 저장 프로세스

5. 공간 DBMS 상에서의 질의

PostGIS에서는 객체의 좌표 값은 3차원으로 저장을 할 수 있지만, 실제로 3차원 좌표에 대한 연산은 2차원 값으로 제공한다. 따라서 3차원 좌표 값을 함수에 대입하고 쿼리문을 실행하여도 결과물은 2차원의 좌표 값을 적용한 것과 같은 결과를 얻는다. 실제 표적인식 시뮬레이션의 기능을 향상시키기 위해서는 건물에서 레이저 신호에 해당하는 벽면을 결과 값으로 시

뮬레이션에 전달해 줘야 하지만 PostGIS에서 기본적으로 제공하는 함수의 제약으로 인해 면 단위의 정보가 아닌 건물단위의 정보를 전달할 수 밖에 없다. 시뮬레이션에서 기본적으로 레이저 신호의 시작점과 끝점의 좌표를 알고 있는 직선이 주어지면 해당 직선과 만나는 건물의 정보를 시뮬레이션에 전달하게 된다. 다음 그림4는 쿼리문 중에서 조건절인 WHERE절을 나타낸다. ST_Touches, ST_Crosses 함수를 사용하면 시작점과 끝점의 좌표 값이 주어진 직선과 face 테이블에 저장된 폴리곤 중에서 이와 만나는 폴리곤의 정보를 획득할 수 있다. 현재 PostGIS에서는 건물 소속 Polygon들로 모두 다 리턴하므로 교차하는 해당 건물정보를 추출하는 것과 같은 효과가 된다.

```

WHERE ST_Touches(GeomFromEWKT('face.Shape'), GeomFromEWKT('linestring(0.5 0.5 100,10 10 100)'))=true;

WHERE ST_Crosses(GeomFromEWKT('face.Shape'), GeomFromEWKT('linestring(50 50 0,50 100 0 0)'))=true;
  
```

그림4. 쿼리문에서의 ST_Touches, ST_Crosses 함수의 사용 예

6. 결론

레이저레이더 시뮬레이션의 성능을 향상시키기 위해서는 시뮬레이션의 결과로 획득되는 공간의 범위에 포함되는 3차원 오브젝트의 위치 및 속성 정보를 정확하고 빠르게 획득해야 한다. 따라서 본 연구에서는 시뮬레이션을 위해 제작한 지형, 건물, 탱크와 같은 3차원 지형지물의 위치정보와 속성 정보를 DB에 저장하고, 질의하여 활용할 수 있도록 DB를 설계하였다. 그리고 각각 객체의 정보들을 file stream 프로그램을 통하여 DB에 저장하였으며, PostGIS에서 제공하는 함수를 이용하여 질의하는 과정을 통하여 위치 및 속성 정보를 정확하고 빠르게 획득하려 하였다. 하지만 PostGIS에서 기본적으로 제공하는 함수가 2차원의 좌표만을 지원함으로써 실제 획득 가능한 객체의 정보

는 제한적이었다. 따라서 레이저레이더 시뮬레이터의 기능을 향상시키기 위해서 앞으로 3차원 좌표까지 지원할 수 있는 함수를 개발하는 것과, 더 나아가 DB에 저장된 객체들의 위치정보를 활용하여 실제 가시화를 하는 것을 추후 연구로 수행할 예정이다.

김성준, 이임평, 민성홍, 이동천, 박진호, 2007. 도화원도를 이용한 3차원 건물모델의 자동생성, 한국지형공간정보학회지

감사의 글

연구는 한국과학기술원 영상정보특화연구센터를 통한 방위사업청과 국방과학연구소의 연구비 지원으로 수행되었습니다. (계약번호 UD070007AD)

참고문헌

Zlatanova, S., 2006, 3D geometries in spatial DBMS, Innovations in 3D Geo Information Systems, pp 1-14

Arens, C., J.E. Stoter, and P.J.M. van Oosterom, 2005, Modelling 3D spatial objects in a geo-DBMS using a 3D primitive, In: Computers & Geosciences, Volume 31, 2, pp. 165-177

Stoter, J.E., Zlatanova, S., 2003. Visualising and editing of 3D objects organised in a DBMS. In: Proceedings EUROS DR Workshop: Rendering and Visualisation, January 2003, Enschede, The Netherlands, 14pp

RICHARD BLUM, 2007, PostgreSQL 8 for Windows, The McGraw-Hill Companies

Paul Ramsey, PostGIS Manual, version 1.3.2

PostGIS (2008), URL: postgis.refractor.net, 2008