순환도메인을 기반으로 하는 PR-화일의 구현 및 성능 평가

김 홍 기† 황 부 현††

요 약

본 논문에서 우리는 공간객체를 취급하는 새로운 동적 공간색인구조인 PR-화일과 계층별 공간국부성 측도인 변형된 계층분산을 제안한다. 다차원 검색공간이 순환 도메인을 갖는다는 가정하에서, PR-화일은 공간적으로 이웃하는 객체들을 결집시키기 위해 변형된 계층분산을 이용한다. PR-화일의 삽입과 분할 알고리즘은 객체의 분포 형태에 관계없이 낮은 계층분산값을 갖는 색인을 유지시킨다. 실험 결과에 의하면, PR-화일은 낮은 계층분산값을 갖는 색인을 사용함으로써 객체의 검색시 적중율을 높이며, 버켓의 용량이 커질수록 버켓이용율을 향상시키는 특성을 보인다.

The Implementation and Performance Evaluation of PR-File Based on Circular Domain

Hong Ki Kim† Bu Hyun Hwang††

ABSTRACT

In this paper, we propose a new dynamic spatial index structure, called PR-file, for handling spatial objects and the modified hierarchical variance which measures the degree of spatial locality at each level. Under the assumption that a multidimensional search space has a circular domain, PR-file uses the modified hierarchical variance for clustering spatially adjacent objects. The insertion and splitting algorithms of PR-file preserve an index which has a low hierarchical variance regardless of object distributions. The simulation result shows that PR-file has a high hit ratio during a retrieval of objects by using an index with low hierarchical variance. And it shows a characteristic that the larger the bucket capacity, the higher the bucket utilization.

1. 서 론

공간 데이타베이스 시스템은 다차원 검색공간에 존재하는 숫자나 문자열과 같은 비공간데이타(aspatial data)와 위치, 모양, 크기등의 속성을 갖는 공간데이타(spatial data)로 구성된 공간객체(spatial object)를 처리한다. 이러한 공간객체(이하 객체라 함)는 컴퓨터지원설계(Computer Aided Design)와 같이 삽입이나 삭제가 빈번히 발생하는 동적인 환경과 지리정보시스템(Geographic Information System)과 같은 상대적으로 정적인 환경 모두에서 취급되어 진다. 따라서 동적 및 정적환경에서 발생하는 객체를 효율적으로 저장하고 검색하는 공간색인방법이 필요하며, 이러한 필요성을 만족시키기 위해 다양한 공간색인구조가 연구되어지고 있다[10].

공간색인방법은 다차원 검색공간에서 객체를 점으로 표현하여 색인하는 방법[2, 7, 8, 9, 11]과 객체가 차지하는 영역을 사각형 형태로 표현하여 색인하는 방법으로 구분되어 진다. 사각형에 의한 표현 방법은 영역 경계가 겹치는 문제를 가지고 있다. 이러한 문제점을 해결하기 위해 사각형을 임의 차원 검색공간의 한 점으로 사상시

† 정 회 원 : 동신대학교 전산통계학과 조교수
†† 정 회 원 : 전남대학교 전산학과 교수
논문접수 : 1995년 10월 10일, 심사완료 : 1995년 11월 27일

켜 점으로 취급하는 방법[4, 5, 13, 14], 겹치지 않는 영역으로 표현하는 방법[13, 15], 그리고 겹침을 허용하는 방법[1, 3, 12, 13]이 있으며 이들은 기억공간 이용율, 검색 효율, 색인갱신비용 면에서 각각 장단점을 가지고 있다. 본 논문에서는 R-트리 유형 중에서 겹침을 허용하는 공간색인방법[1, 3, 12]을 고려한다. 이는 본 논문에서 제시하는 순환(circular)이라는 특성과 변형된 계층분산(hierarchical variance)이라는 측도를 가장 잘 보여줄 수 있는 색인방법이기 때문이다.

기존의 공간색인방법들이 가정하는 검색공간에서 도메인은 객체의 속성이 가질 수 있는 최소와 최대값 사이의 선형적 순서 범위로 구성된다. 그러나 컴퓨터지원설계 및 지리정보시스템과 같은 응용에서 취급되는 객체 또는 시간등의 개념이 포함되어 있는 객체등은 순환적인 속성을 가질 수 있으며, 이러한 객체의 속성을 효율적으로 반영하기 위해서는 순환적인 성질을 갖는 도메인으로 구성된 검색공간이 요구된다.

다음은 도메인의 범위가 각각 0°에서 360°, 0시에서 24시인 순환적 환경에서 발생하는 영역질의(region query)와 최근접질의(nearest neighbor query) 예이다.

i) "X가 350°부터 20°사이이고 Y가 15°부터 30°사이에 있는 객체들을 검색하라"

ii) "23시에 가장 근접한 시간에 발생된 객체를 검색하라"

i)의 영역질의는 순환적인 성질이 포함된 질의범위와 순환적이지 않은 질의범위로 이루어진 예이며, ii)는 근접을 반드시 순환적인 환경에서 고려해야 하는 경우이다. 순환적인 환경의 검색공간이 주어졌을 때, 기존의 방법들은 인접한 객체들이 서로 다른 버켓에 분산되어 저장됨으로, 예에서와 같은 질의형태에 효율적으로 반응하지 못한다.

검색공간 상에 인접한 객체들이 동일한 버켓에 저장된다면 최소의 지연시간으로 이웃하는 객체들을 검색할 수 있다. 객체들이 인접하는 정도를 측정하기 위해 [6]에서는 계층분산이라는 측도를 제시하였다. 계층분산은 부검색공간의 대표값으로 객체집합의 중심을 이용한다. 따라서 이는 오차를 가지고 있는 공간국부성(spatial locality) 측도이다.

본 논문에서는 순환적인 성질을 갖는 도메인이 포함된 다차원 검색공간에서 객체를 취급하는 공간색인방법인 PR-화일(Pointed Region-File)을 제안한다. 또한 객체 및 검색공간들 간의 계층별 공간국부성을 측정하기 위해, 계층분산이 가지고 있는 오차를 제거한 변형된 계층분산이라는 측도를 제안한다. PR-화일은 R-트리에 순환적인 특성을 확장시킨 색인구조를 가지고 있으며, 변형된 계층분산이 적용된 연산 알고리즘에 의해 색인을 구축한다.

논문의 구성은 다음과 같다. 관련연구인 2장에서는 공간색인방법을 공간국부성 측면에서 알아보고, 기존에 제시된 계층분산이 가지는 문제점에 대해 논한다. 3장에서는 본 논문에서 제안한 PR-화일의 기본개념 및 구조 그리고 연산 알고리즘을 기술한다. 4장에서는 PR-화일의 성능을 R-트리와 비교 분석하고, 끝으로 5장에서 결론을 내린다.

2. 관련 연구

본 장에서는 기존의 공간색인방법들이 가지고 있는 문제점을 공간국부성 측면에서 알아보고, 계층분산이라는 측도와 이를 이용한 객체의 결집(clustering) 방법이 가지는 문제점을 기술한다.

객체를 정보의 손실없이 점으로 변환시키거나 또는 동일한 비율 및 크기를 갖는 단위 셀(cell)로 구성시켜 셀의 중심점을 이용한다면, 점으로 객체를 표현할 수 있다. 본 논문에서는 공간국부성과 계층분산이라는 개념을 단순화하기 위해 객체를 점으로 나타내어 설명하며, 질의 유형으로는 가장 일반적인 영역질의 만을 고려한다.

2.1 공간국부성

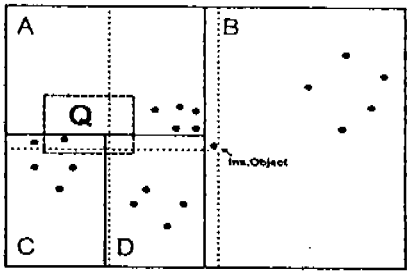
화일에서 일련의 객체들을 접근할 때 만일 임의의 객체가 그 전위 객체에 공간적으로 인접해 있다면 최소의 지연시간으로 해당 객체를 접근할 수 있는데, 이처럼 객체들이 공간상에 서로 이웃하고 있는 정도를 공간국부성이라 한다. 다차원 검색공간에서 인접하는 객체들을 동일한 버켓에

결집시킴으로써 검색의 효율을 향상시킬 수 있다. 따라서 각 버켓에 객체가 저장될 때 효율적인 결집이 이루어지기 위해서는 공간국부성(이하 국부성이라 함)이 반드시 고려되어야 한다.

BANG 화일[2], hB-트리[7], NIBGF[9]등과 같은 객체를 점으로 표현하는 공간색인방법들에서, 검색공간의 분할방법은 국부성 개념이 기본적으로 포함되어 있다. 그러나 이러한 방법들은 인접하는 객체들을 결집시키려는 노력보다는 색인의 크기를 줄이는 데에 관점을 맞추고 있다. 또한 영역질의 처리시 의미없는 버켓이 반응하는 단점을 가지고 있다.

(그림 1)에서는 버켓용량(bucket capacity)이 6일 경우, 분할 결과 생성된 버켓영역이 분할 후 삽입될 객체에 미치는 영향과 영역질의에 반응하는 버켓을 보여준다. 삽입될 객체는 버켓영역 B보다는 A에 포함됨으로써 좋은 국부성을 갖는다. 그러나 객체를 A에 포함시키기 위해서는 검색공간의 분할 위치를 재 설정하여야 한다(점선 참조). 이는 이웃하는 모든 버켓영역에 영향을 줌으로 결국 색인을 다시 구축하여야 하며, 색인의 크기가 증가될 수도 있다. 또한 예에서는 영역질의Q가 발생했을 때 의미없는 두개의 버켓영역(;A,D)을 접근함으로써 검색 효율이 저하됨을 보여준다.

R-트리[3], R^+-트리[15], R^*-트리[1]등과 같이 객체를 사각형으로 표현하는 공간색인방법들은 버켓영역 및 부검색공간을 최소경계사각형(Minimum Bounding Rectangle : MBR)을 취해 색인을 구축한다. MBR을 이용함으로써 질의 처리시 조건에 만족하지 않은 버켓영역들을 미리 제거하여 불필요한 버켓의 접근 횟수를 줄일 수



(그림 1) 버켓영역과 공간국부성
(Fig. 1) Bucket region and spatial locality

있다. 또한 객체가 속할 버켓영역은 확장이 가능함으로 인접하는 객체들을 동일한 버켓에 포함시키기 위해 색인을 재구축할 필요가 없다. R^*-트리에서는 검색의 효율에 영향을 주는 변수로 다음과 같은 사항들을 고려하였다.

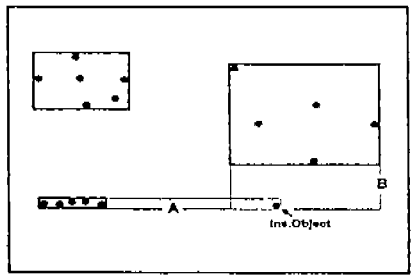
i) MBR의 면적을 최소화한다.

ii) MBR들 간의 겹치는 영역을 최소화한다.

iii) MBR의 둘레(margin) 합을 최소화한다.

iv) 저장공간 이용을 최적화한다.

i), ii), iii)은 iv)와 손익관계가 있으며, Packed R-트리[12]에서는 i)과 ii)를 동시에 만족시킬 수 없음을 보였다. 객체의 삽입 및 버켓의 분할에 있어서, i), iii)은 버켓내에 존재하는 객체들간의 거리가 대우 멀어지는 MBR을 형성할 가능성이 높기 때문에 좋은 국부성을 보장할 수 없다.



(그림 2) 객체삽입시 버켓영역 선택
(Fig. 2) Selection of bucket region at an object insertion

(그림 2)에서는 객체가 삽입될 버켓영역을 선택하는 두 가지 경우를 보여준다. A는 생성될 MBR의 면적 및 둘레의 합을 고려한 경우이고, B는 국부성을 고려한 경우이다. A를 선택한다면, 생성된 버켓은 인접하지 않은 객체가 포함됨으로써 국부성이 낮아진다. 그리고 다른 MBR들과 겹치는 영역의 면적은 줄어들 수 있지만, 겹치는 MBR의 갯수가 많아질 확률이 매우 높다.

2.2 객체의 동적결집

동적결집(dynamic clustering)에 의한 공간색인방법[6]에서는 국부성의 측정방법으로 전체분산(total variance)과 계층분산을 도입하였으며, 계층분산값이 낮을수록 질의에 반응하는 버켓의

수가 감소하고 적중율이 높아짐을 증명하였다.

전체분산은 검색공간상에 존재하는 객체들의 분포정도를 나타내고, 전체분산값이 낮을때 객체들간의 국부성은 강하다. 전체분산 V_T^2은 다음과 같이 정의되었다.

$$V_T^2 = \frac{1}{n}[P-G]W[P-G]^T$$

여기에서

$P=[p_1, p_2, \cdots, p_n]$

(p_i : i번째 객체의 공간적 위치),

$G=[g, g, \cdots, g]$

(g : 검색공간에 존재하는 객체집합의 중심)

W : 가중행렬.

버켓용량과 검색의 효율성을 고려하여 검색공간이 부검색공간들로 분할되고 또한 계층을 갖는다면, 전체분산으로는 이러한 부검색공간들간의 국부성 정도를 표현할 수 없다. 계층분산은 객체들간의 국부성 및 계층별 부검색공간들간의 국부성을 고려한 국부성 측정방법이다. i계층의 검색공간 A에 따른 계층분산 $V_{HA}^2(i)$은 다음과 같이 정의되었다.

만약 A가 최하위 계층에 존재하면

$$V_{HA}^2(i)=\frac{1}{n(A)}\sum_{k=1}^{n(A)} V_{Tk}^2(i+1)+ \frac{1}{n}[H-G]W[H-G]^T$$

그렇지 않으면

$$V_{HA}^2(i)=\frac{1}{n(A)}\sum_{k=1}^{n(A)} V_{Hk}^2(i+1)+ \frac{1}{n}[H-G]W[H-G]^T$$

여기에서

$V_{Tk}^2(i+1)$: k번째 부검색공간의 전체분산,

$V_{Hk}^2(i+1)$: k번째 부검색공간의 계층분산,

$H=[g_1, g_2, \cdots, g_{n(A)}]$

(g_k : k번째 부검색공간에 존재하는 객체 집합의 중심),

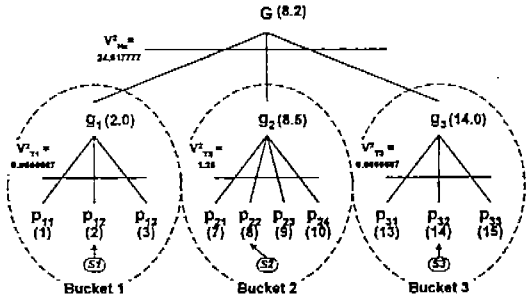
$G=[g, g, \cdots, g]$

(g : 검색공간 A에 존재하는 객체집합의 중심),

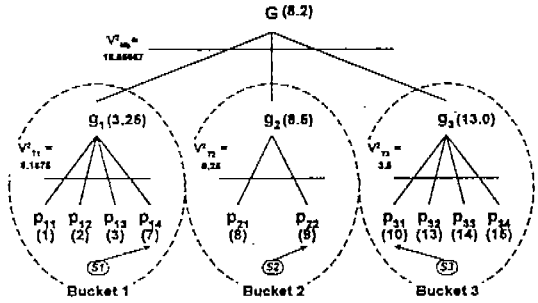
$n(A)$, n : i계층의 부검색공간 수.

제시된 공간색인방법은 검색공간의 분할시 낮은 계층분산값을 갖는 부검색공간을 생성하여 색인을 구축함으로써 질의의 효율성을 높이고자 하는 것이다. 그러나 다음과 같은 문제점을 가지고 있다.

- 계층분산은 모든 버켓에 동일한 갯수의 객체가 존재하고 균형트리를 갖는 색인을 유지하였을때 국부성을 나타내는 정확한 측도가 된다.
- 부검색공간의 대표값으로 객체집합의 중심값을 사용하였으며, 객체는 중심값에 가장 근접한 버켓에 결집된다. 중심값은 해당 부검색공간에 존재하는 객체 전체의 국부성을 측정하는데 오차를 포함하고 있다. 따라서 계층분산은 상위계층으로 갈수록 오차가 누적되므로



(a) V^2_{Ha} = 24.917777



(b) V^2_{Hb} = 18.85667

(그림 3) 객체의 결집에 따른 계층분산 결과

(Fig. 3) Hierarchical variance results by objects clustering

정확한 국부성 측도가 되지 못한다. (그림 3)은 객체집합의 중심값을 이용한 객체의 결집 방법을 보여주는 두가지 예이다. 예에서 (a)는 (b)보다 더 좋은 국부성을 유지하고 있음을 알 수 있다. 그러나 두집단에 따른 계층분산의 관계는 $V_{Ha}^2 > V_{Hb}^2$ 이 되므로, 검색공간에 존재하는 객체의 수를 고려하지 않고 중심값만을 이용하는 계층분산은 국부성의 정확한 측도가 되지 못한다.

- 정적객체로 구성된 데이타 취급을 전제로 하였다. 최근의 응용에서는 정적인 환경뿐만 아니라 동적인 환경에서 발생하는 데이타를 취급하는 색인구조의 필요성이 요구되고 있다.
- 검색공간의 분할결과 생성되는 부검색공간들은 겹치지 않는다고 가정하였다. 이러한 가정은 최적의 계층분산을 유지시키는데 강한 제약조건이 된다.

3. PR-화일

3.1 기본 개념

PR-화일은 순환적인 성질을 갖는 도메인이 포함된 다차원 검색공간에서 객체를 취급하는 공간색인방법이다. PR-화일의 연산 알고리즘은 변형된 계층분산을 이용한다. 본 절에서는 PR-화일의 이론적 기반이 되는 순환 성질을 갖는 도메인 및 영역질의를 정의한다. 또한 기존의 계층분산이 가지고 있는 문제점을 해결한 변형된 계층분산을 제안하고 그 특성을 보인다.

3.1.1 순환 도메인과 순환 영역질의

기존의 공간색인방법에서, N차원 검색공간을 구성하는 임의의 도메인 $D_i(i=1, 2, \cdots, N)$은 해당 검색공간에 존재하는 객체들의 i번째 속성이 가질 수 있는 최소와 최대값사이의 서로 다른 m개의 값들을 갖는 선형적 순서 범위로 이루어져 있다. 즉, $D_i=(d_0, d_1, \cdots, d_{m-1})$이다.

객체의 속성값을 매개 변수로 하여 임의 속성값의 다음 또는 이전 속성값을 반환하는 함수를 각각 *next*, *prev*라 하자. 본 논문에서는 다음과 같은 정의에 의해 도메인을 순환 도메인(circular domain)과 선형 도메인(linear domain)으로 구분한다.

[정의 1] 도메인 D_i이 다음과 같은 성질을 만족할 때, D_i를 순환 도메인이라 한다.

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

[정의 2] 도메인 D_i이 다음과 같은 성질을 만족하고 $next(d_{m-1})$, $prev(d_0)$에 의해 반환되는 속성값이 존재하지 않는다면, D_i를 선형 도메인이라 한다.

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

PR-화일에서의 N차원 검색공간은 S개의 순환 도메인 및 N-S개의 선형 도메인의 카테시언 곱(cartesian product)이다(단, $0\le S\le N$). S가 0이면, 기존의 공간색인방법들이 전제로한 선형 도메인으로 구성된 검색공간이다.

N차원 검색공간에서, 영역질의 Q는 질의 범위들의 카테시언 곱으로 표현된다. 즉, 도메인 D_i에 발생하는 질의 범위를 $I_i=[s_i, e_i]$ ($i=1,2, \cdots, N$, s_i, e_i : 질의 범위의 시작과 끝)라 할 때, $Q=I_1\times I_2\times\cdots\times I_N$로 나타낼 수 있다. 본 논문에서는 다음 정의에 의해 영역질의를 순환 영역질의(circular region query)와 선형 영역질의(linear region query)로 구분한다.

[정의 3] 영역질의 Q를 구성하는 하나 이상의 질의 범위 I_i가 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.

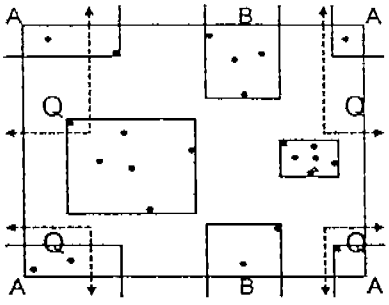
[정의 4] 영역질의 Q를 구성하는 모든 질의 범위 I_i가 순환 또는 선형 도메인 D_i 상에 발생하고 $s_i < e_i$인 성질을 가지면, Q를 선형 영역질의라 한다.

순환 도메인이 포함된 다차원 검색공간이 존재한다면, 순환적인 환경을 전제로 객체를 결집시킴으로써 버켓이용 및 국부성을 높이고 질의 처리시 접근되는 버켓의 수를 줄일 수 있다.

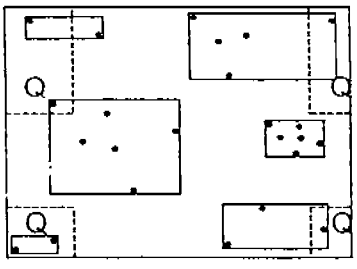
(그림 4)는 2차원 검색공간에서 순환 영역질의 Q가 발생했을 때의 예이다. (a)는 순환을 고려하여 객체를 결집시키는 PR-화일을 적용한 경우로 2번의 버켓접근을 통해 질의를 처리할 수 있다. 순환을 고려하지 않은 색인방법을 적용한 (b)에서는 객체의 결집 후 생성된 버켓의 수가 (a)의 경우보다 많다. 또한 하나의 순환 영역질의가 4개의 선형 영역질의로 분할되어 취급되어져야 하며, 5번의 버켓접근이 요구된다.

3.1.2 변형된 계층분산

일반적인 응용에서 다차원 검색공간에 존재하는 모든 객체들은 동일하게 취급되며, 객체들 간의 인접정도는 유클리드거리(euclidean distance)로 측정된다. 따라서 본 논문에서는 전체분산과 계층분산에 포함되어 있는 가중행렬을 단위행렬로 가정한다.



(a) 순환 도메인으로 구성된 검색공간



(b) 선형 도메인으로 구성된 검색공간

(그림 4) 검색공간에서의 순환 영역질의
(Fig. 4) Circular region query in search space

[6]에서 제시된 계층분산은 부검색공간들 간의 국부성 정도를 측정하기 위해 부검색공간의 대표값으로 객체집합의 중심값을 이용하였다. 그러나 대표값은 오차를 가지고 있기 때문에 이와같은 국부성 측정방법은 실질적으로 다루어져야 할 부검색공간들에 존재하는 객체들간의 국부성을 정확히 반영하지 못한다. 이러한 문제점을 해결하기 위해 본 논문에서는 변형된 계층분산 V_H^2을 다음과 같이 제안하며, 이하 논문에서의 계층분산은 변형된 계층분산을 의미한다.

[정의 5] i계층의 임의의 검색공간 A에 따른 계층분산 $V_{HA}^2(i)$은 다음과 같이 정의한다.

만약 A가 최하위 계층에 존재하면

$$V_{HA}^2(i)=\frac{1}{n}\sum_{k=1}^{n}V_{Tk}^2(i+1)+V_{TA}^2(i),$$

그렇지 않으면

$$V_{HA}^2(i)=\frac{1}{n}\sum_{k=1}^{n}V_{Hk}^2(i+1)+V_{TA}^2(i),$$

여기에서

$V_{Tk}^2(i+1)$: k번째 버켓의 전체분산,

$V_{Hk}^2(i+1)$: 부검색공간의 계층분산,

$V_{TA}^2(i)$: 검색공간 A의 전체분산,

n : i계층의 버켓 수 또는 부검색공간 수.

계층분산은 임의 검색공간에 존재하는 모든 객체의 국부성과 부검색공간들의 평균 국부성을 고려한 계층별 국부성 척도이다. 본 논문에서 정의한 계층분산은 [6]에서 제시된 계층분산에 포함되어 있는 오차를 제거한 것으로, 버켓에 존재하는 객체의 수 및 색인구조로 사용된 트리형태에 관계없이 국부성 정도를 정확히 측정할 수 있다. (그림 3)의 예를 정의한 계층분산에 의해 측정하면 $V_{Ha}^2 : V_{Hb}^2 = 23.42111 : 25.53917$이 되어 (a)가 더 좋은 국부성을 가지고 있음을 알 수 있다.

정리 1에서는 객체가 버켓에 삽입 또는 삭제될 때 해당 버켓에 따른 기존 정보를 이용함으로써 전체분산의 연산 시간을 줄일 수 있음을 보인

다. 그리고 정리 2에서는 변경된 버켓의 전체분산은 최상위 계층의 계층분산에 까지 영향을 미침을 보인다. 따라서 버켓이 분할될 때 분할 후 생성될 버켓의 전체분산이 최소가 되게 함으로써 낮은 계층분산을 유지할 수 있다.

[정리 1] 객체가 버켓에 삽입 또는 삭제될 때, 버켓의 전체분산은 해당 버켓에 존재하는 기존의 객체 수와 객체집합의 중심 그리고 전체분산에 의존한다.

증 명) 임의의 버켓에 따른 기존의 전체분산을 V_T^2라 하자. 새로운 객체 p가 해당 버켓에 삽입된다면 변경된 전체분산 $V_T^{2\prime}$은 다음과 같다.

$$V_T^{2\prime}=\frac{1}{n+1}[P'-G'][P'-G']^T$$

$$=\frac{1}{n+1}[P'-G+G-G'][P'-G+G-G']^T$$

$$=\frac{1}{n+1}\sum_{i=1}^{n+1}((p_i-g)^2+2(p_i-g)(g-g')+(g-g')^2)$$

$$=\frac{n}{n+1}V_T^2+\frac{1}{n+1}(p-g)^2-\frac{2}{(n+1)^2}(p-g)^2+\frac{1}{(n+1)^2}(g-p)^2$$

$$=\frac{n}{n+1}V_T^2+\frac{1}{n+1}(p-g)^2\left(1-\frac{1}{n+1}\right)$$

$$=\frac{n}{n+1}V_T^2+\frac{n}{(n+1)^2}(p-g)^2 \quad \cdots\cdots\cdots\cdots (1)$$

여기에서

$P'=[p_1, p_2, \cdots, p_{n+1}]$, $p\in P'$

(p_i : i번째 객체의 공간적 위치),

$G'=[g', g', \cdots, g']$

(g' : 버켓에 존재하는 객체집합의 중심).

동일한 방법으로, 버켓에 존재하는 객체 p가 삭제된다면 변경된 전체분산 $V_T^{2\prime}$은 다음과 같다.

$$V_T^{2\prime}=\frac{n}{n-1}V_T^2-\frac{n}{(n-1)^2}(p-g)^2 \quad \cdots\cdots (2)$$

따라서 버켓의 전체분산 $V_T^{2\prime}$은 기존 버켓에 존재하는 객체의 수 n와 객체집합의 중심 g 그리고 전체분산 V_T^2에 의존한다.

[정리 2] 버켓이 새로운 두 개의 버켓으로 분할될 때, 계층분산은 분할에 참여한 버켓 수와 분할 후 생성된 버켓들의 전체분산에 의존한다.

증 명) $i-1$계층의 임의 검색공간 B가 q개의 부검색공간을 가지며 부검색공간 중 하나를 A라 하자. A가 최하위 계층에 존재한다고 할 때, A에 있는 n개의 버켓 중 m개의 버켓이 분할에 참여한다면, 분할 후 버켓의 수는 $n+m$개가 된다. 버켓의 분할 후에도 A의 전체분산 $V_{TA}^2(i)$에는 변화가 없다. 따라서 버켓의 분할 후 변경된 A의 계층분산 $V_{HA}^{2\prime}(i)$은 다음과 같다.

$$V_{HA}^{2\prime}(i)=\frac{1}{n+m}\left\{\sum_{k=1}^{n-m}V_{Tk}^2(i+1)+\sum_{S=1}^{2m}V_{TS}^{2\prime}(i+1)\right\}+V_{TA}^2(i)$$

여기에서

$V_{TS}^{2\prime}(i+1)$: 버켓 분할 후 생성된 s번째 버켓의 전체분산.

A의 상위계층 검색공간이 B이므로, 버켓의 분할에 따른 변경된 B의 계층분산 $V_{HB}^{2\prime}(i-1)$은 다음과 같이 표현된다.

$$V_{HB}^{2\prime}(i-1)=\frac{1}{n}\left\{\sum_{k=1}^{n-1}V_{Hk}^2(i)+V_{HA}^{2\prime}(i)\right\}+V_{TB}^2(i-1)$$

또한 A내에 있는 버켓 수의 증가로 인해 A가 r개의 검색공간으로 분할되었다고 하면, B의 부검색공간의 수가 증가하여도 B의 전체분산은 동일하므로, 검색공간 A의 분할에 따른 변경된 B의 계층분산 $V_{HB}^{2\prime}(i-1)$은 다음과 같다.

$$V_{HB}^{2\prime}(i-1)=\frac{1}{q+r-1}\left\{\sum_{k=1}^{q-1}V_{Hk}^2(i)+\sum_{S=1}^{r}V_{HAS}^{2\prime}(i)\right\}+V_{TB}^2(i-1)$$

여기에서

$V_{HA_s}^{2\prime}(i)$: 검색공간 분할 후 생성된 s번째 검색공간의 계층분산.

이는 해당 경로를 따라 최상위 계층의 계층분산에 까지 계속 영향을 미친다. 따라서 버켓의 분할이 발생했을 때, 계층분산은 분할에 참여한 버

켓 수 m과 분할 후 생성된 버켓들의 전체분산 $V_{TS}^{2}{}'(i+1)$에 의존한다.

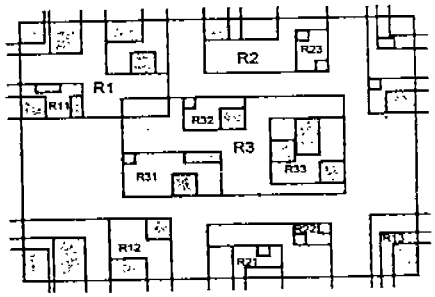
정리 1, 2는 PR-화일의 색인구축을 위한 기본 개념이다. 즉, 객체의 삽입은 해당 객체를 버켓에 포함시켰을 때 버켓의 전체분산 증가가 최소가 되는 버켓을 선택하여 이루어지고, 버켓의 분할방법은 분할 후 생성된 버켓의 전체분산이 최소가 되는 전략을 갖게함으로써 최적의 계층분산을 갖는 PR-화일의 색인이 구축된다.

3.2 PR-화일의 구조

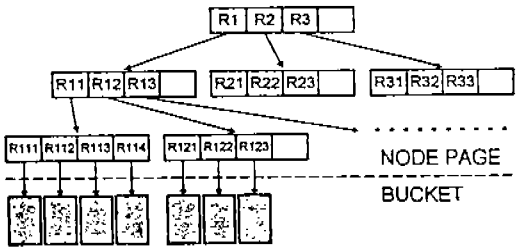
PR-화일은 R-트리와 유사한 색인 구조를 따르며, 실질적인 객체들이 저장되는 버켓과 버켓을 접근하기 위해 경로 정보를 제공하는 노드페이지(node page)로 구성된 높이균형트리이다.

노드페이지NP는 하나의 헤더와 (R, P_{child})인 엔트리들 E로 구성된 일차원 배열 구조를 갖는다. R은 각 차원의 폐쇄 간격(closed interval)들로 이루어진 N차원 MBR이다.

즉, $R=[RI_S, RI_E]_1 \times [RI_S, RI_E]_2 \times \cdots \times [RI_S, RI_E]_N$로 나타낼 수 있으며, 여기에서 RI_S는 폐쇄 간격의 시작값을 RI_E는 폐쇄 간격의 끝값을 나타낸다.



(a) 검색공간



(b) 색인구조

(그림 5) PR-화일의 구조
(Fig. 5) A structure of PR-File

노드페이지가 트리의 리프에 존재한다면, R은 검색공간에서 하나의 버켓이 차지하는 MBR을 나타내며 P_{child}는 버켓의 주소지시자이다. 노드페이지가 리프에 존재하지 않는다면, R은 자식 노드페이지에 속한 모든 객체를 포함하는 MBR이며 P_{child}는 자식 노드페이지의 주소지시자이다.

객체의 효율적인 결집을 위해 노드페이지 및 버켓의 헤더는 포함된 객체의 수와 객체집합의 중심 그리고 전체분산등의 정보를 가지고 있다.

만약 주기억장치의 용량이 충분하지 못하여 보조기억장치에 색인이 위치한다면, 객체의 삽입 및 삭제시 헤더에 포함된 정보를 갱신하기 위해, 잦은 보조기억장치의 접근이 요구된다. 따라서 헤더 정보는 주기억장치내에 상주하는 것이 바람직하다.

하나의 노드페이지에 들어갈 수 있는 최대 엔트리 수를 M이라 하고, $m(\leq \frac{M}{2})$을 최소 엔트리 수라고 했을 때 PR-화일은 다음과 같은 성질을 갖는다.

- 모든 리프 노드페이지는 해당 노드페이지가 루트가 아닐 때 최소 m, 최대 M개의 엔트리를 갖는다.
- 리프 노드페이지의 엔트리 (R, P_{child})에서, P_{child}가 가리키는 버켓내의 객체들이 순환적으로 결집되었다면 R은 $RI_S > RI_E$인 관계를 가지는 폐쇄 간격을 포함한다.
- 루트와 리프가 아닌 모든 노드페이지는 최소 m, 최대 M개의 자식 노드페이지를 갖는다.
- 리프가 아닌 노드페이지의 엔트리 (R, P_{child})에서, P_{child}가 가리키는 자식 노드페이지의 엔트리들이 순환적으로 결집되었다면 R은 $RI_S > RI_E$인 관계를 가지는 폐쇄 간격을 포함한다.
- 리프가 아닌 루트 노드페이지는 적어도 2개의 자식 노드페이지를 갖는다.
- 모든 리프 노드페이지는 동일한 계층에 존재한다.

(그림 5)는 순환도메인을 갖는 2차원 검색공

간에서 노드페이지의 최대 엔트리 수가 4일 때, PR-화일에서 버켓영역들의 포함관계와 그에 따른 색인구조를 나타낸 것이다. (그림 5)에서 R1과 R2는 순환적 성질을 갖는 MBR이다.

3.3 알고리즘

본 절에서는 PR-화일의 기본적인 연산 알고리즘인 검색, 삽입 그리고 삭제 알고리즘을 기술한다. PR-화일의 연산 알고리즘에서 순환 도메인이 포함된 검색공간은 논리적인 검색공간으로 확장되어 해석된다. 예를 들면, 폐쇄 간격이 $RI_S > RI_E$인 관계를 갖는 MBR은 폐쇄간격의 끝값 RI_E에 해당 도메인의 최대값을 더하여 논리적인 MBR로 취급된다.

3.3.1 검색 알고리즘

PR-화일에서 영역질의에 만족하는 버켓을 찾기 위한 검색 연산은 루트 노드페이지부터 시작해서 하위 계층의 노드페이지까지 재귀적인 방법으로 이루어 진다. 순환 영역질의는 순환 성질을 갖는 질의 범위의 끝값에 해당 도메인의 최대값을 더함으로써 논리적인 영역질의로 처리된다. 검색 연산은 P를 루트 노드페이지로 놓고 다음과 같은 알고리즘을 적용한다. 알고리즘에서 사용된 Get_NodePage는 주소지시자가 가리키는 노드페이지를 읽어오는 절차(procedure)이며, overlap은 질의와 엔트리의 MBR을 입력으로 하여 겹침을 판별하는 함수이다. 그리고 Return_Objects는 주소지시자가 가리키는 버켓에서 질의에 속한 객체들을 출력하는 절차이다.

```
Procedure Search(Q:query;P:pointer);
{ Input : Range query Q(I1,I2, ... ,IN),
          Ii denotes a range [si,ei] of Di
  Output : All objects overlapping Q }
BEGIN
  FOR each Ii DO  IF si > ei THEN ei := ei + high_value(Di);
  Get_NodePage(P,NP);
  IF NP.type = leaf_node_page THEN
    FOR all E DO  IF overlap(Q,E.I,R) THEN Return_Objects(Q,E.Pchild);
  ELSE
    FOR all E DO  IF overlap(Q,E.I,R) THEN Search(Q,E.Pchild);
  END; (* IF *)
END;
```

3.3.2 삽입 알고리즘

PR-화일의 삽입 알고리즘은 계층분산을 이용함으로써 버켓영역 및 부검색공간에 포함되는 객체들 간의 높은 국부성을 보장한다. 객체의 삽입과 버켓의 분할은 계층분산에 영향을 미친다. 알고리즘에서 Adjust_Index는 변경된 버켓의 경로상에 존재하는 상위 노드페이지에 속한 엔트리의 MBR과 헤더에 포함되어 있는 전체분산값을 조정하고, 노드페이지의 분할 및 병합이 요구되면 상위 경로를 따라 재귀적으로 색인을 갱신하는 절차이다.

```
Procedure Insert(O:object);
{ Input : An object O
  Output : None }
VAR
  B,B1,B2 : pointer; (* Bucket pointer *)
BEGIN
  ChooseBucket(O,B); Put_Object(O,B);
  IF overflow(B) THEN
    Assign_New_Buckets(B1,B2); SplitBucket(B,B1,B2);
  END; (* IF *)
  Adjust_Index(B,B1,B2);
END;
```

객체의 삽입은 객체를 포함시켰을 때 계층분산 증가분이 최소가 되는 버켓을 선택하여 이루어진다. 정리 1의 (1)식에 의해, 버켓 및 노드페이지의 전체분산은 기존 헤더정보를 이용하여 연산되어짐으로 계층분산의 연산시간을 줄일 수 있다. 삽입 대상 버켓을 찾는 알고리즘은 다음과 같다. 알고리즘에서 Min_Increase_TotalVariance는 객체를 포함시켰을 때 최소의 전체분산 증분을 갖는 엔트리 및 버켓을 찾는 절차이다. Candidate_Buckets은 객체가 포함될 후보 버켓들과 버켓수를 찾는 절차이다. 그리고 min_object_bucket은 후보 버켓들 중에서 최소의 객체를 갖는 버켓의 주소를 반환하는 함수이다.

```
Procedure ChooseBucket(O:object;B:pointer);
{ Input : An object O
  Output : Bucket pointer B }
VAR
  SE  : node_page;
  CB  : array of node_page;
  NB  : integer; (* No of candidate bucket *)
BEGIN
  Get_NodePage(ROOT,NP);
```

```
    WHILE NP.type <> leaf_node_page DO
      Min_Increase_TotalVariance(O,E,SE);
      Get_NodePage(SE.P_child,NP);
    END; (* WHILE *)
    Candidate_Buckets(O,E,CB,NB);
    CASE NB OF
      0 : Min_Increase_TotalVariance(O,E,SE);
          B := SE.P_child;
      1 : B := CB1.P_child;
      ELSE B := min_object_bucket(CB);
    END; (* CASE *)
  END;
```

정리 2에서 설명된 바와 같이, 버켓이 분할될 때 계층분산은 분할 후 생성된 버켓의 전체분산에 의해 영향을 받는다. 따라서 버켓의 분할 알고리즘은 분할 후 생성될 두 개의 버켓이 최소의 전체분산값을 갖도록 하는 전략을 갖는다. 알고리즘에서 Max_Distance_Objects는 버켓에 존재하는 객체들 중 가장 멀리있는 두 개의 객체를 선택하는 절차이다. Put_Rest_Objects는 넘람된 버켓에 남아있는 모든 객체를 새로운 버켓에 저장하는 절차이다. Candidate_Object는 할당되지 않은 각 객체 중에서 두 개의 새로운 버켓에 각각 포함시켰을 때 버켓의 전체분산 증가분이 가장 큰 객체를 선택하는 절차이다. 그리고 Select_Bucket_&_Put_Object는 전체분산 증가분이 적은 버켓에 객체를 저장시키는 절차이다.

```
Procedure SplitBucket(B,B1,B2:pointer);
{ Input : An overflow bucket B, New buckets B1,B2
  Output : None }
VAR
  O1,O2 : object;
BEGIN
  Max_Distance_Objects(B,O1,O2);
  Put_Object(O1,B1); Delete_Object(O1,B);
  Put_Object(O2,B2); Delete_Object(O2,B);
  WHILE no_of_objects(B) <> 0 DO
    IF no_of_objects(B1) > min_objects THEN
      Put_Rest_Objects(B,B2); EXIT;
    ELSE IF no_of_objects(B2) > min_objects THEN
           Put_Rest_Objects(B,B1); EXIT;
         ELSE
           Candidate_Object(B,B1,B2,O1);
           Select_Bucket_&_Put_Object(O1,B1,B2);
           Delete_Object(O1,B);
    END; (* IF *)
  END; (* WHILE *)
END;
```

3.3.3 삭제 알고리즘

객체의 삭제가 요구될 때, 삭제 알고리즘에서는 먼저 삭제될 객체가 속해있는 대상 버켓을 찾고 해당 객체를 삭제한다. 삽입 알고리즘에서와 마찬가지로 객체의 삭제와 버켓의 병합은 계층분산에 영향을 미친다. 헤더에 포함되어 있는 전체분산의 조정은 정리 1의 (2)식을 이용함으로써 연산시간을 줄일 수 있다. 삭제 알고리즘에서 find_bucket은 객체가 포함된 버켓의 주소지시자를 반환하는 함수이며, find_buddy는 이웃하는 버켓 중 병합 후 최소의 전체분산 증분을 갖는 버켓의 주소지시자를 반환하는 함수이다. 그리고 Merge_Bucket은 두 개의 버켓을 병합하는 절차이다.

```
Procedure Delete(O:object);
{ Input : An object O
  Output : None }
VAR
  B1,B2 : pointer;
BEGIN
  B1 := find_bucket(O);
  IF B1 = null THEN RETURN;
  Delete_Object(O,B1);
  IF underflow(B1) THEN
    B2 := find_buddy(B1); Merge_Bucket(B1,B2);
    Delete_Bucket(B2);
  END; (* IF *)
  Adjust_Index(B1,B2,);
END;
```

4. 성능 평가

본 장에서는 PR-화일의 성능을 계층분산, 적중율 그리고 버켓이용율에 따라 평가한다. PR-화일의 비교 대상으로는 순환적인 특성과 계층분산을 가장 잘 비교할 수 있는 R-트리를 선택하였으며, R-트리의 2차 함수비용(quadratic cost) 및 선형 함수비용(linear cost)의 분할 알고리즘을 각각 고려하였다. 성능평가를 위해 사용된 적중율과 버켓이용율은 다음과 같다[4].

$$\text{적중율} = \frac{\text{발견된 객체의 수}}{\text{버켓 용량} \times \text{버켓 접근 횟수}}$$

$$\text{버켓 이용율} = \frac{\text{저장된 객체의 수}}{\text{버켓 용량} \times \text{버켓 수}}$$

실험에서는 2개의 순환도메인으로 구성된 검색공간을 가정하였으며, 크기가 80바이트이고 중복되지 않은 20,000개의 객체들을 데이타로 사용하였다. 도메인의 범위가 [0,1]이라 할 때, 실험에서 사용된 객체의 분포는 다음과 같다(그림 6 참조).

-균일분포(uniform distribution) : 객체들이 검색공간에 균일하게 분포함.

-군집분포(cluster distribution) : 평균 0.5, 분산 0.02를 갖는 정규분포로 이루어진 4개의 군집.

균일분포는 객체의 분포 형태가 이상적인 경우일때, 그리고 군집분포는 균일분포보다 순환적 성질과 국부성 정도가 더 반영되는 경우일때 PR-화일의 특성을 보이기 위한 것이다. 노드페이지와 버켓은 동일한 크기를 가지며 다음과 같이 구성되어 있다고 가정한다.

-노드페이지 및 버켓의 헤더 크기:16바이트.

-MBR 크기:16바이트.

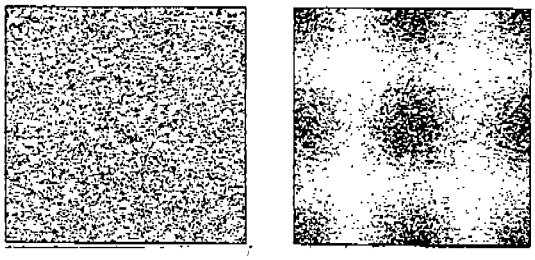
-주소지시자 크기:4바이트.

색인 구축을 위해 노드페이지의 분기율과 버켓용량은 다음과 같이 계산된다.

$$\text{분기율} = \frac{\text{노드페이지 크기} - \text{노드페이지 헤더 크기}}{MBR\text{크기} + \text{주소지시자 크기}}$$

$$\text{버켓용량} = \frac{\text{버켓 크기} - \text{버켓 헤더 크기}}{\text{객체 크기}}$$

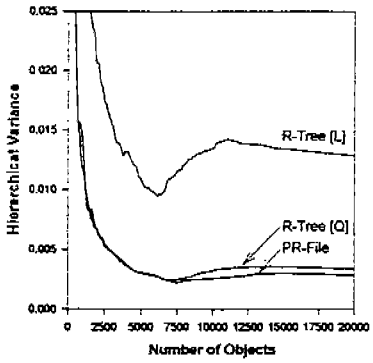
본 논문에서는 이와 같은 가정을 전제로 노드페이지 및 버켓의 크기를 1K바이트부터 시작하여 31K바이트까지 1K바이트씩 증가시키면서 반복적으로 실험하였다.



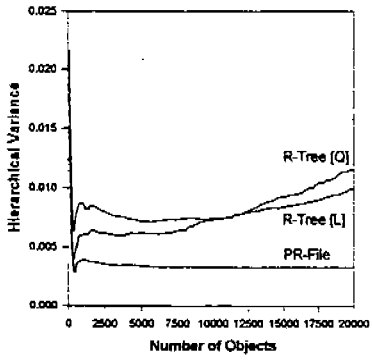
(a)균일분포 (b) 군집분포

(그림 6) 실험에서 사용된 객체 분포
(Fig. 6) Object distributions used in the experiments

실험결과에 의하면 PR-화일은 객체의 분포형태와 버켓의 크기에 관계없이 객체가 삽입될수록 낮은 계층분산값을 유지함을 알 수 있다(그림 7 참조). 상대적으로 R-트리는 높은 계층분산값을 가지며 객체의 분포형태에 영향을 받는다. R-트리가 높은 계층분산값을 갖는 이유는 2.1절에서 설명된 바와 같이 객체의 삽입 및 버켓의 분할방법이 MBR의 면적을 줄이려는 전략을 위주로 하기 때문이다. 이러한 전략은 결과적으로 바늘모양의 MBR을 생성함으로써 좋은 국부성을 보장할 수 없다. 예로 (그림 8)은 버켓의 크기가 1K바이트이고 군집분포를 따르는 1,000개의 객체가 삽입되었을때, PR-화일과 2차 함수비용의 분할 알고리즘을 사용하는 R-트리[Q]의 검색공간을 보여준다.



(a) 균일분포(버켓용량:51)



(b) 군집분포(버켓용량:12)

(그림 7) 계층분산의 변화
(Fig. 7) Variation of hierarchical variance

(a) PR-화일 (b) R-트리[Q]

(그림 8) 검색공간에 생성된 MBR
(Fig. 8) MBR generated on search space

(그림 7)의 (b)에서는 삽입되는 객체의 수가 증가할수록 선형 함수비용의 분할 알고리즘을 사용하는 R-트리[L]가 R-트리[Q]보다 더 좋은 계층분산값을 유지함을 보여준다. 이는 버켓용량이 작고 객체가 검색공간상의 일부분에 집중적으로 발생할 때, R-트리[L]이 R-트리[Q]보다 MBR 면적을 줄이려는 제약조건이 더 약함으로써 바늘모양의 MBR이 적게 발생하기 때문이다. 그러나 버켓용량이 커질수록 이러한 현상은 발생하지 않음을 반복적인 실험을 통하여 알 수 있었다.

실험에서는 다음과 같은 요소를 가지고 100개의 영역질의를 균일하게 발생시켜 계층분산과 적중율의 관계 및 순환 영역질의와 적중율의 관계를 알아 보았다.

-영역질의 크기 : 전체검색공간의 0.01, 0.1, 0.5, 1%

-순환 영역질의 비율 : 0, 5, 10, 30%

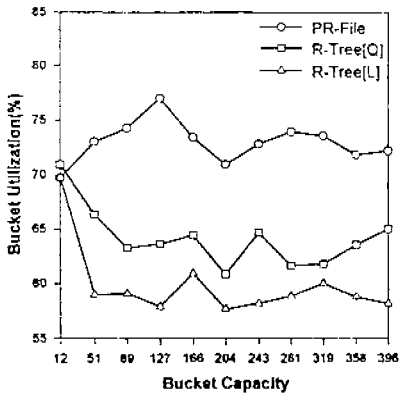
〈표 1〉 계층분산과 적중율
〈Table 1〉 Hierarchical variance and hit ratio

버켓용량 (버켓크기)	요 소		균일분포			군집분포		
			PR-화일	R-트리[Q]	R-트리[L]	PR-화일	R-트리[Q]	R-트리[L]
12 (1KByte)	계층분산		0.0041	0.0062	0.0110	0.0033	0.0116	0.0099
	질의 크기	0.01%	11.9	10.4	8.9	11.6	7.5	8.2
		0.1%	30.3	27.6	26.1	31.6	21.1	23.4
		0.5%	45.7	41.6	39.8	44.5	36.0	38.1
		1.0%	50.6	47.3	46.3	50.3	38.8	42.0
25 (2KByte)	계층분산		0.0024	0.0041	0.0061	0.0027	0.0035	0.0046
	질의 크기	0.01%	5.1	5.7	2.6	5.3	5.0	3.4
		0.1%	18.3	19.0	11.0	18.7	17.2	13.6
		0.5%	33.8	33.4	23.8	31.0	30.4	24.7
		1.0%	39.6	39.7	30.1	38.3	36.7	31.5
51 (4KByte)	계층분산		0.0028	0.0034	0.0129	0.0021	0.0031	0.0100
	질의 크기	0.01%	1.6	1.7	0.5	1.9	1.8	0.6
		0.1%	8.7	8.8	3.2	9.3	8.6	3.6
		0.5%	19.9	19.7	8.6	18.6	17.0	8.6
		1.0%	26.5	25.4	12.9	27.2	23.4	13.1

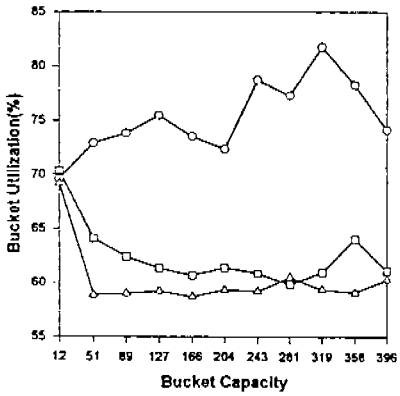
표 1)은 순환 영역질의 비율이 0%이고 버켓의 크기가 1, 2, 4K바이트일 때의 적중율을 보여 준다. PR-화일과 R-트리에서 적중율은 질의의 크기가 커질수록 높아지고, 버켓의 용량이 커

〈표 2〉 순환 영역질의와 적중율(버켓용량:51, 영역질의크기:1.0%)
〈Table 2〉 Circular region query and hit ratio (bucket capacity:51, region query size:1.0%)

객체분포	색인구조	순환 영역질의 비율		
		5%	10%	30%
균일분포	PR-화일	27.0	27.2	27.6
	R-트리[Q]	24.7	25.6	25.5
	R-트리[L]	13.4	12.7	13.5
군집분포	PR-화일	29.5	31.0	34.0
	R-트리[Q]	25.0	25.8	26.7
	R-트리[L]	14.2	13.9	15.9



(a)균일분포



(b) 군집분포

(그림 9) 버켓이용율의 변화
(Fig. 9) Variation of bucket utilization

질수록 낮아지는 성질을 갖는다. 또한 버켓용량의 각 경우에 있어서 낮은 계층분산은 높은 적중율을 산출함을 알 수 있다. R-트리와 비교하여 볼 때 PR-화일은 객체의 분포형태에 관계없이 높은 적중율을 가지며, 특히 균일한 객체의 분포에서 보다는 밀집되어 있는 분포에서 상대적으로 더 높은 적중율을 보였다.

〈표 2〉는 영역질의 크기가 1%이고 버켓 크기가 4K바이트일 때, 순환 영역질의의 비율이 높아짐에 따른 적중율의 변화를 나타낸다. PR-화일에서는 순환 영역질의의 횟수가 많아질수록 적중율이 증가되며, 객체의 분포가 군집분포를 따를 때 증가 정도가 더욱 높아지는 특성을 보인다. 그러나 R-트리의 경우는 PR-화일에 비해 적중율이 좋지 않으며, 또한 적중율의 증가 정도가 일정하지 않음을 알 수 있다.

PR-화일은 객체들의 분포 형태에 관계없이 R-트리보다 높은 버켓이용율을 갖는다(그림 9 참조). 특히 군집분포인 경우에는 객체들의 결집에 순환적 성질이 더욱 많이 반영됨으로써 균일분포보다 버켓이용율이 더 높아진다. 그러나 R-트리는 버켓용량이 증가함에 따라 버켓이용율이 낮아지는 경향을 보이며, 객체들이 군집분포를 형성할 때에는 균일분포보다 낮은 버켓이용율을 갖는다.

5. 결 론

본 논문에서는 공간객체를 취급하는 새로운 동적 색인구조인 PR-화일을 제안하고 그 특성을 실험을 통하여 보였다.

PR-화일은 순환 도메인과 선형 도메인으로 구성된 다차원 검색공간을 전제로 하고 있으며, 이러한 검색공간상에서 순환 영역질의 및 선형 영역질의를 취급한다.

공간적으로 인접한 객체들이 동일한 버켓에 결집되면 공간국부성이 좋아짐으로 검색의 효율을 향상시킬 수 있다. 본 논문에서는 계층별 공간국부성을 측정하기 위한 방법으로 변형된 계층분산을 제안하고 그 특성을 정리하였다. 변형된 계층분산은 임의 검색공간에 존재하는 모든 객체의 국부성과 부검색공간들의 평균 국부성을 고려한 계층별 공간국부성 측도이다. PR-화일의 연산 알고리즘에서 객체의 삽입과 버켓의 분할은 변형된 계층분산을 이용한다.

실험에서는 PR-화일의 성능을 계층분산, 적중율 그리고 버켓이용율에 따라 평가하였다. 실험 데이타로는 균일분포와 군집분포를 이루는 객체들을 이용하였으며, 비교 대상으로는 R-트리를 선택하였다.

반복된 실험결과에 의하면, PR-화일은 객체의 분포형태와 버켓의 크기에 관계없이 낮은 계층분산값을 유지함으로써 높은 적중율을 산출하였다. 또한 순환 영역질의의 횟수가 많아질수록 적중율이 증가되었다. PR-화일은 R-트리보다 높은 버켓이용율을 가지며, 특히 군집분포인 경우에는 객체들의 결집에 순환적 성질이 더욱 많이 반영됨으로써 균일분포보다 버켓이용율이 더 높음을 보였다.

순환도메인이 포함된 다차원 검색공간이 존재한다면, 순환적인 환경을 고려하여 낮은 계층분산을 갖도록 객체를 결집시킴으로써 공간국부성 및 버켓이용율을 높이고 적중율을 향상시킬 수 있다.

참 고 문 헌

[1] 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 Conf., pp.322-331, May. 1990.

[2] M.Freeston, "The BANG File : a New Kind of Grid File," In Proc. Intl. Conf. on Management of Data, SIGMOD, San Francisco, California, pp. 260-269, 1987.

[3] A.Guttman, "R-tree : a Dynamic Index Structure for Spatial Searching," Proc. ACM SIGMOD Conf., pp.599-608, June. 1984.

[4] A.Henrich, H.W.Six and P.Widmayer, "The LSD-tree : Spatial Access to Multi-dimensional Point and NonPoint Objects," Proc. of 5th VLDB Conf., pp.

45-53, 1989.

[5] 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.

[6] L.K.Joune and L.Robert, "The Spatial Locality and a Spatial Indexing Method by Dynamic Clustering in Hypermap System," 2nd Sym. on Large Spatial Databases (SSD), pp.207-223, 1991.

[7] D.B.Lomet and B.Salzberg, "The hB-tree : a Multiattribute Indexing Method with Good Guaranteed Performance," ACM Trans. on Database Systems, Vol.15, No. 4, pp.625-658, Dec. 1990.

[8] J.Nievergelt and H.Hinterberger, "The Grid File : an Adaptable, Symmetric Multikey File Structure," ACM Trans. on Database Systems, Vol.9, No.1, pp. 38-71, Mar. 1984.

[9] M.A.Ouksel and O.Mayer, "A Robust and Efficient Spatial Data Structure," Acta Informatica 29, pp.335-373, 1992.

[10] H.G.Ralf, "An Introduction to Spatial Database Systems," VLDB Journal, No. 3, pp.357-399, Aug. 1994.

[11] J.T.Robinson, "The K-D-B Tree : a Search Structurefor Large Multi-dimensional Dynamic Indexes," Proc. ACM SIGMOD Conf., pp.10-18, 1981.

[12] N.Roussopoulos and D.Leifker, "Direct Spatial Search on Pictorial Databases Using Packed R-trees," Proc. ACM SIGMOD Conf., pp.17-31, 1985.

[13] B.Seeger, "Performance Comparison of Segment Access Methods Implemented on Top of the Buddy-tree," 2nd Sym. on Large Spatial Databases(SSD), pp. 276-296, 1991.

[14] B.Seeger and H.P.Kriegel, "Techniques for Design and Implementation of Efficient Spatial Access Methods," In Proc. 14th Intl. Conf. on VLDB, pp.10-17, 1988.

[15] T.Sellis, N.Roussopoulos and C.Faloutsos, "The R⁺-tree : a Dynamic Index for Multidimensional Objects," Proc. 13th VLDB Conf., pp.507-518, Sept. 1987.

김 홍 기

1984년 전남대학교 계산통계학과(이학사)
1986년 전남대학교 대학원 계산통계학과(이학석사)
1990년~현재 전남대학교 대학원 전산통계학과 박사과정
1991년~현재 동신대학교 전산통계학과 조교수

관심분야 : 공간데이타구조, 공간데이타베이스, 컴퓨터그래픽스

황 부 현

1978년 숭실대학교 전산학과(학사)
1980년 한국과학기술원 전산학과(공학석사)
1994년 한국과학기술원 전산학과(공학박사)
1980년~현재 전남대학교 전산학과 교수

관심분야 : 분산 시스템, 분산 데이타베이스 보안, 객체지향 시스템