

공간 데이터 웨어하우스에서 시공간 데이터의 효율적인 집계 정보 관리 기법

°유호선*, 유병섭*, 박순영*, 이재동**, 배해영*

*인하대학교 컴퓨터·정보공학과

**단국대학교 정보컴퓨터학부

e-mail : {°hsyou, subi, sunny}@dblabinha.ac.kr, letsdoit@dankook.ac.kr, hybae@inha.ac.kr

Efficient Aggregate Information Management of Spatiotemporal Data in Spatial Data Warehouses

°Ho-Sun Ryu*, Byeong-Seob You*, Soon-Young Park*, Jae-Dong Lee**, Hae-Young Bae*

*Dept. of Computer Science & Information Engineering, Inha University

**Division of Information and Computer Science, Dankook University

요 약

다차원 분석을 위한 OLAP 연산에서는 사용자의 요청에 빠르게 응답하기 위해 집계 값을 미리 계산하여 저장해 두는 사전 집계 방식을 이용한다. 시공간 데이터에 대한 사전 집계 기법으로는 R-트리의 각 노드에 대한 과거 집계 값을 요약 테이블로 관리하는 기법과 R-트리의 노드에서 현재 집계 값을 관리하는 기법이 있다. 그러나 이 기법들은 현재와 과거 모두의 집계 정보를 필요로 하는 시스템에서는 성능이 저하되며, 특히 과거 집계 정보의 경우 시간에 따른 계층화가 되어있지 않아 시간에 대한 계층 분석에 어려움이 있다.

본 논문에서는 시공간 데이터의 현재와 과거 집계 정보를 효율적으로 관리하는 기법을 제안한다. 제안 기법은 aR-tree를 이용하여 해당 영역에 대한 현재 집계 정보를 저장하고, 각 노드에 과거 집계 정보에 대한 연결을 위하여 링크를 추가하였다. 과거 집계 정보는 각 노드의 과거에서 현재까지의 집계 정보를 계층 구조로 유지하는 시간 요약 집계 테이블을 만들어 저장한다. 따라서 제안한 기법은 현재와 과거 집계 정보를 모두 유지할 수 있으므로 현재와 과거 집계 정보에 대한 처리 성능을 향상시킨다. 또한 제안 기법에서는 공간 정보를 공간 인덱스인 R-트리로 유지하고, 과거로부터의 시간 정보를 시간 요약 집계 테이블을 이용하여 계층화시켜 유지하므로 시간과 공간에 대한 계층 분석이 용이하다.

1. 서론

공간 데이터 웨어하우스는 여러 운영 데이터에서 추출된 공간 및 비공간 데이터를 주제별로 통합하여 저장하는 시스템으로서, 통합된 데이터는 OLAP(On-Line Analytical Processing) 연산을 통한 다차원 분석으로 기업이나 일반 사용자의 의사결정에 도움을 줄 수 있다[1, 2]. 최근 교통 감시 시스템이나 이동 통신 시스템과 같은 많은 응용에서 이러한 공간 데이터 웨어하우스를 이용한 다차원 OLAP 분석 요구가 증가되면서 현재 이와 관련된 많은 연구가 활발히 진행되고 있다[3, 4]. 다차원 분석을 위한 OLAP 연산에서는 집계 함수와 Group-by 연산을 이용한 집계 정보가 많이 사용되는데 분석 요구가 있을시 마다 대용량의 데이터에 대해 집계 정보를 계산하기에는 많은 시간과 비용이 필요하다. 특히 교통 상황 분석과 같이 신속한 현재와 과거 집계 정보가 필요한 응용에서는 OLAP 연산을 지원하기가 더욱 어렵다. 그러므로 다차원 분석을 위한 OLAP 연산에서는 사용자의 요청에 빠르게 응답하기 위해서 자주 사용되는 질의의 집계 값을 미리 계산하여 저장해 두는 사전 집계 방식을 이용하여 효율적인 OLAP 연산을 지원한다.

공간 데이터 웨어하우스에서 시공간 데이터에 대한 사전 집계 관리 기법에는 R-트리의 각 노드에 대한 과거 집계 값을 별도의 요약 테이블로 관리하는 기법과 R-트리의 노드에서 현재 집계 값을 직접 관리하는 기법이 있다. 그러나 이러한 기법들은 현재 정보나 과거 정보 중 한쪽에 대해서만 집계 정보를 제공할 수 있기 때문에 현재와 과거 정보 모두를 필요로 하는 응용 분야에서

성능을 저하시키는 문제점이 있다. 특히 과거 집계 정보의 경우에는 시간에 대한 계층이 존재하지 않기 때문에 시간에 따른 다양한 분석에 어려움이 있다.

본 논문에서는 공간 데이터 웨어하우스에서 시공간 데이터의 집계 정보를 효율적으로 관리하는 기법을 제안한다. 제안 기법은 공간 데이터의 계층 개념을 지원하는 기존 R-트리의 노드 구성을 확장한 aR-tree를 이용하여 해당 영역에 대한 현재 집계 정보를 저장하고, 각 노드에 과거 집계 정보에 대한 연결을 유지하기 위하여 링크를 추가하였다. 또한 과거부터 현재까지의 집계 정보를 가지는 시간 요약 집계 테이블을 생성하여 트리의 해당되는 각 노드와 연결을 유지하고 과거에 대한 집계 정보를 시간 차원의 계층에 따라 저장한다. 그러므로 제안 기법은 현재와 과거 정보를 모두 필요로 하는 다양한 응용 분야에서 우수한 성능을 나타내며 과거 정보에 대해 시간에 따른 다양한 분석이 용이한 장점을 가진다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 설명을 하고, 3장에서는 시공간 데이터의 현재와 과거 집계 정보를 제공하는 확장된 트리 구조와 시간 요약 집계 테이블의 구성에 대해 살펴본다. 계속해서 4장에서는 현재와 과거 집계 정보에 대한 갱신 및 검색 기법에 대해 설명을 한다. 마지막으로 5장에서 성능평가를 한 후 6장에서 결론 및 향후연구를 기술하겠다.

2. 관련연구

2.1 데이터 웨어하우스에서의 차원 계층 구조

데이터 웨어하우스에서는 각 차원에 대해 계층 구조를 가진다.

1) 본 연구는 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었음

예를 들어 상품에 대한 차원 테이블이 존재한다고 하면, 하나의 상품은 고유의 이름을 가지는 동시에 상품 분류, 제조 회사 등 상위개념에 포함되게 된다. 그러므로 각 차원의 계층 구조를 따라 Roll-Up이나 Drill-Down과 같은 OLAP 연산을 제공할 수 있다[5]. 공간 차원에 대해서도 공간 인덱스의 특징을 이용하여 계층 구조를 나타낼 수 있다. 즉, 트리의 각 노드들은 자신의 자식 노드들의 모든 영역을 공간적으로 포함하고 있기 때문에 자식 노드들의 상위 계층이 되고, 자식 노드들은 부모 노드의 부분 영역을 나타내는 하위 계층이 된다[6]. 따라서 상위 노드의 측정값은 하위 노드들의 측정값의 합으로 구할 수 있게 되므로 공간 차원에 대해서도 OLAP 연산이 가능해진다.

2.2 OLAP-Favored Search Tree

OLAP-Favored Search Tree[7]는 관련연구 2.1과 같은 특징을 가진 공간 인덱스 중 가장 많이 사용되는 R-트리[6]를 이용하여 공간 차원의 계층 구조를 유지하며 각 노드 영역에 존재하는 객체의 과거부터 현재까지의 총 집계 값을 요약 테이블에서 관리한다. 질의에 대한 처리는 질의 범위에 포함되는 노드들의 식별자를 가지고 요약 테이블을 검색함으로써 해당 노드의 집계 값을 얻을 수 있다. 그러나 요약 테이블이 각 노드와 대응되는 집계 값에 대해 단순히 과거로부터 현재까지의 총 집계만을 가지고 있기 때문에 공간 영역에 존재하는 객체들의 현재 집계 정보는 물론 OLAP에서 많이 사용되는 특정 시간에 대한 분석이나 연도별 집계 등 시간 차원에 따른 분석은 지원하지 못한다.

2.3 aR-tree

aR-tree[8]도 R-트리를 사용하여 각 노드의 해당 영역에 현재 존재하는 객체의 집계 정보를 트리의 노드에 함께 저장한다. 질의에 대한 처리는 질의 범위에 완전히 포함되는 노드들에 대해서는 더 이상의 하위 노드 검색 없이 현재 위치의 노드에 함께 저장되어 있는 집계 값을 질의 결과 값에 포함시키고, 부분적으로 겹치는 노드들에 대해서는 하위 노드들을 계속 검색하여 질의 범위에 포함되는 노드들의 집계 값을 얻는다. 만약 최하위 노드까지 검색한 결과 최하위 노드도 질의 범위에 부분적으로 겹친다면 사실 데이터가 저장되어 있는 기본 테이블에 직접 접근하여 결과를 얻거나 통계적 예측 방법을 이용하여 근접한 결과 값을 얻는다[7, 8, 9].

aR-tree 기법은 질의 요청 시 질의 범위에 해당하는 노드들에 대해서 노드에 함께 저장되어 있는 집계 값을 결과로 주기 때문에 현재 데이터에 대한 집계 정보 요청에 대하여 빠르게 응답을 제공할 수 있다. 그러나 트리의 노드에서 현재의 정보만을 가지고 있기 때문에 과거로부터의 집계 정보 요청에는 응답 할 수 없는 문제점이 있다.

3. 시공간 데이터의 효율적인 집계 정보 관리

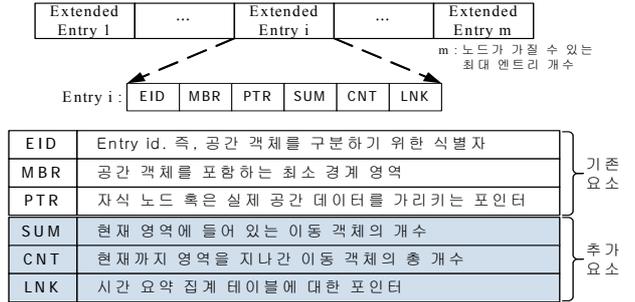
시공간 데이터의 사전 집계 관리를 위한 공간 데이터 웨어하우스의 주요 차원으로는 도로 영역을 나타내는 공간 차원과 도로 영역에 존재하는 이동 객체에 대한 차원이 있다. 그 밖에도 과거로부터 현재까지의 집계 정보를 나타내기 위한 시간 차원이 존재한다.

본 장에서는 앞서 언급한 주요 차원들을 기반으로 효율적인 OLAP 연산을 위한 시공간 데이터의 현재와 과거 집계 정보 관리 구조를 설명하겠다.

3.1 노드의 구조

공간 영역 안에 존재하는 객체의 현재 집계 정보를 관리하기 위하여 하나의 노드는 최대 m개의 엔트리를 가질 수 있다. 기본적으로 노드의 각 엔트리는 R-트리의 노드 엔트리에 포함되어 있는 엔트리 식별자, 각 엔트리가 포함하고 있는 객체들의 최소 경계 영역(MBR: Minimum Bounding Rectangle), 자식 노드들에 대한 포인터를 가진다. 그리고 현재 각 엔트리의 영역 안에 들어 있는 이동 객체들의 집계 값과 지금까지 해당 엔트리의 영역을

거처간 이동 객체의 총 이동량이 OLAP 연산을 위해 추가되었고, 각 노드의 영역별 과거 집계 정보를 가지는 시간 요약 집계 테이블에 대한 링크가 존재한다. 확장된 노드의 구조는 아래의 (그림 1)과 같다.



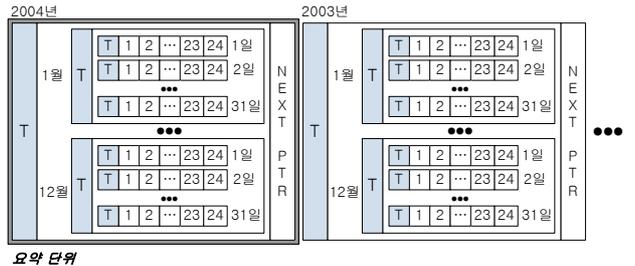
(그림 1) 현재 집계 정보 관리를 위한 확장된 R-트리 노드

3.2 시간 요약 집계 테이블의 구조

기존 공간 데이터 웨어하우스의 관련 연구에서 사전 집계된 요약 테이블에 대해 다루었지만 단순히 노드의 한 영역에서 측정되는 모든 값의 합만을 나타내었다. 그러나 시공간 데이터와 관련해서는 시간에 따른 집계 정보가 더 필요하며 의미가 있다. 그러므로 본 논문에서는 각 노드마다 과거로부터 축적된 집계 정보를 시간 차원과 연계하여 미리 계산한 시간 요약 집계 테이블을 제안하였다.

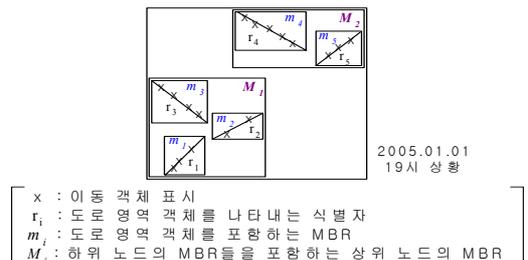
제안하는 시간 요약 집계 테이블은 여러 개의 “요약 단위”로 구성이 되며, 하나의 요약 단위는 사용자 요구가 있거나 질의가 빈번한 시간 차원의 레벨을 최소 집계 단위로 하여 상위 계층의 집계 정보까지 시간에 따라 나타낸다. 예를 들어 (그림 2)의 2004년도 집계를 보면 시간 차원의 최소 집계 단위가 “시간”이 되며, “시간”에 대한 집계를 기반으로 “일”, “월”, “연도”에 대한 각 집계(T)가 하나의 요약 단위가 된다. 만약 최소 요구 레벨이 “월”이라면 시간 요약 집계 테이블의 하나의 요약 단위는 “월”과 “연도”에 대한 집계를 가진다.

시간 요약 집계 테이블은 하나의 영역에 대한 과거 집계 정보를 필요한 내용만 요약하여 유지하기 때문에 이동 객체에 대한 모든 과거 정보를 가지는 것보다 상당히 적은 양의 저장 공간을 사용한다. 그리고 시간 차원에 따른 요약 단위로 관리되기 때문에 Roll-Up이나 Drill-Down 등의 OLAP 연산에 유용하다.

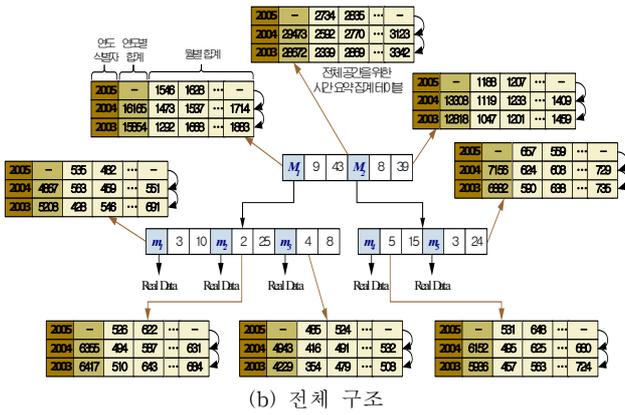


(그림 2) 과거 집계 정보 관리를 위한 시간 요약 집계 테이블

다음의 (그림 3)은 시공간 데이터에 대한 예(a)와 공간 데이터 웨어하우스에서 시공간 데이터의 현재와 과거 집계 정보를 관리하는 전체 구조(b)를 나타낸다.



(a) 시공간 데이터에 대한 예



(b) 전체 구조

4. 집계 정보의 갱신 및 검색 기법

4.1 집계 정보의 갱신 기법

일반적으로 데이터 웨어하우스에서의 갱신은 데이터 변경이 있을 때마다 갱신하기에는 그 비용이 너무 크고 갱신 시간이 많이 들기 때문에 갱신 요청이 있거나 일정 주기에 따라 갱신을 한다. 본 절에서는 위와 같은 갱신 주기를 따르며, 도로 지역을 나타내는 공간 차원에는 변경이 거의 일어나지 않는다는 가정 하에 이동 객체의 이동에 따른 노드와 시간 요약 집계 테이블의 집계 정보 갱신 기법을 설명한다.

4.1.1 현재 데이터에 대한 노드 갱신 기법

노드에 대한 갱신은 이동 객체가 이동할 경우 이동 객체의 이전 위치와 새로운 위치를 이동 객체의 식별자와 함께 별도의 변경 내용 테이블에 저장하여 관리한다. 질의가 들어오거나 일정 주기가 되면 저장해 놓은 변경 내용 중 이동 객체의 이전 위치 값이 포함되는 노드의 집계 정보 값을 감소시키고, 새로운 위치 값이 포함되는 노드의 집계 정보 값을 증가시킴으로써 갱신이 이루어진다.

갱신 기법은 크게 두 단계로 나누어 진행이 되는데 첫 번째 단계는 저장해 놓은 변경 내용 테이블과 도로 지역을 나타내는 공간 차원 집합에 대해 공간 조인을 수행한다. 그 결과 변경이 필요한 도로 영역의 집합과 각각의 도로 영역에 대한 변경된 집계 값을 얻는다. 예를 들면 어떤 한 도로 영역 r3에 이동객체 3개가 들어오고 5개가 영역을 벗어났다고 한다면 조인 수행 결과로 (r3, -2)를 얻게 된다. 두 번째 단계는 조인 수행 결과를 변경이 필요한 노드들에 반영함으로써 갱신을 수행한다. 즉, 도로 영역 r3의 기존의 집계 값에 결과 값 -2를 더해줌으로써 노드 갱신이 이루어진다. 이러한 노드 갱신 방법은 어떤 한 노드에 계속적으로 갱신이 집중 될 경우 매번 노드에 접근하여 계산을 하지 않고 같은 노드에 관련된 갱신 내용을 한번에 처리할 수 있는 장점이 있다.

각 노드의 영역을 지나간 이동 객체의 총 이동량에 대한 갱신도 저장해 놓은 변경 내용 테이블에서 새로운 위치 필드만을 이용하여 새로운 위치 값이 포함되는 노드의 총 이동량 값을 1씩 증가해 주면 된다.

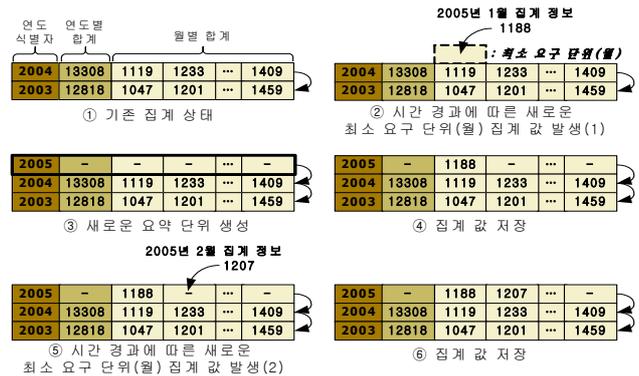
4.1.2 과거 데이터에 대한 시간 요약 집계 테이블 갱신 기법

시간 요약 집계 테이블에 대한 갱신은 기존의 요약 테이블이나 이와 유사한 성격의 실제화 뷰에서 갱신 발생시 전체에 대한 재계산을 수행하거나 갱신 데이터를 전파하는 것과는 달리 새로 생성된 요약 집계 정보를 기존의 시간 요약 집계 테이블에 추가만 하면 된다. 기존 기법들은 현재 집계 정보만을 나타내거나 과거로부터 현재까지 누적된 집계 정보를 다루기 때문에 현재 집계에 대한 변화가 발생하면 현재 집계뿐만 아니라 과거 집계에도 영향을 미친다. 하지만 제안 기법에서는 현재 집계와 과거 집계를 분리하여 관리하기 때문에 갱신 주기가 되면 이미 변화가 반영된 현재 집계 노드로부터 집계 값을 가져와 노드와 연결된 시간 요

약 집계 테이블에 저장시키면 된다.

갱신 기법은 (그림 4)에 나타난 것처럼 기존 집계 상태(①)에서 시간 경과에 따른 새로운 최소 요구 단위 집계 값이 발생(②)되면 먼저 새로 발생한 집계 값이 요약 단위의 어떤 위치에 저장되어야 하는지 살펴본다. 만약 새로운 집계 값이 (그림 4)의 ②처럼 하나의 요약 단위가 완성된 후 다음 요약 단위의 첫 번째 최소 요구 단위 집계 값이라면 새로운 요약 단위를 생성(③)하고 새로운 최소 요구 단위 집계 값을 요약 단위의 지정된 위치에 저장한다. 즉, (그림 4)에서 요약 단위가 “월”, “연도”로 구성되어 있으므로 새로운 집계 값 2005년 1월 집계 정보는 새로 생성된 2005년도 요약 단위에서 연도 식별자(요약 단위 식별자)와 연도별 집계 다음에 위치하게 된다. 그렇지 않고 새로운 최소 요구 단위 집계 값이 (그림 4)의 ⑤처럼 이미 생성된 요약 단위의 중간 부분에 해당하는 집계 값이라면 집계 값이 저장되어야 할 위치를 계산하여 저장한다. 저장될 위치 계산은 요약 단위가 사용자 정의에 따라 고정된 크기의 구조를 가지므로 요약 단위 식별자와 각 집계 값을 저장할 저장소의 크기만을 가지고 구할 수 있다. 예를 들어 요약 단위가 “월”과 “연도” 두 레벨로 이루어져 있다면 새로운 집계 값이 저장될 위치는 (연도 식별자 저장소의 크기 + (연도별 집계 값 저장소의 크기 + (월별 집계 저장소의 크기 × (새로운 집계 값이 저장될 순번(월) - 1))) 방식으로 계산된다.

새로운 요약 단위는 기존 시간 요약 집계 테이블의 맨 처음의 요약 단위 앞에 연결을 시킴으로써 최근 집계 정보를 앞쪽에 저장시킨다. 이것은 과거 정보 중에서도 최근의 과거 정보가 더 자주 사용되므로 검색을 좀 더 쉽게 하기 위함이다.



(그림 4) 시간 요약 집계 테이블의 갱신 과정

4.2 집계 정보의 검색 기법

공간 데이터 웨어하우스에서 공간 데이터에 대한 질의는 특정 범위를 대상으로 하여 그 안에 포함되는 데이터에 대한 집계 값 등을 구하는 영역 질의(Window-query)의 형태를 가진다.

본 절에서는 이러한 영역 질의에 대해 제한한 구조를 검색하는 기법을 현재 집계와 과거 집계 검색으로 나눠서 설명하겠다.

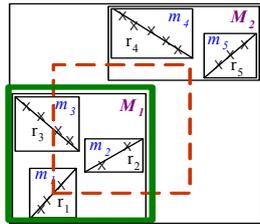
4.2.1 현재 데이터에 대한 집계 정보 검색 기법

현재 집계 정보에 대한 질의는 교통 감시 시스템에서 긴급 환자가 발생했거나 화재가 발생하였을 경우 긴급 차량이 신속히 이동할 수 있도록 도로 정보를 제공해야 하는 경우에 많이 사용된다. 예를 들면 “특정 지역의 현재 총 차량 수는?”이나 “특정 병원을 중심으로 반경 2Km 내의 가장 교통이 한산한 지역은?”과 같은 질의가 있다. 이러한 질의는 질의 영역에 포함되는 노드들을 트리에서 검색한 후 선택된 노드들이 현재 가지고 있는 집계 값을 합하여 결과를 얻을 수 있다.

영역 질의에 포함되는 노드들에 대한 검색은 기존 R-트리의 검색과는 달리 질의 영역에 완전히 포함되는 노드에 대해서는 자식 노드를 검색할 필요 없이 현재 계층에 있는 집계 값을 결과로 주고, 부분적으로 겹치는 노드에 대해서만 자식 노드를 반복적으로 검색하면 된다. 리프 노드까지 검색한 결과 리프 노드도 질의 영역과 부분적으로 겹쳐진 경우는 직접 원본 데이터에 접근하여

결과를 주게 된다. 그러나 실시간으로 빠른 응답을 요구하는 OLAP 연산에서는 질의에 대한 정확한 결과보다는 빠른 시간 내에 근접한 결과를 요구하는 경우가 있다. 이 경우에는 부분적으로 겹치는 노드에 이동 객체가 고르게 분포한다는 가정 하에 질의 영역에 포함되는 노드 영역의 비율(%)을 계산하여 현재 노드가 가지고 있는 집계와 곱하므로 근접 결과를 줄 수 있다[7].

예를 들면 (그림 5)에서 점선으로 된 사각형이 질의 영역일 경우 m_2 노드는 질의 영역에 완전히 포함되므로 가지고 있는 집계 값인 2를 그대로 결과 값에 더하면 되고, m_1 은 위에서 설명한 비율 계산 방법에 의하여 질의 영역에 약 30% 겹친다고 가정하면 결과 값은 $3 \times 0.3 = 1$ 이 된다. 또 다른 예로 (그림 5)의 굵은 선으로 된 사각형이 질의 영역일 경우 M_1 노드가 질의 영역에 완전히 포함되기 때문에 노드 M_1 의 자식 노드들을 검색할 필요 없이 M_1 의 집계 값이 결과가 된다.



(그림 5) 질의 영역 예

4.2.2 과거 데이터에 대한 집계 정보 검색 기법

과거 집계 정보에 대한 질의는 교통의 흐름을 분석하여 교통 혼잡 지역의 제어 및 개선을 효과적으로 수행하기 위해 많이 사용된다. 예를 들면 “특정 지역의 최근 2년간 차량 이동량은?”이나 “특정 지역의 2004년 데이터 중 차량 이동량이 가장 많은 달은?”과 같은 질의가 있다.

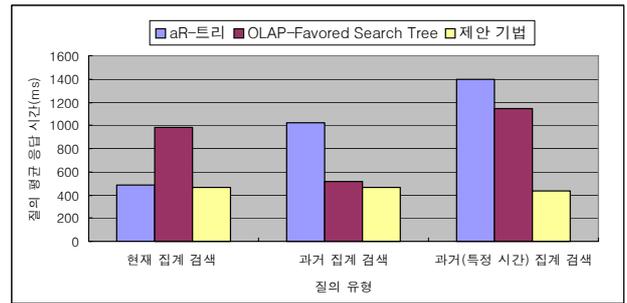
과거 집계 정보에 대한 검색도 현재 집계 정보의 검색과 마찬가지로 먼저 질의 영역에 포함되는 노드들을 트리에서 검색한다. 검색된 각 노드들에 연결된 시간 요약 집계 테이블에 접근하여 질의에 해당되는 기간을 찾음으로써 정보를 구할 수 있다. 만약 질의가 최근 2년(2004년, 2003년)간 차량 이동량을 구하는 것이라면 먼저 시간 요약 집계 테이블에서 요약 단위 식별자들을 검사하여 2004년도 요약 단위를 찾는다. 찾은 요약 단위에서 필요한 연도별 집계 값을 검색한 후 다음 요약 단위로 이동하여 다음 연도인 2003년도의 연도별 집계 값을 합해주면 된다. 다음 요약 단위로의 이동은 요약 단위 마지막에 존재하는 포인터를 이용하면 되고, 요약 단위 내에서의 검색은 시간 요약 집계 테이블 갱신에서 사용한 위치 계산 방법을 이용하면 된다.

5. 성능평가

본 장에서는 제안한 기법의 성능을 측정하기 위하여 aR-tree와 OLAP-Favored Search Tree 그리고 제안 기법을 현재 데이터와 과거 데이터에 관한 질의 처리로 나누어 실험 하였다. 현재와 과거 질의 모두에 응답 할 수 없는 기존 기법들은 지원되지 않는 질의 유형에 대해서 원본 데이터에 직접 접근하여 질의 결과를 얻게 하였다. 실험은 각 질의 유형별로 전체 영역의 0.01%의 크기를 갖는 1,000개의 임의 영역 질의에 대한 평균 응답 시간을 구함으로써 수행되었다. 실험에 사용된 데이터는 공간 데이터를 이용한 실험에서 많이 사용되고 있는 TIGER/Line file[10]을 사용하였으며 이동 객체는 약 100만개 점을 균일 분포로 임의 생성하였다. 실험을 위한 환경은 Pentium4 3.0GHz CPU, 1GB 메모리, 120GB HDD를 가진 Windows XP OS가 되었다.

실험 결과는 아래의 (그림 6)에서 보이는 것과 같이 현재 데이터에 대한 집계에서는 aR-tree와 제안 기법이, 과거 데이터에 대한 집계에서는 OLAP-Favored Search Tree와 제안 기법의 성능이 해당 질의 유형을 지원하지 못하는 기법들에 비해 약 두 배 정도 응답시간이 빠른 것으로 나타난다. 이것은 제안 기법이 현재 집계와 과거 집계 모두를 지원할 수 있기 때문에 지원되지 않

는 집계 정보에 대해 원본 데이터에 직접 접근하여 결과를 얻어 오는 기존 기법들에 비해 뛰어난 성능을 보이게 된다.



(그림 6) 질의 유형별 응답 시간 비교

6. 결론 및 향후연구

본 논문에서는 공간 데이터 웨어하우스에서 시공간 데이터의 집계 정보를 관리하는 기법에 대해서 제안하였다. 제안 기법은 확장된 aR-tree를 이용하여 시공간 데이터의 현재 집계 정보를 관리하고 시간 요약 집계 테이블을 이용하여 과거 집계 정보를 관리하므로 시공간 데이터의 현재와 과거 집계 정보 모두에 대해 빠른 질의응답을 제공할 수 있도록 하였다. 또한 다양한 질의 유형에 대해 성능 평가를 하여 제안 기법의 우수성을 확인하였다.

향후 연구로는 자주 발생되는 Roll-Up이나 Drill-Down 등의 OLAP 연산 요구에 따른 시간 요약 집계 테이블의 재구성 기법에 대한 연구가 필요하며, 질의 크기 변화에 따른 성능 비교 등의 실험을 추가하여 제안 기법의 성능평가 보완이 필요하겠다.

참고문헌

- [1] 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.
- [2] S. Chaudhuri and U. Dayal, "An Overview of Data Warehousing and OLAP Technology", ACM SIGMOD, 1997.
- [3] N. Stefanovic, J. Han, and K. Koperski, "Object-Based Selective Materialization for Efficient Implementation of Spatial Data Cubes", TKDE, 2000.
- [4] D. Zhang and V.J. Tsotras, "Improving Min/Max Aggregation over Spatial Objects", ACM GIS, 2001.
- [5] M. Wang and B. Iyer, "Efficient roll-up and drilldown analysis in relational database", ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, 1997.
- [6] A. Guttman, "R-tree: a dynamic index structure for spatial searching", ACM SIGMOD, 1984.
- [7] F. Rao, L. Zhang, X.L. Yu, Y. Li, and Y. Chen, "Spatial Hierarchy and OLAP-Favored Search in Spatial Data Warehouse", DOLAP workshop on Data Warehousing and OLAP, 2003.
- [8] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, "Efficient OLAP Operations in Spatial Data Warehouses", Lecture Notes in Computer Science, 2001.
- [9] I. Lazaridis and S. Mehrotra, "Progressive Approximate Aggregate Queries with a Multi-Resolution Tree Structure", ACM SIGMOD, 2001.
- [10] TIGER/Line Files, 2000 Technical Documentation, U.S. Bureau of Census, California, accessible via URL http://arcdata.esri.com/data/tiger2000/tiger_statelayer.cfm?sfips=06.