

# 레이저레이더 시뮬레이션을 위한 3차원 데이터 모델의 성능 테스트 Performance Tests of 3D Data Models for Laser Radar Simulation

김근한\* · 김혜영\*\* · 전철민\*\*\*

Kim, Geun Han · Kim, Hye Young · Jun, Chul Min

## 要 旨

레이저레이더 탐색기를 장착한 유도무기를 개발하고 실험하기 위하여 실제로 레이저레이더 탐색기가 장착된 유도 무기를 사용하기는 현실적으로 어렵다. 따라서 레이저레이더 탐색기를 사용하여 표적을 촬영한 것처럼 시뮬레이션 할 수 있는 컴퓨팅 환경을 구축해야 하며, 이를 위해서는 지표의 지형, 지물에 대한 3차원 모델이 필요하다. 또한 레이저레이더 시뮬레이션은 대용량 데이터를 다뤄야 하고 벽면의 재질이나 반사도와 같은 속성을 저장, 검색, 분석하여야 한다. 그러나 현재 사용되고 있는 대부분의 3차원 모델은 파일 기반의 데이터 구조를 가지고 있으며 토폴로지 구조를 가지고 있지 않고, 가시화에 중점을 두고 있다. 따라서 본 연구에서 테스트하고자 하는 레이저레이더 시뮬레이션에는 연산 속도나 가시화 성능 문제로 인하여 기존의 3차원 모델을 적용하기에는 한계가 있다. 이와 같은 문제를 해결하기 위해 본 연구에서는 공간 데이터베이스의 사용과 이를 위한 3차원 데이터 모델링 기법을 제시하였으며 이를 레이저레이더 시뮬레이션에 적용하였다. 기존에 연구되었던 3차원 모델은 서로 다른 목적에 맞도록 모델링 되었기 때문에 어떠한 모델이 레이저레이더 시뮬레이션에 가장 적합한 모델인지 알 수 없다. 따라서 본 연구에서는 레이저레이더 시뮬레이션을 위한 네 가지 종류의 3차원 데이터 모델을 정의하고 각각의 모델들에 대하여 성능 테스트를 수행하여 테스트 결과를 획득하였다. 네 가지 모델 중 레이저레이더 시뮬레이션에 가장 적합하다고 판단되는 Body-Face 모델을 선정하였다.

핵심용어 : 3차원 데이터 모델, 성능 테스트, 레이저레이더 시뮬레이션, 공간 데이터베이스, PostGIS

## Abstract

Experiments using real guided weapons for the development of the LADAR(Laser radar) are not practical. Therefore, we need computing environment that can simulate the 3D detections by LADAR. Such simulations require dealing with large sized data representing buildings and terrain over large area. And they also need the information of 3D target objects, for example, material and echo rate of building walls. However, currently used 3D models are mostly focused on visualization maintained as file-based formats and do not contain such semantic information. In this study, as a solution to these problems, a method to use a spatial DBMS and a 3D model suitable for LADAR simulation is suggested. The 3D models found in previous studies are developed to serve different purposes, thus, it is not easy to choose one among them which is optimized for LADAR simulation. In this study, 4 representative 3D models are first defined, each of which are tested for different performance scenarios. As a result, one model, "Body-Face", is selected as being the most suitable model for the simulation. Using this model, a test simulation is carried out.

Keywords : 3D data model, performance test, Laser radar simulation, spatial databases, PostGIS

## 1. 서 론

레이저레이더 시뮬레이션에서는 가상의 비행물체에 서 레이저신호를 지상의 객체에 송수신하여 3차원 표

적의 속성 및 위치정보를 획득한다. 그리고 시뮬레이션 에서 이용되는 실험 범위는 광범위하고 실험 객체들도 복잡한 형태를 지니며 객체의 벽면 재질이나 반사도와 같은 속성을 저장, 검색, 분석 하여야 한다. 하지만 일

2009년 8월 6일 접수, 2009년 9월 8일 채택

\* 서울시립대학교 대학원 공간정보공학과 석사(nani0809@uos.ac.kr)

\*\* 정희원 · 서울시립대학교 대학원 공간정보공학과 박사과정(mhw3n@uos.ac.kr)

\*\*\* 교신저자 · 정희원 · 서울시립대학교 공간정보공학과 교수(cmjun@uos.ac.kr)

반적으로 사용하는 3차원 모델링 툴들은 파일기반 구조를 가지며 객체들을 한꺼번에 메모리에 올려놓고 가시화를 하게 되어 객체수가 많아지거나 광범위한 지역을 다룰 때에는 메모리 오버플로우(overflow) 문제가 발생하거나 시뮬레이션 속도가 느려지는 문제가 발생할 수 있다. 또한 시뮬레이션 과정 중 비행물체에서 발생하는 레이저신호와 교차하는 표적을 빠르게 추출해야 하는데 현재 일반적으로 활용되고 있는 3차원 모델은 파일단위로 구성, 관리되며 토폴로지 구조를 가지고 있지 않아 3차원 공간분석 및 응용에 있어 한계점이 있다(김근한, 2008).

이와 같은 문제점을 해결하기 위해 본 연구에서는 3차원 모델의 저장을 위해 공간 데이터베이스를 활용하였다. 공간 데이터베이스를 이용하여 건물 내 각 공간의 공간정보와 속성 정보를 테이블 단위로 저장하면 데이터베이스를 통한 빠른 질의와 연산 및 분석이 가능해지고, 공간 정보의 변경 및 유지 관리가 용이해진다. 또한 데이터베이스로부터 추출한 공간 데이터를 2차원 및 3차원으로 가시화할 수도 있다.

이에 본 연구에서는 레이저레이더 시뮬레이션에 공간 데이터베이스를 적용할 수 있는 3차원 데이터 모델을 선정하기 위해 우선 네 가지 종류의 3차원 데이터 모델을 정의하고 각 데이터 모델에 대하여 공간 데이터베이스를 기반으로 데이터를 구축한 후 성능 테스트를 수행하였다. 그리고 이들 모델에 대한 테스트 수행 결과를 비교 분석하여 레이저레이더 시뮬레이션에 효과적으로 적용될 수 있는 3차원 데이터 모델을 선정하였다.

## 2. 연구방법의 개요

본 연구에서는 레이저레이더 시뮬레이션을 위한 실외 공간 개체의 효과적인 3차원 시각화 및 공간 연산을 위하여 공간 데이터베이스를 활용하였으며 레이저레이더 시뮬레이션에 가장 적합한 3차원 데이터 모델을 제시하였다. 이를 위하여 기존의 3차원 데이터 모델들과 레이저레이더 시뮬레이션의 요구사항들을 분석한 후 네 가지의 3차원 데이터 모델을 제시하였다. 그리고 이들 3차원 데이터 모델에 대한 다양한 성능 테스트를 시행하였다. 3차원 데이터 모델에 대한 성능 테스트 항목은 다음과 같다.

- 3차원 가시화의 속도
- 공간 연산이 포함되지 않는 쿼리 획득 속도
- 공간 연산이 포함된 쿼리 획득 속도
- 레이저 신호의 range 범위에 따른 쿼리 획득 속도

이와 같은 테스트들을 수행한 후 획득되어진 3차원 데이터 모델들의 성능 테스트 결과와 3차원 객체의 데이터베이스로의 저장 용이성, 다양한 공간 연산의 활용 가능 여부 등의 사항을 비교 분석하여 레이저레이더 시뮬레이션에 가장 적합한 3차원 데이터 모델을 선정하였다. 또한 선정된 3차원 데이터 모델을 기반으로 공간 데이터베이스를 이용하여 실제 영등포구 지역을 대상으로 하여 데이터를 저장하였다. 그리고 저장된 데이터를 이용하여 레이저레이더 시뮬레이션을 위한 3차원 가시화 및 공간 연산 테스트를 수행하였다.

본 연구에서는 공간 데이터베이스로서 PostgreSQL (PostGIS)을 사용하였고, 3차원 가시화 라이브러리는 OpenGL을 활용하였으며 전체 인터페이스의 개발은 C#언어로 개발하였다.

## 3. Data Modeling

레이저레이더 시뮬레이션에 적용할 3차원 데이터 모델을 제시하기에 앞서 기존에 연구되었던 3차원 데이터 모델들을 분석하였다. 기존 연구에서 나타난 3차원 데이터 모델들은 대체로 다음과 같이 세 가지로 분류될 수 있다.

- A 타입. Body-Face-Edge-Node : 다면체와 다면체를 이루고 있는 면, 면을 이루고 있는 선, 선을 이루고 있는 점의 완전한 위상적 관계를 정의
- B 타입. Body-Face-Node : 다면체, 다면체를 이루고 있는 면, 면을 구성하고 있는 점의 관계를 정의
- C 타입. Body-Face : 다면체와 다면체를 이루고 있는 면들로 구성하여 관계를 정의

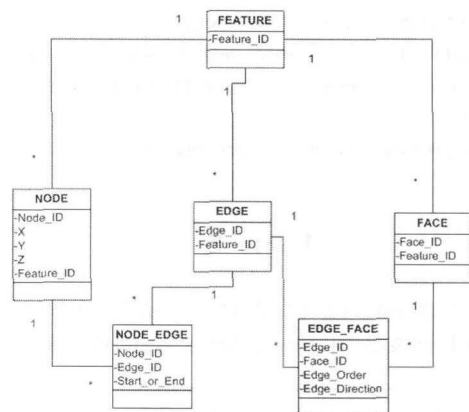


그림 1. A 모델의 UML 클래스 다이어그램

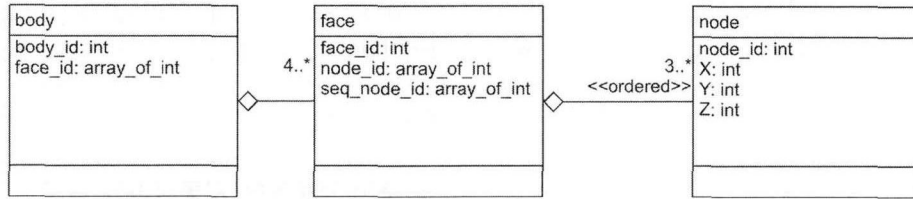


그림 2. B 모델의 UML 클래스 다이어그램

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

그림 3. C 모델의 다면체와 면을 구성하는 테이블

A 타입은 3차원 볼륨을 표현하는 방법으로서 B-Rep (Boundary Representation)이 많이 사용된다. B-Rep은 다면체를 면(Face), 선(Edge), 점(Node)을 사용하여 나타내며 이들 클래스간의 토폴로지 관계를 표현한다. 위에서 A 타입은 다면체를 정의하는데 이 세 가지 클래스를 모두 사용하여 완전한 토폴로지 관계를 정의하고 있다. 그림 1은 A 타입 모델을 UML 클래스 다이어그램으로 나타낸 것이다(Elul, 2008). A 타입은 공간 데이터베이스에서의 3차원 도시 모델을 표현하기 위한 연구(Gröger, 2004)에서도 유사하게 적용하고 있다.

B 타입은 A 타입에서 라인 클래스를 제외한 데이터 모델로서 지오메트리는 A 타입과 동일한 방법으로 최종 클래스인 Node에 저장된다. B 타입은 Zlatanova (2000)가 제시한 데이터 모델을 예로 들 수 있다. Zlatanova는 3차원의 객체를 데이터베이스에 구현하기 위해서 그림 2와 같은 데이터 모델을 제시하였다. 이러한 데이터 모델링 기법은 3차원 공간 객체를 2차원 공간 DBMS로의 통합 연구(Stoter, 2002)와 공간 DBMS에서의 3차원 공간 객체의 모델링에 관한 연구(Arens, 2005), 공간 데이터베이스에서의 지오메트리아 토폴로지 비교한 3차원 공간연산에 관한 연구(Chen, 2008) 등 최근의 연구에서도 찾아볼 수 있다.

C 타입은 DBMS를 활용한 3차원 객체들의 가시화 (Stoter, 2003)의 연구에서 활용되었다. 그림 3과 같이 하나의 다면체를 다면체와 이를 이루고 있는 면들로만 구성하여 다면체를 정의하였다. 하나의 면에는 면을 구성하는 점의 좌표 값들을 포함한 지오메트리가 저장된다.

이와 같이 기존의 연구들은 3차원 객체를 위한 데이터 모델을 제안하였다. 이들 모델들은 저자들이 서로 다른 목적에 적용하기 위하여 제안되었기 때문에 모델 간의 우열관계를 판단내릴 수 없다. 본 연구에서는 기존 연구에서 제시한 3차원 데이터 모델들을 기반으로 레이저레이더 시뮬레이션을 위한 네 가지의 3차원 데이터 모델을 정의하고, 이 모델들에 대하여 테스트를 수행하였다. 또한 테스트 결과 값을 비교 분석하여 레이저레이더 시뮬레이션의 요구사항에 부합하는 최적의 데이터 모델을 제시하였다.

#### 4. 레이저레이더 시뮬레이션

레이저레이더란 레이저 신호를 이용하여 대상의 정밀한 3차원 거리를 획득하는 센서로 레이저의 주요 응용기술 가운데 하나이다. 초기의 레이저레이더는 레이저의 기술적 문제와 시스템의 복잡성으로 인해 개발에

많은 비용이 요구되었다. 그럼에도 불구하고, 다른 센서들과 달리 넓은 지역을 짧은 시간동안 정확하고 세밀하게 관측할 수 있다는 장점으로 인해 주목받아 왔다. 이러한 장점으로 레이저레이더는 선진국에서 국방 분야에서 장거리 유도 미사일이나 유무인 정찰기 등에 탑재되어 활용하기 위한 연구가 활발하게 진행되고 있다 (민성홍, 2008).

이러한 레이저레이더 탐색기를 장착한 유도무기의 표적인식 알고리즘 개발을 위하여 먼저 표적특성을 모의시험 할 수 있는 레이저레이더 시뮬레이션의 연구가 필요하다. 표적인식 알고리즘을 시험하기 위해서 실제로 레이저레이더 탐색기가 장착된 유도무기를 사용하면 천문학적인 비용을 감수해야 하고 다양한 표적에 대하여 시험하기는 매우 어렵다. 따라서 레이저레이더 시뮬레이션을 통하여 실제 표적을 촬영한 것처럼 시뮬레이션 할 수 있는 컴퓨터 환경을 구축한다면, 매우 다양한 환경, 표적들에 대하여 시험이 가능하며, 유도무기, 레이저레이더 탐색기, 표적인식 알고리즘 등의 개발과 시험을 용이하게 할 수 있다.

레이저레이더 센서의 원리 및 구성을 살펴보면 레이저레이더는 레이저 신호를 생성하여, FM 변조를 수행한 후, 표적을 향해 송신한다. 표적에 의해 반사된 신호는 레이저레이더의 광학계 및 검출기를 거쳐 디지털 데이터로 변환되고, 이를 처리하여 센서와 표적간의 거리(range) 및 표적의 반사도(intensity)를 복원한다. 레이저레이더는 레이저를 이용하여 표적의 반사도 및 3차원 입체정보를 복원할 수 있는 능동 센서이다. 특히 최근 들어 레이더 센서는 군수산업에 폭넓게 응용되고 있다.

레이저레이더 시뮬레이터는 실제 레이저레이더를 모의하는 장치이며 이를 위해 주어진 가상의 3차원 표적이 있을 때, 그 표적으로부터 반사된 신호를 자체적으로 생성할 수 있어야 한다. 여기서 주어진 표적이란 일반적으로 3차원으로 모델링 된 모형을 의미하며, 레이저레이더 센서 역시 3차원 공간상에 위치하게 된다.

외국에서는 레이저레이더 시스템뿐만 아니라 시뮬레이터에 관한 연구도 활발하게 이루어지고 있다. 예를 들면, 미국의 첫 번째 위성용 라이다 시스템인 GLAS (Geoscience Laser Altimeter System)의 개발에 이용된 NASA/BPRC 시뮬레이터나 Utah 주립 대학의 USU LadarSIM 등이 있으며, 주로 하드웨어의 설계와 성능평가 및 분석을 위해 활용되고 있다(민성홍, 2008). 하지만 국내에서는 아직 레이저레이더 탐색기와 관련 깊은 연구가 수행된 적이 없었다. 그러나 국내에서 지능형 정밀 유도 무기 개발의 필요성이 점점 더 증대되고 있는 추세이고 이의 개발 핵심 기술인 자동 표적 인식

시스템의 지속적인 개발, 검증 및 개선도 필수적이다. 레이저레이더 영상 탐색기는 이러한 시스템의 대표적인 입력 센서이며 이에 시뮬레이션의 수요도 급속히 증대되고 있다.

## 5. 레이저레이더 시뮬레이션 요구사항 분석

레이저레이더 시뮬레이션은 3차원의 표적에 레이저 신호를 발사하여 표적의 반사 성질을 획득하고, 사물의 반사도를 기반으로 표적의 정보를 획득하게 된다. 본 연구에서 테스트하고자 하는 레이저레이더 시뮬레이션은 다음과 같은 기능을 제공해야 한다.

- 공간 데이터베이스에 저장된 표적과 지형을 3차원으로 가시화
- 일정 범위의 레이저 신호와 대응하는 표적의 면 특성 정보를 공간 연산 쿼리를 이용하여 획득

가시화 과정에서는 공간 데이터베이스에 저장된 표적을 쿼리를 이용하여 해당 객체의 지오메트리를 획득하고 이를 이용하여 3차원으로 가시화 한다. 따라서 레이저레이더 시뮬레이션을 위한 3차원 데이터 모델은 지오메트리를 획득하는 속도가 빨라야 가시화 속도도 빠르게 된다. 뿐만 아니라 획득되는 지오메트리 형식도 가시화에 쉽게 사용될 수 있어야 한다.

3차원 표적의 속성정보의 획득은 공간 데이터베이스에 저장된 표적의 속성정보를 공간연산 쿼리를 이용하여 획득한다. 속성 획득을 위한 공간연산은 range 쿼리를 이용한다. 실제 레이저 신호는 비행물체에서 표적으로 직선으로 쏘아지고, 반사도와 같은 속성 값을 획득하며, 비행물체에서 표적으로 발사되는 레이저 신호는 매우 짧은 시간동안 수많은 레이저 신호가 발사된다. 예를 들어, 레이저레이더 탐색기의 한 종류인 Dragon Eye의 pulse frequency는 50,000-500,000 Hz이다 (Professional Surveyor Magazine, 2009). 이는 비행물체에서 1초에 50000에서 500000번의 레이저 신호를 발사하는 것인데 이를 시뮬레이션에 적용하기에는 어렵다. 따라서 본 연구에서의 시뮬레이션에서는 일정 시간동안 발사된 레이저 신호의 범위를 활용한다. Range의 범위는 1초 동안에 비행기가 움직인 이동거리와 레이저 신호의 발사범위를 계산하여 해당 범위를 획득하고, 이를 range 공간 연산 쿼리에 적용한다.

## 6. 3차원 데이터 모델 정의

본 연구에서는 선행연구에서 조사된 모델들을 기반

으로 하여 다음과 같이 네 가지의 모델을 레이저레이더 시뮬레이션을 위한 데이터 모델로 제시하였다.

- A 타입 : Body-Face-Edge-Node
- B 타입 : Body-Face-Node
- C 타입 : Body-Face
- D 타입 : Body-Face-Edge

그림 4와 같이 A 타입은 Full Topology를 제공하는 3차원 데이터 모델이다. 하나의 Body는 여러 개의 Face들로 구성되고, 하나의 Face는 여러 개의 Edge로 구성된다. 하나의 Edge는 두 개의 Node로 구성되게 된다. 하나의 Node는 여러 개의 Edge에 포함되고, 하나의 Edge는 여러 개의 Face에 포함되기 때문에 Node와 Edge, Edge와 Face는 다대다 관계가 성립된다. 따라서 Edge와 Node, Face와 Edge간의 인터섹션 테이블이 필요하게 되며 이는 표적의 데이터를 획득하기 위한 쿼리를 수행할 때 필요한 조인질의에 사용되게 된다. 지오메트리는 Node 테이블에 포인트의 형식으로 저장되며, 공간연산 쿼리는 이 포인트 지오메트리를 활용하여 연산된다. A 타입은 Body, Face, Edge, Node 테이블 각각에 속성 데이터를 저장할 수 있는 장점이 있는 반면, 표적의 데이터를 획득하기 위한 쿼리를 수행할 때 필요한 조인질의가 복잡해진다는 단점이 있다.

B 타입은 그림 5와 같이 A 타입에서 Edge 클래스가 생략된 데이터 모델이다. 본 모델에서의 지오메트리는 A 타입과 같이 Node 테이블에 포인트의 형식으로 저장되며, 공간연산 쿼리는 이 포인트 지오메트리를 활용하여 연산한다. Body, Face, Node 클래스 각각에 속성 자료를 저장할 수 있지만, Edge 클래스가 없기 때문에 Edge를 기반으로 하는 쿼리를 수행할 수 없다. 예를 들면 임의의 Edge를 포함하는 Face들은 검색할 수 없다.

C 타입은 그림 6과 같이 A, B 타입보다 더 축소된 모델이다. 지오메트리는 Face 테이블에 폴리곤의 형식으로 저장되며, 공간연산 쿼리는 이 폴리곤의 지오메트리를 활용하게 된다. Body-Face 모델은 Body와 Face 간에 인터섹션 테이블이 필요하지 않는 일대다 관계의 단순한 구조를 지니고 있다. 따라서 표적의 데이터를 획득하기 위한 쿼리를 수행할 때 필요한 조인질의가 단순하다는 장점을 가지고 있다. 하지만 라인, 포인트의 속성 자료를 저장할 수 없고, Edge와 Node를 활용한 쿼리는 수행이 불가능하다는 단점을 가지고 있다.

D 타입은 그림 7과 같이 A 타입에서 Node 클래스가 생략된 데이터 모델이다. 지오메트리는 Edge 테이블에 라인의 형식으로 저장되며, 공간연산 쿼리는 라인의 지오메트리를 활용하여 연산된다. 본 모델은 Node 클레

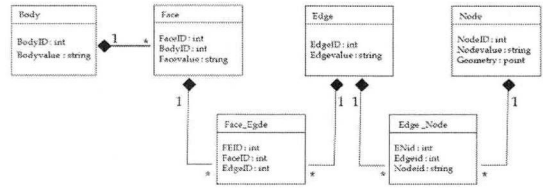


그림 4. Body-Face-Edge-Node의 3차원 데이터 모델

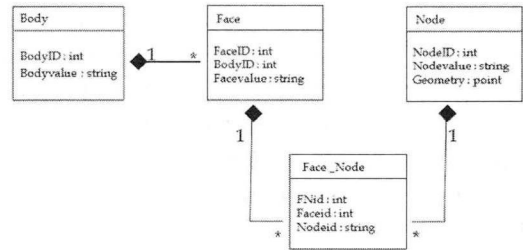


그림 5. Body-Face-Node의 3차원 데이터 모델

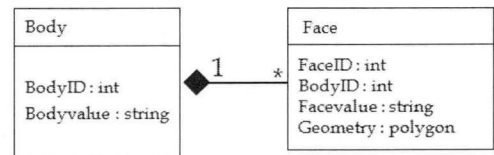


그림 6. Body-Face의 3차원 데이터 모델

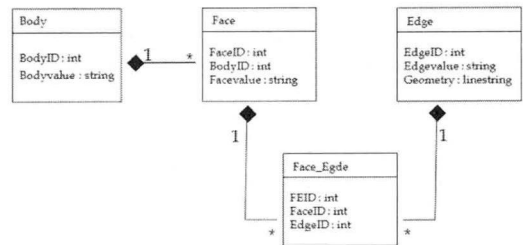


그림 7. Body-Face-Edge의 3차원 데이터 모델

스가 없기 때문에 Node를 활용한 쿼리는 수행이 불가능하다는 단점을 가지고 있다. 기존 선행연구에서는 이 모델과 같이 라인의 지오메트리를 가지는 3차원 데이터 모델이 파악되지 않았지만 본 연구에서는 Body-Face-Edge 데이터 모델을 다른 모델들과 함께 비교분석해 보고자 한다.

## 7. 3차원 데이터 모델의 성능 테스트

앞서 언급한 바와 같이 네 가지의 3차원 데이터 모델들에 대해 시뮬레이션의 요구사항 분석을 기반으로 다

음과 같은 성능 테스트를 수행하였다.

- 3차원 가시화를 위한 데이터 획득 속도
- 비 공간 쿼리를 이용한 데이터 획득 속도
- 공간 쿼리를 이용한 데이터 획득 속도
- Range 사이즈에 따른 데이터 획득 속도

성능 테스트에서는 크게 비 공간 연산 쿼리와 공간 연산 쿼리의 두 종류로 나누어 실험하였다. 비 공간 연산 쿼리는 어떤 조건을 만족하는 튜플들을 얻기 위하여 튜플들 간의 조인 관계를 이용하는 반면 공간 연산 쿼리에서는 객체들의 공간적인 관계를 이용하여 질의를 하게 된다. SQL 문법에서 보면 비 공간 연산 쿼리는 WHERE 절에 일반 조인을 위한 구문만이 사용되지만 공간 연산 쿼리에서는 WHERE 절에 객체들 간의 공간적인 연산을 위한 함수가 포함되게 된다. 표 1은 Body-Face 모델에서의 비 공간 연산 쿼리와 공간연산 쿼리의 SQL 문의 차이를 나타낸다.

이러한 성능 테스트에서는 정육면체로 이루어진 가상의 3차원 객체를 자동 생성하여 공간 데이터베이스에 저장하였다. 생성된 각 개체들은 일정한 간격을 유지하도록 하였고 X축과 Y축에 동일한 개수가 위치하도록 하였다. 데이터 획득 속도 측정에 활용되는 객체는 10000개, 40000개, 90000개, 160000개, 250000개를 생성하여 적용하였다. 10000개의 객체생성을 예로 들면 가로에 100개, 세로에 100개의 정육면체를 생성하였다. 각 정육면체 객체들은 한 번의 길이가 10이고, 객체들 간의 사이의 간격도 10으로 하였다.

각 3차원 데이터 모델의 데이터베이스에 10000개부터 250000개의 표적을 저장하고, 각각의 케이스마다 성능 테스트를 수행하였다. 해당 모델의 테스트에서는 각 표적의 개수에 따른 테스트를 10번씩 수행하였고, 결과 값으로는 이들의 평균을 사용하였다.

### 7.1 3차원 가시화를 위한 데이터 획득 테스트

첫 번째 테스트는 3차원 가시화를 위한 데이터 획득 속도 테스트이다. 3차원 가시화는 데이터베이스에 저장된 객체들의 면을 구성하는 포인트 좌표 값을 활용하여

가시화한다. 그러나 가시화 과정에서 회전과 이동시마다 데이터베이스에 재접속하여 지오메트리 정보를 획득하게 되면 3차원 가시화의 속도가 매우 떨어지게 된다. 따라서 이와 같은 문제점을 해결하기 위해서 프로그램이 처음 실행될 때 모든 지오메트리 정보를 획득하여 배열에 저장하고, 가시화의 다양한 이벤트에서 배열에 저장된 정보를 이용하는 방법을 사용하였다. 이렇게 배열에 저장하는 방법을 사용하기 위하여 우선 제시한 3차원 데이터 모델들마다 객체의 ID, 객체를 구성하는 면의 ID 및 이를 구성하는 지오메트리를 데이터베이스에서 획득하고 배열에 저장하는 작업의 속도를 테스트하였다.

각 네 가지의 모델들의 가시화를 위한 구성요소들이 배열에 저장되는 시간을 측정하였다. 본 테스트에서는 각각의 모델마다 400(20×20), 1600(40×40), 3600(60×60), 6400(80×80), 10000(100×100)개의 개체를 생성하여 성능 테스트를 수행하였다.

### 7.2 비 공간 연산 쿼리를 이용한 데이터 획득 테스트

비 공간 연산 쿼리는 앞서 언급한 바와 같이 조인질을 통해 조건을 만족하는 튜플을 얻는 쿼리이다. 비 공간 연산 쿼리를 수행하였을 때 해당 객체들의 ID와 객체를 구성하는 면들의 ID와 해당 지오메트리를 획득하는데 소요되는 시간을 테스트 하였다. 이는 조인 질의가 복잡해짐에 따라 결과를 획득하는 속도의 성능 차이를 살펴보고자 함이다.

본 테스트에서는 모델마다 10000(100×100), 40000(200×200), 90000(300×300), 160000(400×400), 250000(500×500)개의 개체를 이용하였다.

### 7.3 공간 연산 쿼리를 이용한 데이터 획득 테스트

세 번째로 수행한 공간 연산 쿼리 속도 테스트에서는 range 쿼리(ST\_Intersect)를 수행했을 때 이 범위에 포함되는 면들의 ID와 이 면들을 포함하는 객체의 ID를 획득하는데 소요되는 시간을 측정하였다. ST\_Intersects는 PostGIS에서 제공하는 함수로서 공간 객체들이 서

표 1. 비 공간 연산과 공간 연산의 SQL 예(Body-Face의 경우)

비 공간 연산	공간 연산
<pre>SELECT Distinct solid.solidid, face.faceid, ST_AsEWKT(shape) FROM Solid INNER JOIN face ON solid.solidid = face.solidid ORDER BY body.bodyid, face.faceid;</pre>	<pre>SELECT Distinct solid.solidid, face.faceid FROM Solid INNER JOIN face ON solid.solidid = face.solidid WHERE ST_Intersects(GeomFromEWKT(face.Shape), GeomFromEWKT('POLYGON(((0 0 0,350 0 0,350 55 0,0, 55 0,0 0 0)))')=true;</pre>



로 중첩된다면 True 값을 반환한다 (Paul Ramsey, 2005). 이는 가상의 비행물체에서 레이저 신호를 일정 시간 동안 지상의 표적에 발사하였을 때, 이 레이저 신호의 범위에 포함되는 표적들을 검색하는데 소요되는 시간을 살펴보고자 한 테스트이다. 본 테스트에서 사용된 range의 범위는 다음과 같은 가정에 근거하여 설정하였다.

- 가상의 비행물체가 상공 1000m에서 운항한다고 가정한다.
- 가상의 비행물체에서 발사하는 레이저 신호는 20°의 범위로 발사된다.
- 비행물체는 시속 200km/h로 운항한다고 가정한다.
- 레이저 신호는 비행물체에서 1초 간격으로 발사한다고 가정한다.

위 가정을 만족하는 범위를 살펴보면 가로 약 350m, 세로 약 55m임을 알 수 있다. 가로 방향은 비행물체의 고도와 비행물체에서 발사하는 레이저 신호의 각도를 계산하여 거리를 구하였으며, 세로 방향은 비행물체의 속도와 레이저 신호의 발사 간격을 기준으로 구하였다. 본 테스트에서는 각 모델마다 10000(100×100), 40000(200×200), 90000(300×300), 160000(400×400), 250000(500×500)개의 개체를 이용하였다.

#### 7.4 Range 사이즈에 따른 공간 쿼리 데이터 획득 테스트

가상의 비행물체가 고도를 변경하게 되면 지상에 도달하는 레이저신호의 범위는 변하게 된다. 따라서 네 번째 테스트에서는 range의 변화에 따른 공간 연산 쿼리의 성능을 테스트하고자 range의 범위를 일정하게 증가시키며 앞에서의 테스트와 동일한 공간 연산 쿼리에 소요되는 시간을 측정하였다. 본 테스트에서 사용된 range의 범위는 다음과 같은 가정을 기반으로 사용되었다.

- 가상의 비행물체가 상공 1000m와 1500m 사이에서 운항된다.
- 가상의 비행물체에서 발사하는 레이저 신호는 20°의 범위로 발사된다.
- 비행물체는 시속 200km/h로 운항한다고 가정한다.
- 레이저 신호는 비행물체에서 1초 간격으로 발사한다고 가정한다.

가로 방향은 비행물체의 고도와 비행물체에서 발사하는 레이저 신호의 각도를 계산하여 거리를 구하였는데, 가로 방향은 대략 350m~530m로 계산된다. 세로 방향은 비행물체의 속도와 레이저 신호 발사 간격을 기

준으로 거리를 구하였으며, 대략 55m임을 알 수 있다. 본 테스트에서는 40000(200×200)개의 개체를 활용하였으며 초기 350m×55m의 range부터 시작하여 가로 방향으로 530m가 될 때까지 30m씩 증가시키면서 테스트하였다.

## 8. 성능테스트 결과

### 8.1 3차원 가시화를 위한 데이터 획득 테스트

3차원 가시화를 위한 데이터 획득 테스트 결과를 살펴보면 그림 8과 같다. Body-Face 모델이 가장 빠른 검색 속도를 나타낸다. Body-Face-Node 모델과 Body-Face-Edge 모델은 거의 동일한 검색 속도를 보였으며 이 두 모델은 Body-Face 모델보다 검색속도가 현저하게 늦은 결과를 나타내었다. Body-Face-Edge-Node 모델은 네 가지의 모델 중에서 가장 검색속도가 느린 모델로 나타났다. 테스트 결과를 살펴보면 3차원 객체의 수가 증가할수록 네 가지 모델 모두 쿼리 수행 시간이 증가하였으며, Body-Face 모델은 객체의 수가 증가할수록 쿼리 수행 시간의 증가율이 가장 작았으며, Body-Face-Edge-Node 모델은 증가율이 가장 크게 나타났다. Body-Face 모델이 나머지 모델과 비교하여 대략적으로 최소 20배에서 최대 40배까지의 차이를 보여준다. 위 테스트 결과를 살펴보면 쿼리를 수행하는데 조인의 횟수가 많아지고, 데이터베이스에 저장된 개체의 수가 증가할수록 쿼리 시간이 급격하게 증가함을 알 수 있다.

### 8.2 비 공간 연산 쿼리를 이용한 데이터 획득 테스트

데이터베이스에 저장된 다면체와 다면체를 이루고 있는 면들의 쿼리 획득 결과를 살펴보면 그림 9와 같이 나타난다. 여기에서는 Body-Face, Body-Face-Node, Body-Face-Edge, Body-Face-Edge-Node의 순으로 검색속도가 빠르게 나왔다. 여기에서도 앞에서의 객체의 수가 증가할수록 네 가지 모델 모두 쿼리 수행 시간이 증가하였으며, Body-Face 모델은 증가율이 가장 작았으며, Body-Face-Edge-Node 모델은 증가율이 가장 크게 나타났다. Body-Face 모델이 나머지 모델과 비교하여 대략적으로 최소 1.5배에서 최대 15배까지의 차이를 나타내는 것을 알 수 있다. 객체의 수가 증가할수록 모델별 성능의 격차가 점점 커짐을 알 수 있다.

본 테스트에서도 3차원 가시화를 위한 데이터 활용과 마찬가지로 조인의 횟수와 개체수가 증가할수록 쿼리 시간이 증가하는 것을 알 수 있다.

### 8.3 공간 연산 쿼리를 이용한 데이터 획득 테스트

공간 연산 쿼리를 이용한 데이터 획득 테스트 결과를 살펴보면 그림 10과 같다. 여기에서는 Body-Face-Node, Body-Face-Edge-Node, Body-Face, Body-Face-Edge의 순서대로 검색속도가 나타났다. 여기에서도 마찬가지로 3차원 객체의 수가 증가할수록 네 가지 모델 모두 쿼리 수행 시간이 증가하였으며, 데이터베이스에 저장된 객체의 수가 증가할수록 모델별 쿼리 수행 시간의 격차가 점점 커짐을 알 수 있다.

앞에서 수행하였던 테스트 1, 2와 같이 본 테스트에서 수행한 공간 쿼리를 이용한 데이터 획득 테스트에서도 조인 질의를 사용하지만 테스트 1, 2의 데이터 모델들의 성능 차이와 본 테스트의 데이터

모델들의 성능 차이가 다르게 나오는 것을 알 수가 있다. 이를 통해 조인 질의보다는 지오메트리의 종류가 따라 공간 연산 쿼리 속도에 더 큰 영향을 미친다고 판단할 수 있다.

### 8.4 Range 사이즈에 따른 공간 쿼리 데이터 획득 테스트

Range 사이즈에 따른 공간 연산 테스트 결과를 살펴보면 그림 11과 같다. 여기에서는 앞서 수행하였던 테스트 3의 결과와 동일한 순서와 검색속도를 보여주었다. 그리고 각각의 모델마다 range의 범위를 350부터

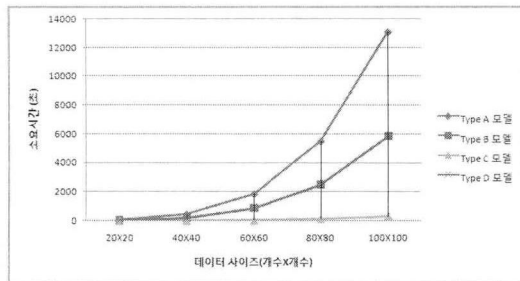


그림 8. 모델별 3차원 가시화를 위한 데이터 획득 소요시간

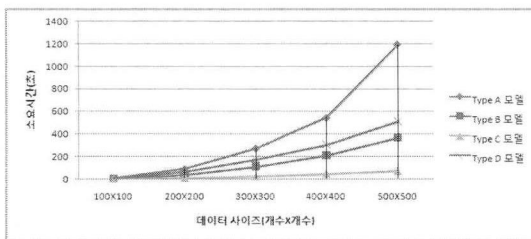


그림 9. 모델별 비 공간 연산 쿼리를 이용한 데이터 획득 소요시간

530까지 증가시키면서 테스트하였지만 속도에는 영향을 미치지 못하였다. 물론 range의 범위를 상당히 넓은 범위로 테스트 할 때는 쿼리 수행 시간이 더 소요되긴 했지만 본 테스트에서와 같이 레이저레이더 시뮬레이션에서 활용되는 range 안에서는 range의 차이에 따른 성능의 차이가 거의 없었다.

### 9. 3차원 데이터 모델 선정

앞에서 제시한 네 가지의 3차원 데이터 모델의 성능 테스트 결과와 레이저레이더 시뮬레이션에서 추가적으로 고려해야 하는 사항들을 모델들에 대하여 비교분석한 후 시뮬레이션에 가장 적합한 3차원 데이터 모델을 선정하였다. 본 연구의 시뮬레이션에서는 일정 범위의 지형 및 표적을 3차원으로 가시화해야 한다. 이를 위하여 3차원 가시화를 위한 데이터 획득 성능 테스트를 수행하였으며, 테스트 결과 Body-Face 모델이 가장 빠른 속도를 나타내었다.

가상의 비행물체는 레이저신호를 지상의 표적에 발사하게 되고 레이저신호와 교차하는 객체들의 면 ID와 속성정보를 획득해야 한다. 이에 본 연구에서는 일정 시간 동안 발사한 레이저 신호를 range로 형성하고 공간 연산 쿼리를 수행하여 쿼리 속도를 테스트하였다. 테스트 결과 가장 빠른 속도를 나타낸 모델은 Body-Face-Node 모델이었다. 그리고 레이저레이더 시뮬레이션에서 사용하는 가상의 비행물체는 1km~1.5km 범위의

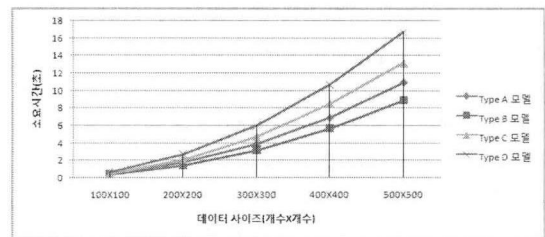


그림 10. 모델별 공간 연산 쿼리를 이용한 데이터 획득 소요시간

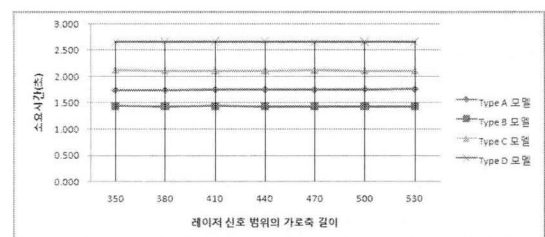
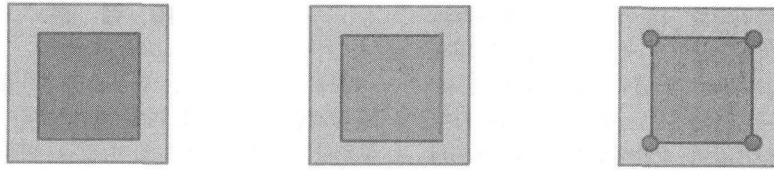
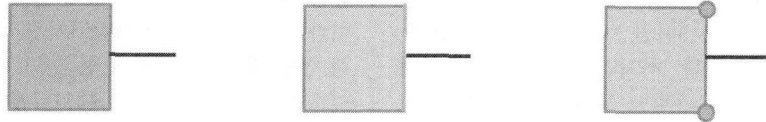


그림 11. range 변화에 따른 공간 연산 쿼리 소요시간





면과 면의 Intersect(연산 가능)    라인과 면의 Intersect(연산 가능)    포인트와 면의 Intersect(연산 가능)



면과 라인의 Intersect(연산 가능)    라인과 라인의 Intersect(연산 가능)    포인트와 라인의 Intersect(연산 불가)

그림 12. 지오메트리와 공간연산의 활용

표 2. 모델별 복잡도

	전체 테이블 개수	인터섹션 테이블 개수
Body-Face-Edge-Node	4	2
Body-Face-Node	3	1
Body-Face	2	0
Body-Face-Edge	3	1

상공에서 레이저신호를 발사하게 됨에 따라 지상으로 도달하는 레이저신호의 범위가 변화하게 되는데, 이를 공간 연산 쿼리에 적용하여 range의 차이에 따른 성능 차이를 확인해보았다. 테스트 결과 Body-Face-Node 모델이 가장 빠른 속도를 나타내었다.

본 연구에서 수행한 성능 테스트 이외에도 데이터 모델 선정 시에는 3차원 데이터 모델의 복잡도와 다양한 공간 연산의 활용 가능 여부도 고려해야 할 것이다. 지형과 표적의 지오메트리 정보와 속성정보를 데이터베이스에 저장할 때, 데이터 모델이 복잡할수록 객체의 정보를 저장하기 어렵고, 저장 시 오랜 시간이 소요된다. 또한 Face의 속성 및 지오메트리 정보를 얻기 위해 Edge, Node로의 조인이 필요해지고, 이때 데이터 가시화에 성능 저하가 발생할 수 있다.

따라서 표 2에서 나타난 바와 같이 Body-Face 모델이 가장 간단한 구조로 이루어져 있어 데이터 모델의 복잡도를 고려할 때도 우수하다고 할 수 있다.

본 연구에서는 현재 컴퓨터 성능의 한계로 인해 레이저 신호를 range 단위로 변환하여 테스트하였지만, 실제 레이저 탐색기에서 발사하는 레이저 신호는 직선으로서 반사되어 돌아오는 신호를 통하여 객체의 정보를 획득하게 된다. 하지만 추후 컴퓨터 성능이 개선되어 빠른 공간 연산이 가능하게 되면 range의 공간 연산뿐만 아니라 직선을 활용한 공간 연산도 사용 가능하게 될 것이다. 따라서 range나 직선을 이용한 공간 연산의 적용 가능 여부도 3차원 데이터 모델 선정에 포함될 수 있다.

앞에서 제시된 네 가지 데이터 모델에서는 각각의 모델에 따라 Face, Edge, Node 클래스에 지오메트리를 저장되게 되는데 그림 12와 같이 range에 해당하는 폴리곤은 폴리곤, 라인, 포인트의 지오메트리와 공간 연산이 가능하다. 그리고 그림 12와 같이 레이저 신호에 해당하는 라인은 폴리곤과 라인의 지오메트리와 공간 연산이 가능하지만 라인과 포인트의 공간 연산은 라인이 포인트와 정확하게 중첩되지 않을 경우 연산이 불가능하다. 따라서 지오메트리가 폴리곤으로 이루어진

표 3. 성능테스트 결과

평가 항목	A모델	B모델	C모델	D모델
3차원 가시화를 위한 데이터 획득 성능 속도	4번째	2번째	1번째	2번째
비 공간 쿼리를 이용한 데이터 획득 속도	4번째	2번째	1번째	3번째
공간 쿼리를 이용한 결과 데이터 획득 속도	2번째	1번째	3번째	4번째
Range 사이즈에 따른 데이터 획득 속도	2번째	1번째	3번째	4번째
3차원 데이터 모델의 복잡도	4번째	2번째	1번째	3번째
다양한 공간 연산의 활용 가능 여부	3번째	3번째	1번째	1번째

Body-Face 모델과 라인으로 이루어진 Body-Face-Edge 모델은 직선과의 공간 연산이 가능하다고 할 수 있지만, Body-Face-Edge-Node 모델과 Body-Face-Node 모델은 직선과의 공간 연산에서는 활용이 불가능하다.

위 내용들을 표 3과 같이 종합해보면 레이저레이더 시뮬레이션에 적합한 3차원 데이터 모델은 두 가지로 요약할 수 있다. 가시화에 적합한 데이터 모델은 Body-Face 모델이고, 공간 연산을 고려한 데이터 모델은 Body-Face-Node 모델이다. 본 연구에서 테스트하고자 하는 시뮬레이션은 3차원 가시화와 공간 연산 두 가지를 다 고려해야 한다. 하지만 개체수가 증가함에 따라서 성능 면에서 더 유리한 모델을 선정하는 것이 바람직 할 것이다. 가시화를 위한 성능 테스트에서 나타난 결과를 보면 개체수가 증가함에 따라서 다른 모델들의 소요 시간이 크게 증가하는 반면에 Body-Face 모델은 큰 증가를 보이지 않았다.

한편 공간 연산을 위한 테스트 결과를 보면 개체수가 증가함에 따라서 모든 모델들에서 소요시간이 증가해 갔으나 Body-Face-Node 모델이 가장 우수한 것으로 나타났었다. 따라서 가시화가 필요한 경우에는 Body-Face 모델이 우수하며 공간 연산만을 적용할 때에는 Body-Face-Node 모델이 우수할 것으로 판단된다. 하지만 실험 모델과 실험 객체의 수에 따라 첫 번째 실험에서의 소요시간이 세 번째 실험에서의 소요시간 보다 더 큰 차이를 보였으므로 본 연구의 시뮬레이션에 가시화를 적용하기 위하여 Body-Face 모델을 이용하였다.

여기에서는 그림 13과 같이 하나의 다면체와 이를 이루고 있는 면들의 관계를 정의하여 부분적인 토폴로지 형태를 갖도록 하였다. 이러한 데이터 모델을 활용하여 실제 질의를 수행할 때, Face와 Edge, Point의 조인질을 생략하고 Body와 Face의 조인질에만 이루어 질 수 있도록 함으로써 데이터의 접근 속도를 향상시켜 시뮬레이션의 수행 속도를 높이하고자 하였다.

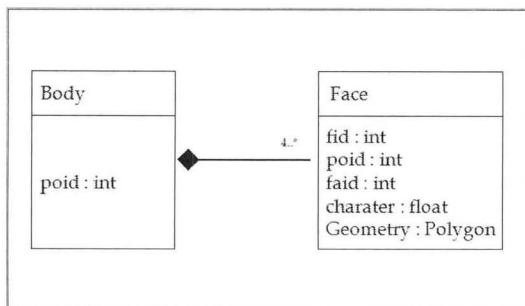


그림 13. 레이저레이더 시뮬레이션을 위한 3차원 데이터 모델

## 10. 결 론

현재 사용되고 있는 대부분의 3차원 모델은 가시화에 중점을 두고 있고, 파일 기반의 데이터 구조로서, 토폴로지 구조를 가지고 있지 않다. 따라서 본 연구에서 테스트하고자 하는 레이저레이더 시뮬레이션에 기존의 3차원 모델을 적용하기에는 연산 속도, 가시화 성능, 속성 정보 이용 등의 문제로 인하여 한계가 있다. 이와 같은 문제점을 해결하기 위해 본 연구에서는 레이저레이더 시뮬레이션에 적합한 3차원 데이터 모델을 다양한 성능 테스트를 통하여 선정하고 이를 공간 데이터베이스에 적용하였다. 공간 데이터베이스를 이용하여 건물 내 각 공간의 공간정보와 속성 정보를 저장하면 데이터베이스를 통한 빠른 질의와 연산 및 분석이 가능해지고, 공간 정보의 변경 및 유지 관리가 용이해진다. 또한 데이터베이스로부터 추출한 공간 데이터를 2차원 및 3차원으로 가시화할 수도 있다.

우선 네 가지 종류의 3차원 데이터 모델을 정의하고 각 모델에 대하여 공간 데이터베이스를 기반으로 데이터를 구축한 후 성능 테스트를 수행하였다. 그리고 이 모델들에 대한 테스트 수행 결과를 비교 분석하여 레이저레이더 시뮬레이션에 가장 적합하다고 판단되는 Body-Face 모델을 선정하였다. 선정된 데이터 모델은 건물과 건물의 면 사이의 조인질에만 이루어질 수 있도록 함으로써 데이터 접근 성능을 향상시켜 시뮬레이션의 수행 속도를 높이하고자 하였다.

본 연구에서 제시한 3차원 데이터 모델 및 공간 데이터베이스 적용 기법은 레이저레이더 시뮬레이션뿐만 아니라 다양한 3차원 분석 및 가시화 분야에 응용될 수 있을 것으로 판단된다.

## 감사의 글

본 연구는 한국과학기술원 영상정보특화연구센터의 II-21 “레이저 영상 신호특성모델링 기법 연구” 과제를 통한 방위사업청과 국방과학연구소의 연구비 지원으로 수행되었습니다(계약번호 UD070007AD).

## 참고문헌

1. 김근한, 전철민, 2008, “레이저레이더 시뮬레이션을 위한 3차원 객체 모델링”, 한국지형공간정보학회, 제 15권, pp.3-14.
2. 민성홍, 김성준, 이임평, 2008, “Linear FM 신호분석을 통한 Ladar Range 영상의 시뮬레이션”, 한국지형공간

- 정보학회, 제 16권, pp.87-95.
3. Arens, C., Stoter, J.E., and van Oosterom, P.J.M., 2005, "Modelling 3D spatial objects in a geo-DBMS using a 3D primitive", *Computers & Geosciences*, Vol. 31, No. 2, pp.165-177.
  4. Chen, T.K., A. Abdul-Rahmana, and S. Zlatanova, 2008, "3D Spatial Operations for geo-DBMS: geometry vs. topology", *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVII, Part B2, Beijing.
  5. Ellul, C., Haklay, M., 2008, Using a B-rep structure to query 9-intersection topological relationships in 3D GIS reviewing the approach and improving performance, In J. Lee and S. Zlatanova, (eds.), *3D Geo-information Sciences*, Springer-Verlag, Berlin: pp.127-151.
  6. Gröger, G., M. Reuter, and L. Plümer, 2004, "Representation of a 3-D city model in spatial object-relational databases", In *Proc. of the 20th Congress of International Society for Photogrammetry and Remote Sensing*, Istanbul, Turkey.
  7. Paul Ramsey, 2005, "PostGIS Manual", version 1.3.2, p.40.
  8. Stoter, J.E., and van Oosterom, P.J.M., 2002, "Incorporating 3D geo-objects into a 2D geo-DBMS", *ACSM-ASPRS 2002 ANNUAL CONFERENCE PROCEEDINGS*.
  9. Stoter, J.E., Zlatanova, S., 2003. "Visualising and editing of 3D objects organised in a DBMS". *Proceedings EUROSDR Workshop : Rendering and Visualisation*, pp.14-29.
  10. Zlatanova, S., 2000, "3D GIS for urban development", PhD thesis, TUGraz , Austria, p.222.
  11. Professional Surveyor Magazine, 2009, URL:<http://www.profsurv.com/assets/magazines/articles/70254/AirborneLidarSensors.pdf>.