

대표점 필터를 이용한 공간조인 처리

(Spatial Join Processing using Representative Point Filter)

주 인 학 [†] 고 명 철 [†] 최 윤 철 ^{††} 고 권 ^{†††}
 (In-Hak Joo) (Myeong-Cheol Ko) (Yoon-Chul Choy) (Kyun Koh)

요 약 공간조인은 지리정보시스템에서 가장 중요한 연산 중의 하나이다. 본 논문에서는 공간조인 처리를 위하여 여과-정제 방법을 사용할 때 여과단계에서의 효과를 높이기 위하여 점 기반 approximation인 Representative Point (RP) 필터를 제안하였다. RP 필터는 실객체에서 선택된 몇 개의 점들로 구성된다. 이 점들을 객체의 근사치로 사용함으로써 공간조인의 여과단계에서 후보객체의 수를 줄이며, 따라서 공간조인의 수행 성능을 높일 수 있다. 또한 RP 필터의 여러 종류들을 제안하였으며 그 종류들의 성질과 성능을 비교분석하였다. RP 필터는 공간조인을 위한 기존의 approximation들에 비하여 생성 및 유지 비용이 매우 적으며 더 높은 필터율을 가진다. 또한 RP 필터는 기존의 approximation들과 달리 선객체에 대하여도 정의될 수 있기 때문에, 다각형 객체거리의 공간조인 뿐만 아니라 다각형 객체와 선 객체간의 공간조인에도 적용할 수 있다.

Abstract Spatial join is one of the most important operations in Geographic Information Systems (GIS). To improve the efficiency of the spatial join, this paper suggests Representative Point (RP) filter which is a point-based approximation. It is consisted of points selected from the vertex set of an object. By using the points as an approximation, we can reduce the number of candidates in filter step of spatial join processing. Consequently, RP filter can improve the performance of the spatial join. We suggest several kinds of RP filter and compare their performances. RP filter has less cost in creation and maintenance and higher filter ratio than established progressive approximations, resulting in more efficient processing of spatial join. Unlike the progressive approximations, it is applicable to spatial joins between arc object set and polygon object set as well as between two polygon object sets, because it is defined for both arc and polygon objects.

1. 서 론

지리정보시스템(GIS, Geographic Information System)은 다양한 분야에서 그 중요성이 날로 증가하고 있다. 대부분의 GIS 응용분야에서 공간데이터의 양은 매우 크고 공간객체는 복잡한 구조를 가지며, 연산의 수행에는 매우 많은 시간이 걸린다 [1,2,3,4]. 이러한 특성 때문에, GIS에서는 효율적인 공간데이터의 관리 및 검색이 매우 중요하다.

GIS에서의 공간질의 종류는 매우 많지만 공간조인은 가장 중요한 연산 중의 하나이다 [1,2,3,4]. 일반적으로 공간조인의 처리를 위하여는 한 객체의 기하 데이터를 반복적으로 읽어와야 하기 때문에 객체의 수가 많아지면 처리시간이 급격히 증가하며 [2,3,5,6], 공간조인은 GIS나 공간 데이터베이스 시스템에서 가장 비용이 큰 공간질의이다 [1,2,3,4,5,7].

공간조인 처리의 성능을 높이기 위하여 객체의 근사치를 사용하는 여과-정제 방법이 널리 연구되고 사용되고 있다 [1,2,3,4,5,6]. 여과-정제 질의처리 방법의 성능을 향상시키기 위해서는 여과단계에서 많은 객체들을 여과하고 후보객체의 수를 줄이는 것이 중요하며 [2,3,4,6,8,9] 이를 위한 필터를 고안하는 분야에 많은 연구가 진행되고 있다. 지금까지 연구된 필터 중 공간조인에서 필터율을 높이고 성능을 향상시키는 데 핵심적인 것은 공간객체에 포함되는 필터(progressive approxima-

[†] 비 회 원 : 연세대학교 컴퓨터과학과
tomcat@rainbow.yonsei.ac.kr
zoo@rainbow.yonsei.ac.kr

^{††} 종신회원 : 연세대학교 컴퓨터과학과 교수
yechoy@rainbow.yonsei.ac.kr

^{†††} 종신회원 : 칭주대학교 컴퓨터정보공학과 교수
kyunkoh@alpha94.chongju.ac.kr

논문접수 : 1999년 3월 3일

심사완료 : 2000년 7월 20일

tion)인 Maximum Enclosed Circle (MEC), Maximum Enclosed Rectangle (MER), Maximum Enclosed Line (MEL) 등이며 이들을 사용하면 정제단계를 거치지 않고 여과단계에서 조인조건을 만족하는 객체를 찾을 수 있다 [2,3,4].

그러나 현재까지 연구된 progressive approximation 들은 가늘고 긴 다각형 객체에 적용시 효율이 떨어지며 선 객체에는 적용할 수 없다는 문제점이 있다 [8,9,10]. 최근에 연구된 4CRS (Four-Color Raster Signature) 라고 하는 래스터 기반 approximation은 가늘고 긴 다각형 객체에도 적용할 수 있으면서 필터율을 높일 수 있지만 [7] 역시 선 객체에 적용할 수 없으며 생성·유지·연산하는데 드는 비용이 벡터기반 approximation에 비하여 크다는 단점이 있다.

본 논문에서는 이러한 approximation들의 단점을 극복하기 위하여 공간조인 수행에서 적은 비용으로 여과 단계에서의 처리 효과를 높일 수 있고 가늘고 긴 다각형과 선 객체에도 효과적으로 적용가능한 점 기반 approximation인 Representative Point (RP) 필터를 제안했으며 이를 R-트리기반의 Spatial Access Method (SAM)와 결합한 다중필터 방법을 제안하였다. RP필터는 여러 종류들을 가지며, 그 중 Maximum-Minimum Points (MMP) [8,9]와 Nearest-to-Corner Points (NCP) [10]는 이미 연구된 바와 같이 영역질의에 있어서 필터율을 높이고 수행성능을 향상시키는데 매우 효과적인 것으로 밝혀져 있다. 본 논문에서는 RP 필터를 이용해 공간조인을 처리하는 방안 대하여 연구하고, 이들이 영역질의 뿐만 아니라 공간조인의 처리에도 효과적임을 보인다. 제안된 RP 필터는 선객체에 대하여도 정의될 수 있기 때문에, 다각형 객체끼리의 공간조인 뿐만 아니라 다각형 객체와 선 객체간의 공간조인에도 적용할 수 있다. 또한 RP 필터는 생성하고 관리하는 비용이 기존의 필터의 비하여 매우 적다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구 및 연구배경에 대하여 설명하고, 3장에서 RP 필터 개념을 제안하였다. 4장에서는 이를 이용한 공간조인 방법을 제시하였다. 5장에서 제안한 방법의 실험자료와 성능평가를 제시하였으며 마지막으로 6장에서 향후연구 방향을 제시하며 결론을 맺었다.

2. 관련 연구

GIS에서 공간조인은 가장 중요한 연산 중의 하나이다 [1,2,3,4]. 공간조인은 두개의 map에 대하여 정의되며, 두 map의 카테시안 곱(Cartesian product)에 대하여 어

떠한 공간적 조건에 따라 그 부분집합을 계산하는 것이다. 이때 만족시켜야 할 공간적 조건을 조인조건(join predicate)이라고 한다 [1].

가장 전형적이고 많이 쓰이는 공간조인은 객체끼리 공통부분을 가지는 객체 쌍을 찾는 것으로 조인조건이 intersection인 intersection join이다 [1,2,3,4].

Intersection 외에도 조인조건으로 다른 공간 relation, 예를 들어 containment 등을 생각할 수도 있으나 공간조인을 수행하는 접근방법은 비슷하다[1]. 본 논문에서는 intersection join에 대하여 연구하였다. 공간조인의 다른 분류로는 MBR-조인, ID-조인, 객체-조인(object-join)이 있다[3]. MBR-조인은 객체의 Minimum Bounding Rectangle (MBR)끼리 겹치는 id 쌍을 찾는 것이며 주로 나머지 두 종류의 조인의 구현을 위한 여과단계로 사용된다. ID-조인은 객체끼리 겹치는 id 쌍을 찾는 것이며, 객체-조인은 ID-조인과 같으면서 겹치는 공통부분의 결과 객체까지 구하는 것이다 [3].

일반적으로 GIS에서 공간객체는 매우 복잡하며 많은 수의 vertex를 가지므로 보조기억장치로부터 읽어와야 하는 한 객체의 기하 데이터는 크기가 매우 크고 또한 공간객체 사이에 조인조건에 의하여 수행하는 연산도 매우 복잡하다 [1,2,3,4,5,7]. 또한 공간조인의 처리를 위하여는 한 객체의 기하 데이터를 반복적으로 읽어와야 하기 때문에 객체의 수가 많아지면 처리시간이 급격히 증가한다 [3,5,6]. 그러므로 공간조인은 GIS나 공간 데이터베이스 시스템에서 가장 비용이 큰 공간질의이다[1, 2,3,4,6,7]. 공간조인 처리의 성능을 높이기 위하여 객체의 근사치를 사용하는 여과-정제 방법이 널리 사용되고 있다 [1,2,3,4,5,6,7,8,9,10]. 여과-정제 방법은 대상객체 중에서 후보객체를 찾아내는 여과단계(filter step)와 후보객체들 중에서 최종 조인조건을 만족하는 객체들을 찾아내는 정제단계(refinement step)로 나누어 처리하는 2단계 질의처리방법이다.

여과단계에서는 SAM에 의하여 MBR-조인을 수행한다. MBR-조인은 조인조건을 만족하지 않는 객체 쌍 중 대부분을 빠른 속도로 걸러내며 그 결과로 조인조건을 만족할 가능성이 있는 후보객체 쌍의 집합을 얻는다. 공간조인에서 MBR-조인 단계의 처리를 위하여는 SAM이 제공되어야 한다. Quadtree [11] 또는 다차원의 Point Access Method (PAM)인 kd-tree [12] 또는 grid file [13] 등 다양한 종류의 SAM이 사용될 수 있으며 각각 장단점을 가지고 있다. SAM에 대한 자세한 분류와 정리는 [14,15]에 소개되어 있다. 공간조인 수행

을 위하여 가장 널리 쓰이는 것은 R-tree [16] 계열의 인덱스 구조이며 [2,3,5,6] 그중에서도 intermediate join index를 이용한 breath-first R-tree join 방법이 MBR-조인단계를 수행하는데 가장 우수한 것으로 알려져 있다 [17].

정제단계에서는 이들의 실제의 기하적 데이터를 보조 기억장치로부터 읽어와서 조인조건에 의하여 연산한다. 일반적으로 근사치를 다루는 여과단계보다 실제 기하 데이터를 다루는 정제단계의 비용이 훨씬 크기 때문에 질의처리의 성능을 향상시키기 위해서는 여과단계에서 많은 객체들을 여과하고 최소한의 후보객체들을 찾아내는 것이 중요하다 [2,3,4,6,8,9]. 여과-정제 방법에 따른 질의처리 방법의 성능을 결정하는 것은 필터율이다 [2]. 필터율이 낮으면 정제단계에서 처리해야 할 객체의 수가 많아지므로 성능이 떨어지게 되며, 필터율이 높으면 후보객체의 수가 줄어들므로 성능이 좋아지게 된다 [2]. 여과-정제 방법은 공간조인 외에도 다양한 종류의 공간 질의처리에 적용할 수 있으며 개념이 간단하고 구현이 용이하다는 장점이 있다.

또한 여과단계에서 더 많은 객체를 여과시키고 후보 객체의 수를 줄이기 위하여, 여과-정제 방법은 여러 개의 서로 다른 기하적 필터를 이용하여 여과단계를 여러 번 수행하는 다단계필터 방법 (multi-step filter method)으로 확장될 수 있으며 적절한 필터를 선택하면 정제단계에서의 비용을 더욱 줄이고 성능을 더욱 향상시킬 수 있다 [1,2,4,5,6,8,9]. 다단계필터 방법에서는 주로 첫번째 필터로 MBR이 사용되며 두번째 필터로 객체를 더 정확히 근사하는 필터가 사용된다. 필터의 선택은 근사 정확도와 저장공간의 적절한 절충 및 응용분야의 특성에 의한다 [1,2,4]. 공간조인 처리는 다음 세 단계로 이루어진다. 먼저 MBR-조인이 수행되어 근사적인 공간조인을 수행한다. 여기서는 검색 공간을 줄여주기 위하여 SAM의 지원이 필요하며, 조인 조건을 만족하지 않는 객체쌍 중에서 대부분을 판별해 내고 그 결과로 후보 객체쌍의 집합을 구한다. 둘째 근사 정확도가 더 높은 기하적 필터가 적용되어 실제 기하 데이터에 접근하지 않고 hit (또는 false hit) 을 식별해 내며, 마지막으로 실제의 기하 데이터를 이용하여 조인조건을 검사하여 후보객체쌍 집합으로부터 최종 질의결과를 얻는다 [1,2,3,4,5,6].

지금까지 연구된 필터는 공간객체를 포함하는 필터 (conservative approximation)와 공간객체에 포함되는 필터(progressive approximation)의 두 가지 부류로 분류된다 [2,4,6]. Conservative approximation에는

MBR, Rotated Minimum Bounding Rectangle (RMBR), Minimum Bounding Circle (MBC), Minimum Bounding Ellipse (MBE), Convex Hull (CH), Minimum Bounding n-corner (n-C) 등이 있다. Progressive approximation에는 MEC, MER, MEL 등이 있다 [2,4,6]. 각 필터들은 서로 다른 특성과 장단점을 가지며 [2]에서 이들 필터에 대한 비교를 제시하였다. 공간조인의 성능을 높이는 방법은 conservative approximation을 사용하여 false hit을 식별해내는 방법과 progressive approximation을 사용하여 hit를 식별하는 방법 두 가지가 있다. 두 가지 방법 모두 후보객체의 수를 줄이기 위한 것이며 일반적으로 후자가 더 효과적인 것으로 알려져 있다 [2,6]. 그러므로 progressive approximation은 정제단계를 거치지 않고 여과단계에서 조인조건을 만족하는 객체를 찾을 수 있는 필터로서, 다단계 여과방법에서 필터율을 높이고 성능을 향상시키는 데 핵심적인 요소이다. 본 논문에서 제안한 RP필터는 공간조인에서 hit를 많이 식별함으로써 공간조인의 수행 성능을 높인다.

그러나 현재까지는 다각형 객체에 대한 공간분석을 위한 progressive approximation들만 연구되었으며 이들은 정사각형에 가까운 모양의 다각형 객체에 적용시에는 효율적이지만 가늘고 긴 다각형 객체에 적용시 효율이 떨어지며 다각형 객체에 포함되는 면이나 선으로 정의되기 때문에 선 객체에는 적용을 할 수 없다는 문제점이 있다 [8,9,10].

최근에는 4CRS라고 하는 래스터 기반 approximation도 연구되었다 [7]. 4CRS에 의한 공간조인 방법은 각 객체에 대하여 4CRS를 구하고 이들 래스터의 중첩에 의하여 객체간 공통부분이 있나 검사하는 것이다. [7]에 의하면 공간조인의 시간성능을 약 40% 정도 향상시킬 수 있다. 그러나 4CRS는 역시 선 객체에 적용할 수 없으며 생성,유지하는데 드는 비용이 벡터기반 approximation에 비하여 크며 래스터 셀로의 변환, 이를 압축, 복원하는 과정, 중첩할 때의 좌표 조정 등 복잡한 과정이 필요하므로 큰 overhead가 된다. 특히 공간 데이터의 수정이 자주 필요한 경우에는 이러한 overhead는 더욱 큰 약점이 된다.

본 논문에서는 공간조인 수행에서 여과단계에서의 처리 효과를 높이기 위하여 Representative Point (RP) 필터를 제안했으며 이를 R-트리기반의 SAM과 결합한 다중필터 방법을 제안하였다. RP필터는 여러 종류들을 가지며, 기존의 점 기반 필터인 MMP [8,9]와 NCP [10]는 RP필터의 범주에 포함시킬 수 있다. 이들은 영역질

에 있어서 필터율을 높이고 수행성능을 향상시키는데 매우 효과적인 것으로 밝혀져 있다. 본 논문에서는 RP 필터를 이용해 공간조인을 처리하는 방안에 대하여 연구하고, 이들이 영역질의 뿐만 아니라 공간조인의 처리에도 효과적임을 보인다. 제안된 RP 필터는 기존의 progressive approximation (MEC 31~32%, MER 34~36% 정도) [2,4,6] 에 비하여 더 높은 약 43~45% 정도의 필터율을 가진다. 제안된 RP 필터는 선객체에 대하여도 정의될 수 있기 때문에, 다각형 객체끼리의 공간조인 뿐만 아니라 다각형 객체와 선 객체간의 공간조인에도 적용할 수 있다. 또한 RP 필터는 객체의 MBR 생성시에 함께 생성할 수 있으므로 필터를 생성하고 관리하는 비용이 기존의 필터의 비하여 매우 적다.

3. RP 필터

3.1 정의

공간객체를 a 라고 할 때 a 의 RP 필터는 공간객체 a 의 정점(vertex)들 중에서 추출한 몇 개의 점들의 순서화된 리스트로 정의되는 필터이다.

$$RP(a) = \{ P_1, P_2, \dots, P_n \}$$

where $P_i \in \text{vertex set of } a \text{ for } i=1, 2, \dots, n$

RP필터를 구성하는 정점 P_i 를 RP 점이라고 한다. RP 필터는 실객체 데이터의 일부분이기 때문에 intersection join 에 대하여 두 객체간에는 다음 성질이 성립한다. RP 필터의 연산과 필터 적용은 모두 이 성질에 근거하여 수행된다.

• 한 객체의 RP점 중 적어도 하나가 다른 객체의 내부에 있으면 두 객체는 공통부분을 가진다.

$$\text{Inside}(a, RP(b)) \Rightarrow a \cap b \neq \emptyset$$

$$\text{Inside}(b, RP(a)) \Rightarrow a \cap b \neq \emptyset$$

한편 위의 성질과 대칭적으로, 한 객체의 RP점 중 적어도 하나가 다른 객체의 외부에 있으면 그 객체는 다른 객체에 포함되지 않는다는 것도 알 수 있다. 이 성질은 조인조건이 containment인 공간조인의 경우에 사용될 수 있다.

3.2 RP필터 점들의 추출 원칙

RP필터의 성능을 높이기 위하여, 즉 필터율을 높이기 위하여는 RP 점들의 추출 방법이 가장 중요하다. 즉 3.1절에서 논의한 inside 조건을 만족할 가능성이 상대적으로 높은 점을 선택하는 것이 목적이 된다. 이를 위하여 가장 먼저 고려되어야 할 것은 두 객체가 겹치는 형태이다. 두 객체의 MBR이 서로 겹치는 형태는 MBR의 변 또는 구석점이 서로 겹치는 경우를 고려하지 않는다면 크게 [그림1]의 네 가지로 분류할 수 있다([10]

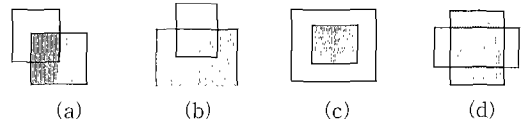


그림 1 두 객체의 MBR이 서로 겹치는 형태

의 분류를 참고정리).

[그림1]과 객체의 모양으로부터 3.1절에서 논의한 inside 조건을 만족할 가능성이 상대적으로 높은 점을 선택하기로 한다. [그림2(a)]의 경우에는 한 객체를 구성하는 점들 중에서 객체MBR(회색 부분)의 왼쪽 위 구석점에서 가장 가까운 점이 다른 객체에 포함될 가능성이 가장 크다고 할 수 있다. [그림2(b)]의 경우는 객체 MBR의 위쪽 변의 중점에서 가장 가까운 점이며, [그림2(c)]의 경우는 객체MBR의 중심점에서 가장 가까운 점이 각각 다른 객체에 포함될 가능성이 가장 크다고 할 수 있다. [그림2(d)]는 무조건 객체끼리의 겹침이 보장되는 형태이므로 [2] 고려하지 않아도 된다.

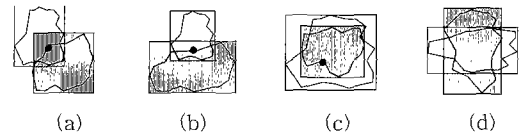


그림 2 Inside 조건을 만족할 가능성이 상대적으로 높은 점

[그림2(a)]에서는 왼쪽 위를 예로 들었지만 이러한 점들은 왼쪽 아래, 오른쪽 위, 오른쪽 아래에서도 대칭적으로 구할 수 있다. 이러한 점 네 개를 모아서 RP필터를 구성하는 방법을 생각할 수 있다. [그림2(b)]의 경우에도 RP필터는 마찬가지로 점 네 개로 구성될 수 있으며, [그림2(c)]의 경우는 하나의 점으로 구성된다. 이러한 경우처럼 어떠한 하나의 원칙을 선택하여 선정된 RP 점들의 순서화된 리스트를 RP필터의 종류 라고 하기로 한다. 여기서 두개의 map의 객체 크기, 즉 객체의 MBR의 크기는 대체로 비슷하며, 객체의 모양도 원 또는 불록다각형에 가깝다고 가정한다면 [그림2(a)]와 [그림2(b)]의 경우가 대부분을 차지하며 또한 inside 조건도 만족될 가능성이 높다고 할 수 있다. 본 연구에서의 실험에 의하면 두 MBR이 서로 겹치는 경우 중에서 [그림2(a)]와 [그림2(b)]의 경우가 각각 57%와 34%를 차지한다.

이상에서 논의한 내용들은 성능이 우수한 RP 필터의 종류를 만들어 내기 위한 중요한 착안점이 된다. RP필터의 종류에 대하여는 다음 절에서 자세히 설명하였다.

3.3 RP필터의 종류

RP 필터는 그를 구성하는 RP 점들의 개수와 추출 원칙에 따라 여러가지 종류를 가질 수 있다. 3.2절에서 논의한 내용에 의하여 몇 가지 종류들을 제안하였다.

3.3.1 Random 추출

RP 필터를 구성하는 점들이 객체의 vertex 점들 중에서 무작위로 추출된 경우이다. 이 방법은 추출을 위하여 연산이 전혀 필요없는 장점이 있는 반면 공간적인 특성이 전혀 없다는 단점이 있다. 점들의 수가 너무 적으면 필터율이 낮아지며 너무 크면 점들을 위한 저장 공간의 overhead를 가져온다.

3.3.2 MMP (Maximum-Minimum Points)

객체의 정점들 중 x, y 각 변수에 대하여 최대값과 최소값을 가지는 점 4개로 정의된다 (그림3).

$$(P_{max_x}, P_{min_x}, P_{max_y}, P_{min_y})$$

where

$$P_{max_x} = (x, y) \in V \text{ such that } x \geq x' \quad \forall (x', y') \in V;$$

$$P_{min_x} = (x, y) \in V \text{ such that } x \leq x' \quad \forall (x', y') \in V;$$

$$P_{max_y} = (x, y) \in V \text{ such that } y \geq y' \quad \forall (x', y') \in V;$$

$$P_{min_y} = (x, y) \in V \text{ such that } y \leq y' \quad \forall (x', y') \in V.$$

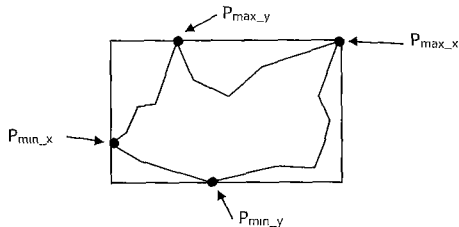


그림 3 다각형 객체의 MMP

정의에 따라 구해진 네 개의 점들 중 두 개 이상은 같은 점이 될 수도 있으나 $(P_{max_x}, P_{min_x}, P_{max_y}, P_{min_y})$ 는 순서가 있는 리스트이므로 이러한 경우는 같은 점을 중복해서 가진다. 괄호 안에 표기한 점들은 집합이 아니며, 순서화되고 중복이 허용되는 리스트를 이룬다. 이런 원칙은 RP필터의 다른 종류들의 경우에도 마찬가지이다. 또한 각 점의 정의를 만족하는 점이 두 개 이상인 경우도 존재한다. MBR의 한 변에 접하는 점들이 두 개 이상일 경우를 말한다. 이러한 경우에는 그 중 하나를 임의적으로 선택하여도 문제가 없으며 정의에도 위배되지 않는다. 역시 RP필터의 다른 종류들의 경우에도 마찬가지로 적용된다.

3.3.3 NCTP (Nearest-to-Center-Point)

NCTP 는 객체의 정점들 중 객체의 MBR의 중심점

으로부터 가장 가까운 점으로 정의된다 (그림4).

$$(N_{center})$$

where

$$MBR = (left, top, right, bottom),$$

$$Center = \left(\frac{left + right}{2}, \frac{top + bottom}{2} \right),$$

$$N_{center} = (x, y) \in V \text{ such that } (x, y) \sim Center \leq (x', y') \sim Center \quad \forall (x', y') \in V$$

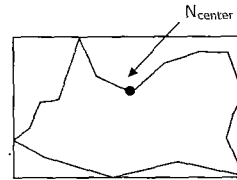


그림 4 다각형 객체의 NCTP

NCTP 는 parameter의 수가 2로 RP필터의 다른 종류들에 비하여 적지만 필터율은 그만큼 떨어진다. NCTP는 [그림2(c)]에 해당하는 경우이다.

3.3.4 NSP (Nearest-to-Side Points)

NSP는 객체의 정점들 중 객체의 MBR의 각 네 변의 중점에서 가장 가까운 점 4개로 정의된다.

$$(S_{left}, S_{top}, S_{right}, S_{bottom})$$

where

$$MBR = (left, top, right, bottom),$$

$$S_{left} = (x, y) \in V \text{ such that } (x, y) \sim \left(left, \frac{top + bottom}{2} \right) \leq (x', y') \sim \left(left, \frac{top + bottom}{2} \right) \quad \forall (x', y') \in V$$

$$S_{top} = (x, y) \in V \text{ such that } (x, y) \sim \left(\frac{left + right}{2}, top \right) \leq (x', y') \sim \left(\frac{left + right}{2}, top \right) \quad \forall (x', y') \in V$$

$$S_{right} = (x, y) \in V \text{ such that } (x, y) \sim \left(right, \frac{top + bottom}{2} \right) \leq (x', y') \sim \left(right, \frac{top + bottom}{2} \right) \quad \forall (x', y') \in V$$

$$S_{bottom} = (x, y) \in V \text{ such that } (x, y) \sim \left(\frac{left + right}{2}, bottom \right) \leq (x', y') \sim \left(\frac{left + right}{2}, bottom \right) \quad \forall (x', y') \in V$$

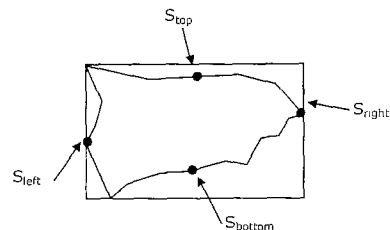


그림 5 다각형 객체의 NSP

NSP는 [그림2(b)]에 해당하는 경우이다. [그림5]는 다각형 객체의 NSP의 예를 보여준다.

3.3.5 NCP (Nearest-to-Corner Points)

공간객체의 정점들 중 그 MBR의 네 구석점에서 가장 가까운 거리를 갖는 점 4개로 정의된다 (그림6). NCP는 [그림2(a)]에 해당하는 경우이다.

$$(N_{left_top}, N_{left_bottom}, N_{right_top}, N_{right_bottom})$$

where

$$MBR=(left, top, right, bottom),$$

$$N_{left_top}=(x, y) \in V \text{ such that } (x, y) \sim (left, top) \leq (x', y') \sim (left, top) \quad \forall (x', y') \in V$$

$$N_{left_bottom}=(x, y) \in V \text{ such that } (x, y) \sim (left, bottom) \leq (x', y') \sim (left, bottom) \quad \forall (x', y') \in V$$

$$N_{right_top}=(x, y) \in V \text{ such that } (x, y) \sim (right, top) \leq (x', y') \sim (right, top) \quad \forall (x', y') \in V$$

$$N_{right_bottom}=(x, y) \in V \text{ such that } (x, y) \sim (right, bottom) \leq (x', y') \sim (right, bottom) \quad \forall (x', y') \in V$$

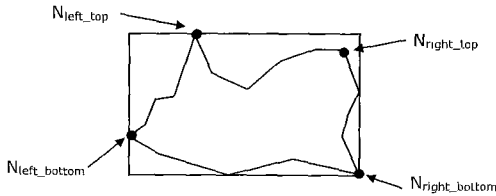


그림 6 다각형 객체의 NCP

3.3.6 AMP (Area-Maximizing Points)

공간객체의 AMP는 공간객체의 정점들 중 다음 식을 만족하는 점 4개로 정의된다.

$$(M_{left_top}, M_{left_bottom}, M_{right_top}, M_{right_bottom})$$

where

$$MBR=(left, top, right, bottom),$$

$$M_{left_top}=(x, y) \in V \text{ such that } |x-right| |y-bottom| \geq |x'-right| |y'-bottom| \quad \forall (x', y') \in V$$

$$M_{left_bottom}=(x, y) \in V \text{ such that } |x-right| |y-top| \geq |x'-right| |y'-top| \quad \forall (x', y') \in V$$

$$M_{right_top}=(x, y) \in V \text{ such that } |x-left| |y-bottom| \geq |x'-left| |y'-bottom| \quad \forall (x', y') \in V$$

$$M_{right_bottom}=(x, y) \in V \text{ such that } |x-left| |y-top| \geq |x'-left| |y'-top| \quad \forall (x', y') \in V$$

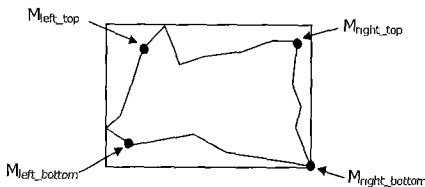


그림 7 다각형 객체의 AMP

AMP도 [그림2(a)]에 해당하는 경우이다. [그림7]은 다각형 객체의 AMP를 예로 나타내고 있다.

3.4 RP필터의 생성 비용

RP 점들의 선택 방법에서 또 하나 중요한 요소는 계산 비용이 크지 않아야 한다는 것이다. 특히 공간 데이터의 수정이 자주 필요한 경우에는 approximation의 수정 비용이 클 경우 overhead를 가져올 수 있다. 일반적으로 공간객체는 많은 수의 vertex를 가지며, vertex의 수가 커지면 공간객체의 approximation을 생성,수정하는 비용이 커진다.

본 논문에서 제안한 RP필터의 종류들은 모두 vertex의 수에 선형비례하는 시간에 생성하고 수정할 수 있다. 이는 RP필터 각 종류들의 정의로부터 명백하다. 즉 vertex들을 단일스캔하며 간단한 값만을 계산하고 그값을 최대 또는 최소로 될 때의 점을 선택하는 것이므로 vertex의 수에 비례하는 연산을 필요로 한다.

객체의 vertex의 수를 n이라 할 때 RP필터의 생성 비용은 O(n) 이며, 기존의 approximation들과의 생성 비용은 일반적인 다각형의 경우 MER이 O(nlog²n) [18], MEC가 O(nlogn) [19] 이다.

3.5 RP필터의 적용

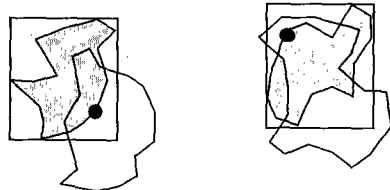
RP필터의 적용은 다음 두 가지로 이루어진다 (그림 8).

- Inside test.

Intersection join의 경우 한 객체의 RP를 구성하는 점들 중 적어도 하나가 다른 객체에 포함되는지 검사한다. 이 검사가 성공하면 이 객체 쌍은 조인조건을 만족하는 hit이다. 이를 Inside(p)로 나타낸다.

- Outside test.

한 객체의 RP를 구성하는 점들 중 적어도 하나가 다른 객체의 외부에 있는지 검사한다. 이 검사는 containment join의 경우에 사용하면 된다. 검사가 성공하면 이 객체 쌍은 조인조건을 만족하지 않는 false hit이다. 이를 Outside(p)로 나타낸다. 이 검사는 의미상 위의 것과 대칭적이다.



(a) Inside(p)

(b) Outside(p)

그림 8 RP 필터의 두 가지 적용

주로 많이 사용되는 intersection join의 경우 Inside(p)를 적용하고, containment join의 경우 Outside(p)를 적용한다. 두 경우 모두에 대해 검사가 성공하지 못하면 객체 쌍은 여전히 후보객체로 남게 되어 정제단계로 보내진다. [그림6]의 경우 Inside(p)는 성공하여 hit로 판별된 경우이며 Outside(p)는 성공하지 못하여 결과를 아직 결정하지 못한 경우이다. 이런 성질은 RP필터가 실객체를 구성하는 점의 일부이기 때문에 성립한다.

3.6 RP 필터를 R-tree 에 포함

공간조인 수행을 위하여 RP필터는 R-tree 구조에 한 구성요소로서 포함된다 (그림 9). 이것은 R-tree에 의한 MBR-조인 단계 이후에 보조기억장치에 대한 접근 없이도 RP필터를 가져올 수 있게 해준다. R-tree의 루트노드와 중간노드 구조는 기본적으로 변화가 없으며, 단지 리프노드에 있는 객체 id와 MBR 에 추가하여 RP 필터가 저장된다. RP 필터를 위한 저장공간은 최대의 경우 MBR의 두 배이며 리프노드에만 저장되므로 저장공간상의 큰 overhead는 되지 않는다. 다른 approximation들도 이런 식으로 리프노드에 저장될 수 있으며 저장공간에 대하여는 5.5절에서 간단히 비교하였다.

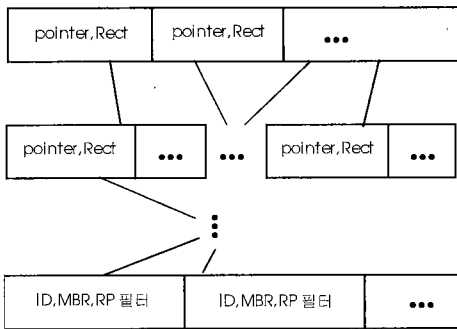


그림 9 RP필터를 포함한 R-tree의 구조

4. RP 필터를 이용한 공간조인 방법

4.1 전통적인 MBR-조인 기반 방법

MBR-조인을 수행한 후 필터단계 없이 모든 후보 객체 쌍들에 대하여 두 객체의 기하 데이터를 각각 읽어와서 두 객체의 공통부분이 있는가를 검사한다. 이 방법은 매우 많은 양의 디스크 접근을 필요로 한다.

4.2 One-sided RP 필터

MBR-조인을 거친 후보객체 쌍에 대하여 실제 기하 데이터를 읽어와 연산하기 전에 RP필터에 의한 여과단계를 수행한다. RP필터는 다른 approximation들과 달

```

JOIN-1 : conventional MBR-조인
MBR-join( );
For all candidate pairs ( a, b )
{
    Get Geometry(a)
    Get Geometry(b)
    If ( Overlap ( a, b ) == TRUE )
        pair ( a, b ) is a hit
    Else
        pair ( a, b ) is a false hit
}
    
```

```

JOIN-2 : one-sided RP 필터
MBR-join( );
For all candidate pairs ( a, b )
{
    Get RP(a)
    Get Geometry(b)
    If ( Inside ( b, RP(a) ) == TRUE )
        Pair ( a, b ) is a hit
}
For all candidate pairs ( a, b )
{
    Get Geometry(a)
    Get Geometry(b)
    If ( Overlap ( a, b ) == TRUE )
        Pair ( a, b ) is a hit
    Else
        Pair ( a, b ) is a false hit
}
    
```

리 점으로 구성된 필터이기 때문에 양쪽 map에 대하여 모두 근사치를 가져와 근사치끼리만 연산해서 hit를 판별해 낼 수는 없다. 이를 해결하기 위하여 한쪽 map만 보조기억장치로부터 실제 기하 데이터를 가져오고 다른 쪽 map은 RP필터를 가져와서 사용한다. 이때 RP필터는 주기억장치에 존재하는 R-tree의 리프노드에 포함되어 있으므로 보조기억장치로부터 읽어올 필요가 없다. 이러한 RP필터 적용을 one-sided RP 필터 방법이라고 부르기로 한다.

4.3 Two-sided RP 필터

4.2의 방법과 약간 다르게, RP필터 단계로부터 나온 후보객체 쌍에 대하여 정제단계를 수행하기 이전에 여과단계를 한번 더 수행하는 방법이다. 두 번째 여과단계에서는 양쪽 map의 역할, 즉 근사치(RP필터)를 가져오는 쪽과 기하 데이터를 가져오는 쪽을 서로 반대로 한다. 이러한 RP필터 적용을 two-sided RP 필터 방법이라고 부르기로 한다.

위에서 논의된 JOIN-2와 JOIN-3 방법의 문제점은 여과단계에서 hit로 판별되지 않은 경우 여과단계에서 가져온 기하 데이터를 정제단계에서 한번 다시 가져오게 된다는 것이다. 이것은 필터율이 매우 높지 않은 이

JOIN-3: two-sided RP 필터
<pre> MBR-join(); For all candidate pairs (a, b) { Get RP(a) Get Geometry(b) If (Inside (b, RP(a)) == TRUE) Pair (a, b) is a hit Else { Get RP(b) Get Geometry(a) If (Inside(a, RP(b)) == TRUE) Pair (a, b) is a hit } } For all candidate pairs (a, b) { Get Geometry(a) Get Geometry(b) If (Overlap (a, b) == TRUE) Pair (a, b) is a hit Else Pair (a, b) is a false hit } </pre>

상 I/O 시간면에서 상당한 중복이 되며 성능을 오히려 저하시킬 수 있다. 이를 해결하기 위해 여과단계에서 b의 기하 데이터를 읽어와서 RP필터의 inside test를 수행하고 hit가 되지 않을 경우 바로 a의 기하 데이터를 읽어와서 객체끼리 공통부분이 있는지 연산을 한다. 이 방법에 의하면 필요없이 중복된 기하 데이터 접근 횟수를 줄임으로써 정제단계 수행시간을 약 반 정도로 줄여 줄 수 있다. 다음의 JOIN-4와 JOIN-5는 각각 one-sided RP필터 방법인 JOIN-2와 two-sided RP필터 방법인 JOIN-3을 개선한 방법이다.

4.4 One-sided RP 필터와 정제단계가 결합된 방법

JOIN-4: one-sided RP 필터+정제단계가 결합된 방법
<pre> MBR-join(); For all candidate pairs (a, b) { Get RP(a) Get Geometry(b) If (Inside (b, RP(a)) == TRUE) Pair (a, b) is a hit Else { Get Geometry(a) If (Overlap (a, b) == TRUE) Pair (a, b) is a hit Else Pair (a, b) is a false hit } } </pre>

4.5 Two-sided RP 필터와 정제단계가 결합된 방법

JOIN-5: two-sided RP 필터+정제단계가 결합된 방법
<pre> MBR-join(); For all candidate pairs (a, b) { Get RP(a) Get Geometry(b) If (Inside (b, RP(a)) == TRUE) Pair (a, b) is a hit Else { Get RP(b) Get Geometry(a) If (Inside (a, RP(b)) == TRUE) Pair (a, b) is a hit Else { If (Overlap (a, b) == TRUE) Pair (a, b) is a hit Else Pair (a, b) is a false hit } } } </pre>

4.6 선 객체에 적용

공간조인을 위하여 연구된 기존의 필터들은 다각형 객체에만 적용가능하고 선 객체에는 적용할 수 없다. 그러나 본 연구에서 제안한 RP필터는 다각형 객체끼리의 공간조인 뿐만 아니라 다각형 객체와 선 객체간의 공간조인에도 적용할 수 있다. RP필터는 선 객체에도 정의될 수 있으므로, one-sided RP 방법(JOIN-2 또는 JOIN-4)에서 선 객체를 a로 하고 다각형 객체를 b로 하여 마찬가지로 수행이 가능하다. 한편 선 객체끼리의 공간조인에는 RP필터도 여전히 적용할 수 없다. 이것은 다른 기존의 progressive approximation을 적용할 때도 마찬가지인 한계점이다.

4.7 각 공간조인 방법의 성능 비교

위에서 논의한 JOIN-1, JOIN-2, JOIN-3, JOIN-4, JOIN-5 각 알고리즘의 시간 성능을 다음과 같이 분석한다. 각 방법 모두에서 수행시간의 대부분은 객체의 기하 데이터를 읽어오는 시간과 실제 객체간의 intersection 연산 시간이다. 객체의 기하 데이터는 보조 기억장치에 저장되어 있으며 크기도 근사치에 비하여 매우 크기 때문에 기하 데이터를 다루는 시간은 근사치(RP)를 다루는 시간에 비하여 매우 크다. 근사치를 access하는 시간과 RP의 inside test에 걸리는 시간은 상대적으로 무시하기로 한다.

그러면 각 방법에 소요되는 시간은 한 객체의 기하 데이터를 읽어오는 횟수와 실제 객체간의 intersection 연

표 1 각 공간조인 방법의 성능 비교

c : 기하 데이터 읽는 시간, d: 객체간의 intersection test 시간

조인 방법	수행 시간
JOIN-1	$n * (2c + d)$
JOIN-2	$n * ((3-2r)c + (1-r)d)$
JOIN-3	$n * ((5-5r+2r^2)c + (1-r)^2d)$
JOIN-4	$n * ((2-r)c + (1-r)d)$
JOIN-5	$n * ((3-3r+r^2)c + (1-r)^2d)$

산 횟수로 나타낼 수 있게 된다. MBR-조인에서 나온 후보 객체 쌍의 수를 n, RP필터의 필터율을 r이라 할 때 각 알고리즘에 소요되는 시간을 위의 두 가지 시간 요소로 표현하면 [표1]과 같다.

이 경우 c와 d 간의 관계는 객체를 구성하는 점의 수 및 실험 환경에 따라 달라질 수 있으며 본 연구의 실험에 의하면 10:1 정도이다. 이러한 가정하에, 여과단계를 전혀 두지 않은 JOIN-1을 기준으로 할 때 JOIN-2와 JOIN-3 이 더 적은 시간을 소요하기 위하여는 r = 50% 이상의 필터율이 필요하다. JOIN-4의 경우 다른 시간요소들을 무시한다면 항상 다른 방법보다 우수하며 약간의 무시된 시간요소들을 고려하더라도 JOIN-1보다 우수하기 위하여 필요한 최소 필터율은 10% 정도에 불과하다. JOIN-5의 경우 JOIN1보다 적은 시간을 소요하기 위한 필터율은 r = 40% 정도 이상이다. 결국 공간조인 수행 시간성능은 주로 RP필터의 필터율에 좌우되며, 종합하여 보면 시간성능 면에서 가장 좋은 방법은 JOIN-4라고 할 수 있다.

필터율에 따른 이들 방법들의 시간성능을 도식적으로

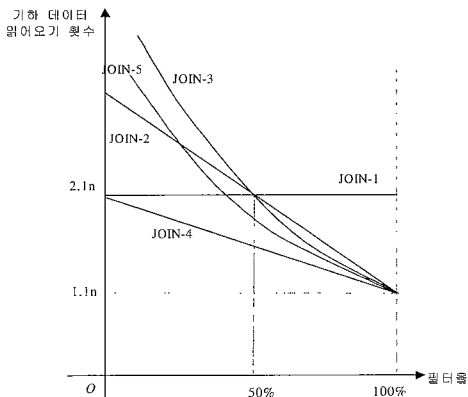


그림 10 필터율에 따른 각 공간조인 방법의 성능

비교해 보면 [그림10]과 같다. 5장에서 실제로 측정된 값과 비교하여 다시 논의될 것이다.

4.8 RP 필터를 조합하여 사용

Approximation들을 조합하여 사용하는 방법은 [2]에서 언급된 바 있다. 이 경우 관리비용이 점점 더 커지기 때문에 문제가 된다고 할 수 있으나 필터율을 높일 수 있다면 충분히 가치가 있다 [2].

제안된 RP 필터를 조합하여 사용하면 저장공간이 더 필요하고 관리비용이 약간 더 커지지만 역시 필터율을 높일 수 있으며 수행시간도 줄일 수 있다. RP 필터를 조합하여 사용하는 공간조인 알고리즘은 JOIN-4의 방법에서 간단히 확장된다. RP필터를 조합하여 사용한 경우의 필터율과 수행시간 성능, 그리고 가능한 조합 중에서 어느것이 상대적으로 우수한지는 5장에서 논의하였다.

```

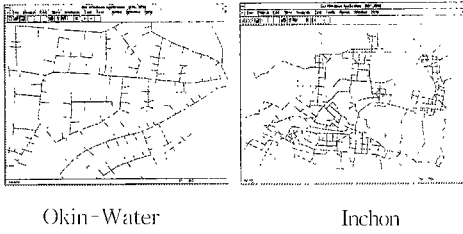
RP 필터를 두 개 조합하여 사용한 방법
MBR-join( );
For all candidate pairs ( a, b )
{
    Get RP(a)
    Get Geometry(b)
    If ( Inside ( b, RP1(a) ) == TRUE )
        Pair ( a, b ) is a hit
    Else if ( Inside( b, RP2(a) ) == TRUE )
        Pair ( a, b ) is a hit
    Else
    {
        Get Geometry(a)
        If ( Overlap ( a, b ) == TRUE )
            Pair ( a, b ) is a hit
        Else
            Pair ( a, b ) is a false hit
    }
}
    
```

5. 실험 및 성능평가

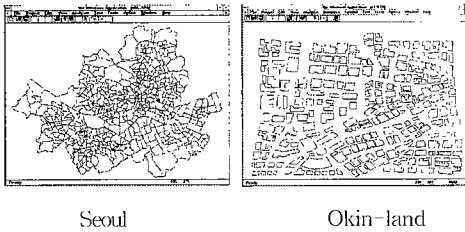
5.1 실험 환경

제안된 RP필터의 성능평가를 위한 실험은 64MB RAM과 166MHz Pentium-II 프로세서를 가지는 PC에서 이루어졌으며 사용된 언어와 개발환경은 Visual C/C++ 6.0 이다. 성능평가를 위한 파라미터는 필터율과 질의처리 수행시간이며 질의처리 시간은 CPU time을 milisecond단위로 측정하였다. 사용된 데이터셋은 [그림 11]의 네 종류이며 그 특성은 [표2]와 같다.

먼 객체 데이터의 경우 원래의 지도를 평행이동하고 회전하여 공간조인에 사용할 두 번째 지도를 얻었으며 평행이동과 회전의 parameter를 다르게 하여 다양한 조



(a) 선 속성 데이터



(b) 면 속성 데이터
그림 11 실험에 사용된 데이터

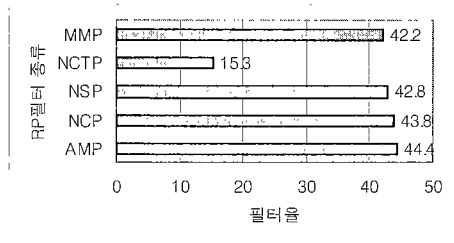
표 2 데이터 특성

데이터	객체의 속성	객체의 수	객체당 정점의 수 (평균)
Okin-water	선	515	36.5
Incheon	선	614	54.3
Seoul	면	1268	135.3
Okin-land	면	557	72.2

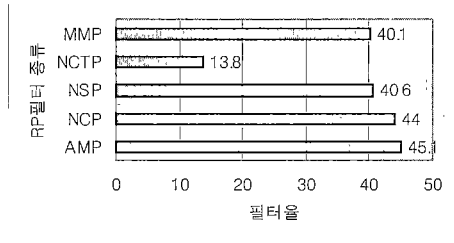
합의 지도에 대하여 실험하였다(예를 들어 Seoul 원본 데이터와 평행이동,회전한 Seoul 데이터의 공간조인을 Seoul * Seoul 로 표시). 선 객체 데이터의 경우 다른 면 객체 데이터와 공간조인을 수행하였다(예를 들어 Incheon 데이터와 Seoul 데이터의 공간조인을 Incheon * Seoul 로 표시). RP필터 종류들과 그 조합들간의 성능을 필터율에 의하여 비교하였으며 JOIN-1, JOIN-2, JOIN-3, JOIN-4, JOIN-5 각 방법에 대하여 수행시간을 측정하여 비교하였다.

5.2 RP 필터 종류들간의 비교

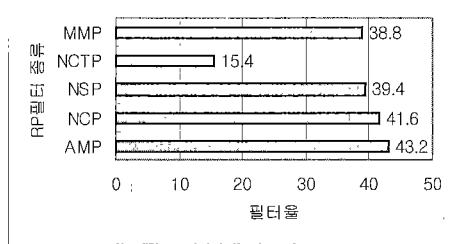
RP 필터 종류들간의 필터율 비교는 [그림12] 와 같다. NCTP를 제외한 RP필터들의 필터율은 43~45% 정도로 기존 progressive approximation의 필터율 31~32% (MEC), 34~36% (MER) 보다 높은 것으로 나타났다.



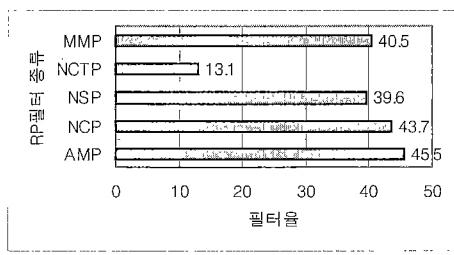
(a) Okin-Water * Okin-land



(b) Incheon * Seoul



(c) Seoul * Seoul



(d) Okin-land * Okin-land

그림 12 RP 필터 종류들간의 필터율 비교

어떤 조인 방법을 사용하던지 다른 과정은 모두 동일하게 수행되므로 저장공간 측면을 생각하지 않는다면 이들간의 우열은 오직 필터율에 의해서만 결정된다. NCTP를 제외한 네 개의 다른 종류들은 커다란 차이를 보이지 않고 거의 같은 필터율을 가진다. NCTP는 parameter수가 적은 장점 대신에 낮은 필터

율을 가지는 것으로 분석되었다. 네 가지 RP필터의 종류는 저장공간도 동일하며 필터율에서 거의 차이를 보이지 않았다.

5.3 RP필터 조합들간의 비교

RP 필터 자체의 필터율을 평균 44%라고 하면 RP 필터의 종류 두 개를 조합하여 사용한 경우 필터율은 RP 필터 하나만을 사용한 경우에 비해 최소 약2%, 최대 약 27% 향상되었다. RP 필터의 종류 두 개의 가능한 조합의 필터율을 [그림13]에서 나타내었다. 필터율을 향상시키기 위한 RP 필터의 최적 조합은 NCP-NSP

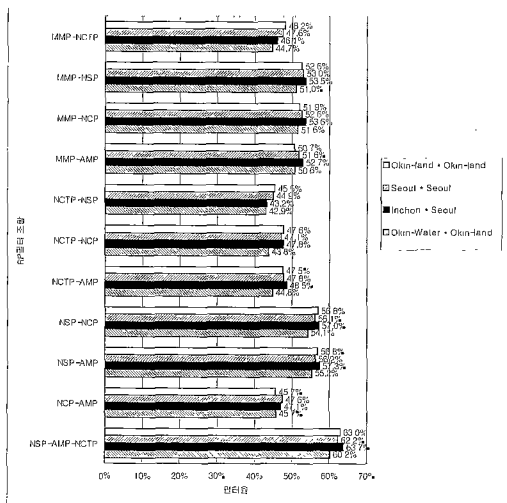


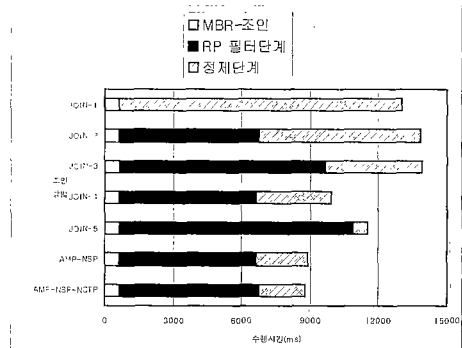
그림 13 RP 필터 종류 두 개 이상을 조합하여 사용한 경우 필터율

또는 AMP-NSP로 나타났다. 이것은 NCP와 AMP가 개념상으로 거의 같은 것이며 NSP와는 반대되는 개념인 것으로부터 충분히 예상할 수 있는 것이다. 즉 서로 겹치는 MBR 쌍 중에서 가장 많이 나타나는 형태가 [그림2(a)] 와 [그림2(b)] 이며 여기에 해당하는 것이 각각 NCP (또는 AMP)와 NSP이기 때문이다. NCTP가 포함된 조합은 NCTP 자체의 낮은 필터율 때문에 여전히 상대적으로 낮은 필터율을 나타낸다. 또한 AMP와 NCP는 개념상 중복되는 것이므로 조합해서 사용할 경우 필터율 향상을 거의 가져오지 못한다. 한편 세 개 이상의 RP필터를 조합해서 사용한 경우 약 50~63% 정도의 필터율을 나타냈으며 가장 높은 조합은 AMP-NSP-NCTP로 나타났다.

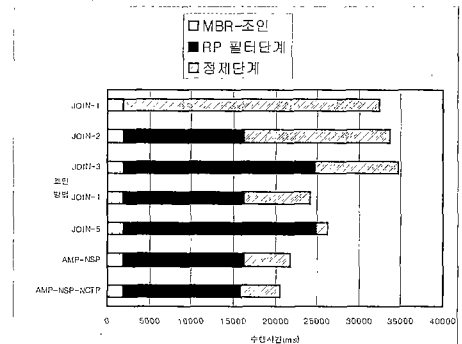
5.4 공간조인 알고리즘의 성능비교

앞장에서 논의한 JOIN-1, JOIN-2, JOIN-3, JOIN-4,

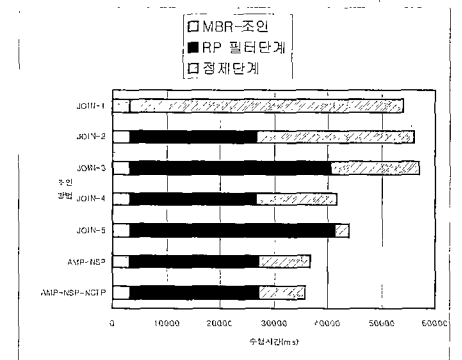
JOIN-5 다섯 가지 공간조인 방법의 시간성능을 [그림 14]에서 비교하였다. MBR-조인에 사용된 SAM으로는 R-tree를 선택하였다. 다섯 가지 방법 모두에서 MBR-조인 과정은 공통으로 적용되었으므로 수행시간도 같으며, RP 필터단계와 정제단계에 소요된 시간만 차이를 보였다. 각 방법들의 상대적 시간성능은 대략적으로 [그



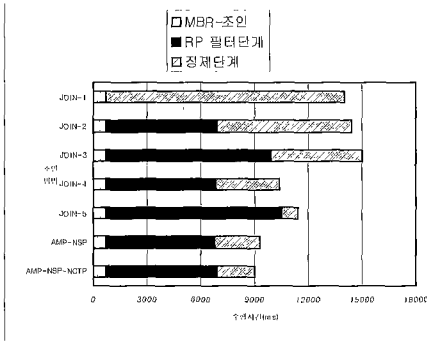
(a) Okin-water * Okin-land



(b) Incheon * Seoul



(c) Seoul * Seoul



(d) Okin-land * Okin-land

그림 14 공간조인 수행시간 비교

림 10]에서 살펴본 바와 일치하였다. RP 필터의 필터율이 50% 이하이기 때문에 전체 수행시간 면에서는 JOIN-2와 JOIN-3은 오히려 JOIN-1(MBR-조인만을 수행)보다 시간이 더 걸리는 것으로 나타났으며 정제단계를 RP 필터단계와 통합하여 객체접근 횟수를 줄인 JOIN-4와 JOIN-5는 나머지 방법보다 상대적으로 적은 수행시간을 나타냈다. 각 방법들의 수행시간 감소 성능은 추가적인 필터단계가 없는 SAM에 의한 MBR-조인만을 수행한 시간을 기준으로 하여 얼마나 수행시간을 감소시킬 수 있는가를 나타낸 것이며 여러 번 반복 측정된 값의 평균을 취하였다. 4.7절에서 논의한 바와 같이 JOIN-4 방법이 가장 우수하며 JOIN-1 방법과 비교하여 약 25% 정도 수행시간을 줄이는 효과를 나타냈다. 두 개의 종류를 조합하여 사용할 경우 가장 성능이 좋은 조합은 AMP-NSP이며 정제단계의 시간을 더 줄임으로써 전체적으로는 약 34% 정도 수행시간을 감소시켰다. 세 개의 종류의 조합 중에서 필터율이 가장 높은 조합인 AMP-NSP-NCTP의 경우 수행시간의 감소는 약 37% 정도로 나타났다.

5.5 기존 방법과의 성능비교

제안한 RP필터 방법의 성능을 5C-MER-MEL을 조합하여 사용하는 방법[2]과 4CRS 방법[15]과 비교하였다. [표3]은 본 연구에서 실험한 이들 방법들의 수행시간 감소 성능과 저장공간에 대한 비교를 보여준다. 본 연구에서 제안한 RP필터의 수행시간 감소 효과는 하나의 종류만을 사용할 경우 약 25%, 두 개 또는 세 개의 종류를 조합하여 사용할 경우 각각 약 34%와 약 37%이다. RP필터는 두 개 이상 조합해서 사용할 경우 5C-MER-MEL을 조합하여 사용하는 방법이나 4CRS에 비하여 수행시간 면에서는 비슷하거나 약간 우수하

며 저장공간 면에서 다른 것들보다 우수하며 생성,수정 연산과 자체 연산이 매우 간단하다는 장점이 있다. 4CRS의 경우 80% 정도의 필터율을 보이지만[7] 필터 단계에서의 연산 시간이 많이 걸리므로, 즉 래스터 셀을 중첩하는 과정에서 좌표 조정 등의 연산이 시간을 많이 필요로 하므로 정제단계의 시간이 적음에도 불구하고 전체 수행시간 면에서는 RP필터를 사용했을 경우와 큰 차이를 보이지 않았다.

표 3 Approximation 방법들의 성능비교

Approximations	저장공간 (bytes)	수행시간 감소
MER-MEL	40	19.3%
5C-MER-MEL	80	32.7%
4CRS	104.5 (42.5 *)	35.4%
AMP	32	25.1%
AMP-NSP	64	34.8%
AMP-NSP-NCTP	72	37.1%

* = 압축된 경우

6. 결론 및 향후연구 방향

본 논문에서는 공간객체에 대한 효율적인 공간조인 처리에 대하여 연구하였다. 공간조인은 지리정보시스템에서 가장 많이 쓰이며 또 중요한 연산 중의 하나이다. 공간조인 처리는 많은 양의 데이터를 다루는 시간이 많이 걸리는 작업이므로, 질의처리에 따른 시간과 비용을 줄이는 것이 가장 중요하다. 공간조인 처리의 성능을 높이기 위하여 객체의 근사치를 사용하여 근사적인 연산을 하는 여과-정제 방법이 널리 사용되고 있다. 이때 많은 시간이 소모되는 정제단계에서의 비용을 줄여 주어야 하므로, 사전에 여과단계에서 후보객체의 수를 최소화시켜줄 필요가 있으며 이 부분에 많은 연구가 진행되어 왔다.

본 논문에서는 점 기반 approximation인 RP필터와, 이를 공간조인에 효과적으로 적용하는 다중필터기법을 제시하여 공간조인의 수행에서 정제단계에서의 후보객체들의 수를 줄였으며 전체적인 수행시간을 25% ~37% 정도 감소시켰다. 또한 RP필터의 종류로서 몇 가지 필터들을 제안하고 사용하였으며 이들을 두 개 이상 조합하여 사용하는 방법에 대하여도 논의하였다. 그리고 RP필터를 공간조인에 적용하는 몇 가지 다른 방법을 제시하고 각 방법에 대하여 성능을 비교분석하였다.

제안한 RP필터는 공간조인에서 hit를 많이 식별함으로써 공간조인의 수행 성능을 높일 수 있는 일종의

progressive approximation이다. RP필터는 기존의 MER이나 MEC 등 면으로 구성된 progressive approximation에 비해서 필터의 생성과 관리에 따른 비용이 훨씬 적으면서 더 높은 필터율을 가진다. 또한 RP 필터는 점으로 구성된 필터이기 때문에 다각형 객체에 대하여만 정의되는 기존의 면이나 선으로 구성된 필터들과 달리, 다각형 객체간의 공간조인 뿐만 아니라 선 객체와 다각형 객체간의 공간조인에도 효과적으로 적용될 수 있다.

향후 연구과제로는 첫째, RP 필터간의 연산만에 의하여 여과단계를 수행할 수 있는 방안이 매우 중요하다고 할 수 있으며 둘째, RP 필터 점들을 추출하는 더 좋은 방법을 고안하여 필터율을 더욱 높이는 것이 요구되며 셋째, 가능한 RP필터의 종류들 중 그리고 종류들의 조합들 중 필터율을 높일 수 있는 최선의 것이 어느 것인지 이론적으로 검증되어야 하며, 마지막으로 선객체끼리의 공간조인에도 적용할 수 있는 방안이 필요하다.

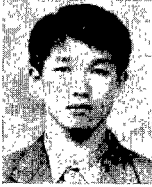
참 고 문 헌

- [1] T. Brinkhoff, H. P. Kriegel, R. Schneider, and B. Seeger, "Multi-Step Processing of Spatial Joins," In *Proceedings of International Conference on Management of Data*, ACM SIGMOD, pp. 197-208, 1994.
- [2] T. Brinkhoff, H. P. Kriegel, "Approximations for a Multi-Step Processing of Spatial Joins," *Proceedings of International Workshop on Advanced Research in Geographic Information Systems (IGIS94)*, Lecture Notes in Computer Science, Springer, 1994.
- [3] T. Brinkhoff, H. P. Kriegel, and B. Seeger, "Efficient Processing of Spatial Joins Using R-trees," In *Proceedings of International Conference on Management of Data*, ACM SIGMOD, pp. 237-24, 1993.
- [4] T. Brinkhoff, H. P. Kriegel and R. Schneider, "Comparison of Complex Objects Used for Approximation-based Query Processing in Spatial Database System," *Proceedings of 9th International Conference on Data Engineering*, Vienna, Austria, pp.40-49, 1993.
- [5] T. Brinkhoff, H. P. Kriegel and R. Schneider, "Efficient Spatial Query Processing in Geographic Database Systems," *IEEE Data Engineering Bulletin*, Vol. 16, No. 3, pp. 10-15, 1993.
- [6] H. P. Kriegel, R. Schneider, and T. Brinkhoff, "Potentials for Improving Query Processing in Spatial Database Systems," *Proceedings of 9th Journals Bases de Donnes Avances*, Toulouse, 1993.
- [7] Geraldo Zimbrão, Jano Moreira de Souza, "A Raster Approximation for the Processing of Spatial Joins," *Proceedings of the 24th VLDB conference*, New York, USA, 1998.
- [8] 이 인석, 강 신봉, 최 윤철, "다중 필터를 이용한 효율적인 공간질의어 처리", 한국 컴퓨터 그래픽스 학회 추계 학술대회, pp.74-78, 1995.
- [9] 강 신봉, 주 인학, 최 윤철, "MMP 다중필터를 이용한 효율적인 공간질의어 처리 방법," 한국정보과학회 논문지, 제24권 제5호, pp.476-486, 1997.
- [10] Joo In-Hak, Choy Yoon-Chul, "A Study on the Efficient Spatial Query Processing Method using MBR operator and NCP filter," *Proceedings of GIS98/RT 98*, pp.241-245, 1998.
- [11] H. Samet, "The Design and Analysis of Spatial Data Structures," Addison-Wesley, MA, 1990.
- [12] Bentley, J. L., "Multidimensional binary search in database applications," *IEEE Transaction on Software engineering*, Vol 4, No 5, pp. 333-340, 1979.
- [13] Nievergelt, J., Hinterberger, H., and Sevcik, K. C., "The grid file: An adaptable, symmetric multikey file structure," *ACM Transactions on Database System*, Vol 9, No 1, pp. 38-71, 1984.
- [14] Volker Gaede and Oliver Geunter, "Multidimensional Access Methods," *ACM Computing Surveys*, Vol 30, No 2, June 1998.
- [15] H. Lu and B. C. Ooi, "Spatial Indexing: Past and Future," *IEEE Data Engineering Bulletin*, Vol 16, No. 3, pp.16-21, Sep 1993.
- [16] A. Guttman, "R-trees: A Dynamic Index Structure for Spatial Searching," *Proceedings of ACM SIGMOD International Conference on Management of Data*, Boston, MA., pp.47-57, 1984.
- [17] Yun-Wu Huang, Ning Jing, Elke A. Rundensteiner, "Spatial Joins Using R-trees: Breadth-First Traversal with Global Optimizations," *Proceedings of the 23rd VLDB conference*, pp 396-405, 1997.
- [18] Karen Daniels, Victor Milenkovic and Dan Roth, "Finding the largest Area Axis-Parallel Rectangle in a Polygon," *Proceedings of the fifth Annual Canadian Conference of Computational Geometry*, 1993.
- [19] D.T. Lee, "Medial Axis Transformation of a Planar Shape", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4, No. 4, 363-369, July 1982.



주 인 학

1992년 연세대학교 전산학과(이학사).
1994년 연세대학교 전산학과(이학석사). 1994년 ~ 현재 연세대학교 컴퓨터학과 박사과정. 관심분야는 지리정보시스템(GIS), 공간데이터베이스, 공간색인 등



고 명 철

1994년 제주대학교 정보공학과(공학사).
1997년 연세대학교 컴퓨터학과(공학석사). 1997년 ~ 현재 연세대학교 컴퓨터학과 박사과정. 관심분야는 컴퓨터그래픽스, 지리정보시스템(GIS), 가상현실 등



최 윤 철

1973년 서울대학교 전자공학과(공학사).
1975년 Univ. of Pittsburgh(공학석사).
1979년 Univ. of California Berkeley Dept. of IE & OR(공학박사). 1979년 ~ 1982년 Lockheed사 및 Rockwell International사 책임연구원. 1982년 ~ 1984년 Univ. of Washington 전산학과 박사과정. 1990년 ~ 1992년 Univ. of Massachusetts 연구교수. 1984년 ~ 현재 연세대학교 컴퓨터학과 교수. 관심분야는 멀티미디어, 컴퓨터그래픽스, 지리정보시스템(GIS), 가상현실 등



고 건

1987년 2월 연세대학교 전산학과 졸업(이학사). 1989년 2월 연세대학교 수학과(전산전공) 졸업(이학석사). 1993년 9월 日本 東京大學校 共學部 졸업(공학박사). 1995년 3월 ~ 현재 청주대학교 컴퓨터정보공학과 교수. 관심분야는 컴퓨터비전, 그래픽스, 가상현실 등.