

객체의 순환 속성을 지원하는 CR*-트리

선 휘 준[†] · 김 홍 기^{††}

요 약

공간 데이터베이스 시스템에서 검색의 성능을 높이기 위해서는 공간국부성을 고려한 공간색인 방법이 요구되며, 공간국부성은 객체들의 위치 속성과 관계가 있다. 기존의 공간색인 방법들에서는 객체가 가질 수 있는 순환적인 위치 속성이 고려되지 않았다. 본 논문에서는 순환 및 선형 도메인들로 구성된 검색공간에서 객체의 순환적인 위치 속성을 고려한 공간색인 구조인 CR*-트리를 제안하고 그 성능을 평가하였다. 실험결과에 의하면 CR*-트리는 객체의 분포형태와 버킷 용량에 관계없이 낮은 디스크 접근 횟수와 높은 버킷 이용률을 보였다.

The CR*-Tree Supporting a Circular Property of Objects

Hwi-jun Seon[†] · Hong Ki Kim^{††}

ABSTRACT

To increase the retrieval performance in spatial database systems, it is required to develop spatial indexing methods considered the spatial locality. The spatial locality is related to the location property of objects. The previous spatial indexing methods are not considered the circular location property that objects will be taken. In this paper, we propose a dynamic spatial index structure called CR*-tree, and evaluate the performance of the proposed index structure. This is a new spatial index structure considered the circular location property of objects in which a search space is constructed with the circular and linear domains. By the simulation results, the CR*-tree shows that the number of disk access is low and the bucket utilization is high regardless of object distribution and bucket capacity.

키워드 : 순환속성(Circular Property), 순환도메인(Circular Domain), 선형 도메인(Linear Domain), Spatial Index Structure, CR*-트리 (CR*-tree)

1. 서 론

공간 데이터베이스 시스템에서 공간객체(이하 객체라 함)를 관리하기 위해서는 많은 저장 공간과 처리시간이 요구되며, 적절한 색인구조와 질의처리 기법을 적용함으로써 공간 질의를 효율적으로 처리할 수 있다[1, 2, 5-8, 12].

기존의 공간색인 방법들이 가정하는 검색공간에서 도메인은 객체의 속성이 가질 수 있는 최소와 최대 값 사이의 선형적 순서 범위로 구성된다[3, 4]. 그러나 컴퓨터지원설계 및 지리정보 시스템과 같은 최근의 응용에서 취급되는 객체 또는 시간의 개념이 포함되어 있는 객체 등은 순환적인 위치 속성을 가질 수 있다. 이러한 객체의 순환적 위치 속성을 효율적으로 반영하기 위해서는 순환적인 성질을 갖는 도메인으로 구성된 검색공간이 요구된다. 그리고 순환적인 성질이 포함된 질의범위를 갖는 영역질이나 최대근접질의 처리시 검색의 성능을 높이기 위해서는 공간적으로 인접한 객체

체들을 동일한 버킷에 저장시킬 수 있는 공간 색인방법이 필요하다.

다음은 도메인이 0°에서 360°인 경우 순환적인 성질이 포함된 질의범위와 순환적이지 않는 질의범위로 이루어진 영역질의 예이다

“X가 355° 부터 17° 사이이고 Y가 27° 부터 42° 사이에 있는 객체들을 검색하라”

순환적인 환경이 주어졌을 때 기존의 공간색인 방법들에서는 인접한 객체들이 서로 다른 버킷에 분산되어 저장됨으로 예에서와 같은 질의 형태에 대하여 효율적인 검색이 이루어지지 못한다.

본 논문에서는 순환적인 성질을 갖는 도메인이 포함된 검색공간에서 객체들의 위치 속성을 고려하여 객체를 결집시키는 새로운 공간색인 구조인 CR*-트리(Circular R*-tree)를 제안한다. CR*-트리는 R*-트리[2]에 순환적인 특성을 확장시킨 색인구조이다. 실험에서는 CR*-트리의 성능을 디스크 접근횟수, 저장공간 사용량, 그리고 버킷 이용률 관점에

[†] 정 회 원 : 서남대학교 컴퓨터정보통신학과 교수

^{††} 정 회 원 : 동신대학교 컴퓨터학과 교수

논문접수 : 2003년 3월 18일, 심사완료 : 2003년 9월 1일

서 R*-트리와 비교 평가한다.

2. 순환 도메인과 순환 영역질의

공간색인 방법에서 순환적인 위치 속성을 갖는 객체를 효율적으로 취급하기 위해서는 순환적 순서범위를 갖는 도메인으로 구성된 검색공간이 요구되며, N차원 검색공간에서 영역질의 구성하는 질의범위는 순환적인 범위를 가질 수 있다

기존의 공간색인 방법에서 N차원 검색공간을 구성하는 임의의 도메인 $D_i (i = 0, 1, \dots, N-1)$ 은 해당 검색공간에 존재하는 객체들의 i번째 속성이 가질 수 있는 최소와 최대 값 사이의 서로 다른 m 개의 값들을 갖는 선형적 순서 범위로 이루어져있다. 즉, $D_i = (d_0, d_1, \dots, d_{m-1})$ 이고, 여기에서 $d_i < d_{i+1}, i = 0, 1, \dots, m-2$ 이다.

객체의 위치 속성값을 매개 변수로 하여 임의의 위치 속성값의 다음 또는 이전 위치 속성값을 반환하는 함수를 각각 next, prev라 할 때, [13]에서는 다음과 같은 정의에 의해 N차원 검색공간을 구성하는 도메인을 순환도메인(circular domain)과 선형도메인(linear domain)으로 구분하였다.

【정의 1】 도메인 D_i 가 다음과 같은 성질을 만족할 때, D_i 를 순환 도메인이라 한다.

- i) $next(d_k) = d_{k+1} (0 \leq k < m-1)$ 그리고 $next(d_{m-1}) = d_0$
- ii) $prev(d_k) = d_{k-1} (0 < k \leq m-1)$ 그리고 $prev(d_0) = d_{m-1}$

【정의 2】 도메인 D_i 가 다음과 같은 성질을 만족하고 $next(d_{m-1}), prev(d_0)$ 에 의해 반환되는 위치 속성값이 존재하지 않는다면, D_i 를 선형 도메인이라 한다.

- i) $next(d_k) = d_{k+1} (0 \leq k < m-1)$
- ii) $prev(d_k) = d_{k-1} (0 < k \leq m-1)$

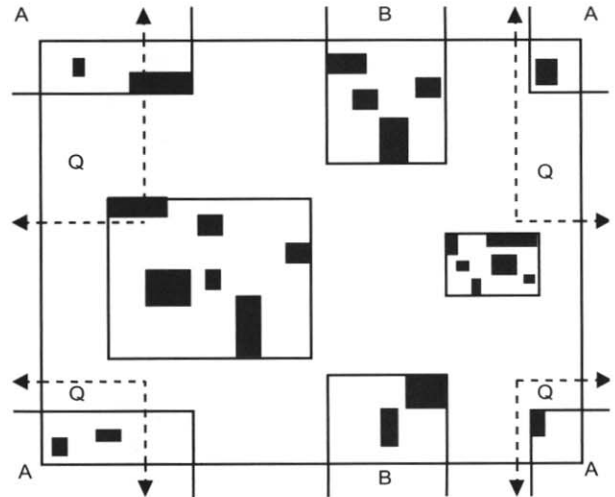
N차원 검색공간에서 영역질의 Q는 질의범위들의 카테시안 곱(cartesian product)으로 표현된다. 즉, 도메인 D_i 에 발생하는 질의범위를 $I_i = [s_i, e_i] (i = 0, 1, \dots, N-1, s_i, e_i : \text{질의범위의 시작과 끝})$ 라 할 때, $Q = I_0 \times I_1 \times \dots \times I_{N-1}$ 로 나타낼 수 있다.

[13]에서는 다음과 같은 정의에 의해 영역질을 순환 영역질의(circular region query)와 선형 영역질의(linear region query)로 구분하였다.

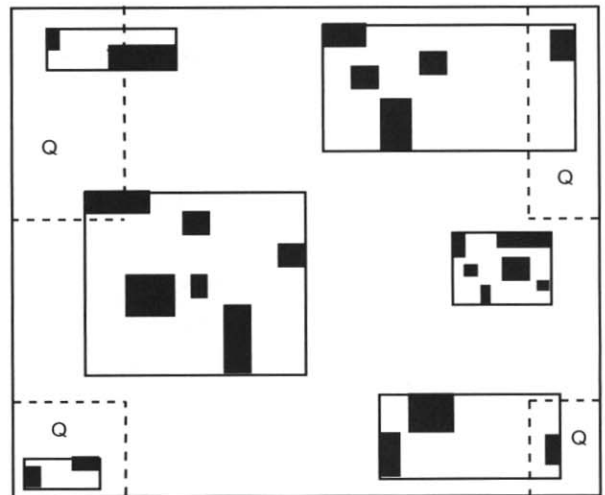
【정의 3】 영역질의 Q를 구성하는 하나 이상의 질의범위 I_i 가 순환 도메인 D_i 상에 발생하여 $d_0, d_{m-1} \in I_i$ 이고 $s_i > e_i$ 인 성질을 가지면, Q를 순환 영역질의라 한다.

【정의 4】 영역질의 Q를 구성하는 모든 질의범위 I_i 가 순환 또는 선형 도메인 D_i 상에 발생하고 $s_i < e_i$ 인 성질을 가지면, Q를 선형 영역질의라 한다.

순환 도메인이 포함된 다차원 검색공간이 존재한다면, 순환적인 환경을 전제로 객체를 결집시킴으로써 버킷 이용률 및 공간국부성을 높일 수 있으며, 영역질의 처리시 접근되는 버킷의 수를 줄일 수 있다.



(a) 순환 도메인으로 구성된 검색공간



(b) 선형 도메인으로 구성된 검색공간

(그림 1) 객체의 결집과 순환 영역질의

(그림 1)(a), (그림 1)(b)는 각각 순환도메인과 선형도메인으로 구성된 2차원 검색공간에서 위치 속성을 고려하여 객체들을 결집시키는 예이다. (그림 1)(a)에서 A, B는 순환적으로 객체를 결집시키는 공간색인 방법을 적용함으로써 생성된 버킷의 부검색공간이다. 기존의 R-트리 유형의 색인

방법을 적용한 (그림 1)(b)에서는 객체를 걸집시킬 때 좋은 공간극부성을 유지하기 위해 (그림 1)(a)의 경우보다 더 많은 버킷이 요구됨을 알 수 있다. 즉, (그림 1)(a)에서는 4개의 버킷이 필요하며, (그림 1)(b)에서는 6개의 버킷이 필요하게 된다.

또한 순환 영역질의 Q가 발생했을 때, (그림 1)(a)에서는 2개의 버킷을 접근함으로써 질의를 처리할 수 있지만 (그림 1)(b)에서는 순환 영역질의가 4개의 선형 영역질의로 분할되어 취급되어야 하며, 5번의 버킷접근이 요구된다.

3. CR*-트리

이 장에서는 순환 및 선형 도메인들로 구성된 검색공간에서 객체들의 순환적인 위치 속성을 지원하기 위한 공간색인 구조인 CR*-트리의 구조 및 알고리즘을 기술한다. CR*-트리의 구조 및 알고리즘은 R*-트리에 순환적인 속성을 포함한 것이다.

3.1 CR*-트리의 구조

CR*-트리는 R*-트리와 유사한 색인구조를 가지며, 객체들이 저장되는 버킷과 버킷을 접근하기 위해 경로 정보를 제공하는 노드로 구성된 높이 균형 트리이다.

CR*-트리에서 N 차원 검색공간은 S 개의 순환 도메인과 $N-S$ 개의 선형 도메인으로 구성된다(단, $0 \leq S \leq N$). 만약 S 가 0이면, 기존의 공간색인 방법들이 전제로한 선형 도메인으로만 구성된 검색공간이다.

CR*-트리의 버킷에는 헤더와 하나 이상의 객체가 저장되며, 검색을 위해 각 객체에는 영역값 R 이 할당된다. O_{id} 를 객체 식별자라고 할 때 하나의 객체는 다음과 같이 표현된다.

$$(R, O_{id})$$

R 은 해당 객체가 검색공간에 차지하는 영역을 N 차원 최소 경계 사각형으로 근사시킴으로써 구해진다.

$$R = (R_0, R_1, \dots, R_{N-1})$$

여기에서 $R_i (i = 0, 1, \dots, N-1)$ 는 객체가 차원 i 에서 차지하고 있는 범위를 나타내는 폐쇄 경계 구간(closed bounded interval)이다.

CR*-트리의 노드는 헤더와 다음과 같은 엔트리들로 구성된 일차원 배열 구조를 갖는다.

$$(R, flag, P_{child})$$

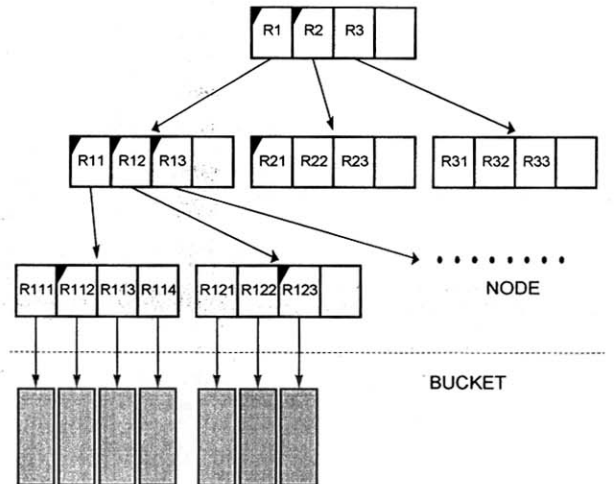
위에서 기술된 바와 같이 R 은 각 차원의 폐쇄 경계 구간들로 이루어진 N 차원 부검색공간이다. $flag$ 는 R 이 순환적인

성격을 갖는 부검색공간인가 아닌가를 나타낸다. 그리고 P_{child} 는 하위 레벨의 노드 또는 버킷의 주소지시자이다.

엔트리가 트리의 리프 노드에 존재한다면, R 은 하나의 버킷이 차지하는 부검색공간을 나타내며 P_{child} 는 해당 버킷의 주소지시자이다. 엔트리가 리프 노드에 존재하지 않는다면, R 은 자식 노드에 속한 모든 객체를 포함하는 부검색공간이며 P_{child} 는 자식 노드의 주소지시자이다.



(그림 2) CR*-트리의 검색공간



(그림 3) CR*-트리의 색인구조

(그림 2)와 (그림 3)은 두 개의 순환도메인으로 구성된 검색공간에서 노드의 최대 엔트리 수가 4일 때, CR*-트리에서 버킷 및 부검색공간들의 포함 관계와 그에 따른 색인구조를 나타낸 것이다. (그림 2)에서 R1과 R2는 순환적 성질을 갖는 부검색공간이다.

3.2 CR*-트리의 알고리즘

CR*-트리의 모든 알고리즘에서 순환 도메인에 놓이는 부검색공간은 논리적인 부검색공간으로 변환되어 취급된다. RI_s 를 폐쇄 경계 구간의 시작값, RI_e 를 구간의 끝값이라고 했

을 때, 부검색공간을 구성하는 임의의 폐쇄 경계 구간이 $RI_S > RI_E$ 인 관계를 갖는다면, 해당 부검색공간은 RI_E 에 폐쇄 경계 구간이 속한 도메인의 최대 값을 더하여 논리적인 부검색공간으로 변환된다.

CR*-트리의 알고리즘은 부검색공간의 취급방법을 제외하고는 R*-트리와 동일하기 때문에 본 절에서는 CR*-트리의 검색 및 삽입 알고리즘을 중심으로 기술한다.

CR*-트리에서 영역질의에 만족하는 버킷을 찾기 위한 검색 연산은 루트 노드부터 시작해서 하위 레벨의 노드까지 재귀적인 방법으로 이루어 진다. 검색 알고리즘에서 순환 영역질의는 순환 성질을 갖는 질의 범위의 끝값에 해당 도메인의 최대값을 더함으로써 논리적인 영역질의로 처리된다.

```

Algorithm SEARCH(node N, Region Query Q)
/* Perform Region Query Q(I0, I1, ..., IN-1), Ii denotes a
   range [si, ei] of Di, Ni.child is a child node of node Ni */
{
  S1. [Convert a circular region query to a logical region query]
  For all I in the region query Q
    If si is more than ei, then
      ei plus the maximum value of domain Di equals ei.
  S2. [Search nonleaf nodes]
  For every entry Ni in N
    Invoke SEARCH(Ni.child, Q)
    for every entry whose MBR intersects the region query Q.
  S3. [Search objects in bucket]
  Output all objects that intersects the region query Q.
}
    
```

객체의 삽입에 따른 전략은 공간국부성 측면에서 매우 중요하다. CR*-트리의 삽입 알고리즘은 R*-트리의 삽입 전략에 객체의 순환적인 결집을 지원함으로써 버킷 및 부검색공간에 포함되는 객체들 간의 높은 공간국부성을 보장한다.

```

Algorithm INSERT(node N, object O)
/* Insert object O into a tree rooted at node N */
{
  I1. [Find the bucket for new object O]
  Use ChooseBucket(O, ptr) to choose a bucket B in which
  to place O.
  I2. [Add object O to bucket B]
  Insert O into B.
  If B is first overflow, then
    invoke ReInsert(O)
  else invoke SplitBucket(B, O), which will return new bucket
  if split was inevitable.
  I3. [Propagate changes upward]
  Form a set S that contain B, its adjacent siblings,
  and the new bucket (if any).
  Invoke AdjustTree(S).
  I4. [Grow tree taller]
  If node split propagation caused the root to split, then
  create a new root whose children are two resulting nodes.
}
    
```

객체의 삽입이 요구되면 CR*-트리의 삽입 알고리즘에서는 ChooseBucket을 사용하여 객체가 저장될 버킷을 두 단계로 찾는다. 첫 번째 단계는 루트 및 중간단계에서 객체를 포함시켰을 때 순환적인 위치 속성을 고려한 부검색공간의 면적 증분이 최소가 되는 노드를 경유하여 리프노드를 찾는다. 두 번째 단계는 해당 리프노드에서 객체를 포함시켰을 때 순환적인 위치속성을 고려하여 접치는 부 검색공간의 영역이 최소가 되는 버킷에 객체를 저장한다.

만약 임의의 버킷 및 노드에서 두 번 이상의 범람으로 부검색공간의 분할이 요구된다면, 해당 부검색공간은 Split Bucket에 의해 분할 후 생성될 두 개의 부검색공간에 따른 면적, 둘레(margin), 겹침의 정도가 최소가 되도록 분할된다.

다음은 CR*-트리에서 객체를 저장하기 위한 버킷의 주소 지시자를 찾는 알고리즘이다.

```

Algorithm ChooseBucket(object O, pointer p)
/* Returns the bucket in which to place a new object O */
{
  C1. [Initialize]
  Set N to be the root node.
  C2. [Leaf check]
  If N is a leaf, return N.
  C3. [Choose subtree]
  If childpointers in N point to leaves, then
  considering the location attribute value of all entries in N
  circularly and non-circularly, choose the entry (O, p) whose
  rectangle needs least overlap enlargement to include the
  new object O; on tie, choose the entry with the least
  area enlargement.
  If childpointers in N do not point to leaves, then
  considering the location attribute value of all entries in N
  circularly and non-circularly, choose the entry (O, p) with
  the least area enlargement to include the new object O;
  on tie, choose the entry with the smallest area.
  C4. [Descend until a leaf is reached]
  Set N to the node pointed by p and repeat from C2.
}
    
```

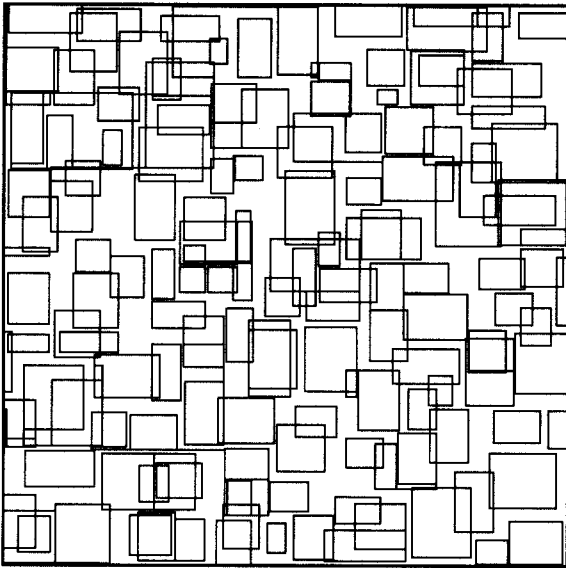
4. CR*-트리의 성능평가

본 장에서는 순환 속성을 지원하는 CR*-트리의 성능을 디스크 접근 횟수, 저장공간 사용량 그리고 버킷 이용률(bucket utilization)에 따라 R*-트리와 비교 평가한다.

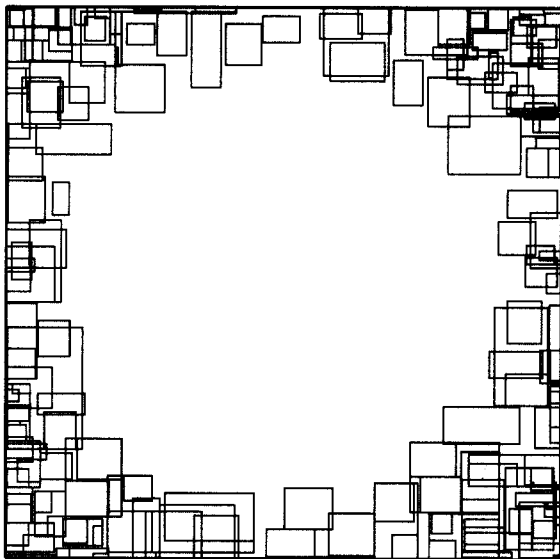
실험에서는 두 개의 순환 도메인으로 구성된 이차원 검색공간을 가정하였으며, 80바이트의 크기를 갖는 중복되지 않는 20,000개의 사각형 객체를 사용하였다. 또한 객체들의 면적은 전체 검색공간의 0.1%인 경우를 고려하였다. 이는 색인에 삽입되는 대부분의 객체가 어느 정도 겹치는 경우에 대해서 CR*-트리와 R*-트리의 성능을 규명하기 위해서이다.

하나의 도메인의 범위가 [0, 1]이라할 때, 실험에서 사용된 객체의 분포는 다음과 같다(그림 4) 참조).

- 균일 분포(uniform distribution): 객체들의 중심점이 중복되지 않고 균일하게 분포
- 모서리 분포(corner distribution): 평균 0.5인 이차함수 분포



(a) 균일 분포



(b) 모서리 분포
(그림 4) 객체 분포

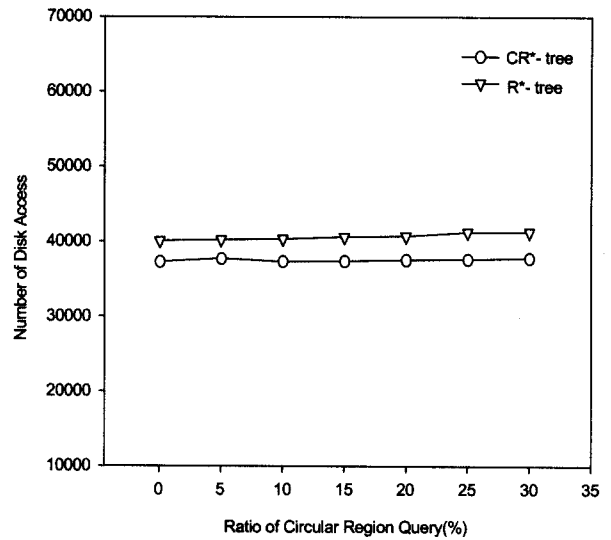
균일 분포는 객체의 분포가 이상적인 상태에서 CR*-트리와 R*-트리의 성능을 비교하기 위한 것이다. 그리고 모서리 분포는 객체의 분포가 밀집되어 있고 순환적인 성질이 많이 반영된 상태에서 CR*-트리와 R*-트리의 성능을 규명하

기 위한 것이다.

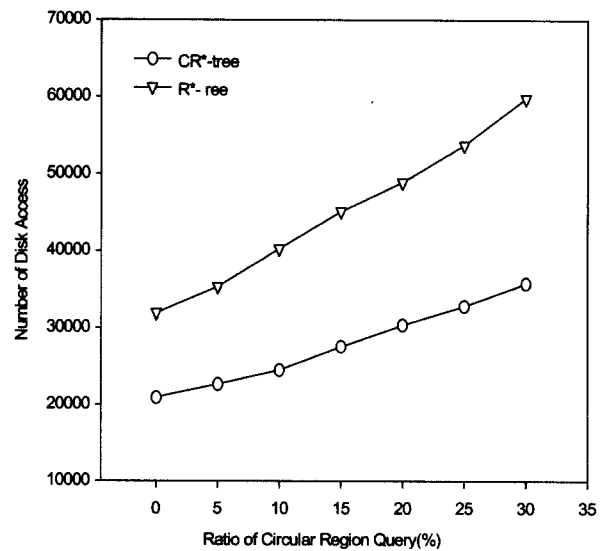
질의처리에 따른 성능 실험에서는 다음과 같은 매개 변수 값들을 입력으로 하여 순환 및 선형 영역질의를 균일하게 발생시켰다.

- 영역질의 크기: 전체 검색공간의 0.01%, 0.1%, 0.5%, 1%
- 영역질의 수: 1000
- 순환 영역질의 비율: 0%, 5%, 10%, 15%, 20%, 25%, 30%

영역질의의 크기가 크고 버퍼의 크기가 작을 때 디스크 접근 횟수는 최대가 된다. 본 논문에서는 순환 영역질의의 비



(a) 균일 분포



(b) 모서리 분포

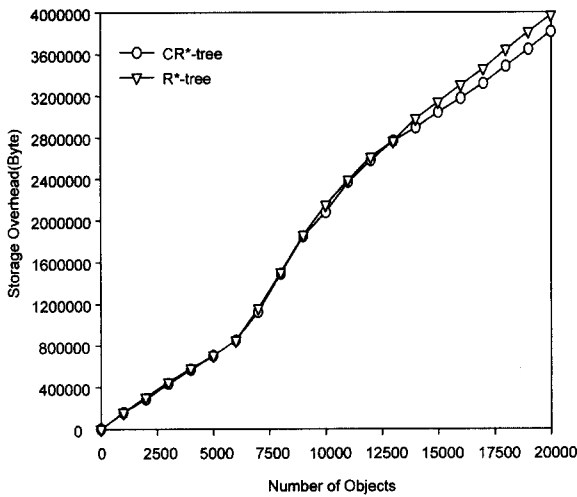
(그림 5) 디스크 접근 횟수

율과 디스크 접근 횟수의 관계를 최대의 디스크 접근 횟수를 갖는 실험 환경에 맞추어 기술한다.

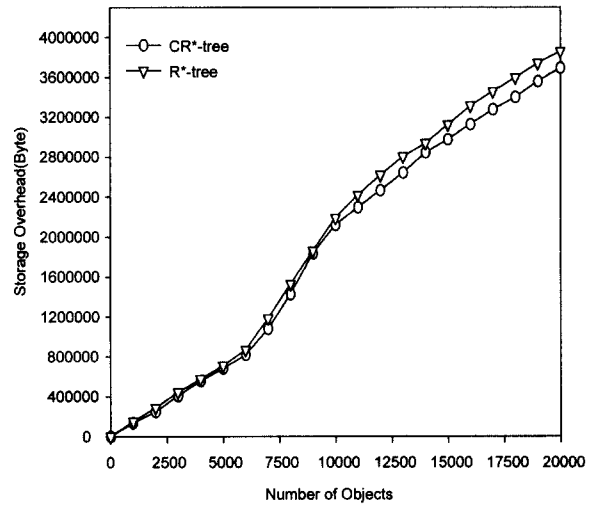
(그림 5)는 균일 분포와 모서리 분포에서 영역질의 크기가 1%이고 버퍼의 크기가 2K 바이트일 때, 순환 영역질의 비율이 높아짐에 따른 디스크 접근 횟수를 나타낸 것이다. 그림에서와 같이 CR*-트리와 R*-트리 모두가 균일 분포에서는 일정한 디스크 접근 횟수를 보이나 지수분포에서는 순환 영역질의 비율이 높아짐에 따라 디스크 접근 횟수가 많아진다. 이러한 이유는 순환적인 특성이 많이 적용된 분포일수록 객체들이 검색공간의 가장자리에 많이 존재함으로써 순환 영역질에 반응하는 버킷의 수가 많아지기 때문이다. 여러 객체 분포에 따른 반복된 실험 결과에 의하면 순환 영역질의 비율이 0%일 때 CR*-트리와 R*-트리

의 디스크 접근 횟수는 유사하였고, 순환 영역질의 비율이 높아질 때 CR*-트리는 R*-트리보다 디스크 접근 횟수가 상대적으로 낮은 증가를 보였다.

(그림 6)은 균일분포와 모서리분포에서 버퍼의 크기가 4K 바이트일 때 객체의 삽입에 따른 저장공간 사용량을 나타낸 것이다. 그림에서는 객체의 수가 6,000개일 때 저장공간 사용량의 증가에 변화를 보인다. 이는 삽입되는 객체의 수가 많아짐에 따라 해당 검색공간에서 범람이 발생하는 버킷의 수가 많아지기 때문이다. 여러 객체분포에 따른 반복된 실험결과에 의하면 CR*-트리의 저장공간 사용량의 증가형태는 버퍼의 크기에 관계없이 R*-트리와 유사하였다. 그러나 객체의 분포가 밀집되어 있고 순환적인 결집이 가능한 분포일 때, 또는 버퍼의 크기가 클 경우에 CR*-트리

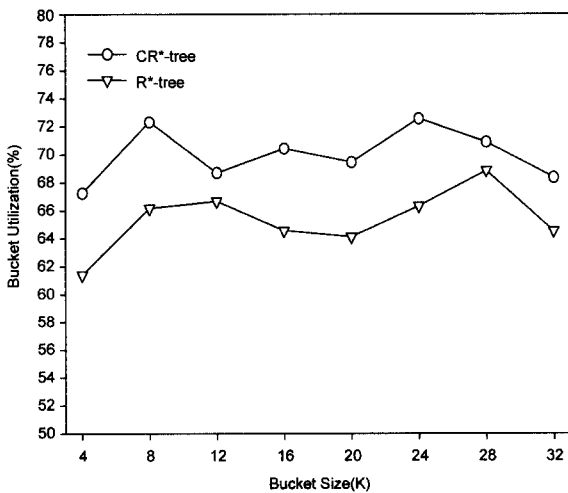


(a) 균일 분포

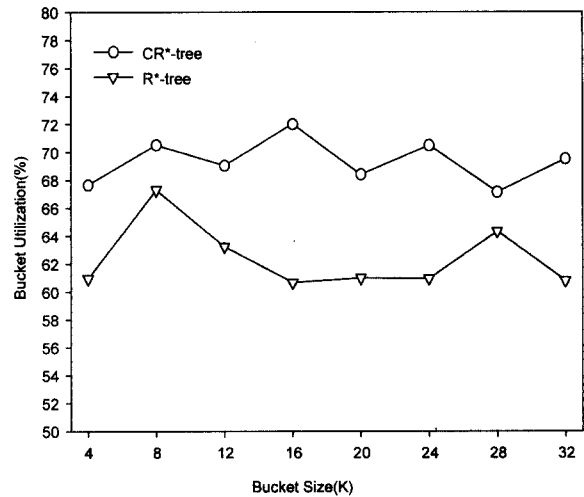


(b) 모서리 분포

(그림 6) 저장공간 사용량



(a) 균일 분포



(b) 모서리 분포

(그림 7) 버킷 이용률

의 저장공간 사용량이 R*-트리에 비해 낮음을 보였다.

(그림 7)은 버킷 용량과 버킷 이용률간의 관계를 알아보기 위해 버킷의 크기를 4K 바이트부터 32K 바이트까지 4K 바이트씩 증가시키고, 각 경우에 20,000개의 객체를 삽입한 후 균일 분포와 모서리 분포에 따른 버킷 이용률을 나타낸 것이다.

반복된 실험 결과에 의하면 CR*-트리는 R*-트리보다 높은 버킷 이용률을 유지하며, 버킷 용량의 증가에 관계없이 균일분포에서는 67%에서 73%, 모서리 분포에서는 68%에서 72% 사이의 버킷 이용률을 나타냈다. CR*-트리가 R*-트리보다 높은 버킷 이용률을 갖는 이유는 버킷 용량의 각 경우에 있어서 CR*-트리가 생성하는 버킷의 수가 R*-트리에 비해 적기 때문이다. 이는 순환 도메인이 포함된 검색공간에서 CR*-트리는 객체들의 순환적인 위치 속성을 고려하여 인접한 객체를 동일한 버킷에 저장시킴으로써 버킷의 수를 줄였기 때문이다.

5. 결 론

공간 데이터베이스 시스템에서 검색의 성능을 향상시키기 위해서는 색인의 구축시 객체가 가질 수 있는 순환적인 위치 속성이 고려되어야 한다. 기존의 공간색인 방법들에서는 객체가 가질 수 있는 순환적인 위치 속성이 고려되지 않았다.

본 논문에서는 객체의 순환적인 위치 속성을 지원하는 공간색인 구조인 CR*-트리를 설계 및 구현하고 실험을 통하여 그 성능을 평가하였다. CR*-트리는 순환 도메인과 선형 도메인으로 구성된 다차원 검색공간을 전체로 하고 있으며, 이러한 검색공간 상에서 순환 영역질의 및 선형 영역질의가 처리되는 공간색인 구조이다.

실험에서는 CR*-트리의 성능을 디스크 접근 횟수, 저장공간 사용량 그리고 버킷 이용률에 따라 평가하였다. 실험 데이터로는 균일 분포와 모서리 분포를 이루는 사각형 객체를 이용하였으며, 비교대상으로는 R*-트리를 선택하였다.

반복된 실험결과에 의하면, CR*-트리는 R*-트리에 비해 디스크 접근 횟수가 적으며, 순환 영역질의의 비율이 높아질 때 R*-트리보다 디스크 접근 횟수가 상대적으로 낮은 증가를 보였다. 또한 CR*-트리와 R*-트리의 저장공간 사용량의 증가형태는 여러 유형의 객체분포에 관계없이 유사하였다. 그리고 CR*-트리는 R*-트리보다 높은 버킷 이용률을 갖으며, 버킷 용량의 증가에 관계없이 67%에서 73%사이의 변화의 폭이 크지 않는 버킷 이용률을 유지하였다. 이러한 결과는 CR*-트리의 알고리즘에 객체들의 순환적인 위치 속성을 고려하여 인접한 객체들을 동일한 버킷에 저장하는 전략이 포함되어 있기 때문이다.

참 고 문 헌

- [1] W. G. Aref and H. Samet, "Optimization Strategies for Spatial Query Processing," Proc. of the 7th Int. Conf. on VLDB, pp.81-90, 1991.
- [2] N. Beckmann, H. Kriegel, R. Schneider and B. Seeger, "The R*-tree : an Efficient and Robust Access Method for Points and Rectangles," Proc. ACM SIGMOD Int. Conf. on Management of Data, pp.322-331, 1990.
- [3] S. Berchtold, C. Bohm, H. Kriegel, "Improving the Query Performance of High-Dimensional Index Structures Using Bulk-Load Operations," 6th. Int. Conf. on Extending Database Technology, 1998.
- [4] S. Berchtold, D. Keim, H. Kriegel, "The X-tree : An Index Structure for High-Dimensional Data," 22nd Conf. on Very Large Databases, pp.28-39, 1996.
- [5] T. Brinkhoff and H. P. Kriegel, "The Impact of Global Clustering on Spatial Database Systems," Proc. of the 20th VLDB Conf. pp.168-179, 1994.
- [6] R. H. Güting, "An Introduction to Spatial Database Systems," VLDB Journal, No.3, pp.357-399, Aug., 1994.
- [7] A. Henrich and H. W. Six, "How to Split Buckets in Spatial Data Structures," Geographic DB Management Systems, Capri (Italy), pp.212-244, May, 1991.
- [8] H. V. Jagadish, "Linear Clustering of Objects with Multiple Attributes," Proc. ACM SIGMOD Int. Conf. on Management of Data, pp.332-342, 1990.
- [9] L. K. Joune and L. Robert, "The Spatial Locality and a Spatial Indexing Method by Dynamic Clustering in Hypermap System," Proc. of the 2nd Sym. on Large Spatial Databases, pp.207-223, 1991.
- [10] N. Katayama, S. Satoh, "The SR-tree : An Index Structure for High-Dimensional Nearest Neighbor Queries," Proc. ACM SIGMOD Int. Conf. on Management of Data, pp.369-380, 1997.
- [11] B. U. Pagel, H. W. Six, H. Toben and P. Widmayer, "Towards an Analysis of Range Performance in Spatial Data Structures," Proc. of the 12th ACM SIGACT-SIGMOD-SIGART Sym. on Principles of Database Systems, pp.214-221, 1993.
- [12] Y. Theodoridis, K. Timos, "A Model for Prediction of R-tree Performance," Proc. of 15th ACM SIGACT-SIGMOD-SIGART Sym. on Principles of Database Systems, pp.161-171, 1996.
- [13] 김홍기, 황부현 "순환도메인을 기반으로 하는 PR-화일의 구현 및 성능 평가", 정보처리논문지, 제3권 제1호, pp.63-76, 1996.



선 휘 준

e-mail : hiseon@tiger.seonam.ac.kr

1988년 목포대학교 전산통계학과(이학사)

1990년 전남대학교 대학원 전산통계학과
(이학석사)

1998년 전남대학교 대학원 전산통계학과
(이학박사)

1997년~현재 서남대학교 컴퓨터정보통신학과 조교수
관심분야 : 공간자료구조, 멀티미디어 시스템, 지리정보시스템



김 흥 기

e-mail : hkkim@blue.dongshinu.ac.kr

1984년 전남대학교 계산통계학과(이학사)

1986년 전남대학교 대학원 계산통계학과
(이학석사)

1996년 전남대학교 대학원 전산통계학과
(이학박사)

1991년~현재 동신대학교 컴퓨터학과 부교수
관심분야 : 공간데이터베이스, 컴퓨터그래픽스, 멀티미디어 시스템