

# 비트맵 근사 표현을 이용한 효율적인 공간 조인 (The Processing of Spatial Joins using a Bit-map Approximation)

홍 남 희\*      김 희 수\*\*  
(Nam-Hee Hong) (Hee-Soo Kim)

## 요 약

공간 질의 처리에서는 효율적인 처리를 위하여 대상 공간 객체들 중에서 후보객체를 찾아내는 여과단계와 후보객체들 중에서 최종 질의를 만족하는 객체들을 찾아내는 정제단계로 나누어 처리하는 방법이 연구되었다. 이때 정제단계의 비용이 여과단계에서의 비용보다 훨씬 크기 때문에 정제단계의 계산비용을 줄일 수 있는 방법을 찾아내는 것이 중요하다. 본 논문에서는 정제단계에서의 계산비용을 줄이기 위한 방법으로 비트맵 근사 표현 기법을 제안한다. 비트맵 근사 표현은 conservative 표현과 progressive 표현을 모두 적용하여 정확한 공간 객체 교차 알고리즘을 수행하기 전에 false hit과 true hit을 사전에 찾아냄으로써 정제 단계의 계산비용을 줄이는 방법이다. 또한 비트맵 근사 표현을 R-트리에 적용하여 정제단계에서 계산비용을 줄이는 다양한 방법들간의 비교를 통하여 성능을 평가하였다.

## ABSTRACT

This paper studies on the processing of spatial joins. The spatial join operation is divided into filter and refinement steps in general. The processing of spatial joins can be greatly improved by the use of filters that reduce the polygons in order to find the intersecting ones. As a result, three possible sets of answers are identified: the positive one, the negative one and the inconclusive one. To identify all the interesting pairs of polygons with inconclusive answers, it is necessary to have access to the representation of polygons so that an exact geometry test can take place. We introduce a bit-map approximation technique to drastically reduce the computation required by the refinement step during refinement processing. Bit-map representation are used for the description of the internal, the external and the boundary regions of the polygon objects. The proposed scheme increases the chance of trivial acceptance and rejection of data objects, and reduces unnecessary disk accesses in query processing. It has been shown that the reference to the object data file can be cut down by as much as 60%.

## 1. 서론

컴퓨터 처리 능력이 향상됨에 따라 지리정보시스템(geographic information system), 이미지 처리 시스템, CAD/CAM 등과 같은 분야의 시장 수요가 계속 증가하고 있다.

지리정보시스템은 공간상에 존재하는 객체의 위치를 표현하기 위한 2차원 또는 3차원의 수치 데이터(공간 데이터)와 공간객체의 속성을 기술하는 문자 위주의 텍스트 데이터(비공간 데이터)로 구성된 멀티

\* 정희원 : 광주보건대학 전산정보처리과 교수  
\*\* 정희원 : 광주보건대학 전산정보처리과 교수

논문접수 : 2001. 1. 29.  
심사완료 : 2001. 2. 9.

미디어 데이터 중의 하나로 간주할 수 있다. 이러한 공간 데이터와 비공간 데이터를 통합하여 효과적으로 관리할 수 있는 시스템을 공간 데이터 베이스 시스템(spatial database system)이라고 정의할 수 있다.

공간 데이터를 이용한 많은 응용 분야들은 공간 객체들간의 다양한 공간 조인(spatial join) 연산을 포함한 질의어를 가지고 있기 때문에 공간 데이터베이스에서는 특히 효율적인 공간 조인 방법이 가장 중요한 요소 중의 하나라고 생각할 수 있다. 특히 공간 조인은 빈번하면서도 많은 비용을 요구하므로 공간 조인을 효과적으로 처리하는 방법에 관한 많은 연구가 이루어져 왔다. 관계형 데이터베이스 시스템에서의 조인 연산은 다음과 같이 정의한다. 두 개의 릴레이션(relation) R과 S의 열(column)의 i와 j에 대한  $\theta$ -조인이란 카티션 곱(cartesian product)  $R \times S$ 에 있는 튜플(tuple)들 중 R의 i 번째 열과 S의 j 번째 열이  $\theta$  관계에 있는 것들을 나타내는 것을 말한다. 공간 데이터베이스 시스템에 있어 공간 조인은 R의 i 번째 열과 S의 j 번째 열이 공간 데이터 타입이고  $\theta$ 가 공간 연산자(거리, 방향, 위치 교차 등)이면 공간 조인이라 정의할 수 있다. 공간 조인은 크게 거리 연관 조인 (distance associated joins), 방향 연관 조인 (orientation associated joins), 공간 객체 교차 조인(spatial intersection joins)으로 구분할 수 있다.

공간 조인의 효율을 높이기 위하여 일반적으로 공간 조인 처리는 여과(filter)와 정제(refinement)의 2단계로 나누어 처리한다[7].

가. 단계1 : 여과 단계

MBR(Minimum Bounding Rectangle)과 같은 객체의 기하적 근사값(geometric approximation)만을 이용하여 조인을 수행한다. 여과 단계의 결과는 기하적 근사값에 의한 결과이므로 실제 공간 데이터로는 조인이 안될 경우도 있다. 여과 단계를 거쳐서 나온 결과를 후보 세트(candidate set)라고 한다.

나. 단계2 : 정제 단계

정제 단계에서는 여과 단계의 결과로 얻은 후보 세트의 모든 객체에 대해서 객체들의 실제 공간 표현을 이용하여 두 공간 객체들이 실제로 교차하는지를 정확하게 검사(exact match)하여 그 결과를 제시한다. 객체들의 실제 공간

표현을 보조기억장치로부터 읽어들이고 기하 계산 알고리즘을 사용하기 때문에 많은 처리 비용을 요구한다. 두 공간 객체가 실제로 교차하는지에 사용되는 알고리즘으로는 plane sweep 알고리즘이 많이 사용되고 있다. Plane sweep 알고리즘의 계산 시간(time complexity)은  $O(n * \log(n))$  이다. 여기서 n은 공간 객체를 표현하는 포인트 좌표의 개수이다. 또 다른 처리 방법으로는 객체를 분해하여 분해된 구성 요소들에 연산을 수행하는 방법이 있다[2].

실제 GIS 데이터를 이용한 공간 조인의 비용은 여과 단계에서보다는 정제 단계에서의 비용보다 높게 나타남을 최근의 많은 연구 결과로부터 알 수 있다[4]. 즉 공간 데이터 처리에는 I/O 위주의 비용보다는 CPU 위주의 계산 비용이 많이 차지하기 때문에 CPU 위주의 계산 비용을 얼마나 줄이느냐가 공간 데이터를 효율적으로 처리하는 주요 요인이 된다.

본 논문에서는 정제 단계의 계산 비용을 줄이기 위한 방법으로 비트맵 근사 표현 방법(Bit-map approximation)을 제안한다. 비트맵 근사 표현 방법은 다각형 객체의 래스터(raster) 표현을 이용하여 다각형 객체가 서로 교차하는지를 사전에 검사하여 정제 단계에 적용되는 후보 세트를 가능한 많이 줄임으로써 공간 데이터를 효율적으로 처리하는 방법이다.

본 논문의 구성은 다음과 같다. 2장에서는 공간 조인과 관련된 기존의 연구에 대해서 기술하고, 3장에서는 비트맵 근사 표현 방법에 대해서 기술하며, 4장에서는 비트맵 근사 표현 방법을 이용한 질의 처리 과정에 대해서 기술하며, 5장에서는 실험 결과를 보고 마지막으로 6장에서는 결론을 기술하고자 한다.

## 2. 관련 연구

지금까지의 여과 단계 처리 방법은 근사값을 변환하는 방법과 변환하지 않는 방법으로 구분할 수 있으며 또한 공간 색인 구조를 이용하는 방법과 이용하지 않는 방법으로 구분할 수 있다.

공간 객체에 대한 근사값을 변환하여 사용하는 방법으로는 Z-ordering을 이용한 기법이 제안되었다[6]. 여기에서는 Z-ordering 기법을 이용하여 다차원 근사값을 1차원 값으로 변환시켜 이 값을 B+-트리 구조에 저장시킨 후에 B+ 트리 구조상에서 공간 조인을 수행시키는 기법이다.

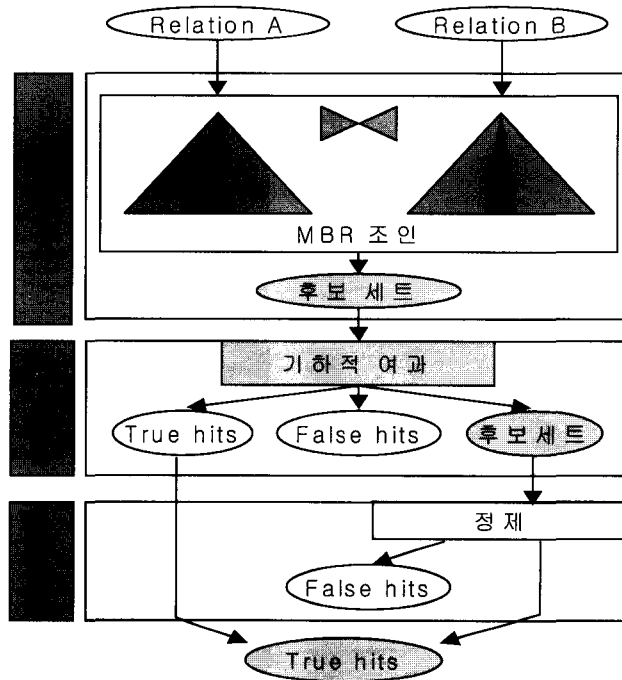
근사값을 변환하지 않으면서 공간 색인을 사용하지 않는 방법으로는 공간 해쉬(hash) 조인이 있다[5]. 관계형 데이터베이스의 해쉬 조인과 유사한 처리 방법이다. 또 다른 방법으로는 분할 기반(partition based) 방법을 들 수 있다[8].

색인을 사용하는 방법으로는 R-트리 색인 구조를 들 수 있다[3]. R-트리는 B+-트리와 같이 높이 균형

트리이며, B+-트리는 일차원적 순서를 사용하는 구조인 반면 R-트리는 다차원 상의 구간들로 객체를 표현하는 방식이며 주기적으로 재구성할 필요가 없는 동적 색인 구조이다.

정제 단계에 대한 연구로는 객체를 분할하는 방법과 분할하지 않는 방법으로 나눌 수 있다. 객체를 분할하지 않는 방법으로는 nested loop 방법과 plane-sweep 방법이 있다. nested loop 방법은 관계형 데이터베이스 시스템에서와 유사한 방법이다. Plane-sweep 방법은 두 다각형 객체의 간선들을 한 좌표축으로 정렬한 후 교차 발생 여부를 가상의 sweep line을 이용하여 검사하는 방법이다.

객체를 분할하는 방법은 복잡한 객체를 보다 간단한 구성 요소로 분할하여 공간 조인 연산을 수행하는 방법이다. 복잡한 객체를 분할하는 방법은 여러 가지가 있을 수 있을 뿐만 아니라 분해된 객체 구성 요소들을 효과적으로 사용할 수 있는 공간 액세스 방법도 임의로 채택할 수 있다. [2]에서는 객체를 사다리꼴(trapezoid)로 분할하여 R\*-트리와 유사한



[그림 1] Multi-step Spatial Query Processor (MSQP)  
 [Fig.1] Multi-step Spatial Query Processor (MSQP)

TR\*-트리라는 공간 액세스 방법을 제안하였다. 객체 분할 방법의 문제점은 분할된 객체의 구성 요소 데이터를 저장할 추가적인 저장 공간이 요구되므로 I/O 비용이 증가하는 점과 객체를 저장하기 전에 사전에 객체를 분할하여야 하는 비용이 발생되며 또한 분할되어 저장된 객체를 백터 형식으로 변환하여 응용 프로그램에 사용되어야 한다는 단점을 가지고 있다.

[2]에서는 Multi-step Spatial Query Processor (MSQP) 구조가 제시되었다. 이 구조에서는 공간 조인 처리 과정을 기능별로 분리된 3단계로 구분하여 처리하고 있다.

가. 단계1 : MBR 조인

MBR(Minimum Bounding rectangle)과 같은 근사값만을 이용하여 공간 조인을 수행한다. MBR 조인을 수행하기 위하여 R\*-트리 구조를 사용한다.

나. 단계2 : Filter Approximation

MBR 조인 결과로부터 조금 더 정제된 결과를 얻기 위하여 MBR 보다 정교한 근사를 이용한다. [BRI94]에서는 MBR 보다 정교한 근사 방법으로 5-corner 근사 표현을 제안하고 있다. 본 논문에서는 이 단계에 적용될 필터 비트맵 근사 표현 구조를 제안한다.

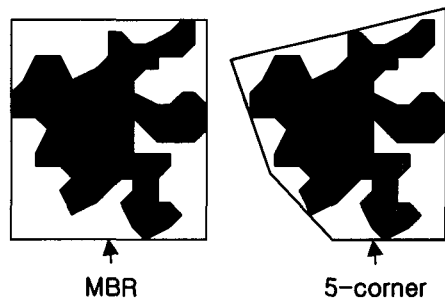
다. 단계3 : Exact Geometry Intersection

단계2로부터 얻은 모든 후보 객체들의 순서쌍 결과에 대해서 정확한 공간 교차가 있는지를 테스트하는 단계로 [2]에서는 plane-sweep 방법과 trapezoid 객체 분할 방법을 제안하고 있으며, [4]에서는 plane-sweep 알고리즘의 수행 시간을 줄이기 위한 방법으로 SID(Symbolic Intersection Detection) 방법을 제시하고 있다.

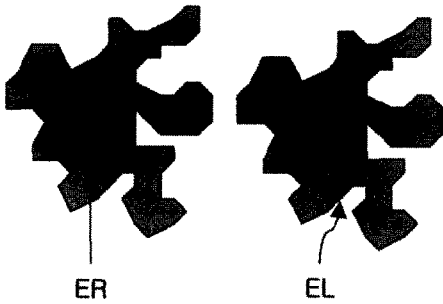
### 3. Bit-map approximation

#### 3.1 공간 객체 근사 표현 방법

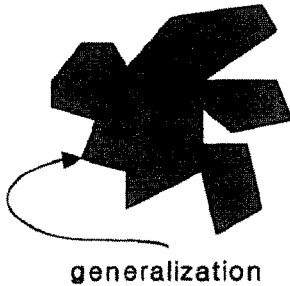
공간 객체의 근사 표현 방법은 conservative, progressive, 그리고 generalizing의 3가지 방법으로 분류될 수 있다[1]. 실제 공간 객체의 경계 부분이 근사 표현에 모두 포함되어 있다면 이러한 방법을 conservative 근사 표현 방법이라 하고[그림2], 근사 표현에 속한 모든 좌표 값들이 실제 객체에 모두 포함된다면 이러한 표현 방법을 progressive 근사 표현 방법이라 한다[그림3]. Generalization 근사 표현 방법은 객체를 표현하는 모든 좌표 값들 중에서 일부분만을 이용하여 객체를 표현하는 방법을 말한다[그림4]. 일반적으로 generalization 근사 표현 방법은 실제 객체와 근사 표현된 객체와의 위상관계(topology)를 유지하지 못한다는 단점을 가지고 있기 때문에 generalization 근사 표현 방법으로는 다각형 공간 객체들간의 교차성 검사에는 사용되지 못한다. Conservative 근사 표현 방법으로는 MBR, n-corner(5-corner) 등이 있으며, conservative 표현 방법에는 Maximum enclosed rectangle(ER), Maximum enclosed rectilinear line segment (EL) 방법 등이 해당된다[1]. Conservative 근사 표현 방법을 이용하면 교차되지 않은 객체들도 결과로 제시될 수 있으며, progressive 근사 표현 방법을 이용하면 교차된 객체들도 결과에서 제외될 수 있다는 특징을 가지고 있다.



[그림 2] Conservative 근사 표현  
[Fig.2] Conservative approximation representation



[그림 3] Progressive 근사 표현  
[Fig.3] Progressive approximation representation



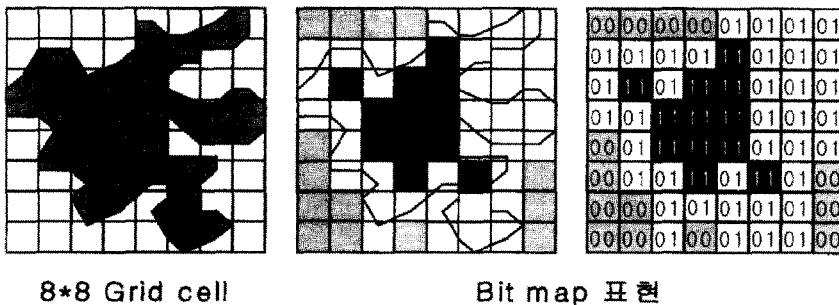
[그림 4] Generalization 근사 표현  
[Fig.4] Generalization approximation representation

### 3.2 Bit-map 근사 표현 방법

본 논문에서는 progressive와 conservative의 두 가지 근사 표현을 하나로 나타낼 수 있는 비트맵 근사 표현 방법을 제안한다. 정제 단계에 사용되는 근사 표현은 공간 색인 구조에 저장되어야 하기 때문에 충분히 작은 양의 정보로 표현되어야 한다. 비트맵 근사 표현 정보는 16바이트를 차지하기 때문에 5-corner 방법에서의 20 바이트보다는 작은 공간을 차지하게 된다. 공간 객체의 비트맵 근사 표현 구조는 다음과 같이 구한다. 먼저 객체의 MBR을 구한다. 각 MBR을 8\*8 그리드 셀(grid cell)로 나누어 각 그리드 셀을 2개의 비트로 셀의 정보를 표현한다.

만약 셀이 실제 객체 안에 완전히 포함된다면 11로 하고, 셀이 실제 객체를 일부분도 포함하지 않는다면 00으로 표현한다. 그리고 셀이 실제 객체를 조금이라도 포함하고 있다면 이러한 셀들은 01로 표현한다(그림 5).

셀 값이 00을 가지면 그 셀은 conservative 표현에 해당되고, 11 값을 가지면 progressive 표현에 해당된다. 01 값을 가진 셀들은 객체의 외형을 나타내는 셀들이 된다. 이러한 비트맵 근사 표현 방법을 이용하여 다각형 공간 객체의 교차를 사전에 검사함으로써(false hit과 true hit) 실제 공간 객체 교차 테스트의 후보를 줄여 효율적인 공간 조인을 수행할 수 있다<표 1>.

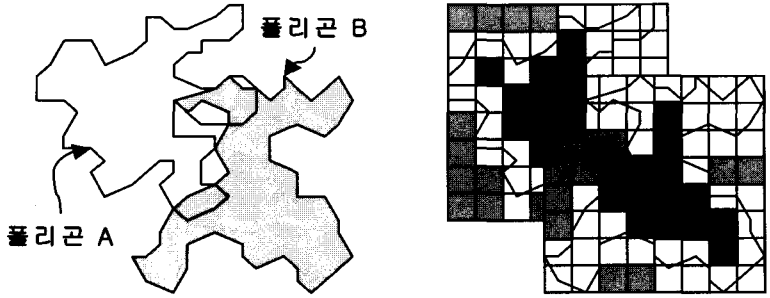


8\*8 Grid cell

Bit map 표현

[그림 5] 비트맵 근사 표현

[Fig.5] Bit-map approximation representation



[그림 6] 비트맵을 이용한 교차 테스트  
[Fig.6] Intersection test using a bit-map

<표 1> 셀 매칭 테이블  
<Table 1> Cell matching table

Grid cell 값	00	01	11
00	교차하지 않음	교차하지 않음	교차하지 않음
01	교차하지 않음	후보	교차
11	교차하지 않음	후보	교차

#### 4. 질의 처리

##### 4.1 포인트 질의

임의의 포인트를 포함한 공간 객체들을 검색하는 포인트 질의인 경우 객체의 MBR이 주어진 포인트를 포함하고 있는 객체를 R-트리를 이용하여 검색하고 R-트리의 leaf 노드에 저장된 16 바이트의 비트맵 근사 표현 정보를 스캔한 후 그 포인트가 속한 그리드 셀을 찾아서 그리드 셀 값이 00이면 false hit으로 바로 판단되어 버려지고, 11이면 true hit이 되어 결과 값으로 제시된다. 만약 그리드 셀 값이 01이면 근사 값만으로는 판단되어질 수 없기 때문에 실제 공간 좌표 값들을 읽어서 plane-sweep 알고리즘을 이용하여 정밀한 교차 검사를 수행한다.

##### 4.2 윈도우 질의

윈도우 질의에는 여러 가지 종류가 있으나 본 논문에서는 단순히 교차 질의(intersection query)만을 논한다. 포인트 질의와 마찬가지로 윈도우 영역과 객

체의 MBR이 교차된 객체들을 R-트리를 이용하여 검색한다. 검색된 객체의 leaf 노드에 저장된 비트맵 근사 표현 정보를 읽어서 윈도우 영역에만 포함된 그리드 셀에 대해서 다음과 같은 조건에 따라 처리한다.

- 가. 윈도우 영역에 포함된 그리드 셀들 중에서 어느 하나라도 값이 11인 셀이 있다면 이 객체는 true hit이 되어서 더 이상의 교차 검사 과정을 거칠 필요가 없다.
- 나. 윈도우 영역에 포함된 모든 그리드 셀들의 값이 00이라면 이 객체는 false hit이 되어 더 이상의 처리를 필요로 하지 않고 버려진다.
- 다. 윈도우 영역에 포함된 그리드 셀들의 값이 00과 01로만 이루어져 있다면 이 객체는 비트맵 근사 표현 정보만으로는 판단할 수 없게 되어 정밀한 교차 검사 처리를 요하게 된다.

#### 5. 성능 평가

비트맵 근사 표현 방법의 성능을 평가하기 위하여 비트맵 근사 표현의 근사율과 비트맵 R-트리의 효율성 및 디스크 접근 회수를 측정하였다.

실험에 사용한 데이터는 미국의 인구 통계국(US Bureau of the Census)에서 제작한 캘리포니아주의 산타바바라 지역의 수로에 대한 데이터를 사용하였다. 이 데이터는 35,428개의 다각형 객체로 구성되어 있다.

### 5.1 비트맵 근사 표현의 근사율

비트맵 근사 표현 방법은 두 개의 근사 표현 방법 (progressive 근사 표현 방법과 conservative 근사 표현 방법)을 하나의 표현으로 나타낸 것이다. Progressive 근사 표현을 이용하면 true hit을 찾을 수 있고, 반면에 conservative 근사 표현 방법을 이용하면 false hit을 찾을 수 있다. 비트맵 근사 표현에서 progressive도 아니고 conservative도 아닌 부분은 근사 표현으로는 두 공간 객체의 교차성 판단이 되지 않는 것으로 정밀한 교차성 검사 과정을 필요로 한다. 이와 같이 progressive 영역도 아니고 conservative 영역도 아닌 영역을 미결정 영역(indecision area)이라고 정의하며 이 영역의 비율이 근사율과 밀접한 관계를 가진다고 할 수 있다(그림 7).

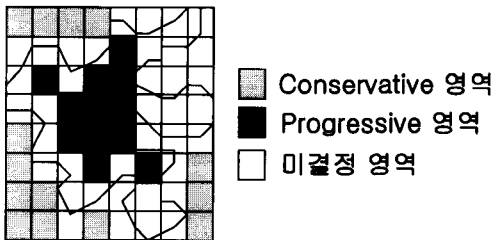


그림 7 근사 표현 영역  
[Fig 7] Approximation representation area

비트맵 근사 표현의 근사율은 다음과 같이 정의할 수 있다.

$$\text{근사율} = \frac{MBR - (\text{progressive} + \text{conservative})}{\text{폴리곤객체영역}}$$

비트맵 근사 표현에서의 근사 표현 영역은 progressive 영역과 conservative 영역을 합한 것을 MBR 영역에서 뺀 값이다. 위에서 정의한 근사율을 가지고 MBR 근사 표현과 5-corner 근사 표현 방법과의 비교를 통하여 비트맵 근사 표현의 근사율을 평가하였다<표 2>.

<표 2> 근사 표현 방법들간의 근사율 비교  
<Table 2> Comparison of approximation between approximation representations

근사 표현 방법	근사율
Bit-map	122.3%
5-Corner	133%
MBR	193%

### 5.2 효율성 및 디스크 접근 회수

비트맵 근사 표현을 R-트리로 구현했을 때 R-트리 leaf 노드에 근사 표현 정보를 저장하기 때문에 leaf 노드의 사이즈가 커지게 된다. R-트리 경우에는 leaf 노드의 크기는 20 바이트를 차지하며 비트맵 R-트리 경우에는 36바이트를 차지한다. 페이지의 크기가 1K인 경우 R-트리 구조는 50개의 leaf 노드를 가질 수 있으며 비트맵 R-트리 구조는 28개의 leaf 노드를 갖는다. 트리의 중간 노드는 모두 50개의 노드를 갖는다.

교차 질의를 위하여 100개의 윈도우 영역을 임의로 생성하여 측정하였다. 효율성은  $t/c$  ( $t$ : 후보 객체 수,  $c$ : 비트맵 근사 표현을 이용한 테스트 후의 후보 객체 수)로 정의하여 측정하였다. 디스크 접근 회수는 R-트리 디스크 접근회수와 비트맵 R-트리 디스크 접근 회수를 비교하여 측정하였다<표 3>.

<표 3> 비트맵 R-트리의 성능  
<Table 3> Performance of a bit-map R-tree

효율성	디스크 접근
64.7%	12.5%

## 6. 결론

본 논문은 간단하면서도 효과적으로 정제 단계의 비용을 줄일 수 있는 비트맵 근사 표현을 이용한 공간 조인 방법에 관하여 기술한 것이다. 이 방법은 기존의 R-트리 구조에 conservative 근사 정보와 progressive 근사 정보를 모두 포함한 비트맵 표현을 사용함으로써 일반적인 R-트리 구조를 사용할 때보다 효율성 면에서는 64.5%의 증가를 가져왔으며 디스크 접근 회수는 12.5% 향상됨을 볼 수 있었다.

비트맵 R-트리는 R-트리에 추가적인 근사 정보를 넣게 되어 트리의 깊이가 증가함에 따라 추가적인 디스크 접근 요인이 있으나 실제 정밀한 교차성 검사를 하게 될 후보 객체 수를 줄임으로서 전체 디스크 접근 회수가 감소됨을 보이고 있다.

### ※ 참고문헌

- [1] T. Brinkhoff, H. Kriegel, B. Seeger, "Efficient Processing of Spatial Joins using R-trees", Proc. ACM SIGMO Int. Conf. on Management of Data, pp. 237 ~ 246, 1993.
- [2] T. Brinkhoff, H. Kriegel, R. Schneider, B. Seeger, "Multi-step Processing of Spatial Joins", Proc. ACM SIGMOD Int. Conf. on Management of Data, pp. 197 ~ 206, 1996.
- [3] A. Guttman, "R-tree: a dynamic index structure for spatial searching", Proc. ACM SIGMOD Int. Conf. on Management of Data, pp. 45 ~ 57, 1984.
- [4] Y. W. Huang, M. Jones, E. A. Rundensteine, "Improving Spatial Interesting Joins Using Symbolic Intersect Detection", 5th International Symposium, SSD 97, pp. 165 ~ 177, 1997.
- [5] M. L. Lo, C. V. Ravishanker, "Spatial Hash-Joins", Proc. ACM SIGMOD Int. Conf. on Management of Data, 1986.
- [6] J. A. Orenstein, F. A. Manola, "PROBE : Spatial Data Mining and Query Processing in an Image Database Application", IEEE Transactions on the Software Engineering, vol.14(5), 1988.

- [7] J. A. Orenstein, "Spatial Query Processing in an Object-Oriented Database System" Proc. ACM SIGMOD Int. Conf. on Management of Data, 1986.
- [8] J. M. Patel, et al, "Partition Based Spatial-Merge Join", Proc. ACM SIGMOD Int. Conf. on Management of Data, 1996.

### 홍 남 희



1984년 전남대학교 계산통계학과  
이학사  
1986년 한국과학기술원 전산학과  
공학석사  
1986년 ~ 1992년  
국방과학연구소 선임연구원  
1992년~ 1995년 포항공과대학교  
정보통신연구소 선임연구원  
1996년~ 1998년  
(주)엔피아시스템즈 부장  
1999년 ~ 현재 광주보건대학  
전산정보처리과 교수

### 김 희 수



1985년 전남대학교 계산통계학과  
이학사  
1989년 조선대학교대학원  
산업공학과 공학석사  
2001년 조선대학교대학원  
전산통계학과 박사과정  
졸업예정  
1985년 ~ 1992년 호남대학교  
전산실장  
1992년 ~ 현재 광주보건대학  
전산정보처리과 교수