

# iSTORM에서의 2D/3D 공간질의어 및 질의 최적화기법

김영삼<sup>U</sup> 임헌기 남광우 류근호  
충북대학교 데이터베이스연구실  
{yskim, hklim, kwnam, khryu}@dblab.chungbuk.ac.kr

## 2D/3D Spatial Query Language and Query Optimization on iSTORM

Young Sam Kim<sup>U</sup>, Kwang Woo Nam, Hun Ki Lim, Keun Ho Ryu  
Dept. of Computer Science, Chungbuk National University

### 요 약

지리정보시스템은 지리적으로 관련된 정보를 효율적으로 저장, 변경, 관리 및 분석을 위한 시스템으로서 하드웨어, 소프트웨어, 지리 데이터 등으로 구성되어 있다. 지리정보시스템은 일반적인 데이터베이스의 연산과 공간정보를 이용한 분석 기능을 통합한 시스템으로서 응용 범위 또한 매우 다양하다. 이 논문에서는 지리정보시스템의 2차원 및 3차원 질의 처리를 위한 2차원 및 3차원 공간 질의어, 그 질의처리 및 최적화에 대해 보인다.

### 1. 서론

지리정보시스템은 지리적으로 관련된 정보를 효율적으로 저장, 변경, 관리 및 분석하기 위한 시스템으로서 하드웨어, 소프트웨어, 지리 데이터 등으로 구성되어 있다. 지리정보시스템은 일반적인 데이터베이스의 연산과 공간정보를 이용한 분석 기능을 통합한 시스템으로서 응용 범위 또한 매우 다양하다. 최근에는 GIS 기술 그룹과 데이터베이스의 그들간의 상호 연관된 연구의 필요성이 인식되고 있다. 이 연구는 그러한 지리정보시스템에서의 데이터베이스 기술 개발을 위한 연구선상에 있다.

이 논문에서는 지리정보시스템의 2차원 및 3차원 질의 처리를 위한 2차원 및 3차원 공간 질의어, 그 질의처리 및 최적화에 대해 보인다.

### 2. 관련 연구

사용자에게 공간 객체에 대한 검색 기능을 제공해 주는 많은 응용 소프트웨어의 구조 즉, 공간 객체의 유지·관리 및 질의 처리를 위한 접근 방법은 네 가지로 분류된다[Laru93].

첫 번째 방법은 초기 구조 형태로 파일 집합으로 시스템을 구성하는 것으로 전체 데이터베이스의 이해 없이 특정 분야에 이용할 목적만 가진다. 이러한 시스템은 데이터 관리에 제한적이어서 수정 및 확장이 어렵고, 또한 데이터 처리능력의 효율성을 감소시킨다.

두 번째 방법은 기존의 데이터베이스 시스템을 이용하여 공간 객체의 비 공간 데이터 부분과 공간 데이터를 분리하여 관리한다. 이러한 이중 구조(dual architecture)의 형태를 지니고 있는 시스템에는 Arc/Info[ESRI95], Sand[Aref91], GeoQL[Ooi90] 등이 있다. 이러한 구조는 서로 다른 데이터 모델을 사용하여 저장하고 있기 때문에 공통의 식별자(identity)가 요구된다.

세 번째는 공간, 비 공간 데이터를 동일한 데이터베이스 시스템 내에 관리하여 하나의 새로운 계층이 데이터베이스 시스템의 상위에 놓이게 된다. 이러한 구조를 기반으로 한 시스템들은 보다 나은 상위 레벨의 통합을 위해 기본적인 공간 데이터 타입과 연산자들을 제공한다. 이와 같은 통합된 접근 방법을 갖는 시스템들로는 Paradise[Dewi94], Graf[Beck92], Postgres[Ston90] 등이 있다. 이와 같은 통합 시스템 구조의 경우에는 효율적인 공간 데이터 액세스가 가능하고 일반적인 프러디킷을 지니는 공간 질의의 선택이 넓어지게 된다. 이 논문의 시스템은 세 번째의 방법을 사용한 시스템[임헌기00]을 기반으로 한다.

### 3. 공간 질의 언어

#### 3.1 공간 데이터 정의 언어

공간 자료 정의어는 기존의 일반 속성뿐만 아니라 2차원과 3차원 공간 속성을 구성요소로 표현한다. 공간 속성은 하나 이상의 원자값으로 구성된다. 예를 들면, POINT 속성은 하나의 공간 좌표 쌍을 가지며, LINestring 속성은 공간 좌표쌍의 연속으로 구성되며, POLYGON은 닫혀있는 공간 좌표 쌍의 연속으로 구성된다. 공간 데이터 타입은 2차원 공간 데이터 타입과 3차원 공간 데이터 타입으로 구성되며 다음과 같이 구성된다.

```
<SpatialType> := <2DSpatialType> | <3DSpatialType>
2DSpatialType := "Point"
                |"LineString"
                |"Polygon"
3DSpatialType := "SBUILDBOX"
                |"SBUILDCON"
                |"SBUILDCYL"
                |"SBUILDEXT"
```

이 연구는 한국과학재단의 99년도 특정 연구 관제의 지원으로 수행되었음

|"SROADFACE"  
|"SPIPEEXT"

위에서 보이는 것과 같이 지원되는 3차원 공간 데이터 타입에는 SBUILDBOX, SBUILDCON, SBUILDXYL, SBUILDEXT, SBOADFACE, SPIPEEXT 등이 있다. 다음의 <그림 1>은 공간 데이터 타입을 갖는 테이블의 생성을 위한 데이터 정의어의 예이다. 대전호수는 2차원 공간 데이터를 갖는 테이블로서 geo라는 POLYGON 속성을 가지며, SeoulApt 테이블은 3차원 공간 데이터 sbuildext 데이터를 가진다.

CREATE TABLE 대전호수 (	CREATE TABLE SeoulApt (
id number(10),	id number(10),
레이어이름 char(50),	o_name char(10),
유형 char(50),	f_id number(20),
색 integer,	o_type number(10),
주소 char(50),	imag char(20),
면적 integer,	sound char(10),
둘레 integer,	details char(30),
폭 integer,	exs number(10),
최대저수량 integer,	name char(30),
최소저수량 integer,	dong number(5),
geo polygon	story number(5),
);	pyung number(5),
	compY number(5),
	geo sbuildext
	);

그림 1. 2차원과 3차원 공간데이터 테이블

3.2 공간 데이터 조작어

공간 자료 조작 언어에는 객체에 대한 삽입, 삭제, 변경 및 검색문 등이 있다. 이들 문장들은 시간지원 연산자와 공간 연산자를 조건절(predicate clause)과 목적절(target clause)에서 포함하며, 이를 통해 객체에 대한 공간관리를 제공한다.

가. 공간 데이터 표현

이미 설명한 것과 같이 제안하는 시스템에서는 2차원 및 3차원 공간 데이터를 지원한다. 즉, 공간 객체-관계 데이터베이스는 공간 데이터를 가지며 질의어는 데이터베이스내의 공간 데이터에 대한 검색 및 갱신을 위해서 접근하기 위한 공간 데이터 표현을 지원해야 한다. 아래는 POINT, LINESTRING, POLYGON 타입의 텍스트 표현을 위한 문법이다.

예를 들면, 각 데이터 타입의 텍스트 표현은 다음과 같다.

POINT ( 10, 10 )  
LINESTRING ( 10 10, 20 20, 30 30 )  
POLYGON (( 100 100, 200 200, 200 100, 400 400 ))

이 시스템에서는 SBUILDBOX, SBUILDCON, SBUILDXYL, SBUILDEXT, SROADFACE, SPIPEEXT 등의 3차원 공간 데이터 타입을 지원한다. 3차원 공간 데이터 타입의 문법은 위와 같다. 3차원 데이터 타입은 아래와 같은 텍스트 형태로 표현될 수 있을 것이다.

SBuildBox( 100 200 300 )  
SBuildCyl( 200 200 200 )  
SBuildExt( 140,100,false, (209 0 446, 209 0 446,209 0 446, 209 0 446) )

나. 공간 검색문

공간 검색문은 데이터베이스에 저장된 객체에 대하여 공간 조건을 논

시에 지정함으로써 객체에 대한 공간 정보를 사용자에게 제공한다. 공간 검색문의 문법은 일반적으로 사용하는 질의문과 같다. 검색문의 목적절에는 일반 속성 뿐만 아니라 공간 속성 값 자체를 출력하는 공간 함수 구문이 사용될 수 있다. 또한 FROM 절에서는 지정된 테이블에 대한 별명을 사용함으로써 2개 이상의 테이블로 구성된 복잡한 질의 구조를 보다 간단하게 표현해준다. WHERE절에서는 일반 속성과 공간 속성을 대상으로 하는 공간 판단 연산자와 공간 함수를 사용하여 검색될 객체에 대한 조건을 기술하게 된다. 공간 데이터 검색을 위해 지원되는 것은 공간 함수와 공간 조건 판단자 두개로 나누어 질 수 있다. 지원하는 판단자는 equal, disjoint, touch, within, overlap, cross, intersect, contain 등이 있다.

공간 데이터 타입을 갖는 대전 아파트 테이블이 있다고 할 때 화면에서 사용자 인터페이스를 통해 입력 받은 POLYGON 내에 속하는 아파트의 이름과 id를 출력하는 공간 SQL은 다음과 같이 사용될 수 있다.

```
[1] POLYGON ((100 100, 200 200, 200 100, 400 400))내에 위치하는 대전아파트의 id와 이름을 검색하라.
SELECT a.id, a.이름
FROM 대전아파트 a
WHERE a.geo equal polygon(( 100 100, 200 200, 200 100, 400 400));
```

공간 질의어에서 buffer, distance, dimension, geometrytype, astext, asbinary, getx, gety, startpoint, endpoint, numpoints, pointn, area, length, minx, miny, maxx, maxy, perimeter, height 등의 함수가 지원된다. 다음은 공간 함수들을 사용한 질의의 예이다.

```
[2] 기준점 POINT(500,500)과의 거리가 500m이내인 대전아파트를 검색해서 id와 기준점과의 거리를 검색하라.
SELECT a.id, distance(a.geo, point(500, 500))
FROM 대전아파트 a
```

4. 질의 그래프 및 질의 최적화

4.1 질의 처리 시스템 구조

질의처리기는 구문분석(parsing), 의미분석(semantic check)을 통해 질의를 내부 질의 표현인 질의 그래프 트리의 형태로 변환하며, 이 질의 그래프 트리에 대하여 최적화(optimization)를 수행해서 질의 계획(query plan)을 만들게 되고, 질의 수행기(query executor)를 통해 질의를 수행하게 된다. 그림 2에서 보여지는 각 단계는 다음과 같이 구성된다.

가. 구문 분석과 의미 분석

사용자나 응용 프로그램으로부터 입력된 질의를 파싱하여 구문 오류(syntax error)와 의미 오류(semantic error)를 검사한다. 또한 이 단계에서 질의의 정규화 작업, 즉 Not 연산을 소거하며 질의를 Disjunctive Normal Form(DNF)의 형태로 변환하게 된다. 오류가 없으면 질의의 그래프(query graph)를 생성한다.

나. 최적화

최적화기는 질의의 그래프에 대하여 공간 데이터 타입을 포함한 추상 데이터 타입(ADT)에 따라 질의그래프 트리를 확장하며, 같은 클래스에 대한 질의들과 비공간 속성들을 하나의 조건 판단자로 통합하고, 이 결과 그래프 트리에 대하여 연산비용을 측정하여 최적 실행 경로를 선택하게 된다. 선택된 최적 실행 경로는 물리적 공간 객체-관계 대수로 구성되는 질의 실행 계획(query evaluation plan)로 전환되어 질의 실행기로 넘겨진다.

다. 질의 평가

최적화 단계를 통해 넘겨진 질의 실행 계획은 물리적 공간 객체 관계 대수들의 트리 형태를 갖는다. 이 물리적 공간 객체-관계 대수는 open,

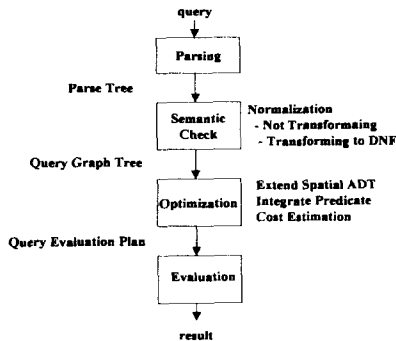


그림 2. 공간 질의 처리 과정

next, close로 구성되는 반복자(iterator)에 의한 스트리밍 형태로 구현되었다. 그러므로, 질의 수행기는 대수 트리의 루트에 대한 레퍼런스를 통해 반복자를 호출함으로써 질의물 수행하게 된다. 질의의 수행에 앞서 질의의 종류에 따라 추가적인 대수가 루트에 더해지기도 한다.

#### 4.2 질의 정규화

공간 질의에서는 조건의 표현을 위해 NOT, AND, OR 등의 논리 연산자의 사용은 질의의 처리를 복잡하게 하게 되므로 정규화 작업을 통해 단순화 해야한다. 일반적으로 사용되는 질의 정규화에는 NOT의 소거와 AND, OR 정규화가 있다. NOT의 소거는 질의에서 사용되는 NOT연산자를 드모르간 규칙을 이용하여 조건 판단자 내부로 옮겨서 소거하는 것이며 AND, OR 정규화는 질의를 AND에 의해 접속되는 Conjunctive Normal Form(CNF) 또는 OR에 의해 연결되는 Disjunctive Normal Form(DNF)의 형태로 정규화하는 것이다.

#### 4.3 질의 그래프 및 최적화

구문 분석 및 의미 분석을 통과한 질의는 질의 그래프의 형태로 변환된다. 질의 그래프는 질의의 대상 테이블을 노드(node)로 갖고 WHERE절의 조건에 의해 접근되는 경로는 노드들 상의 연결선(edge)으로 표현되는 그래프이다. 질의 그래프는 질의가 수행되기 위한 모든 경로를 포함하게 되므로 질의 공간(query space)을 형성하게 되며 질의 최적화는 이 질의 공간상에서 최적 수행경로를 찾는 것으로 정의할 수 있다. 질의 그래프에 대한 최적화는 이 시스템에서 다음과 같은 단계를 갖는다.

- STEP 1: 질의 그래프의 구성
- STEP 2: 공간 ADT 및 복합 레퍼런스의 분해
- STEP 3: 클러스터링 객체 연산의 통합
- STEP 4: 최적경로 탐색
- STEP 5: 물리적 객체-관계 대수 연산 트리의 생성

구문 분석과 의미 분석과정을 통과한 질의는 질의 해석(query calculus)로 볼 수 있으며 질의 그래프는 이 질의 해석을 입력으로 생성된다(STEP 1). 이렇게 생성된 질의 그래프에서 각 연결선의 속성값이 공간 ADT나 복합 객체 일 경우 별도의 노드를 생성하도록 분해된다(STEP 2). 클래스 스키마 정보를 사용하여 클러스터링된 객체를 발견한 후 그 노드들과 연결선을 통합하게 된다(STEP 3).이 시스템에서는 공간 ADT는 별도의 클래스로 구성되며 일반적인 속성에 대한 연산과 구별되는 공간 연산을 통해 수행된다. STEP 3에서 클러스터링 객체의 통합 과정에서 비공간 질의에 대한 통합을 수행하게 되고, 이러한 클러스터링된 질의 그래프에 대하여 최적 질의 수행 경로를 탐색하게 된다. 우선 각 연결선의 처리비용이 계산되며, 처리비용의 순으로 정렬된다. 이 방법 외에 최적의 수행경로를 찾기 위해 유전자 알고리즘이나 질의의 대수 변환알고리즘이 사용될 수도 있다(STEP 4). 끝으로 최적 수행경로

가 생성되면 물리적 공간 객체-관계 대수로 변환하는 과정을 거쳐서 질의 수행 계획을 생성해서 질의 수행기에 넘겨주게 된다(STEP 5)

#### 5. 물리적 공간 객체-관계 대수 연산자

공간 객체 관계 대수는 공간/비공간 혼합 질의 처리를 위한 대수이다. 일반 객체 관계 대수에 공간 조건 판단자까지 처리할 수 있도록 확장된 것이다. 이 질의 처리 시스템 내부에서 사용자 질의는 질의 그래프의 형태로 표현되며 최적 수행 경로 탐색에 의해 선택된 경로는 실제 수행을 위해 물리적 공간 객체 관계 대수로 변환된다.

#### 6. 결론 및 향후 연구 방향

이 논문에서는 지리정보시스템의 2차원 및 3차원 질의 처리를 위한 2차원 및 3차원 공간 질의어, 그 질의처리 및 최적화에 대해 보았다. 현재로서 이 연구는 초기 모델로서 그 SQL3에 기반한 가장 기본적인 질의에 대한 처리만을 다루고 있기 때문에 데이터베이스 관리 시스템이 필요로 하는 많은 부분이 부족한 상태이다. 또한 객체 모델의 가장 중요한 요소 중의 하나인 베스드 정의에 대한 구현 방법에 대해서는 현재 초기 연구가 진행 중에 있으며 클래스 참조 범위와 또한 질의의 그래프상에서 최소비용 탐색 알고리즘에 대한 많은 연구가 필요하다. 또한 질의의 그래프 상에서 집계합수가 고려되어야 하며, 객체관계 모델에 적합한 query unnesting을 고려한 질의 최적화 방법을 시스템에 추가할 것이다.

#### [참고문헌]

- [Aref91] W. G. Aref, "Optimization Strategies for Spatial Query Processing," Proceedings of 17th International Conference on Very Large Databases, 1991.
- [Beck92] L. Becker, R. H. Guting, "Rule-Based Optimization and Query Processing in an Extensible Geometric Database System," ACM Transactions on Database Systems, Vol. 17, No. 2, 1992.
- [Dewi94] D. J. DeWitt, N. Kabra, J. Luo, J. M. Patel, and J. Yu, "Client-Server Paradise," Proceedings of the 20th International Conference on Very Large Data Bases, 1994.
- [Laru93] T. Larue, D. Pastre, and Y. Viemont, "Strong Integration of Spatial Domains and Operators in a Relational Database System," Proceeding of the 3th International Symposium on Large Spatial Databases, Lecture Notes in Computer Science 692, 1993.
- [Ooi90] B. C. Ooi, "Efficient Query Processing in Geographical Information Systems," Lecture Notes in Computer Science, No. 471, 1990.
- [Ston90] M. Stonebraker, L. A. Rowe, and M. Hirohama, "The Implementation of POSTGRES," IEEE Transactions on Knowledge and Data Engineering, Vol. 2, No. 1, 1990.
- [ESRI95] ESRI, Spatial Database Engine: Developer's Guide Version 2.1, Environmental Systems Research Institute, 1995.
- [임현기00] 임현기, 김영삼, 남광우, 류근호, "iSTORM:인터넷 지향적 공간 객체-관계 데이터베이스 시스템, 제27회 정보과학회 춘계 학술 대회 2000. 4.
- [박경현00] 박경현, 남광우, 박성희, 류근호, "iSTORM에서의 공간 객체-관계 데이터 모델", 제27회 정보과학회 춘계 학술 대회, 2000. 4.