Journal of Korea Multimedia Society Vol. 9, No. 11, November 2006(pp. 1421-1434)

공간 데이터 웨어하우스에서 공간 데이터의 개념계층기반 사전집계 색인 기법

전병윤[†], 이동욱[††], 유병섭[†††], 김경배[††††], 배해영[†††††]

요 약

공간 데이터 웨어하우스는 SOLAP(Spatial On-Line Analytical Processing)을 이용하여 의사 결정에 필요한 분석 정보를 제공한다. SOLAP은 대용량 데이터를 분석하기 때문에 사전집계를 이용하여 분석비용을 줄이기 위한 많은 연구가 진행되었다. 기존 기법들은 고정크기노드를 갖는 색인을 이용하여 개념계층을 지원하였다. 따라서 산개분포 영역에는 빈 공간이 많이 발생하며, 밀집분포 영역에는 개념계층을 지원할 수 없다. 본 논문은 공간 데이터의 개념계층기반으로 사전집계 색인의 동적 구성 기법을 제안한다. 제안 기법은 트리구조를 이용하여 개념계층의 레벨을 트리의 레벨과 같도록 지원한다. 하나의 노드는 데이터가 적을 경우 엔트리를 분할하여 서로 다른 부모 엔트리를 가질 수 있으며, 데이터가 많을 경우 노드의 연결리스트를 이용하여 같은 레벨에 순차적으로 저장한다. 따라서 데이터가 산개된 분포의 노드에 대해서 저장 공간의 낭비를 최소화하며, 데이터가 밀집한 영역의 노드에 대해서도 노드의 연결리스트로 노드가 분할되지 않으므로 개념계층을 지원할 수 있다. 성능평가를 통하여 색인 구축 시간이 다른 기법과 비슷하고, 색인의 저장 공간이 감소하며, 집계정보의 검색 성능이 다른 기법에 비해 우수한 것을 보인다.

Pre-aggregation Index Method Based on the Spatial Hierarchy in the Spatial Data Warehouse

Byung-Yun Jeon[†], Dong-Wook Lee[††], Byeong-Seob You[†††], Gyoung-Bae Kim[††††], Hae-Young Bae[†††††]

ABSTRACT

Spatial data warehouses provide analytical information for decision supports using SOLAP (Spatial On-Line Analytical Processing) operations. Many researches have been studied to reduce analysis cost of SOLAP operations using pre-aggregation methods. These methods use the index composed of fixed size nodes for supporting the concept hierarchy. Therefore, these methods have many unused entries in sparse data area. Also, it is impossible to support the concept hierarchy in dense data area. In this paper, we propose a dynamic pre-aggregation index method based on the spatial hierarchy. The proposed method uses the level of the index for supporting the concept hierarchy. In sparse data area, if sibling nodes have a few used entries, those entries are integrated in a node and the parent entries share the node. In dense data area, if a node has many objects, the node is connected with linked list of several nodes and data is stored in linked nodes. Therefore, the proposed method saves the space of unused entries by integrating nodes. Moreover it can support the concept hierarchy because a node is not divided by linked nodes. Experimental result shows that the proposed method saves both space and aggregation search cost with the similar building cost of other methods.

Key words: Spatial Hierarchy(공간 개념계층), Pre-aggregation(사전집계), SOLAP(Spatial OLAP), Spatial Data Warehouse(공간 데이터 웨어하우스)

※ 교신저자(Corresponding Author) : 배해영, 주소 : 인천광역시 남구 용현동 253(402-751), 전화 : 032)860-8712, FAX : 032)862-9845, E-mail : hybac@inha.ac.kr
접수일 : 2006년 8월 30일, 완료일 : 2006년 10월 20일
[†] 준회원, 인하대학교 컴퓨터정보공학과 석사과정
(E-mail: mysummit@dblab.inha.ac.kr)
[††] 인하대학교 컴퓨터정보공학과 박사과정
(E-mail: dwlee@dblab.inha.ac.kr)
[†††] 인하대학교 대학원 컴퓨터정보공학과 박사과정
(E-mail: subi@dblab.inha.ac.kr)
[††††] 정회원, 서원대학교 컴퓨터교육과 조교수
(E-mail: gbkim@seowon.ac.kr)
[†††††] 정회원, 인하대학교 컴퓨터공학부 교수
※ 본 연구는 대학 IT연구센터 육성 · 지원사업의 연구결과로 수행되었음

1. 서 론

공간 데이터 웨어하우스는 운영계 데이터(operational data)로부터 추출한 공간 및 비공간 데이터를 통합하고, 통합된 데이터를 SOLAP(Spatial On-Line Analytical Processing)을 이용하여 의사 결정을 지원하는 시스템이다[1-3]. SOLAP은 대용량의 데이터를 분석하므로 의사 결정에 필요한 요약 정보를 생성하는 데에 많은 시간이 요구된다. 따라서 질의 처리 성능을 향상시키기 위하여 사전집계(pre-aggregation)기법을 이용한 많은 연구가 진행되었다[4,5].

공간 데이터의 개념계층을 기준으로 사전집계 색인을 구성하는 기법으로는 aR-tree기법과 OLAP-Favored Search 기법이 연구되었다[6,7]. 이 기법들은 R-tree의 부모 노드가 자식 노드의 영역을 포함하는 특성을 이용해 개념계층(concept hierarchy)을 지원하고, 사전 집계 정보를 관리하기 위하여 노드의 엔트리를 확장하거나 별도의 저장 공간을 할당하여 요약테이블을 생성하는 기법을 사용한다[8]. 하지만 R-tree는 노드의 크기가 정적이므로 공간 데이터의 분포가 불균등하면 노드의 팬아웃(fan-out)보다 적은 수의 공간 데이터를 가지는 노드는 노드의 빈 저장 공간이 발생하며, 노드의 팬아웃보다 많은 수의 공간 데이터를 가지는 노드는 하나의 노드에 모든 데이터를 저장하지 못하여 노드 분할이 발생하게 된다.

본 논문에서는 공간 데이터의 개념계층을 기반으로 사전집계 정보를 관리하는 동적 색인기법을 제안한다. 제안 기법의 색인의 구축 방법은 동적 밀도 색인 기법을 이용하여 개념계층에 포함된 자식 노드의 공간 데이터를 색인에 순차적으로 삽입하고 부모 노드에서는 자식노드의 군집된 데이터의 시작위치를 관리한다[9]. 개념계층이 여러 단계로 구성되어있으므로 동적 밀도 색인 기법을 여러 레벨로 구성하여 개념계층의 구조를 지원하며, 개념계층의 하나의 계층을 색인의 하나의 레벨로 구성한다. 하나의 레벨의 데이터들은 여러 개의 노드로 구성되어 있으므로 노드들을 연결리스트를 이용하여 순차적으로 데이터를 관리하며, 하나의 노드를 이용하여 모든 데이터를 관리하지 못하는 자식 노드는 여러 개의 노드로서 데이터를 관리하고 노드를 연결리스트를 이용하여 연결한다. 또한 부모 노드의 포인터는 노드의 중간에 있는 엔트리를 관리하여 노드에 낭비되는 공간이 발생하지 않는다. 따라서 제안 기법은 공간 데이터의 분포가 불균등 하더라도 동적으로 색인을 구성하여 저장 공간의 낭비가 발생하지 않으며, 효율적으로 공간 데이터의 개념계층을 지원한다. 또한 제안 기법에서는 집계 정보의 검색을 개념계층기반으로 하여 검색을 지원하며, 영역 질의(window query)에 대해서는 기존의 기법과 동일한 연산 방법을 이용하여 집계 정보를 검색한다[6,7].

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 공간 데이터 웨어하우스의 개념계층과 R-tree 기반의 사전집계 기법에 대해서 설명하고, 3장에서는 공간 데이터 웨어하우스의 개념계층을 기반으로 사전집계 수행시 고려해야 할 사항에 대해서 기술한다. 4장에서는 본 논문의 제안기법으로 공간 데이터의 개념계층을 기반으로 사전집계 색인 기법을 구축하는 방법에 대해서 설명하고, 5장에서는 제안 기법과 기존 기법을 색인 구축 비용과 검색 연산 성능을 비교한다. 마지막 6장에서는 결론 및 향후 연구를 기술한다.

2. 관련연구

본 장에서는 공간 데이터 웨어하우스에서 공간 데이터의 개념계층을 설명하고, 개념계층기반의 SOLAP 연산에 대해 설명한다. 또한 공간 데이터의 사전집계를 수행하는 기존 기법인 aR-tree 기법과 OLAP-Favored Search 기법을 설명한다.

2.1 공간 데이터 웨어하우스에서 개념계층

공간 데이터 웨어하우스에서는 공간 데이터를 가지고 있는 차원테이블(dimension table)에 개념계층을 정의하여 데이터의 집계 기준을 계층으로 정의한다. 공간 데이터의 개념계층은 영역으로 구성되며, 각 영역들은 특정 그룹으로 계층을 형성하고 있기 때문에 상위 계층을 정의한 영역은 몇 개의 하위 계층 영역을 포함한다[10]. [그림 1]은 공간 데이터의 개념계층을 표현한 것이다.

[그림 1]에서 영역 R1 영역 R2, R3, R4를 포함하는 것과 같이 공간 데이터의 개념계층은 상위 계층에 포함된 하위 계층에 대한 영역을 포함한다. 공간 데이터의 개념계층의 영역에는 분석해야하는 많은 수

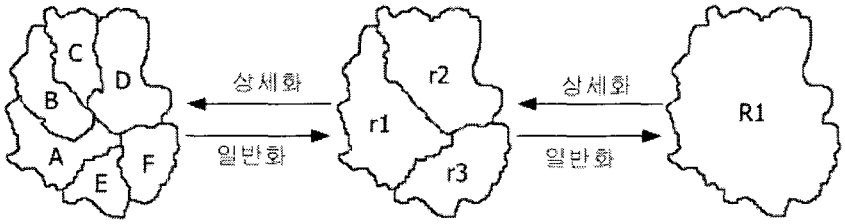


그림 1. 공간 데이터 웨어하우스의 공간 데이터의 개념계층

의 공간 객체가 분포되어 있다. 따라서 개념계층을 정의한 많은 영역들은 자신이 포함된 공간객체의 분석정보를 포함해야 한다.

SOLAP에는 개념계층을 정의하여 데이터들의 계층에 따라 일반화된 분석을 지원하거나 세분화된 연산을 지원한다[1,11,12]. 일반화(roll-up) 연산은 개념계층의 단계를 높여가면서 일반화된 집계정보를 제공하는 연산이고, 상세화(drill-down) 연산은 개념계층의 단계를 낮춰 세분화된 집계정보를 제공한다.

2.2 aR-tree 기법

aR-tree 기법은 R-tree를 구성하는 노드의 엔트리에 집계값 속성을 포함하도록 확장하였다[7]. 이 기법은 R-tree의 엔트리의 구조를 확장함으로써 R-tree의 계층적인 구조를 이용하여 공간 데이터의 개념계층을 관리하고, 확장된 엔트리를 이용하여 집계정보를 관리할 수 있으므로 공간 데이터의 개념계층을 기반으로 집계정보를 관리한다. [그림 2]는 aR-tree 기법의 전체적인 구조를 나타낸다.

aR-tree 기법은 각 노드에 대한 현재 집계정보를 단말 노드에서부터 집계를 수행하여 상위 노드로 거슬러 올라가는 상향식(bottom-up) 방법으로 사전집계를 수행한다. 예를 들어 [그림 2]에서 루트 노드는 노드 A와 노드 B를 포함하고 있으며, 노드 A에 대한 집계정보는 하위 노드의 a1, a4, a3의 집계값인 2, 2, 3을 합한 7을 집계정보로 갖게 된다.

aR-tree 기법은 공간 객체 정보에 대해서는 색인에서 관리하지 않으며 공간 데이터의 개념계층에 대한 집계정보만을 관리한다. 따라서 공간 객체가 집중적으로 분포하는 경우에도 개념계층에 정보를 관리할 수 있으며, 공간 데이터의 개념계층을 기준으로 빠르게 집계정보를 제공한다. 또한 하나의 단계에는 데이터를 관리하는 노드들을 연결하는 리스트를 관리하여 동일한 단계의 집계정보를 요구하는 검색이 요청되면 빠르게 집계정보를 제공한다.

또한 aR-tree 기법의 사전집계를 이용하여 영역질의(window query)에 대한 검색이 요청되면 영역에 포함된 공간 객체에 대한 집계정보를 제공한다. 검색 방법은 질의 범위에 완전히 포함되는 엔트리는 현재 엔트리의 집계정보를 포함시키고 자식 노드의 탐색을 수행하지 않으므로 색인의 탐색비용이 감소하며, 부분적으로 겹치는 엔트리에 대해서는 자식 노드를 탐색하여 질의 영역에 포함되는 엔트리를 재귀적으로 탐색하여 질의 영역에 포함된 영역을 찾아내어 집계정보에 반영한다. 하지만 단말 노드에 도달할 때까지 집계정보를 탐색하지 못하는 경우에는 실제 공간 데이터의 집계정보를 분석해야 하기 때문에 공간 데이터 웨어하우스의 저장소에 있는 레코드에 접근하여 결과를 얻거나 통계적 예측 방법을 이용하여 근사값을 얻는다[13,14]. 그러나 공간 데이터 웨어하

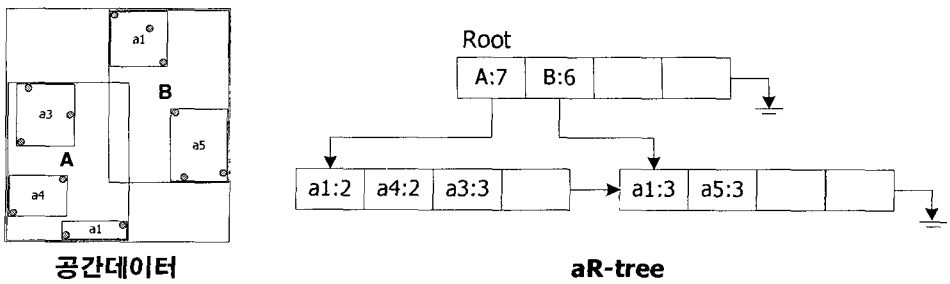


그림 2. aR-tree 기법의 구조

우스의 저장소에 있는 레코드를 검색하여 집계 정보를 분석하는 방법은 많은 검색 비용이 필요한 문제점을 가지고 있으며, 통계적 예측 방법을 수행하는 경우에는 공간 데이터의 개념계층을 기준으로 사전집계 색인을 구축하게 되면 공간 데이터가 노드에 불균등하게 분포하므로 통계적 예측을 이용하여 정확한 예측을 수행할 수 없는 문제점이 있다.

2.3 OLAP-Favored Search 기법

OLAP-Favored Search 기법은 OLAP 연산의 질의 처리 성능 향상을 위하여 공간 데이터의 개념계층을 지원하기 위하여 R-tree를 사용하며, 요약 테이블을 생성하여 집계정보를 관리한다. [그림 3]은 OLAP-Favored Search 기법을 구성하는 R-tree와 요약테이블의 구조를 나타낸다.

[그림 3]에서와 같이 R-tree는 공간 데이터의 개념계층을 관리하는 색인을 생성하며, 요약 테이블에는 집계정보가 R-tree를 후위순회 탐색(postorder traversal)한 순서로 저장된다. 요약 테이블에 저장된 데이터는 색인의 노드의 아이디인 NID(node identification)를 기준으로 관리된다.

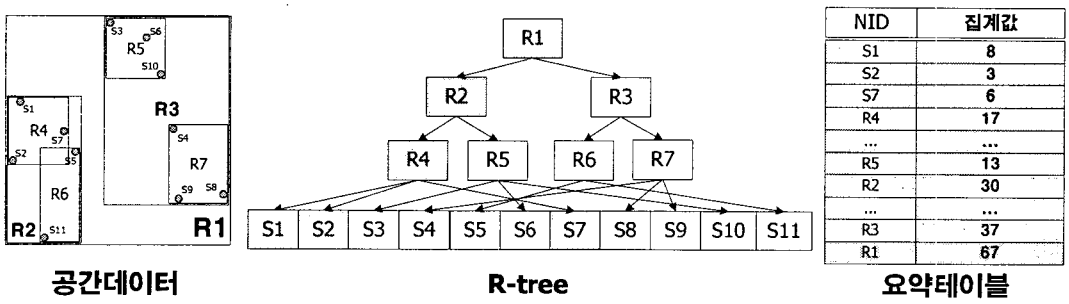
이 기법은 공간 데이터의 개념계층을 기반으로 SOLAP 연산을 수행하면 처음 단계에는 트리의 탐색을 수행하여 NID를 얻고, 그 NID에 대한 레코드를 검색하여 집계정보를 얻는다. 영역 질의인 경우에는 aR-tree 기법의 영역질의와 마찬가지 방법을 이용하여 트리 탐색을 통하여 질의 영역에 포함된 엔트리는 요약테이블에서 집계정보를 가져와 총 집계정보에 포함하고, 부분적으로 겹치는 경우에는 하위 노드로 이동하여 질의 영역을 포함하는 엔트리를 재귀적으로 검색한다. 이 기법은 모든 공간 데이터를 집계 색인에서 관리하므로 단말 노드에 이르게 되면 공간 데이터의 집계정보를 얻을 수 있다.

하지만 OLAP-Favored Search 기법은 공간 데이터의 개념계층을 지원하는 색인인 R-tree와 집계정보를 관리하는 요약 테이블로 분리하였기 때문에 집계정보를 검색하는 추가 비용이 필요하다. 또한 요약 테이블의 구조가 개념계층을 기반으로 구축되어 있지 않으므로 색인을 이용하여 개념계층을 기반으로 SOLAP이 요청되면 트리를 탐색한 뒤 요약테이블을 검색하므로 높은 질의 비용이 필요한 문제점이 있다.

또한 R-tree의 노드의 구조는 정적이기 때문에 공간 데이터의 분포가 불균등하면 색인의 구조가 데이터를 효과적으로 관리할 수 없는 문제점을 가진다. 개념계층 영역에서는 노드의 팬아웃(fan-out)보다 적은 개수의 하위 영역을 관리하므로 대부분의 노드에서 저장 공간이 낭비되는 현상이 발생되는 문제점을 가진다. 또한 한 영역에 공간 데이터가 집중적으로 분포하여 개념계층 영역을 노드의 팬아웃(fan-out)보다 많은 수를 관리하는 개념계층 영역은 하나의 노드에서 개념계층을 지원할 수 없으므로 여러 노드를 사용하여 관리하고 그 노드들을 관리하기 위한 부모 노드를 생성하여야 하기 때문에 개념계층의 단계와 색인의 단계가 불일치하여 효과적으로 개념계층을 색인을 이용하여 관리할 수 없는 문제점을 가진다.

3. 공간 데이터의 개념계층기반 사전집계 수행시 고려사항

기존 기법들은 공간 데이터의 색인으로 많이 사용되는 R-tree를 확장하여 공간 데이터를 계층화하여 집계를 수행하였다[8,9]. 하지만 R-tree는 정적인 크기의 노드를 사용하기 때문에 공간 데이터의 개념계



NID	집계값
S1	8
S2	3
S7	6
R4	17
...	...
R5	13
R2	30
...	...
R3	37
R1	67

그림 3. OLAP-Favored Search Method의 R-tree와 요약테이블 구조

층을 기반으로 사전집계를 수행하면 문제가 발생한다. 따라서 본 장에서는 공간 데이터의 개념계층을 기준으로 사전집계를 수행할 때 고려해야 할 사항이 무엇인지 알아보기 위하여 공간 데이터의 개념계층을 R-tree에 적용하여 문제점을 설명한다.

공간 데이터의 개념계층과 R-tree는 여러 레벨로 구성되는 것과 부모 노드는 자식 노드의 공간 데이터의 영역을 포함하는 공통적인 특징을 가진다. 하지만 몇 가지 차이점을 가지고 있기 때문에 공간 데이터의 개념계층을 R-tree 기법에 적용하게 되면 문제점이 발생한다.

첫째, R-tree는 공간 데이터의 갱신(삽입, 삭제, 변경)이 발생했을 때 분할(split) 또는 병합(merge)연산을 통하여 색인의 전체의 구조를 동적으로 변경하여 공간 데이터의 갱신이 빈번하여도 트리의 균형을 유지할 수 있다. 이것은 색인의 구조를 최소한으로 변화시키고 빠르게 공간 객체의 검색한다[8,15]. 따라서 R-tree는 예측할 수 없는 데이터가 갱신이 발생하여도 효과적으로 데이터를 관리할 수 있는 구조를 가지고 있다.

그러나 일단 공간 데이터 웨어하우스에 적재된 데이터는 갱신되지 않으며, 색인의 구조를 결정하는 공간 데이터와 개념계층에 대한 정보도 갱신이 일어나지 않는다. 따라서 색인의 구조와 데이터가 갱신이 일어나지 않으므로 개념계층기반의 사전집계 색인은 저장 공간의 효율을 높일 수 있는 구조를 가져야 한다.

둘째, R-tree는 데이터의 계층 관계를 형성하는 이유는 상위 노드의 하나의 엔트리가 하나의 노드를 관리하기 때문이다. 따라서 R-tree가 공간 데이터의 개념계층을 지원하기 위해서는 상위 계층에 포함된 하위 계층 영역들이 하나의 노드에서 관리되어야 한다. 하지만 R-tree는 크기가 고정된 노드를 사용하기 때문에 공간 데이터의 분포가 불균등하게 되면 개념계층을 지원할 수 없다. [그림 4]는 개념계층을 기반으로 공간 데이터를 분류한 것이다.

[그림 4](a)는 차원 테이블이 가지고 있는 공간 데이터를 공간 개념계층의 최하위 계층의 영역을 기준으로 표현한 것이고, [그림 4](b)에서는 공간 차원의 개념계층에 대한 정의를 트리구조를 이용하여 표현한 것이다. 공간 데이터의 개념계층의 영역은 6개로 구성되어 있으며, 공간 데이터를 영역을 기준으로 분류하면 분포가 일정하지 않다. 예를 들어 영역 E에는 1개의 공간 객체가 포함되어 있지만 영역 D에는 6개의 공간 객체가 분포한다.

이런 공간 데이터의 개념계층을 4개의 엔트리를 관리하는 노드로 R-tree를 구축한다고 하면, 공간 데이터의 개념계층에 대한 위상적인 관계를 지원하기 위해서는 반드시 하나의 노드를 이용하여 한 계층에 포함된 정보들을 관리하여야 한다. 하지만 영역 E를 관리하는 노드에는 1개의 엔트리만이 사용되어 75%의 공간 낭비가 발생하게 되며, 영역 E에 대한 노드에는 6개의 엔트리가 생성되어야 하므로 모든 데이터에 대한 집계정보를 하나의 노드에서 관리할 수 없다. 그러므로 공간 데이터의 분포가 불균등한 분포이면 R-tree를 이용하여 논리적으로 공간 개념계층을 표현하지 못한다.

따라서 공간 데이터의 개념계층을 기반으로 사전집계 기법은 엔트리의 개수를 동적으로 구축이 가능해야 한다. 이런 구축 기법은 공간 데이터의 분포가 불균등하더라도 개념계층을 기준으로 집계정보를 구축할 때 저장 공간의 낭비 없이 개념계층의 구조를 색인에 반영할 수 있다.

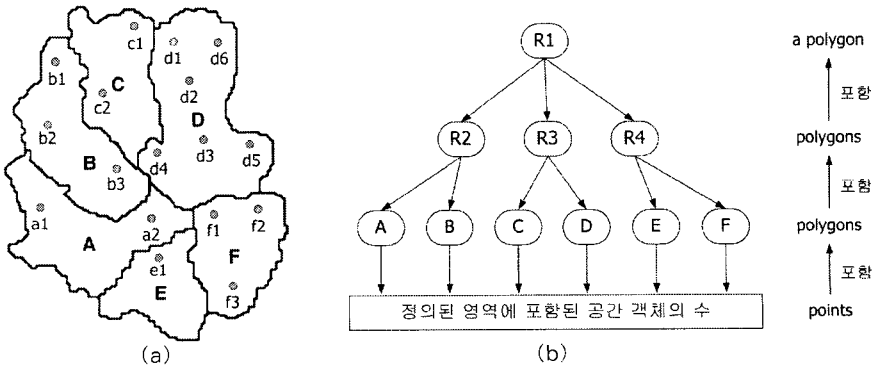


그림 4. 공간 데이터의 개념계층을 기준으로 분류한 공간 객체의 예: (a) 공간 데이터, (b) 공간 데이터의 개념계층

4. 공간 개념계층기반의 사전집계 색인 기법

본 장에서는 공간 데이터 웨어하우스에서 공간 데이터의 개념계층을 기반으로 SOLAP 질의 처리 향상을 위한 사전집계 색인을 제안한다. 제안 기법은 공간 데이터 웨어하우스 시스템의 저장소에 적재된 데이터로부터 SOLAP 연산의 집계정보를 생성하는 것이 많은 비용이 요구되므로 사전집계를 수행하여 질의 처리 성능을 향상시킨다. 본 기법은 밀도 색인 기법은 동적으로 엔트리를 생성하여 하위 노드를 관리하므로 색인의 노드의 빈 공간이 발생하지 않으므로 효율적으로 저장 공간을 사용한다. 또한 여러 단계로 구축되어 있는 공간 데이터의 개념계층을 밀도 색인 기법을 여러 단계로 구축하여 사전집계 정보를 관리하므로 공간 데이터의 분포가 불균등하더라도 저장 공간의 낭비가 최소화되면서 논리적으로 개념계층의 구조를 만족시킨다.

다음에서는 제안 기법의 색인 구조에 대해서 설명하고, 색인 구축 방법에 대해서 기술한다. 또한 구축된 색인의 검색연산에 대해서 기술한다. 검색 연산은 공간 데이터의 개념계층을 기준으로 검색이 요청되었을 경우와 영역 질의(window query)에 대한 검색이 요청되었을 때 영역 질의에 포함되는 공간 객체의 집계정보를 검색하는 기법에 대해서 나누어 설명하도록 한다.

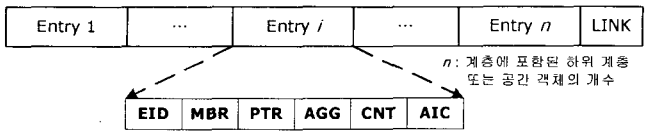
4.1 사전집계 인덱스 구조

제안 기법의 색인 구축은 공간 데이터의 개념계층의 구조와 각 영역에 포함되는 객체의 수에 따라 동적으로 구축되며 개념계층의 구조가 갱신되지 않으므로 색인이 구축되면 갱신이 일어나지 않는다. 제안 색인기법의 기본 단위는 블록과 엔트리로 구성된다. 블록은 데이터를 접근하는 기본단위로서 여러 개의 엔트리와 블록에 대한 연결리스트로 구성되어 있다. 블록의 연결리스트는 동일한 계층에 대한 다음 블록에 위치 정보를 가지고 있으므로 동일한 계층에 대한 집계정보들이 순차적으로 구성된다. 따라서 동일한 계층에 대한 모든 집계정보를 검색하는 연산이 요구되면 순차적으로 모든 집계정보를 얻을 수 있으므로 빠른 연산을 수행한다. 엔트리의 구조는 R-tree의 엔트리를 확장하여 사전집계 정보를 포함하도록 하였다. [그림 5]는 제안 색인 기법의 블록과 엔트리의 구조를 나타낸다.

제안 기법의 엔트리는 기존의 R-tree의 엔트리와 동일한 EID(entry identification), MBR(Minimum Bounding Rectangle), PTR(pointer) 속성과 SOLAP을 지원하기 위하여 AGG, CNT(count), AIC(Area Information Cartridge) 속성으로 구성된다.

R-tree에서 이미 존재하는 속성인 EID는 엔트리를 식별하기 위해서 사용하고 공간 객체를 구분하며, MBR은 하위 노드에 포함된 모든 공간 객체를 영역을 표현한다. PTR은 현재 엔트리가 가지고 있는 하위 계층에 대한 엔트리의 시작포인터이다. 제안 기법의 PTR은 시작 위치의 단위가 엔트리이므로 블록의 번호와 해당 엔트리의 오프셋(offset)으로 구성된다.

사전집계 정보를 관리하기 위해서 추가된 속성은 공간 데이터의 집계정보를 관리하는 AGG와 동적으로 노드를 관리하기 위하여 하위 객체의 수에 대한



EID	엔트리 식별자, 즉, 공간 객체를 구분하기 위한 식별자	기존 요소
MBR	공간 객체를 포함하는 최소 경계 영역	
PTR	자신의 하위 계층의 시작 위치에 대한 포인터	
AGG	현재 영역에 대한 집계 정보 (SUM, MAX, MIN, COUNT, AVG)	추가 요소
CNT	현재 영역에 포함된 하위 단계의 공간 객체의 수	
CAI	현재 영역에 대한 정보 (Current Area Information)	

그림 5. 공간 개념계층 단위의 집계 정보를 관리하는 엔트리 구조

정보를 가지는 CNT 속성을 추가하였다. 또한 SOLAP을 수행하는 경우 집계정보와 공간 객체에 해당하는 설명 정보를 빠르게 얻어오기 위한 AIC를 포함하고 있다. 만약 설명 정보를 색인에서 포함하지 않으면 실제 레코드의 검색에 대한 비용이 필요하기 때문에 이름과 같은 설명이나 레코드의 포인터를 관리함으로써 질의 성능을 향상시킨다.

제안하는 색인 기법의 구조는 엔트리를 기본단위로 동적으로 색인을 구축하는 밀도 색인 기법을 이용한다. 개념계층의 한 단계는 밀도 색인의 기법의 한 단계를 이용하여 순차적으로 집계정보를 관리하며, 개념계층이 여러 단계로 구성되어 있으므로 밀도 색인 기법을 여러 단계로 구축하여 집계정보를 표현한다. 이 기법은 노드를 기본 단위로 하는 기존 색인 기법과는 다르게 엔트리를 기본 단위로 하여 색인을 구축하고 이 엔트리는 블록을 이용하여 관리한다. [그림 6]은 공간 데이터의 개념계층을 지원하는 사전집계 색인의 구조를 [그림 4]의 공간 데이터의 개념계층과 공간 객체를 기반으로 제안 기법의 색인을 구성하였다. 제안 기법은 [그림 6]과 같이 한 영역에 대한 하위 레벨의 데이터를 엔트리 단위를 동적으로 관리함으로써 저장 공간의 낭비가 발생하지 않으며, 노드의 크기에 제한을 받지 않는다. 만약 블록의 개수를 초과하는 영역이 발생하면 새로운 블록을 추가하고 기존의 블록의 연결리스트를 이용하여 다음 블록의 위치 정보를 관리한다. 또한 하위 영역에 존재하는 영역의 시작 위치만을 상위 엔트리에 관리함으로써 동적으로 노드를 구성할 수 있다. 예를 들어 영역 D는 6개의 하위 영역으로 구성되어 있기 때문에 하나의 블록에서 관리가 불가능하지만 엔트리를 여러 블록에 걸쳐서 관리하고, 인접한 블록들을 연결리스트를 이용하여 연결함으로써 동적으로 노드를 생성할 수 있다.

하나의 단계에서 모든 영역에 대한 정보가 삽입되면 추가적으로 TERM(terminal) 엔트리를 삽입한다. TERM 엔트리는 개념계층의 단계의 데이터의 마지막을 인지하도록 하는 기능을 수행하여 집계정보 검색에 대한 오류와 연산 비용을 줄이는 역할을 수행한다.

4.2 사전집계 색인 구축 방법

사전집계 색인은 개념계층을 기반으로 집계정보를 관리하기 위하여 개념계층과 차원 테이블의 공간 객체를 이용하여 색인을 구축하고, 사실 테이블의 집계정보를 이용하여 색인의 집계정보를 구축한다. 사전집계 색인의 구조에 영향을 주는 개념계층과 차원 테이블은 갱신이 발생하지 않으므로 색인의 구조가 변경되지 않는다. 따라서 사전집계 색인은 색인의 관리의 효율성을 고려하기보다 색인의 저장 공간의 효율적으로 사용하도록 색인을 구축한다. 사전집계 색인을 구축 단계는 다음과 같이 세 단계로 구성된다.

첫 번째 단계는 공간 차원의 개념계층에 대한 정보를 이용하여 [그림 6]의 1번째 블록부터 4번째 블록과 같이 비단말노드를 구성한다. 공간 데이터의 개념계층은 트리 구조를 가지므로 넓이 우선 탐색(Breadth First Search) 기법의 순서로 탐색하여 색인을 구성한다. 하나의 영역에 포함된 하위 영역의 수가 블록에서 관리할 수 있는 엔트리의 수보다 많으면 여러 개의 블록을 연결리스트로 연결하여 순차적으로 관리하고, 하나의 영역에 포함된 하위 영역의 수가 블록의 엔트리 수보다 적은 경우에는 블록을 여러 부모 엔트리의 자식 엔트리가 공유하여 사용함으로써 빈 엔트리가 발생하지 않도록 한다. 또한 하나의 계층에 모든 엔트리가 삽입되면 TERM 엔트리를 삽입하여 검색 연산에 대한 비용을 절약한다.

다음 단계는 공간 데이터의 집계정보를 관리하기 위하여 공간 데이터의 엔트리를 단말노드에 추가하

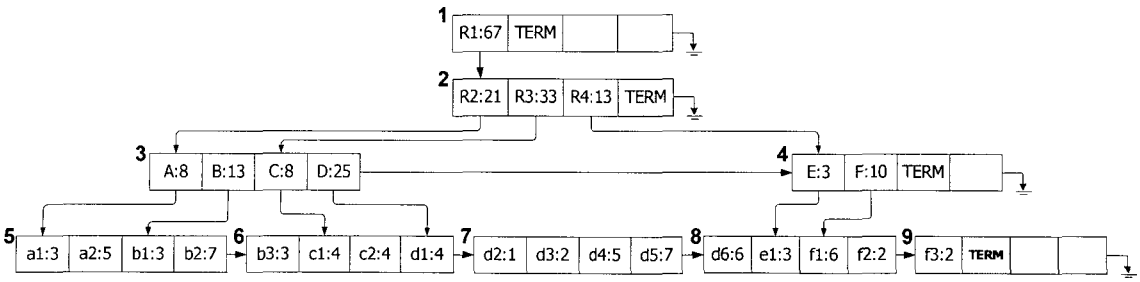


그림 6. 공간 데이터의 개념계층을 지원하는 사전집계 색인 구조

는 단계이다. 엔트리를 추가하기 위해서는 비단말노드에 구축되어 있는 개념계층의 엔트리와 위상관계를 만족하여야 한다. 따라서 개념계층의 마지막 단계에 대한 엔트리를 검색하여 위상관계를 만족하도록 공간 객체를 삽입한다. 또한 데이터를 삽입하는 경우에는 첫 번째 단계에서와 같이 데이터의 분포에 따라 여러 부모 엔트리에 대한 자식 엔트리들이 블록을 공유하거나 하나의 부모 엔트리가 여러 블록에 걸쳐 하위 영역의 엔트리를 삽입한다. [그림 7]은 동적으로 엔트리를 구성하는 부분으로서 [그림 6]의 일부이다.

[그림 7]의 5번째 블록은 영역 B의 자식 엔트리 b3과 영역 C의 공간 객체 c1과 c2를 포함하고 있다. 또한 영역 D에 공간 객체 d1도 5번째 블록에 포함되어 있기 때문에 3개의 부모 엔트리가 5번째 블록을 공유하여 색인을 구성한다. 반면에 영역 D에는 블록의 엔트리 보다 많은 수의 공간 객체를 가지고 있으므로 5번째 블록부터 7번째의 블록에까지 엔트리를 구성한다. 따라서 색인에서 빈 엔트리가 발생하지 않기 때문에 저장 공간을 효율적으로 사용하여 집계정보를 관리할 수 있다.

마지막 단계는 공간 데이터 웨어하우스의 저장소에 저장된 데이터를 엔트리의 집계정보 속성에 저장한다. 집계정보는 색인의 단말 노드에서부터 집계정보를 저장하고 상향식(bottom-up)방식으로 상위 노드에 대한 집계정보에 반영한다. 예를 들어 [그림 6]의 공간 객체 b1의 집계정보가 저장되면 b1을 포함하는 영역 B, 영역 R2, 영역 R1에 대한 엔트리에 집계정보가 반영이 되어야 한다. 따라서 b1의 집계정보 3이 갱신되면 영역 B에서 영역 R2와 영역 R1으로 이동하면서 집계정보를 반영한다.

4.3 검색 연산

본 절에서는 제안 기법에서 지원하는 두 가지 집계정보 검색 연산에 대해 설명한다. 첫 번째 검색 연산은 공간 데이터의 개념계층기반으로 집계정보를 검색하는 연산이다. 일반적으로 SOLAP은 개념계층을 기준으로 일반화 연산과 세분화 연산을 이용하여 집계결과를 얻기 때문에 개념 계층을 기반으로 집계정보를 검색하는 것은 중요하다.

또 하나의 검색 연산은 일반적인 공간 데이터 웨어하우스에서 지원하는 검색 연산으로서 영역 질의에 포함된 공간 객체들이 갖는 집계정보를 합하여 결과를 제공한다. 이 연산은 임의의 영역에 대한 분석을 수행하는 경우에 유용하며 제안 기법의 검색 연산은 기존 기법에서 사용하는 기법을 응용하여 사용하여 집계결과를 빠르게 얻을 수 있다.

공간 데이터의 개념계층기반의 집계정보 검색 연산은 제안 기법이 개념계층과 동일한 단계로 색인을 구성하였고 동일한 레벨에 대한 모든 집계정보는 연결리스트를 이용하여 연결되어 있기 때문에 효과적인 검색을 수행할 수 있다. [그림 8]은 공간 데이터의 개념계층기반의 집계정보 검색 연산을 [그림 6]의 세 번째 레벨에 대해서 집계정보를 검색하는 연산이다.

검색 연산을 수행하는 처음 단계는 루트에서부터

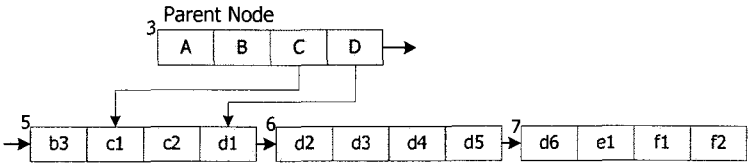


그림 7. 사전집계 색인의 구축

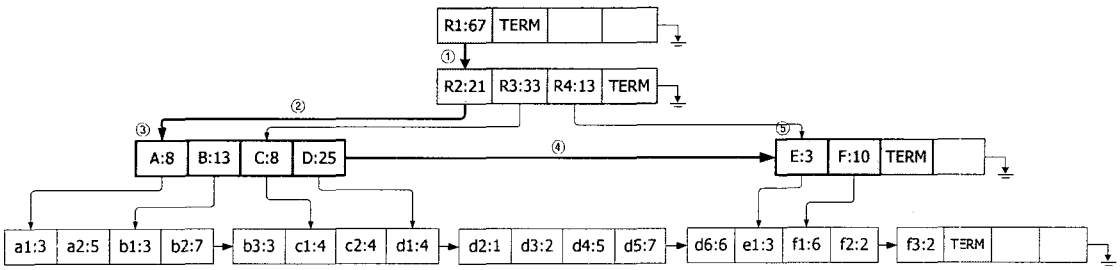


그림 8. 공간 데이터의 개념계층기반의 검색 연산에 대한 질의 결과

검색이 요청된 레벨의 처음 블록을 탐색하는 과정이다. 제안 기법은 개념계층과 색인이 동일한 레벨을 가지고 있으므로 요청된 단계의 시작 엔트리에 접근하기 위하여 색인을 탐색한다. 색인의 탐색은 루트 엔트리에서부터 자식 포인터를 이용하여 개념계층의 하위 단계의 첫 번째 엔트리에 탐색하며 요청된 단계에 도달할 때 까지 동일한 과정을 반복한다.

다음은 이전 과정을 통하여 탐색한 요청된 레벨의 시작 엔트리를 이용하여 요청된 레벨의 집계정보를 제공하는 과정이다. 집계정보는 [그림 8]의 세 번째 단계와 같이 여러 개의 블록으로 구성된다. 따라서 여러 개의 블록을 탐색하여 집계정보를 검색해야 한다. 검색 도중에 블록의 집계정보의 검색을 모두 마치면, 블록의 연결리스트를 이용하여 다음 블록에 있는 집계정보를 검색한다. 이런 과정을 반복하는 도중에 TERM 엔트리나 블록의 연결리스트에서 NULL 값을 만나게 되면 모든 집계정보를 검색한 것이므로 검색 연산을 중단한다. [알고리즘 1]은 사전집계 색인에서 공간 데이터의 개념계층기반의 검색연산을 나타낸다.

[알고리즘 1] 공간 개념계층기반 SOLAP 연산 알고리즘

Input
block : 시작 블록의 위치
level : 개념 계층의 단계
Variable
nEntry : 한 블록이 가지는 엔트리의 개수
Constant
TERM : 모든 계층의 터미널 인트리

```
Function HierarchicalQuery (block, level)
BEGIN
    // 요청된 레벨의 개념 계층을 검색
01 : for ( i = 1 to level )
02 :   block = GotoChildNode(block);
03 :   if ( block == NIL )
04 :     ErrorMessage("개념 계층 존재하지 않음");
05 :     return;
    // 현재 레벨에 대한 데이터를 모두 검색
06 : while ( block ≠ NULL){
07 :   for ( j = 1 to nEntry ){
         // 엔트리의 마지막이면,
08 :     if entry = TERM then
09 :       return ;
10 :     else
         // 집계 정보 얻기
11 :       GetAggregateInfo (entry);
12 :     entry = NextEntry(entry);
13 :   }// end for
14 :   block = NextBlock(block);
15 : }
END.
```

또 하나의 집계정보의 검색 연산은 aR-tree 기법에서의 일반적인 연산과 같이 영역 질의에 포함된 공간 객체들의 총 집계정보를 검색하는 것이다. 이 경우 영역 질의와 MBR 속성의 위상 관계에 따라서 트리를 탐색하는 방법이 달라진다. 질의 영역이 엔트리의 MBR 속성을 포함하면 현재 엔트리의 집계정보를 총 집계정보에 합하고 하위 엔트리를 탐색을 수행하지 않으므로 색인 탐색 비용이 감소하여 집계정보를 연산 비용이 감소하는 장점을 가진다. 질의 영역과 엔트리의 MBR 속성이 부분적으로 겹치는 경우에는 현재의 엔트리의 집계 정보를 이용할 수 없으므로 하위 영역의 엔트리를 탐색하여 질의 영역과 엔트리의 MBR의 위상 관계를 세부적으로 비교한다. 마지막으로 집계 영역과 MBR이 포함 관계를 가지고 있지 않으면 현재 검색하는 엔트리의 하위 영역에도 포함하고 있는 객체가 존재하지 않으므로 탐색을 마친다. 이와 같은 색인의 탐색을 통해 합해진 집계정보는 영역 질의에 포함된 공간 객체들의 집계정보를 합한 것이다. [그림 9]는 [그림 6]의 색인에 영역 질의가 요청되었을 때 질의 영역에 포함된 공간 객체의 총 집계정보를 구하는 예를 보여준다.

[그림 9]에서 질의 영역은 영역 B와 영역 D를 부분적으로 포함하지만, 영역 C는 모든 영역을 포함한다. 따라서 질의 수행 과정은 다음과 같다. 질의 영역은 전체 영역 R1과 부분적으로 겹치므로 하위 영역을 탐색하여 세부적으로 영역을 분석한다. 영역 R2에는 영역 질의가 영역 B의 공간객체 b3을 포함하므로 집계값 3을 총 집계결과에 합하며, 영역 R3도 영역 질의와 부분적으로 겹치므로 하위 영역의 색인을 탐색한다. 영역 C는 영역 질의가 모든 영역을 포함하므로 공간 객체에 대한 집계정보를 검색하지 않고 영역 C의 집계값 8을 집계 정보에 합한다. 영역 D는 모든 객체가 질의영역에 포함되지만 영역 D가 질의 영역에 부분적으로 겹치므로 하위 계층인 공간 객체 d1에서부터 d6에 대한 집계값 4, 1, 2, 5, 7, 6을 각 엔트리로부터 검색하여 총 집계정보에 포함시킨다. 영역 R2의 집계정보를 모두 포함하였으므로 영역 R4에 대한 탐색을 수행한다. 영역 R4는 영역 질의와 겹치는 부분이 존재하지 않으므로 검색을 수행하지 않고 다음 엔트리로 접근한다. 다음 엔트리는 모든 엔트리를 검색을 알려주는 TERM 엔트리를 만나므로 검색 연산을 마친다. 따라서 [그림 8]의 검색 연산을 통하

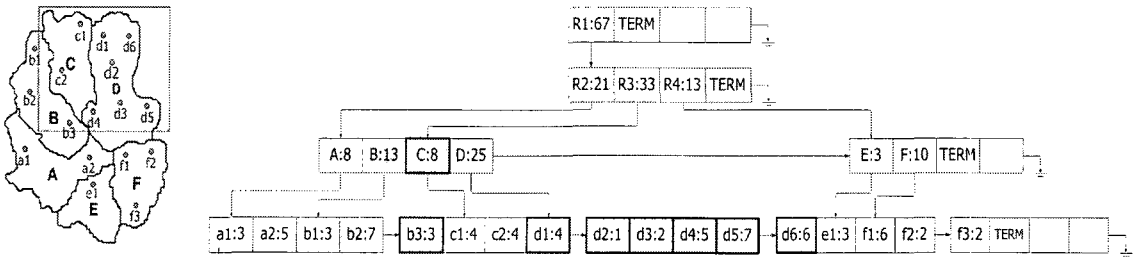


그림 9. 영역 질의에 포함되는 엔트리 검색

여 다음과 같은 집계결과를 도출된다.

총 집계 결과 = 3 + 8 + 4 + 1 + 2 + 5 + 7 = 36

[알고리즘 2]는 영역 질의에 대한 집계정보 검색 알고리즘이다.

영역 질의가 요청이 되면, RangeQuery 알고리즘을 이용하여 색인의 루트 노드의 MBR에서부터 탐색하여 엔트리의 MBR 속성과의 위상관계를 비교한다.

〔알고리즘 2〕 영역 질의에 대한 집계연산을 수행하는 RangeQuery 알고리즘

Input
entry : 시작하는 엔트리
winquery : 영역질의의 영역
Variable
sum : 총 집계 정보
Constant
TERM : 모든 계층의 터미널 엔트리

```
Function RangeQuery (entry, winquery)
BEGIN
01 : sum = 0;
02 : if ( entry == TERM)
03 :   return sum;
     // 블록에 있는 모든 엔트리를 검색하면
04 : if ( entry == _NIL)
05 :   entry = NextBlockEntry(entry);
     // 현재 노드가 단말노드인 경우
06 : if ( entry.PTR == _NIL)
07 :   if ( entry ⊆ winquery )
08 :     sum = sum + entry.SUM;
09 :     return sum;
     // 현재 노드가 단말노드가 아닌 경우
10 : else
       // 엔트리가 질의 영역과 겹치는 부분이 없으면
11 :   if (entry.MBR ∩ winquery = 0)
12 :     return sum;
       // 엔트리가 질의 영역에 포함되면
13 :   if ( entry.MBR ⊆ winquery )
14 :     sum = sum + entry.SUM;
15 :     return sum;
       // 엔트리가 질의 영역에 부분적으로 겹치면
16 :   else
17 :     for ( i = 1 to entry.CNT )
18 :       RangeQuery( entry.PTR, winquery );
19 :       NextEntry(entry);
END.
```

13번째 줄부터 15번째 줄에서는 탐색을 수행하는 과정에서 엔트리의 MBR이 영역 질의에 포함되면 현재 엔트리의 집계정보를 총 집계결과에 포함시키는 과정을 수행하고, 11번째 줄부터 12번째 줄에서는 겹치는 영역이 없는 경우에는 하위노드를 더 이상 검색하지 않는다. 16번째 줄부터 19번째 줄은 부분적으로 겹치는 영역이 존재하는 경우에는 재귀적으로 알고리즘을 수행하여 하위 노드를 검색하는 과정이며, 6번째 줄부터 9번째 줄은 공간 객체에 대한 엔트리를 만나게 되었을 경우에 공간 객체에 대한 집계정보를 집계결과에 포함시키고 재귀호출을 종료하는 과정을 수행한다.

5. 성능평가

본 장에서는 제안 기법과 aR-tree 기법과 OLAP-Favored Search 기법의 검색 성능을 비교한다. 검색 연산은 공간 데이터의 개념계층을 기준으로 검색을 수행하는 연산과 영역 질의의 집계정보에 대한 검색 연산으로 분류된다. 공간 데이터의 개념계층을 기준으로 검색하는 연산의 경우에는 질의에 대한 응답시간을 비교하고 disk I/O를 횟수를 비교하여 검색 연산을 성능을 검증한다. 영역 질의에 대한 검색 연산의 경우 질의의 영역을 점진적으로 확장하면서 질의의 응답시간을 비교하여 검색에 대한 성능을 비교한다.

5.1 평가 환경

실험에서 연산을 수행하기 위한 환경은 한 디스크 블록은 8KB이며 버퍼는 LRU(Least Recently Used)를 이용하여 50개의 블록을 관리한다. 실험에 적용된 데이터는 공간 데이터베이스 분야에서 널리 사용되는 표준 벤치마크 데이터인 TIGER/Line file을 사용하여 50만개 공간 객체를 점 데이터로 임의로 생성하

였다[16]. 그리고 공간 데이터의 개념계층은 각 단계별로 영역의 중복이 발생하지 않도록 직사각형을 이용하여 3단계로 정의하였다. [표 1]은 실험에서 사용된 시스템 환경이다.

5.2 개념계층기반의 색인 구축 성능

본 절에서는 공간 데이터 웨어하우스에 정의된 개념 계층을 기반으로 색인을 구성하는 성능을 비교한다. 성능 평가 방법은 기존 기법인 aR-tree 기법과 OLAP-Favored Search 기법과의 색인 구축 시간을 비교하고, 구축된 색인의 저장 공간을 비교함으로써 색인 구축 성능을 비교한다. [그림 10]은 개념계층기반의 색인구축비용을 기존기법과 비교하여 나타낸다.

[그림 10](a)와 같이 제안 기법의 색인구축비용은 aR-tree 기법보다 5%, OLAP-Favored Search 기법보다 7% 지연된다. 제안 기법이 기존 기법에 비하여 낮은 성능을 가지는 이유는 연산의 단위가 엔트리이므로 노드를 연산의 기본 단위로 하는 기존 기법에 비하여 연산 과정이 복잡하다. 특히 OLAP-Favored Search는 색인에서 집계정보를 관리하지 않으므로 구축 시간이 적으며, 노드에 많은 공간 정보를 추가할 수 있으므로 빠르게 색인 구축이 가능하다.

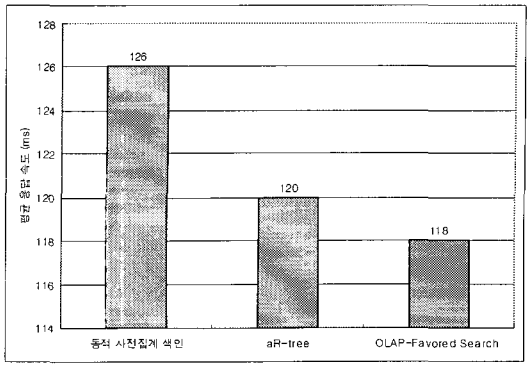
하지만 제안 기법은 엔트리를 기반으로 하기 때문에 노드에 낭비되는 공간 없이 색인을 구축한다. 따라서 노드를 기반으로 개념계층을 구축하는 색인을 구축하는 기법들에 비하여 적은 저장 공간을 사용한다. [그림 10](b)의 실험과 같이 제안 기법은 aR-tree 기법보다 26% 우수한 성능을 보이며, OLAP-Favored Search 기법보다 17% 우수한 성능을 보인다.

5.3 개념계층 기반의 검색 연산 성능

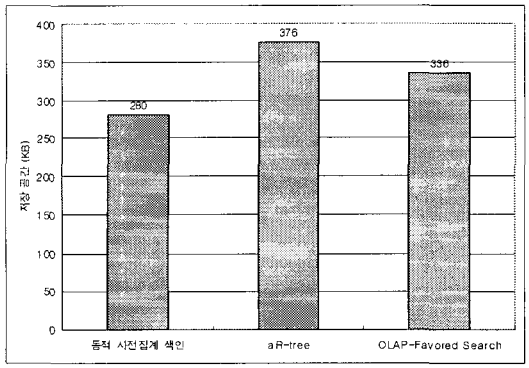
본 절에서는 공간 데이터의 개념계층을 기준으로 질의가 요청이 되는 경우에 검색 성능을 평가한다.

표 1. 실험에 사용된 시스템 환경

기종	IBM PC 호환
CPU	Pentium IV 2.6GHz
메인 메모리	1GB
하드 디스크	120GB, 7200RPM, ATA방식
운영 체제	Windows XP Professional
개발 언어	Visual C++ 6.0



(a)



(b)

그림 10. 개념계층기반의 사전집계 색인구축비용: (a) 개념계층기반의 사전집계 색인구축시간, (b) 개념계층기반의 사전집계 색인의 저장공간

제안 기법, aR-tree 기법, 그리고 OLAP-Favored Search 기법을 개념계층의 각 단계별로 질의를 수행하여 평균 응답시간을 측정한다. 이렇게 각 단계의 질의에 대한 평가를 수행하는 것은 SOLAP의 일반화 연산과 상세화 연산을 수행하는 것과 동일한 연산을 수행되기 때문이다. [그림 11]은 개념계층기반 검색 연산의 평균 응답 시간을 나타낸다.

[그림 11]의 실험은 공간 개념계층을 기준으로 사전집계 정보를 검색하는 평균 시간을 측정하는 실험이다. 제안기법은 aR-tree 기법과 OLAP-Favored Search 기법보다 1단계에서는 각각 25%와 67% 좋은 성능을 보였으며, 2단계에서는 25%와 65% 좋은 성능을 보였다. 하지만 1단계와 2단계에서는 많은 양의 데이터를 검색하지 않으므로 정확하게 성능을 비교하기 어렵다. 따라서 3단계는 제안기법이 aR-tree 기법에 비하여 24% 우수한 성능을 보였으며, OLAP-Favored Search 기법보다는 37% 우수한 성능을 보였다. 제안 기법이 다른 기법에 비하여 우수한 성능을

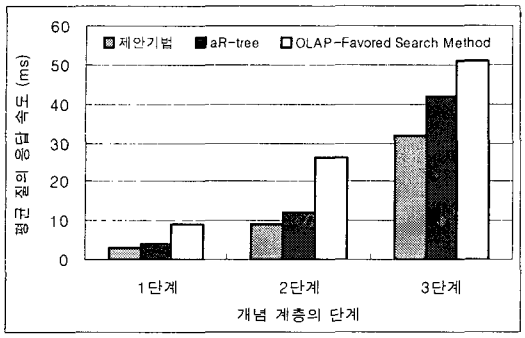


그림 11. 개념계층기반 검색 연산에 대한 평균 응답 시간 측정

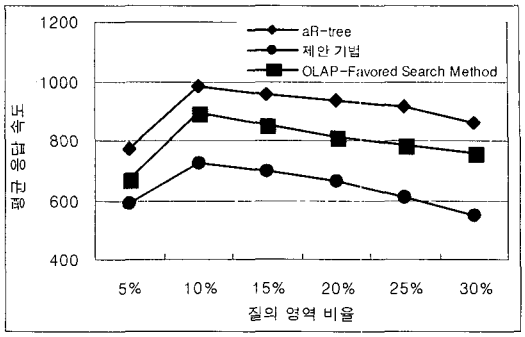


그림 12. 질의 영역 비율별 평균 응답 속도 측정

보이는 이유는 aR-tree 기법과 제안 기법은 동일한 단계에 존재하는 집계정보를 블록의 연결리스트를 이용하여 순차검색을 할 수 있기 때문이다. 반면 OLAP-Favored Search 기법은 트리의 탐색을 이용하여 집계정보에 대한 NID를 얻은 뒤에 요약테이블을 검색해야 하므로 검색을 수행하는 데에 오랜 시간이 걸리게 된다. 또한 제안 기법과 aR-tree 기법이 성능의 차이를 보이는 이유는 aR-tree 기법은 노드의 저장 공간이 낭비가 발생하기 때문이다. 하지만 제안 기법은 노드의 중간에 공백이 존재하지 않으므로 빠르게 검색 결과를 제공한다.

5.4 영역 질의에 대한 검색 연산 성능

본 절에서는 정사각형의 질의 영역을 전체 공간 영역의 5%, 10%, 15%, 20%, 25%, 30%로 생성하여 검색에 대한 평균 응답속도를 측정하는 실험을 수행하였다. 이 실험을 통해 제안 기법, aR-tree 기법, 그리고 OLAP-Favored Search 기법의 영역 질의에 대한 성능을 비교한다. 질의 발생하는 경우 aR-tree 기법은 공간 객체에 대한 집계정보를 가지지 않으므로, 실제 레코드에 접근하여 데이터에 대한 집계정보를 가지고 오도록 하였다. [그림 12]는 질의 영역 비율을 일정한 비율로 증가시키면서 평균 질의 응답 속도를 측정하는 실험이다.

이 실험 결과는 10% 질의 영역을 기준으로 aR-tree 기법과는 26% 좋은 성능을 보였으며, OLAP-Favored Search 기법과 19% 좋은 성능을 나타냈다. aR-tree 기법의 평균 응답 시간은 전반적으로 다른 기법들에 비하여 가장 길게 나타나는 이유는 공간 데이터의 집계 정보를 색인에서 관리하지 않으므로 실제 데이터를 검색해야 하는 경우에 비용이 많이 요구되기 때문이다. 또한 OLAP-Favored Search 기법의 경우는 요약 테이블을 검색하는 비용이 필요하므로 응답 시간이 지연되는 것을 알 수 있다.

또한 세 기법에서 공통적으로 영역 질의가 10%를 초과하는 경우 평균 응답 시간이 짧아지는 것을 볼 수 있다. 그 이유는 질의 영역의 크기가 증가하면 질의 영역에 포함되는 MBR이 많아지기 때문에 하위 노드의 탐색을 수행하지 않아도 집계정보를 얻을 수 있으므로 분석 비용이 감소한다. 따라서 질의 비용은 질의 영역이 10%를 기준으로 하여 그 이하인 경우에는 질의 영역이 늘어날수록 색인의 엔트리의 MBR과 질의 영역이 부분적으로 겹치는 경우가 많아지지만, 10%이상이 되면 엔트리의 MBR이 질의 영역에 포함되는 비율이 점점 증가하기 때문에 탐색 비용이 감소하게 되어 전체적인 검색 성능이 향상되는 것을 볼 수 있다.

6. 결론 및 향후 연구

본 논문은 공간 데이터 웨어하우스에서 SOLAP의 질의 비용을 줄이기 위하여 개념계층기반으로 사전집계 색인을 동적으로 구성하는 기법을 제안하였다. 제안 기법은 개념계층의 단계와 동일한 단계로 트리구조의 색인을 구성한다. 같은 부모노드를 갖는 자식노드들의 데이터가 적을 경우 하나의 노드로 각 자식노드들의 엔트리를 통합한다. 이 때 부모노드는 통합되지 않고 각각의 자식 노드의 시작위치를 갖는다. 많은 양의 데이터가 존재하여 하나의 노드에서 관리할 수 없는 경우 분할되지 않고 하나의 자식 노드에 연결리스트를 이용하여 여러 노드들을 연결함으로써 같은 레벨에 데이터를 순차적으로 저장한다. 또한 각 엔트리는 해당 엔트리에 포함되는 사전집계 정보를 관리

한다. 따라서 제안 기법은 데이터의 분포가 불균등한 영역에서도 노드들의 연결리스트를 이용하여 색인의 레벨과 개념계층의 레벨을 동일하게 유지할 수 있으며 산개 노드들의 통합으로 저장 공간의 낭비가 최소화된다. 또한 트리에 사전집계 정보를 관리하여 분석을 위한 집계정보 요청시 빠른 응답을 제공한다. 성능 평가를 통하여 색인 구축 시간은 제안기법이 aR-tree 기법과 OLAP-Favored Search 기법과 비교해 각각 5%와 7% 지연되었지만, 저장 공간에 대해 제안기법이 aR-tree 기법보다 25%, OLAP-Favored Search 기법보다 17% 감소하여 저장 공간 효율성을 보였다. 또한 검색 성능 평가에서 개념계층기반의 검색질의에 대하여 aR-tree 기법보다 24%, OLAP-Favored Search 기법보다 37% 우수하며, 영역 질의에 대해서는 aR-tree 기법보다 26%, OLAP-Favored Search 기법보다 19% 우수한 성능을 보였다.

향후 연구로는 제안된 색인 기법이 시간과 공간을 모두 고려하여 사전집계를 수행할 수 있도록 연구할 것이다.

참 고 문 헌

[1] S. Chaudhuri and U. Dayal, "An Overview of Data Warehousing and OLAP Technology," *ACM SIGMOD Record*, Vol. 26, No. 1, pp. 65-74, 1997.

[2] E.F. Codd, S.B. Codd, and C.T. Salley, "Providing OLAP(On-Line Analytical Processing) to User-Analysts: An IT Mandate," *Technical Report*, 1993.

[3] S. Rivest, Y. Bédard, M.-J. Proulx, M. Nadeau, F. Hubert, and J. Pastor, "SOLAP Technology: Merging Business Intelligence with Geospatial Technology for Interactive Spatio-Temporal Exploration and Analysis of Data," *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 60, No. 1, pp. 17-33, 2005.

[4] N. Roussopoulos, "Materialized Views and Data Warehouse," *SIGMOD Record*, Vol. 27, No. 1, pp. 21-26, 1998.

[5] N. Stefanovic, J. Han, and K. Koperski, "Object-Based Selective Materialization for Efficient Implementation of Spatial Data Cubes," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 12, No. 6, pp. 1-21, 2000.

[6] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, "Efficient OLAP Operations in Spatial Data Warehouses," *In Proc. of the 7th International Symposium on Spatial and Temporal Databases, Lecture Notes in Computer Science*, pp. 443-459, 2001.

[7] F. Rao, L. Zhang, X.L. Yu, Y. Li, and Y. Chen, "Spatial Hierarchy and OLAP-Favored Search in Spatial Data Warehouse," *Proc. of the 6th ACM international workshop on Data warehousing and OLAP*, pp. 48-55, 2003.

[8] A. Guttman, "R-trees: A Dynamic Index Structure for Spatial Searching," *In Proc. of ACM SIGMOD*, pp. 47-57, 1984.

[9] H. Garcia-Molina, Jeffrey D. Ullman, and J. Widom, *Database System Implementation*, Prentice-Hall Inc., New Jersey, Upper Saddle River, 2000.

[10] L. Zhang, Y. Li, F. Rao, X. Yu, and Y. Chen, "An Approach to Enabling Spatial OLAP by Aggregating on Spatial Hierarchy," *Proc. Data Warehousing and Knowledge Discovery, Lecture Notes in Computer Science*, pp. 35-44, 2003.

[11] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh, "Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-tab, and Sub-totals," Data Mining and *Knowledge Discovery*, Vol. 1, No. 1, pp. 29-53, 1997.

[12] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publisher, San Francisco, Calif., 2001.

[13] I. Lazaridis and S. Mehrotra, "Progressive Approximate Aggregate Queries with a Multi-Resolution Tree Structure," *In Proc. of ACM*

SIGMOD, pp. 401-412, 2001.

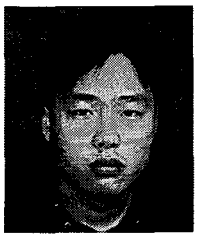
[14] D. Zhang and V.J. Tsotras, "Improving Min/Max Aggregation over Spatial Objects," *Proc. of the 9th ACM International Symposium on Advances in Geographic Information Systems*, pp. 88-93, 2001.

[15] T.K. Sellis, N. Roussopoulos, and C. Faloutsos, "The R+-tree: A Dynamic Index for Multi-Dimensional Objects," *Proc. of 13th International Conference on Very Large Data Bases*, pp. 507-518, 1987.

[16] TIGER/Line Files, 2000 Technical Documentation, U.S. Bureau of Census, Washington DC, accessible via URL : http://www.census.gov/geo/www/tiger/tigerua/ua_tgr2k.html.

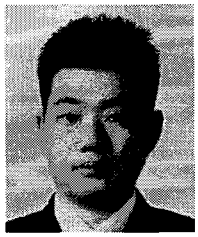
전 병 윤

2001년~2005년 인하대학교 컴퓨터공학부(공학사)
2005년~현재 인하대학교 컴퓨터정보공학과 석사과정
관심분야 : 공간 데이터베이스, 공간 데이터 웨어하우스, 데이터 통합, OLAP



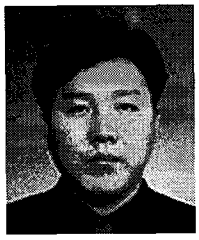
이 동 욱

1996년~2003년 상지대학교 전자계산공학과 학사
2003년~2005년 인하대학교 컴퓨터정보공학과 석사
2005년~현재 인하대학교 컴퓨터정보공학과 박사과정
관심분야 : Spatial Database Warehouse, Spatial Information Management, Ubiquitous 환경을 위한 SDBMS



유 병 섭

2002년 인하대학교 컴퓨터공학부(공학사)
2004년 인하대학교 컴퓨터공학부(공학석사)
2004년~현재 인하대학교 대학원 컴퓨터정보공학과(박사과정)
관심분야 : 공간데이터베이스, 공간 데이터 웨어하우스, Data Stream, 유비쿼터스 컴퓨팅



김 경 배

1992년 인하대학교 전자계산공학과 (공학사)
1994년 인하대학교 대학원 전자계산공학과 (공학석사)
2000년 인하대학교 대학원 전자계산공학과 (공학박사)
2000년~2004년 한국전자통신연구원 선임연구원
2004년~현재 서원대학교 컴퓨터교육과 조교수
관심분야 : 이동 실시간 데이터베이스, 스토리지 시스템, GIS, VOD 등

배 해 영

1974년 인하대학교 응용물리학과(공학사)
1978년 연세대학교 대학원 전자계산학과(공학석사)
1989년 숭실대학교 대학원 전자계산학과(공학박사)
1985년 Univ. of Houston 객원교수
1992년~1994년 인하대학교 전자계산소 소장
1982년~현재 인하대학교 컴퓨터공학부 교수
1999년~현재 지능형GIS연구센터 센터장
2000년~현재 중국 중경우전대학교 대학원 명예교수
2004년~2006년 인하대학교 정보통신대학원 원장
2006년~현재 인하대학교 대학원 원장
관심분야 : 분산 데이터베이스, 공간 데이터베이스, 지리정보 시스템, 멀티미디어 데이터베이스 등