

□ 기술해설 □

지리정보시스템에서의 공간질의 처리방법

부산대학교 이 기 준*

● 목	차 ●
1. 머릿말	3.2 공간변형을 이용한 방법
2. 공간질의의 종류	3.3 여과-정제 방법
3. 공간질의처리방법	3.4 Distance Join Index
3.1 Linear Quadtree를 이용한 공간질의처리방법	4. 공간질의처리방법의 비교적 연구
	5. 맺음말

1. 머릿말

지리정보시스템은 지리적인 자료를 저장하여, 사용자가 원하는 정보를 제공하기 위하여 자료를 처리하고, 분석하는 기능의 집합으로 간단하게 정의될 수 있다[26]. 따라서, 지리적인 정보의 효과적인 저장, 관리, 그리고 처리 기능은 시스템의 주요기능이 되고, 이 기능들의 성능은 전체 시스템의 성능을 결정하는 주요 요소가 된다. 특히, 지리정보시스템이 저장하고 처리해야할 자료가 다른 정보시스템에 비해 상대적으로 커서, 효과적인 저장과 처리가 이루어지지 않으면 그 기능을 제대로 발휘할 수가 없다

지리정보시스템은 지리적인 자료를 저장하고 처리한다. 여기서 지리적인 자료란, 일반적으로 공간적인 자료와 비공간적인 자료의 합을 의미한다. 먼저, 비공간적인 자료란 문자, 숫자와 그외 멀티미디어 등을 말한다. 그런데 이와 같은 자료에 대한 연구는 이미 상당히 이루어졌고, 지리정보시스템만이 가지고 있는 특징이라고 보기는 어렵다. 그러나, 공간적인 자료는 지리정보시스템만이 가지고 있는 독특한 자료라고 할 수

있다. 따라서, 지리정보시스템의 성능을 향상하기 위해서는, 한편으로 지금까지 연구되어 구현된 비공간적인 자료의 처리기능을 사용하면서, 다른 편으로는 지금까지 상대적으로 별로 연구되지 못한 공간적인 자료에 대한 저장 및 처리기능을 강화하여야 한다[4].

공간적인 자료의 저장 및 처리를 위한 시스템을 일반적으로 공간데이터베이스관리시스템이라고 부른다. 본 논문에서는 이 공간데이터베이스관리시스템에 대한 주요기능에 대해서 다루어 보겠다. 공간데이터베이스관리시스템은 대략 그림 1과 같은 기능으로 구성되어 있다. 먼저, 공간색인 및 저장은 효과적으로 공간적인 자료를 보조기억장치에 저장하고, 탐색하는 기능을 말한다. 공간색인방법과 공간자료의 Clustering방법, 병렬디스크의 사용방법등을 이용하여 이 기능을 구현하는 것이 일반적인 접근방법이다. 두 번째의 기능은 동시적인 사용(Concurrent Access)이나 시스템의 장애로부터 자료를 보호하는 기능이다. 이 기능의 구현방법은 일반적인 데이터베이스관리시스템의 구현방법과 유사하나, 자료의 공간적인 특성에서 오는 몇가지의 일반 데이터베이스관리시스템과 다른 접근방식이 요구되기도 한다. 마지막으로, 공간질의처리기능은

*정회원

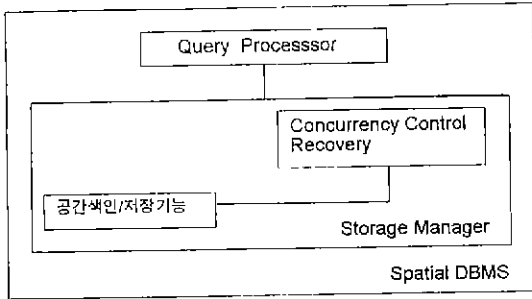


그림 1 공간데이터베이스관리시스템의 주요기능

효과적인 공간데이터베이스관리시스템을 구성하기 위해서는 위의 세가지 기능이 모두 중요한 지리정보시스템의 사용자가 던지는 다양한 공간적 질의를 처리하기 위한 기능이다.

다. 본 논문에서는 이중 세번째 기능, 즉 공간질의 처리기능에 대해 살펴보겠다. 효과적인 공간질의 처리방법을 개발하기 위해서는 우선 공간적 질의가 기존의 질의와 무엇이 다른가에 대한 이해가 요구된다. 따라서, 우선 제 2장에서는 공간질의의 특성과 종류에 대해서 살펴보겠다. 그리고, 제 3장에서는 지금까지 개발된 공간질의 처리방법에 대해서 살펴본다, 그런데 한가지 미리 주목할 것은, 여기서 소개되는 각 공간적인 질의 처리방법은 어떤 공간색인방법을 사용하였는가와 매우 밀접하게 연관되어 있다는 사실이다. 그 다음, 이들 공간질의 처리방법들의 비교에 관한 내용을 제 4장에서 다루어 보겠다. 사실, 3장에서 소개된 공간질의 처리방법에 대한 비교적 연구가 아직, 거의 이루어지지 않았다. 따라서, 제 4장에서는 실제적인 비교라기보다는 앞으로 있을 비교적 연구의 방향을 간단하게 제시하겠다.

2. 공간질의의 종류

공간질의는 기존 데이터베이스시스템의 질의와 매우 다르다[11,22]. 그 주요한 차이점은, 동어반복과 같은 표현이지만, “공간질의는 공간적인 자료에 대한 공간적인 질의”라는 점에서 비롯된다. 결국, 공간질의는 공간을 대상으로 한다는 점에서, 일반 질의와 다르다. 물론, 기존의

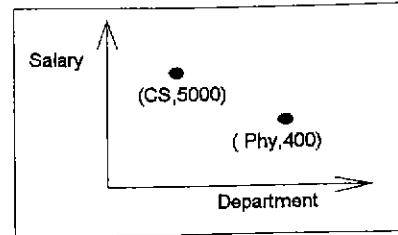


그림 2-1 일반데이터베이스 다차원공간

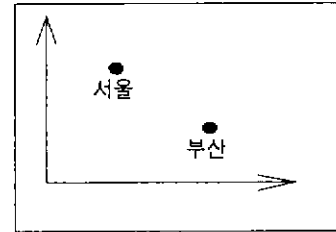


그림 2-2 공간데이터베이스 다차원공간

질의-예를 들어, “Select name From EMP where salary>2000”과 같은 질의-도 공간적인 성질을 일부 가지고 있다. 그러나, 그것은 대부분 일차원적인 공간에 불과하고, 간혹 다차원적인 공간을 이용한 질의의 경우에도, 특별히 그 공간의 특징을 이용하기가 매우 어렵다. 예를 들어, 그림 2-1의 경우와 같이, Department와 Salary 라는 두개의 Attributes를 가지고 있는 관계 EMP에서, {(Department,Salary)}라는 공간은, 우리가 일반적으로 공간에서 이용할 수 있는 중요한 몇가지 성질-예를 들어, 거리-이 존재하지 않는다.

그러나, 지리정보시스템에서 다루는 공간질의의 경우에는, 이차원이상의 다차원공간에 존재하는 모든 성질을 이용할 수가 있다. 물론, 경우에 따라, 공간의 종류가 다르기 때문에 이용할 수 있는 공간적 성질의 차이가 다소간 존재하지만, 일단, 응용분야가 정해지면, 그것에 따른 자료의 공간이 정해지고, 그 공간에서 발견될 수 있는 성질을 공간질의에 이용할 수 있다. 예를 들어, 그림 2-1의 경우, 두점 사이의 거리가 아무런 의미가 있을 수 없으나, 그림 2-2의 공간데이터베이스의 다차원공간에서는 두점의 거리를 이용한 질의가 가능하다.

공간데이터베이스의 공간적인 특성은 두가지 측면에서 이용될 수 있다. 먼저, 공간질의에 공간적인 성질-예를 들어, 거리, 포함관계, 연속성 등-을 이용할 수 있다. 예를 들어[25],

```
Select city.name
From city, airport
Where airport.name="김해" and airport.geometry
INSIDE city.geometry
```

와 같은 공간질의에서는 INSIDE 라는 공간적 성질을 이용하였다. 두번째 측면은, 공간적 성질을 이용하여 공간질의처리방법을 구현한다는 것이다. 예를 들어, "Eucliden공간에 주어진 삼각형의 두변의 합은 다른 변보다 길다."는 사실은 공간질의처리를 개발하는데 매우 중요한 단서가 된다[9,19,21]. 본 장에서는 위에서 언급한 첫번째 특성, 즉 공간질의에 이용되는 공간적 특성에 대하여 살펴보겠다. 두번째 특성인 공간질의처리를 위한 공간적 특성은 제 3장에서 다루기로 한다.

공간질의는 공간질의에 의해 이용되는 공간적인 성질로 나누어 진다. 일반적으로, 공간적인 성질에 의해, 기하적인 질의와 위상적인 질의로 나누어 진다. 기하적인 질의는 공간상의 기하학적인 성질-예를 들어, 거리, 넓이, 둘레길이-등을 이용하는 질의를 의미한다. 이에 반해 위상적인 질의는 기하적인 값을 사용하지 않고 단순히 객체들간의 공간적인 관계-예를 들어, 포함관계, 이웃관계등-을 이용하는 질의이다. 경우에 따라, 연속된 3차원적인 질의-예를 들어, 해발고도, 기울기-도 하나의 독립된 공간질의 형태로 보는

경우도 있으나, 본 논문에서는 이는 기하적인 질의에 포함시키겠다. 대표적인 기하학적인 질의와 위상적인 질의를 정리하면 다음과 같다 [26].

- 기하적 질의
 - 최소거리
 - Buffer Zone
 - 넓이, 둘레, 부피
 - 높이, 기울기
- 위상적 질의
 - 연결성에 관한 질의
 - Up-stream/Down-stream
 - 겹침
 - 포함
 - 인접

공간질의를 위상적인 질의와 기하적인 질의로 나누는 것은 공간질의에 사용된 조건이 어떤 공간적 성질을 이용했느냐는 것을 이용한 것이다. 반면에 공간적 질의를 Single Scan 질의와 Multiple Scan 질의로 나누는 방법도 있다[11]. Single Scan 질의는 조건을 명시할 때, 탐색하고자 하는 공간객체가 하나일 경우에 해당되고, Multiple Scan 질의는 조건에 맞는 두개 이상의 공간객체쌍을 탐색하는 질의를 의미한다.

다음 그림 3-1에의 질의에서는 "부산"이라는 주어진 공간객체에 대해 거리가 30 Km 이하인 객체를 찾는 질의이고 그림 3-2는 거리가 30 Km 이하인(airport.name, city.name)을 탐색하는 질의다. 따라서, 질의 3-1은 city와 airport라는 관계를 각 한번만 조사(Single Scan)하면 되지만,

```
Select airport.name
From airport, city
Where city.name="부산" and DISTANCE(city.geometry, airport.geometry) < 30 Km
```

그림 3-1 Single Scan 질의

```
Select airport.name, city.name
From airport, city
Where DISTANCE(city.geometry, airport.geometry) < 30 Km
```

그림 3-2 Multiple Scan 질의

그림 3-2의 질의는 두 관계를 여러번 조사(Multiple Scan)하여야 된다. 물론, 이와 같은 이유로, Single Scan 질의보다 Multiple Scan에 의한 질의가 훨씬 처리하기 어렵다.

공간질의의 종류를 살펴보면 한가지 주목해야 할 사실은 공간질의가 매우 다양하고 복잡적이라는 사실이다. 위에서 열거한 공간질의의 종류는 가장 대표적이면서 단순한 것들이다. 일반적으로 실제 응용에서 주어지는 질의는 위의 질의의 복잡한 형태를 가지고 있으면서, 매우 다양하다. 예를 들어, 위에서 정의된 거리에 관한 질의는 실제 응용에서 매우 다양하게 해석된다. 즉, 거리는 객체사이의 최소거리로, 무게중심사이의 거리, 또는 최대거리로도 해석될 수 있다. 공간질의처리를 공간데이터베이스에, 또는 지리 정보시스템에 구현하는데 어려움 중 하나는 바로 여기에 있다.

3. 공간질의처리방법

본 장에서는 위에서 나열된 공간질의의 처리 방법 중 대표적인 것들에 대해서 살펴보겠다. 우선 각각의 공간질의처리방법에 대해 살펴보기 전에 공간질의처리와 공간색인방법의 관계를 알아볼 필요가 있다. 이는 공간질의처리방법은 공간색인방법과 매우 밀접하게 연관되어 있기 때문이다. 대부분의 공간질의처리방법은 특정한 공간색인방법을 이용하여 구현되었다. 따라서,

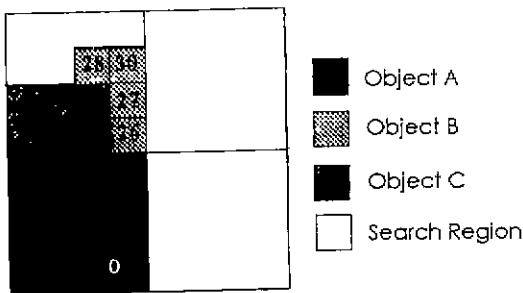
공간질의처리방법을 구분하기 위해서는 공간색인방법의 구분방법을 이용하는 것이 효과적이다. 일반적으로 공간색인방법은 이용하는 공간의 차원에 따라 다음과 같이 나누어 진다.

- 일차원적 변환에 의한 방법 : Linear Quadtree[26-28]
- 높은 차원으로 변환하는 방법 : LSD tree [2], MLGF[16]
- 차원의 변환이 없는 방법 : R-tree계열[1,5], Grid File계열[27,28]

본 논문의 범위가 공간색인방법을 포함하지는 않으므로, 위의 각 공간색인방법에 대한 설명은 생략하겠다. 더 자세한 내용은 참고문헌을 이용하기 바란다. 그러면, 위의 각 공간색인 방법을 이용한 공간질의처리방법을 살펴보기로 한다.

3.1 Linear Quadtree를 이용한 공간질의처리 방법

Linear Quadtree를 이용한 질의처리방법은 Linear Quadtree를 이용한 공간색인방법에 기초를 두고 있다[27,28]. Linear Quadtree를 이용한 공간색인방법의 가장 큰 장점은 기존의 데이터베이스관리시스템을 그대로 이용할 수 있다는 것이다[6]. 예를 들어, Linear Quadtree를 관계형 데이터베이스관리시스템을 이용하여 구현할 경우, Linear Quadtree에 의한 Peano



<Query Processing>

$$R1 = O \bowtie S$$

$$R2 = \Pi_{Object} (R1)$$

Object	Peano Key
A	0
B	26
B	27
B	28
B	30
C	16
C	24
C	25

Peano Key
25
27

S : Search Region

○ : Object Quadrants Table

그림 4 Linear Quadtree를 이용한 공간질의

Key를 직접 Index를 위한 Attribute로 지정하여 사용할 수 있다. 이는 제 2장에서 밝힌 다차원상의 공간이 Space-Filling Curve[26]를 통해 일차원으로 변환되었기 때문이다.

일단 일차원의 Peano Key로 변환이 되면, 관계대수(Relational Algebra)에서 정의된 연산자를 이용해, 원하는 질의를 표현할 수 있다. 그림 4에 나타난 예를 보면, 질의범위(Query Region)과 겹치는 공간객체를 찾기 위해서, Join과 Projection의 연산을 이용하였다. 따라서, Linear Quadtree를 이용한 질의처리방법은 Peano Key를 이용해 일반적인 색인(예를 들어 B+tree)을 구성하고, 관계대수의 연산자를 이용하여, 질의처리를 수행하는 것이다[26].

그런데 이와 같은 방법으로 구현할 수 있는 공간질의처리의 종류는 제한되어 있다. 위에서 예로 소개한 겹침, 포함관계의 질의와, 몇가지의 기하적인 질의-예를 들어 넓이-등 기하학적질의만이 가능하다. 따라서, 이 방법은, 관계형데이터베이스시스템위에 직접 구현할 수 있다는 장점에도 불구하고, 널리 사용되는 방법은 아니고 특수한 응용분야에만 적용가능하다.

3.2 공간변형을 이용한 방법

위의 Linear Quadtree를 이용한 방법은 공간의 차원을 다차원에서 일차원으로 낮춘 반면, 공간변형-예를 들어 Corner Transformation-에 의한 방법은 오히려 객체가 존재하는 공간의 차원을 높힘으로써 수행된다. 즉, 공간에 존재하는 점이 아닌 객체를 점으로 만들기 위해서 차원의 변환을 수행한다[2,3]. 이 차원 변환의 예는 그림 5와 같다.

일반적으로 차원의 변환은 위의 그림 5와 같이 다차원적 구간-일차원의 경우 구간, 이차원의 경우 직사각형-으로 공간객체를 단순화하여 구간의 최소값, 최대값의 좌표를 변환된 공간의 점-편의상 이를 점객체라고 부르겠다.-으로 만드는 과정으로 이루어진다. 이때, 변환된 공간은 재미있는 성질을 가지게 된다. 즉 그림 5의 예에서 나타난 것과 같이, 질의구간내 S에 포함되는 객체는 반드시 (I) 구역에 존재하는 점객체가 된다.

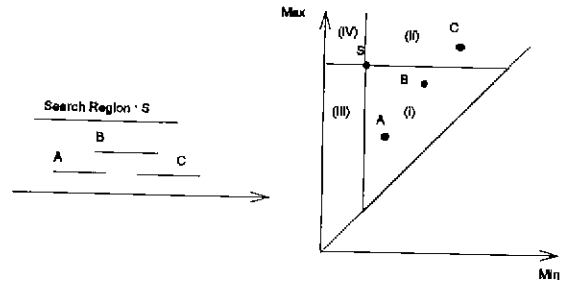


그림 5 차원의 변환

마찬가지로 질의구간과 겹쳐지는 객체는 (II)와 (III) 구간에 존재하게 된다. (IV) 구간에 존재하는 점객체는 질의구간 S와 전혀 겹쳐지 않는 것들이다.

따라서, 질의구간과 겹침, 포함됨과 같은 공간 질의는 변환된 공간에서 해당되는 구역에 존재하는 점객체를 찾아냄으로써 이루어진다. 그런데, 이 질의처리과정중 공간색인의 도움이 요구된다. 즉, 그림 5에서 주어진 질의구간에 포함되는 공간객체를 찾으려할 경우, 구간 (I)에 속하는 점객체를 찾아야 하는데 이는 변환된 공간에 주어지는 또 다른 공간질의가 된다. 물론, 이 새로운 질의는 원래의 질의보다 처리하기가 쉽지만, 공간색인의 도움없이는 질의처리성능이 매우 떨어지게 된다. 따라서, 변형된 공간의 질의처리를 위해서, 점객체에 적합한 공간색인방법을 사용하면 효과적이다. 일반적으로 이를 위해, Tri-Cell [3], LSD-tree[2], Grid File[27][28]이나 MLGF[16]와 같은 공간색인방법이 사용된다.

변환된 공간을 이용하는 질의처리방법은 위에서 예로 든 포함, 겹침과 같은 질의처리를 위해서는 매우 효과적이다. 그러나, Linear Quadtree를 이용한 질의처리방법과 마찬가지로, 처리할 수 있는 공간질의가 매우 제한되어 있다. 예를 들어, 가장 가까운 객체의 탐색, Buffer Zone에 의한 탐색등의 공간질의를 공간변환방법에 의해 구현하기는 매우 어렵다.

3.3 여과-정제(Filtering-Refinement) 방법

위의 두가지 종류의 방법은 공간의 차원을 바꾸는 공간색인을 이용하는 것이다. 공간의 차

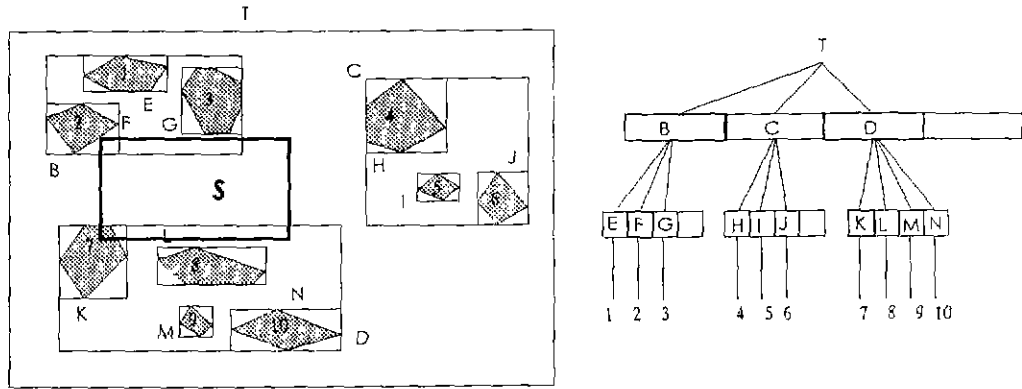


그림 6 R-tree와 여과-정제방법에 의한 공간질의처리

원을 바꾸는 과정에서 발견되는 특수한 성질을 이용한 공간질의처리방법이었다. 그러나, 여과-정제에 의한 방법은 공간의 차원을 변환하지 않는 방법에 기초한다. 공간의 차원을 변환하지 않고, 원래의 차원이 가지고 있는 여러가지 기하적인 성질을 이용하여 공간질의처리를 하는 방법이다[11,24].

여과-정제방법을 이용한 공간처리에서 가장 일반적으로 사용되는 공간색인방법은 R-tree계열이다. 그림 6은 공간안에 주어진 객체들을 R-tree로 만든 것이다. 이 때, 질의구간 S과 겹치는 객체를 찾으라는 질의를 주었다고 가정하자. 그러면, 여과-정제방법에 의한 질의처리는 다음과 같은 두가지 단계를 거쳐 수행된다.

단계1: 여과단계

R-tree 각 단계에서, 질의구간 S를 포함하는 MBR(Minimum Bounding Rectangle)에 해당되는 노드만 선택하여 자식노드로 내려간다. 단계별로, 이 과정이 반복되어 R-tree의 단말노드까지 도달하였을 때, 마지막에 남은 노드는 공간질의의 조건을 분명히 어기는 공간객체를 제외한 후보객체가 된다. 물론, 모든 후보객체가 질의를 만족하는 객체는 아니다. 이 후보객체는 단지, 그의 MBR이 질의구간과 겹친다는 것을 의미할 뿐이다. 실제로, 후보객체가 질의구간에 겹치는지 여부는 다음 단계인 정제단계에서 결정된다.

단계 2: 정제단계

정제단계에서는 후보객체가 실제로 질의구간과 겹치는지를 살펴본다. 이를 위해서는 먼저,

후보객체의 실제 구역에 관한 자료를 읽어와서, 이것이 질의구간과 겹치는지 여부를 조사한다. 이를 위해서는 여과단계에서와 같은 MBR을 통한 조사가 아니라, 실제 공간객체와 질의구간의 겹침을 조사한다. 이 조사에서 질의구간과 겹치는 객체가 질의조건을 만족하는 객체가 되어 사용자에게 전달된다.

여과-정제방법에 따른 질의처리방법의 성능을 결정하는 것은 여과율이다. 이 여과율은 다음과 같이 표현된다.

$$r = \frac{1 - N_c}{N}$$

N_c 는 후보객체의 수, N 은 총 객체의 수

이 때, 여과율이 작으면, 그 만큼 정제단계에서 조사하여야 할 객체의 수가 많아지고, 이에 따라, 많은 디스크입출력과 계산이 요구되어 질의처리의 성능이 떨어지게 된다. 반면에 여과율이 높으면, 그만큼 적은 수의 후보에 대한 정제만 하면되므로 질의처리의 성능이 향상된다. 여과율을 높이기 위하여, 경우에 따라서는 여과단계와 정제단계사이의 추가적인 여과단계를 삽입하기도 한다. 이를 위해서는 추가적인 정보가, 공간색인 과정에 주어져야하는데, 너무 많은 정보를 추가하면, 오히려 질의처리의 성능이 떨어질 수도 있다.

여과정제방법은 앞의 두가지 방법에 비하여 많은 장점을 가지고 있다. 먼저 첫번째 장점은, 다양한 종류의 공간질의를 위한 처리방법의 구

현이 쉽다는 것이다. 위의 두가지 방법은 겹침이나 포함을 위한 공간질의에는 효과적으로 처리될 수 있으나 그외의 질의처리방법에는 적용할 수 없다는 한계가 있는 반면, 여과-정제에 의한 방법은 거의 모든 공간질의에 적용할 수 있다는 장점이 있다. 이와 같은 이유로 많은 연구가 여과-정제방법을 위해 진행되고 있다.

3.4 Distance Join Index

공간질의 중 한가지 종류는 최소거리에 존재하는 객체, 또는 Buffer Zone에 포함되는 공간객체에 관한 질의이다. 예를 들어,

```
Select building.name
From building.station
Where station.name="부산" and DISTANCE(station,building)<400 m
```

과 같은 질의의 경우, 공간질의처리를 위한 방법중 한가지는 모든 building과 station 사이의 거리를 이미 저장하고 있고, 이를 이용하여 원하는 객체를 찾는 것이다. 만일 이와 같은 거리 정보를 이미 Join Index와 같은 형태로 가지고 있다면, 위와 같은 질의에 대한 처리는 쉽게 될 수 있다. 그러나, 현실적으로, 모든 객체사이의 거리를 이미 가지고 있다는 것은 매우 어렵다. 따라서, 적당한 정도로 객체사이의 거리 정보를

가지고, 필요한 경우 나머지 객체사이의 거리를 계산할 수 있도록하면 효과적일 수 있다.

Distance Join Index를 이용한 질의처리방법은 이와 같은 개념에 기초하고 있다[9,19,21]. 즉, 그림 7에서 나타난 것과 같이, 집중된 공간객체를 각기 하나의 집단으로 나누어, 각 집단내의 공간객체 사이의 거리를 Distance Join Index표로 각기 보관한다. 또한 각 집단의 대표객체를 설정하여, 이 대표객체사이의 거리를 또다른 Distance Join Index표로 보관한다. 물론, Distance Join Index는 검색의 속도를 빠르게 하기 위해서, 공간객체의 식별자(Object Identifier)로 B-tree를 구성한다. 만일 위와 같은 질의가 주어지면, 다음과 같은 과정으로 공간처리를 수행한다.

단계 1 :

주어진 객체(station="부산": O)가 속한 집단에 대해, 400 m 이내의 객체를 찾는다.

단계 2 :

다른 집단의 대표객체(O_k)와 주어진 객체와 주어진 객체가 속한 집단의 대표객체(O_i)와의 거리 합이 500 이하인 다른 집단을 선택한다.

즉, {G_k | Distance(O, O_i) + Distance(O_i, O_k) < 400}을 선택한다.

단계 3 :

단계 2에서 선택된 집단 G_k의 모든 객체에 대하여, 주어진 객체와의 거리를 계산하여, 400 m 이내의 것을 선택한다.

이와 같은 방법에 의한 공간질의처리는 거리에 관계된 질의에 대해서는 매우 효과적이나, 그외의 질의에 대해서는 처리가 곤란하다. 따라서, 이 방법도 역시, 거리에 관한 질의처리에만 국한되어 있다는 단점을 가지고 있다.

o1	o11	3
o1	o12	5
o11	o1	3
o11	o12	4
o12	o11	4
o12	o1	5

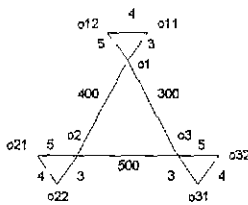
o2	o21	5
o2	o22	3
o21	o2	5
o21	o22	4
o22	o21	4
o22	o2	3

o3	o31	3
o3	o32	5
o31	o3	3
o31	o32	4
o32	o31	4
o32	o3	5

a) 각 공간집단내 객체사이의 거리

o1	o3	300
o1	o2	400
o2	o1	400
o2	o3	500
o3	o1	300
o3	o2	500

b) 각 집단의 대표객체사이의 거리



가) 공간객체사이의 거리

4. 공간질의처리방법의 비교적 연구

제 3장에서는 지금까지 제시된 공간질의처리방법에 대하여 개괄적으로 설명하였다. 그러나 효과적인 공간질의처리방법을 개발하는 것만큼 중요한 것은, 실제 응용분야가 결정되면 위에서 제시된 질의처리방법 중 한가지를 선택할 수 있는 기준을 제시하는 것이다. 그런데 불행히도,

그림 7 Distance Join Index에 의한 공간질의처리

이와 같은 기준을 제시하기 위한, 위의 방법들 사이의 비교적 연구가 거의 이루어지지 않았다 [4,10,14]. 즉, 위의 방법들 중에서 어떤 것이 가장 성능이 우수하다던가 아니면, 어떤 응용분야에 대해서는 어떤 방법이 가장 좋다던가 하는 것에 대한 연구가 지금까지 거의 없다.

좋은 공간질의처리방법이란 물론 성능이 좋은 방법을 말한다. 그런데, 지리정보시스템은 일반적으로 많은 양의 데이터를 가지고 있어, 많은 수의 디스크입출력을 필요로 한다. 따라서, 바람직한 공간질의처리방법이란, 가능한 적은 수의 디스크입출력을 통해, 원하는 결과를 얻어낼 수 있는 방법을 의미한다. 이와 더불어, 가능한 적은 계산(CPU Time)을 통해 원하는 결과를 얻어낼 수 있어야 한다. 전통적인 데이터베이스시스템의 질의처리와 다른 점중 하나가 바로 이점인데, 일반적으로 지리정보시스템에서 요구되는 공간 질의처리는 상대적으로 많은 양의 계산을 요구하게 된다. 대부분의 경우, 이 계산-주로 기하학적계산-이 전체 처리시간에서 차지되는 비율이 적지 않다는 사실은 디스크참조횟수 뿐아니라, 계산시간도 최소화하는 방법이 결국 바람직한 공간질의처리방법이 될 수 있다.

물론, 모든 응용분야에 대해, 완벽한 성능을 발휘하는 공간질의처리방법은 존재하지 않는다. 그러나, 적어도 주어진 응용분야에 대해서는 가장 우수한 공간질의처리방법이 무엇인가는 알아낼 수가 있다. 불행히도, 이것을 위한 비교적 연구도 매우 어려운 작업이다. 그 이유는 크게 다음과 같이 두가지로 요약할 수 있다.

첫번째 이유는, 공간질의처리방법은 공간색인 방법과 밀접하게 연관되어 있다는 점에서 비롯된다. 즉, 공간질의방법에 대한 비교적 연구는 공간색인방법에 대한 비교연구에 바탕을 두게되는데, 이 공간색인방법에 대한 비교적 연구가 아직 완전히 이루어지지 않았다는 것이다. 두번째 이유는, 각 공간질의처리방법의 성능은 주어진 응용분야의 환경(Workload)에 좌우되는데, 이 응용분야의 환경을 분석하는 것 자체가 매우 힘들다는 것이다. 즉, 응용분야의 환경을 특성화(Characterization)할 수 있는 요인(Parameters)을 발견하여, 이를 이용하여 각 공간질의처리방

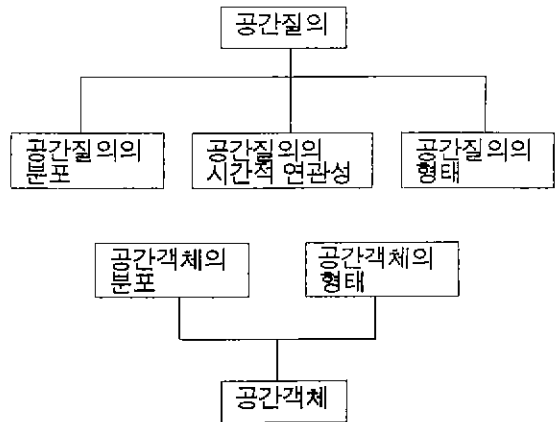


그림 8 응용분야를 결정하는 요인들

법을 비교하여야 하는데, 이 요인이 너무 다양하다는 것이다. 이것을 간단하게 분류하면 다음 그림 8과 같다.

공간질의처리방법에 대한 비교적 연구는 궁극적으로 세가지의 변수를 가진 하나의 표를 만드는 것이다. 즉, 한쪽 축은 그림 8에서 나열된 모든 요인을 수치화한 것이고, 다른 축은 각 공간색인방법이며, 마지막 축은 이에 따른 성능을 나타낸 것이다. 물론, 이와 같은 표를 완벽하게 만든다는 것은 거의 불가능하다. 그러나, 적어도 응용분야에 따른 공간질의처리방법의 선택에 도움을 주는 기준은 마련되어야하고, 이를 위한 연구는 절대적으로 필요하다.

이 비교적 연구의 단서를 제공할 수 있는 연구는 결국 응용분야의 환경에 대한 분석이다. 이것이 가능하여지면, 즉, 응용분야에 대한 특성화가 가능하여지면, 이에 따른 비교적 연구가 가능하여지리라 예상된다.

5. 맺음말

지금까지 지리정보시스템에서 요구되는 공간 질의의 처리방법에 대해서 개괄적으로 알아보았다. 지리정보시스템은 비교적 최근에 나타난 분야이어서 많은 문제가 미해결 상태로 남아 있다. 한가지 재미있는 사실은, 지리정보시스템에 대한 산업적 요구가 매우 많아, 아직 미해결된 여러 가지 문제가 존재함에도 불구하고, 이미 상품화

되어 사용되고 있다는 사실이다. 그러나, 이미 상품화된 많은 시스템이 존재한다는 것이 모든 문제가 해결되었다는 것을 의미하는 것은 아니다. 오히려, 지리정보시스템은 매우 빠르게 확장되는 분야이어서, 많은 문제들이 발견될 가능성이 더욱 많다.

지리정보시스템이 보다 광범위하게 사용되기 위해서 해결되어야 할 중요한 문제점중의 하나가 공간질의처리문제다. 일반적으로 지리정보시스템이 많은 양의 자료를 보관, 처리하기 때문에 응답시간이 길어지게 된다. 따라서, 시스템의 성능을 개량하기 위해서, 디스크참조횟수와 계산 시간을 최소화하는 공간질의처리방법을 개발하는 것이 매우 중요하다. 아직은 공간질의처리방법에 대한 많은 연구가 이루어지지 않았다. 많은 투자자가 이 공간질의처리방법에 이루어져야 한다. 앞으로 많은 관심을 가지고 투자가 이루어져야 할 부분을 열거하면 다음과 같다.

첫번째로, 앞의 장에서 잠깐 언급하였듯이 지금까지 제시된 공간질의처리방법에 대한 비교적 연구가 더욱 이루어져야 한다. 이를 위해서는 공간 통계학의 도움이 많이 요구된다. 공간 통계학은 응용분야에 따른 환경분석에 절대적으로 필요한 방법적 도구이다. 그러나, 불행히도, 이 분야는 통계학에 의해서도, 전산학에 의해서도 충분히 연구되지 않은 분야이다.

두번째, 아직, 구현이 완벽하게 이루어지지 않은 몇가지 종류의 질의-예를 들어, 선이나, 면의 Buffer Zone 질의, 방향성을 갖는 연결성에 관한 질의[7][23] 등-에 대한 처리방법개발과 구현이 이루어져야 한다. 이와 더불어, 응용분야에 따라, 쉽게 질의처리방법을 개발할 수 있는 도구나 환경을 만드는 것도 중요하다. 이는 지리정보시스템이 모든 종류의 질의에 대한 처리방법을 제공할 수는 없기 때문에 경우에 따라, 응용소프트웨어 개발자가 응용분야에 맞는 질의처리방법을 구현해야하기 때문이다.

세번째, 공간질의처리에 대한 병렬화도 중요한 분야이다[8,17]. 병렬화는 공간질의처리방법에 대한 성능개선에 많은 도움을 주리라 예상된다. 물론, 이 병렬 공간질의처리방법은 병렬 공간색인방법과 많은 관계가 있다.

네번째로, 공간적인 질의처리방법과 비공간적인 질의처리방법의 효과적인 결합이다. 일반적으로 많은 질의가, 공간적인 질의와 비공간적인 질의의 복합된 형태로 주어진다. 이 때, 이 두 종류의 질의에 대한 적절한 처리방법은 시스템의 성능을 상당히 개량할 수 있다.

참고문헌

- [1] T. Sellis, N. Roussopoulos, and C. Faloutsos, "The R+-tree: A Dynamic Index for Multidimensional Object", Proc. VLDB 1987, pp. 507-518.
- [2] A. Henrich, H.-W. Six, and P. Widmayer, "The LSD-tree: spatial access to multidimensional point and non point objects", Proc. VLDB 1989, pp. 45-53.
- [3] C. Faloutsos and W. Rego, "Tri-Cell - A Data Structure for Spatial Objects", Information Systems Vol. 14, No. 2, pp. 131-139, 1989.
- [4] O. Guenther and A. Buchmann, "Research Issues in Spatial Databases", ACM SIGMOD Record Vol. 19, No. 4, pp. 61-68, 1990.
- [5] N. B., H.-P. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: An Efficient and Robust Access method for Points and Rectangles", Proc. ACM SIGMOD'90, pp. 322-331.
- [6] H. V. Jagadish, "Linear Clustering of Objects with Multiple Attributes", Proc. ACM SIGMOD'90, pp. 332-342.
- [7] C. P. Kolovson and M. Stonebraker, "Segment Indexes: Dynamic Indexing Techniques for Multidimensional Interval Data", Proc. ACM SIGMOD'91, pp. 138-147.
- [8] I. Kamel and C. Faloutsos, "Parallel R-trees", Proc. ACM SIGMOD'92, pp. 195-204.
- [9] W. Lu and J. Han, "Distance-Associated Join Indices for Spatial Range Search", Proc. IEEE ICDE'92, pp. 284-292.
- [10] E. G. Hoel and H. Samet, "A Qualitative Comparison Study of Data Structures for Large Line Segment Databases", Proc. ACM SIGMOD'92, pp. 205-214.
- [11] T. Brinkhoff, H.-P. Kriegel and B. Seeger, "Efficient Processing of Spatial Joins Using R-trees", Proc. ACM SIGMOD'93, pp. 237-246.

[12] J. A. Orenstein and F. A. Mamola, "PROBE Spatial Data Modeling and Query Processing in an Image Database Application", *IEEE TOSE* Vol. 14, No. 5, pp. 611-629, 1988.

[13] M. Stonebraker, "The Design of the POSTGRES Storage System", *Proc. 13th VLDB 1987*, pp. 289-300.

[14] C. Faloutsos and I. Kamel, "Beyond Uniformity and Independence: Analysis of R-tree Using the Concept of Fractal Dimension", *Proc. ACM PODS'94*, pp. 4-13.

[15] B. Seeger and Hans-P.Kriegel, "The Buddy-Tree: An Efficient and Robust Access Method for Spatial Database System", *Proc. 16th VLDB*, pp. 590-601, 1990.

[16] K. Y. Whang and S. W. Kim, "Dynamic Maintenance of Data Distribution for Selectivity Estimation", *Journal of VLDB*, Vol. 3, pp. 29-51, 1994.

[17] J.Li, J.Srivastrava, and D.Rotem, "CMD: A Multidimensional Declustering Method for Parallel Database Systems", *Proc. 18th VLDB*, pp. 3-14, 1992.

[18] H. V. Jagadish, "On Indexing Line Segments", *Proc. 16th VLDB*, pp. 614-625, 1990.

[19] D. Rotem, "Spatial Join Indices", *Proc. IEEE ICDE'91*, pp. 500-509.

[20] M. L. Lo and C. V. Ravishankar, "Spatial Joins Using Seeded Trees", *Proc. ACM SIGMOD'94*, pp. 209-220, 1994.

[21] T. L. Whan and D. Shasha, "Query Processing for distance Metrics", *Proc. 16th VLDB*, pp. 602-613, 1990.

[22] J. Paredaens, J. V. den Bussche, and D. van Gucht, "Towards a Theory of Spatial Database Queries", *Proc. ACM PODS'94*, pp. 279-288.

[23] R. Agawal and H. V. Jagadish, "Algorithms for Searching Massive Graphs", *IEEE TOKDE* Vol. 6, No. 2, pp. 225-238, 1994.

[24] Hans-Peter Kriegel and Thomas Brinkhoff and Ralf Schneider, "Efficient Spatial Query processing in Geographic Database Systems", *IEEE Data Engineering Bulletin* Vol. 16, No. 3, pp. 10-15, 1993.

[25] M. J. Egenhofer, "Spatial SQL: A Query and Presentation Language", *IEEE TOKDE* Vol. 6, No. 1, pp. 86-95, 1994.

[26] R. Laurini and D. Thompson, "Fundamentals of Spatial Information Systems", Academic Press, 1992.

[27] H. Samet, "The Design and Analysis of Spatial Data Structures", Addison-Wesley, 1990.

[28] H. Samet, "Applications of Spatial Data Structures", Addison-Wesley, 1990.

이 기 준



1980 ~1984 서울대학교 계산통계학과 학사
 1984 ~1986 서울대학교 계산통계학과 석사
 1988 ~1992 프랑스 국립응용과학원(INSA) 전산학박사
 1984 ~1987 금성통신연구소 연구원
 1990 ~1991 프랑스 Logicim 엔지니어

1994 ~현재 (주)경제기획원 GIS 자문위원
 1993 ~현재 부산대학교 전자계산학과 전임강사
 관심 분야: 지리정보시스템 및 공간데이터베이스, 하이퍼미디어, 객체지향 소프트웨어공학