논문 2004-41CI-2-3

공간 데이터베이스에서 질의 결과 크기 추정을 위한 공간 분할

(Spatial Partitioning for Query Result Size Estimation in Spatial Databases)

황 환 규*

(Whan-Kyu Whang)

요 약

질의 최적화기의 중요 기능 중에 하나는 질의가 주어졌을 때 질의 조건을 만족하는 입력 레코드의 개수를 추정하는 일이다. 관계 데이터베이스와 마찬가지로 공간 데이터베이스에서 질의 결과 크기 추정은 입력 데이터 공간을 버켓으로 불리는 작은 영역으로 분할한 후 분할된 영역에 대해서 질의 결과 크기를 추정한다. 추정의 정확도는 작은 영역으로 분할할 때 근사 계산한 데이터와 실제 데이터의 차이에 의해서 결정되며 이것은 공간 분할을 어떻게 분할하는가에 달려 있다. 기존의 방법은 일차원에 많이 사용되는 데이터의 범위를 균일하게 하는 너비 균등 방법과 빈도수의 합을 일정하게 하는 높이 균등 방법을 공간상의 이차원에 적용한 면적 균등 분할과 개수 균등 분할 방법에 기초를 두고 있다. 본 논문에서 제안한 방법은 공간을 분할할 때 데이터의 범위와 빈도수의 곱을 면적으로 나타낸 후 면적 값의 차이가 가장 큰 순서로 버켓을 정하는 방법으로 데이터 범위와 빈도수를 동시에 고려하여 최적의 버켓을 결정한다. 본 논문에서는 제안한 방법과 기존의 방법을 실제 데이터와 인위 데이터를 사용하여 질의 크기, 버켓수, 데이터 개수, 데이터 크기의 변화에 대해서 질의 결과 추정에 대한 정확도를 비교, 분석하여 제안한 방법의 성능 우수성을 확인한다.

Abstract

The query optimizer's important task while a query is invoked is to estimate the fraction of records in the databases that satisfy the given query condition. The query result size estimation in spatial databases, like relational databases, proceeds to partition the whole input into a small number of subsets called "buckets" and then estimate the fraction of the input in the buckets. The accuracy of estimation is determined by the difference between the real data counts and approximations in the buckets, and is dependent on how to partition the buckets. Existing techniques for spatial databases are equi-area and equi-count techniques, which are respectively analogous in relation databases to equi-height histogram that divides the input value range into buckets of equal size and equi-depth histogram that is equal to the number of records within each bucket. In this paper we propose a new partitioning technique that determines buckets according to the maximal difference of area which is defined as the product of data ranges and frequencies of input. In this new technique we consider both data values and frequencies of input data simultaneously, and thus achieve substantial improvements in accuracy over existing approaches. We present a detailed experimental study of the accuracy of query result size estimation comparing the proposed technique and the existing techniques using synthetic as well as real-life datasets. Experiments confirm that our proposed techniques offer better accuracy in query result size estimation than the existing techniques for space query size, bucket number, data number and data size.

Keywords: Query Optimization, Query Result Size Estimation, Spatial Selectivity Estimation

I. 서 론

지리 정보 시스템 (GIS: Geographic Information Systems)은 점, 선, 다각형, 평면 등과 같은 공간 데이터를 저장, 관리한다[1]. 공간 데이터를 관리하는 상용 데이터베이스 시스템으로서 ESRI의 ARC/INFO[2], Inter-Graph의 MGE, MapInfo, Informix[3] 등이 있다. 대부분의 선도적인 데이터베이스 공급 업체는 공간 데이터

* 정회원, 강원대학교 전기전자정보통신공학부
(Kangwon National University, Dept. of Electrical and Computer Eng.)
접수일자: 2003년8월7일, 수정완료일: 2004년3월2일

에 대한 지원을 제공하고 있다.

데이터베이스 시스템의 질의 처리기는 다양한 질의 실행 계획의 비용을 추정하여 가장 효율적인 액세스 계획을 선택한다[4]. 이러한 비용은 질의에 대한 중간 결과 크기의 추정에 기초하여 산출한다. 관계 데이터베이스에서 질의 결과 크기는 보통 데이터베이스 상에 유지되는 릴레이션에 대한 다양한 통계를 사용한다. 지금까지 관계 데이터베이스의 질의 결과 크기를 추정하기 위해서 히스토그램[5], 샘플링[6], 파라메터 기법[7]이 제안되었다. 이 중 히스토그램은 애트리뷰트의 빈도수 분포를 근사 계산하기 위해서 애트리뷰트의 값을 버켓으로 그룹핑하여 각 버켓에 유지된 요약 데이터를 근거로 애트리뷰트의 값과 빈도수를 근사 계산하였다. 다른 방법과 달리 히스토그램의 주된 장점은 실행 시간에 대한 부담이 적고, 데이터가 특정 확률 분포를 따를 필요가 없으며, 작은 공간을 차지하며 (보통 수 백 바이트 차지), 오차율이 적다는데 있다. 이러한 이유로 히스토그램 방법이 실제 상용 데이터베이스 시스템 (DB2, Informix, Ingres, Microsoft SQL Server, Sybase)에서 많이 사용되고 있다.

공간 데이터베이스에서 질의 처리 비용은 관계 데이터베이스와 같이 질의 조건을 만족하는 질의 결과 크기에 의하여 결정된다[8]. 질의 결과 크기를 계산하기 위해서 전체 질의를 수행하는 것은 비실용적이므로 데이터를 근사 계산하기 위해 통계 데이터를 유지하고 이러한 통계를 바탕으로 결과 크기를 추정한다. 공간 데이터베이스에서 요약 데이터를 얻기 위해 공간을 가상으로 분할하여 분할된 영역 내에 존재하는 MBR(Minimum Bounding Rectangle: 최소 경계 사각형)의 개수를 요약 데이터로 유지한다. 이때, 공간을 어떠한 방법으로 분할하느냐에 따라 그 정확성이 달라진다. 본 논문은 공간 데이터를 근사 계산하기 위해 전체 데이터를 버켓이라 불리는 작은 영역으로 분할하고 각 버켓에 데이터의 개수를 유지하는 히스토그램 방법을 사용한다. 기존의 분할 방법은 데이터 값 간의 거리를 일정하게 나누어 분할하는 면적 균등과, 버켓 내의 데이터의 빈도수를 일정하게 하는 개수 균등 분할 방법으로 관계 데이터베이스의 너비 균등, 높이 균등 분할 방법에 해당된다. 본 논문에서 제안하는 공간 분할 방법은 공간을 버켓으로 분할할 때 데이터 값 간의 거리와 빈도수의 곱을 면적으로 나타낸 후 면적 값의 차이가 가장 큰 순서로 버켓을 정하는 방법으로 데이터 값 간의 거리와 빈도수를 동시에 고려하여 최적의 버켓을 결정하는 방법이다. 이 방법은 1차원 데이터에 대한 질의 결과 크기 추정에 가장 우수한 방법으로 보고된 바 있으며[9] 이것을 2차원 공간 데이터베이스에 적용하여 기존의 방법과 질의 결과 크기 추정의 정확성을 비교, 분석하여 제안한 방법의 우수함을 확인하고자 한다.

본 논문의 구성은 다음과 같다. Ⅱ장은 기존의 공간 분할에 의한 요약 데이터 작성 방법에 대해 살펴보고 Ⅲ장에서는 본 논문이 제안한 공간 분할에 기초한 요약 데이터 작성 방법에 대해 논의한다. Ⅳ장은 제안한 방법에 의한 질의 결과 크기 추정 방법을 소개한다. Ⅴ장은 실험을 통하여 기존의 방법과 제안한 방법에 대한 성능 평가를 한다. 마지막으로 Ⅵ장에서 결론을 맺는다.

Ⅱ. 기존의 공간 분할 방법

공간 데이터베이스는 다양한 모양, 서로 다른 크기, 편재된 데이터(skewed data)로 구성되므로 이들을 고려하여 전체 공간을 분할한 후, 분할된 버켓 내에서 최소 경계 사각형으로 표현된 데이터의 개수를 요약 데이터로 유지하게 된다. 모든 데이터는 분할 영역 내에 균일하게 분포되어 있다고 가정한다.

기존의 공간 데이터 분할 방법으로는 균등 분할 기법과 인덱스에 기초한 분할 기법 등이 있다. 균등 분할 기법은 공간 분할에 대한 성질이 모두 같도록 분할하는 방법으로 균등 분할의 기준에 따라 분할 공간의 면적이 같도록 분할하는 면적 균등 기법과 분할 공간의 데이터 개수가 같도록 분할하는 개수 균등 기법으로 분류할 수 있다. 인덱스 기반의 분할 기법은 공간 인덱스 구조에 의해 생성된 분할을 요약 데이터를 유지하기 위한 공간 분할로 사용하는 방법이다.

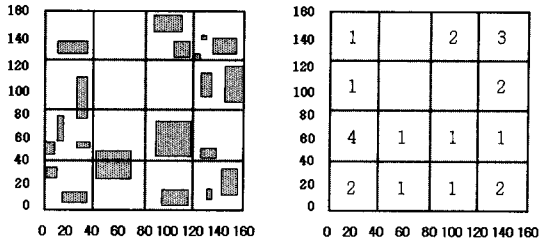
최근에 데이터 밀도에 근거하여 공간 분할을 시도한 방법은[8] 최적의 데이터 분할이 NP-hard 문제가 되어 이를 해결하기 위해서 입력 영역을 수직이나 수평으로 나누는 이진 공간 분할 방법을 사용하였다. 이 방법은 시간 복잡도가 O(N2.5)가 되는 문제점이 있으나 이를 줄이기 위해서 휴리스틱스 기법 (greedy 방법)을 사용하여 지역적으로 최적의 분할을 시도하였다.

1. 면적 균등 분할 방법

면적 균등 분할 기법은 모든 분할 영역의 면적이 같아지도록 공간을 분할하는 방법이다. 이 방법은 공간 분

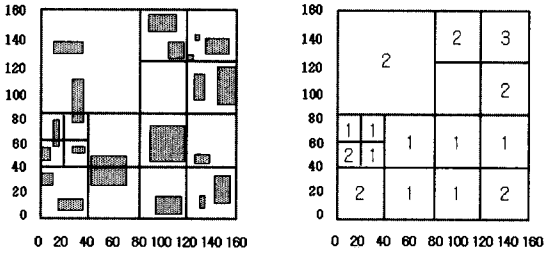
할이 비교적 간단하지만 데이터의 분포와 무관하게 균일한 격자 형태로 공간을 분할하므로 데이터의 분포 특성을 나타내는데 한계가 있다.

그림 1은 면적 균등 분할 방법을 적용하여 공간 데이터를 분할한 예를 보여준다. 각 분할 영역의 숫자는 분할 영역에 존재하는 공간 데이터의 개수를 의미한다. 큰 버켓은 오차율이 커지는 가능성이 있음으로 면적을 균등하게 분할하는 것은 최악의 오류를 최소화하는 방법으로 생각할 수 있다. 분할 영역의 면적을 크게 나누면 데이터가 실제보다 더 넓은 영역에 분포한 것처럼 나타나고 분할 영역의 면적을 작게 나누면 데이터의 개수가 실제보다 더 많이 표현되는 문제가 발생한다. 면적 균등 분할 방법은 데이터의 분포를 전혀 고려하지 않기 때문에 이와 같은 문제가 발생하게 된다. 비록 이 방법은 요약 데이터 작성은 간편하고 빠르나, 실제로 정확한 질의 결과 크기 추정치를 얻어내기에는 한계를 갖고 있다.



(a) 분할 영역 (버켓수=16) (b) (a)의 요약 데

그림 1. 면적 균등 분할.
Fig. 1. Equi-Area Partition.



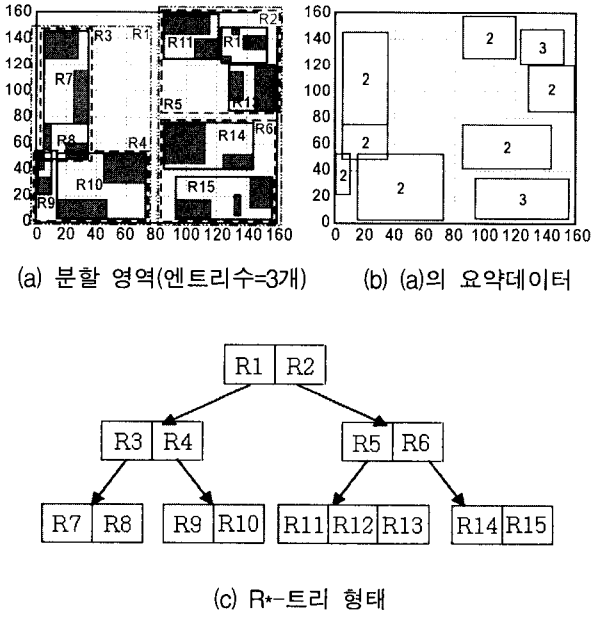
(a) 분할 영역 (버켓수=16) (b) (a)의 요약

그림 2. 개수 균등 분할.
Fig. 2. Equi-Count Partition.

2. 개수 균등 분할 방법

개수 균등 분할 방법은 모든 분할 영역의 공간 데이터 개수가 동일하도록 분할하므로 편재된 영역일수록 더욱 세밀히 분할하게 된다. 이 방법은 면적 균등 분할 방법과는 달리 데이터의 분포를 고려해서 분할하므로 질의 결과 크기 오차를 줄이는 효과가 있다.

그림 2(a)는 모든 분할 영역의 MBR 개수가 동일하도록 분할하기 위해 사분 트리(Quad-Tree) 방식에 의해 MBR 개수가 4개 이상인 영역을 4개로 분할한 결과를 보여준다. 예제 데이터에 대하여 총 3차 분할까지 진행되고 마지막 분할 결과에 대한 요약 데이터를 작성하면 그림2(b)와 같이 나타낼 수 있다. 편재되지 않은 영역에 대해서 실제보다 더 넓은 영역에 데이터가 분포된 것으로 나타난다. 또한 세밀히 분할한 영역의 데이터의 중복 카운트 현상이 발생한다. 따라서 이러한 영역에 대한 공간 질의는 오차율이 증가한다.



(a) 분할 영역(엔트리수=3개) (b) (a)의 요약데이터

(c) R*-트리 형태

그림 3. R*-트리 분할.
Fig. 3. R*-tree Partition.

3. R*-트리 분할

인덱스 분할 기법은 공간 인덱스 구조에 의해 생성된 분할을 요약 데이터를 유지하기 위한 공간 분할로 사용하는 방법이다. R-트리[10]는 공간 데이터에 대한 일반적인 인덱스 구조이다. 인덱스 구조를 형성할 때 R-트리 삽입 알고리즘은 내부 노드의 경계 사각형에 대한 면적, 마진, 겹침을 최소화도록 한다. 본 논문에서는 공간 인덱스 구조로 가장 효율적인 것의 하나로 알려진 R*-트리를 사용한다[11]. R*-트리는 분할 공간 내에 비어있는 공간과 분할 영역들 사이의 겹치는 영역을 최소화한다.

그림 3(a)와 (b)는 R*-트리 인덱스 구조와 트리의 리

프 노드를 최종 분할 영역으로 하였을 때 요약 데이터를 나타낸 것이다. 그림 3(c)는 그림 3(a)에서 공간 분할에 사용된 R*-트리 공간 인덱스의 계층 구조를 보여준다. 분할에 사용한 엔트리의 수를 3으로 하여 인덱싱한 결과이며 최종 분할 영역은 트리의 리프 노드인 R7~R15의 영역이 된다.

공간 인덱스에 기초한 분할 방법은 R*-트리 인덱스 구조의 특성상 리프 노드가 가리키는 버켓 영역의 중첩이 발생한다. 따라서 편재가 심한 영역일수록 더 많이 중첩된 버켓을 생성하게 되어 질의 결과 크기 추정 오차를 증가시키는 요인이 된다. 또한 R*-트리 삽입 알고리즘은 새로운 노드를 생성할 때 전체를 고려하지 않고 지역적인 모양에 의하여 결정됨으로 생성된 최종 버켓은 상당히 편재된 모양을 형성한다.

Ⅲ. 제안한 공간 분할 방법

기존의 분할 방법은 공간 분할시 공간 데이터의 빈도수만을 고려하거나 데이터간의 거리만을 고려하여 분할하였다. 면적 균등 분할 방법은 데이터 간의 거리를 균일하게 하여 모든 분할 면적을 동일하게 분할하였다. 이는 편재된 분포에 대하여 부정확한 질의 결과 크기 추정 결과를 보여준다. 반면, 개수 균등 분할 방법이나 R*-트리 분할 방법의 경우는 공간 데이터의 빈도수만을 고려하여 분할 영역 내 개수를 균일하게 분할하였고 실제 데이터 간의 거리는 고려되지 않은 문제가 있었다.

본 논문에서 제안한 공간 분할 방법은 데이터 간의 거리와 빈도수를 모두 고려하기 위해서 데이터 간의 거리와 빈도수의 곱을 면적으로 나타낸 후 면적 값의 차이가 가장 큰 순서로 버켓을 결정하는 방법이다. 데이터 간의 거리와 빈도수를 동시에 고려하여 최적의 버켓을 결정하게 되므로 위의 문제점들을 극복하게 된다. 이 방법[9]은 1차원 데이터에 대한 질의 결과 크기 추정에 가장 우수한 방법으로 보고된 바 있으며([9]의 MaxDiff (V,A)) 이것을 2차원 공간 데이터베이스에 적용하여 기존의 방법과 질의 결과 크기 추정의 정확성을 정량적으로 비교, 분석하여 그 우수성을 확인하고자 한다.

1. 최대 면적 차이 분할 방법

최대 면적 차이 분할 방법은 데이터 값의 범위 (데이터 값 간의 차이)와 빈도수를 동시에 고려한 분할 방법

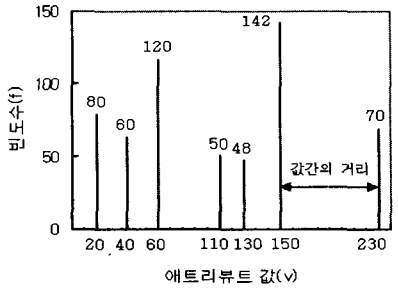


그림 4. 1차원 데이터 집합.
Fig. 4. One-Dimensional Data Set.

표 1. 최대 면적 차이를 이용한 그룹핑 결과.
Table 1. Grouping using Maximum Area Difference.

v	20	40	60	110	130	150	230
f	80	60	120	50	48	142	70
A	1,600	1,200	6,000	1,000	960	11,360	70
ΔA	400	4,800	5,000	40	10,400	11,290	
분할	총 분할 버켓수=5개 버켓1의 범위={20,40}, 평균 빈도수=(80+60)/2=70 버켓2의 범위={60}, 평균 빈도수=120 버켓3의 범위={110,130}, 평균 빈도수=(50+48)/2=49 버켓4의 범위={150}, 평균 빈도수=142 버켓5의 범위={230}, 평균 빈도수=70						

으로 데이터 값의 범위와 빈도수의 곱을 면적으로 나타내며 면적의 크기 순으로 버켓을 분할하는 방법이다. 면적간의 차이가 큰 것부터 허용된 버켓 수(그룹핑 개수)보다 하나 더 작은 수만큼의 차이까지를 분할 경계로 한다. 면적을 구하는 공식은 다음과 같다.

$$A_{x_i} = (v_{i+1} - v_i) \times f_{v_i} \quad (i \geq 1) \tag{1}$$

1차원 데이터베이스에서 최대 면적 차이를 이용한 분할 방법을 예로 들면 데이터 값과 빈도수의 관계가 그림 4와 같이 분포되어 있다고 할 때 표 1은 이 데이터 집합을 (1) 식에 근거하여 면적 값을 계산하고 그 차이를 계산하여 5개의 버켓으로 분할하는 과정을 보여주고 있다.

표 1에서 영역간의 면적 차이 ΔA를 계산한 결과 그 차이가 큰 순으로 보면 11,290, 10,400, 5,000, 4,800, 400, 40 이다. 분할하고자 하는 버켓 수가 5개라면 분할 경계 수는 버켓 수보다 하나 더 작은 4개로 결정해야 한다. 따라서 면적 차이 값이 큰 순서 4개는 11,290, 10,400, 5,000, 4,800이 되며 버켓의 경계는 면적 차이 값에 해당

되는 데이터 값, 즉, 40에서 60사이, 60에서 110사이, 130에서 150사이, 150에서 230사이가 된다.

최대 면적 차이 분할 방법에서 면적은 데이터 범위와 빈도수의 곱으로 나타내며 면적이 큰 순서로 버켓의 경계를 정하게 되므로 버켓 간에는 많은 차이가 나는 반면 버켓 내의 값은 비교적 일정하게 된다.

2. 2차원 공간 데이터베이스에 적용

1차원 데이터베이스에 사용한 최대 면적 차이분할 방법을 2차원 공간에 적용하여 최종 분할 영역을 얻어 내기 위해서는 각 차원에 대한 최대 면적 차이 분할을 시도한 후, 각 차원에서 구한 경계를 교차시켜 최종 분할 영역을 구한다. 다음 식은 일차원 식을 토대로 2차원 공간의 x축과 y축 상의 면적 계산 공식이다. x_i는 x축 상에 i번째 좌표 값을 의미하고, $f_{x_i MBR}$은 좌표 x_i과 좌표 x_{i+1} 사이에 존재하는 MBR 개수를 의미한다. y축에 대해서도 동일하게 구한다.

$$A_{x_i} = (x_{i+1} - x_i) \times f_{x_i MBR}, \quad A_{y_i} = (y_{i+1} - y_i) \times f_{y_i MBR} \qquad (2)$$

면적 차이가 주어진 값 이상인 곳을 분할 경계로 하여 x축 상의 모든 분할 경계를 구하고, y축에 대해서도 이와 동일한 방법으로 분할 경계를 찾아낸다.

<최대 면적 차이 공간 분할 알고리즘>

1) 단계1
- x, y축으로 이진 분할
- 모든 분할 영역의 MBR 개수가 임계치가 될 때까지 분할

2) 단계2
- 각 차원별로 얻어진 이진 분할 영역의 면적을 계산
- 면적간의 차이가 임계 면적 차이 값 이상되는 버켓을 분할 경계로 결정

3) 단계3
- 각 차원에서 구한 분할 경계를 서로 교차하여 이차원 공간상의 최종 분할 영역으로 결정

최종적으로 각 축에 대한 분할 경계의 교차점에 의하여 2차원 공간의 분할 영역이 결정된다. 최대 면적 차이 공간 분할 알고리즘은 아래와 같이 정리할 수 있다.

가. x축 상의 분할 – 단계1, 단계2

그림 5는 분할 알고리즘 단계1에 해당되며, 편재된 데이터 분포를 고려하여 분할 영역 내 데이터의 개수가 임계 MBR 개수를 초과하면 해당 영역만 2진 분할을 시도하도록 하였다 (예제에서 임계 MBR 개수는 5개). 단계2에서는 이진 분할 후 각 분할 영역의 크기와 영역 내 빈도수의 개수를 파악하여 표 2와 같이 최종 분할 경계를 찾을 수 있다. 공간상의 최종 분할 경계를 표시하면 그림 6과 같다.

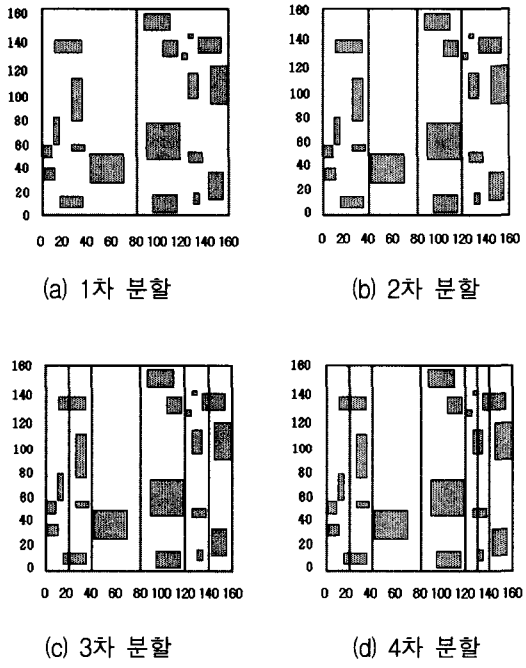


그림 5. x축 상의 2진 분할.
Fig. 5. Binary Partition on x Coordinate.

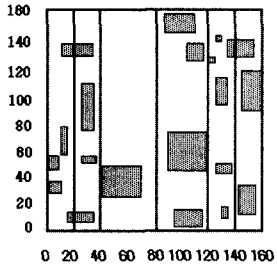
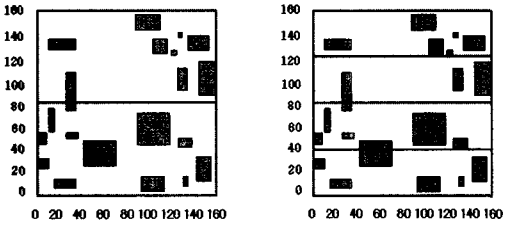


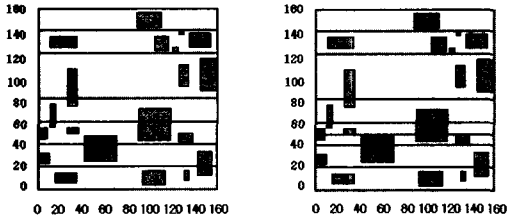
그림 6. x축 상의 최종 분할.
Fig. 6. Final Partition on x Coordinate.

표 2. x축 상의 최종 분할 경계
Table 2. Final Boundary by Binary Partition on x Coordinate.

v	20	40	80	120	130	140	160
f	5	4	1	4	4	4	3
A	100	80	40	160	40	40	60
ΔA	20	40	120	120	0	20	
분할	분할 경계={20, 40, 80, 120, 140}						



(a) 1차 분할 (b) 2차 분할



(c) 3차 분할 (d) 4차 분할

그림 7. y축 상의 2진 분할.
Fig. 7. Binary Partition on Y Coordinate.

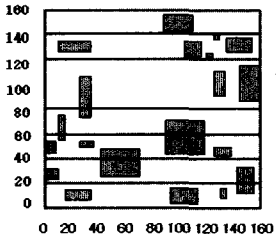
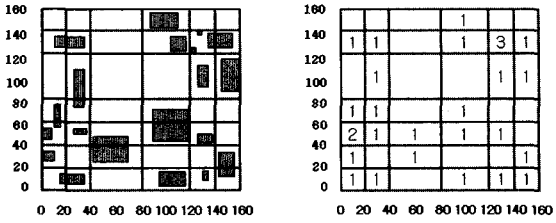


그림 8. y축 상의 최종 분할.
Fig. 8. Final Partition on Y Coordinate.

표 3. y 축 상의 최종 분할 경계.
Table 3. Final Boundary by Binary Partition on Y Coordinate.

v	20	40	50	60	80	120	140	160
f	4	3	4	4	3	3	5	1
A	80	60	40	40	60	120	100	20
ΔA	20	40	0	20	60	20	80	
분할	분할 경계={20, 40, 60, 80, 120, 140}							



(a) 분할 영역 (b) (a)의 요약데이터

그림 9. 최대 면적 차이 공간 분할.
Fig. 9. Spatial Partition by Maximum Area Difference.

나. y축 상의 분할 - 단계1, 단계2

y축 상의 분할은 x축 상의 분할과 같은 방법으로 수행하며 그림 7은 분할 알고리즘 단계1에 해당된다. 편재된 데이터 분포를 고려하여 분할 영역 내 데이터의 개수가 임계 MBR개수인 5개를 초과하면 해당 영역만 2진 분할을 시도하도록 하였다. 단계 2에서는 이진 분할 후 각 분할 영역의 크기와 영역 내 빈도수의 개수를 파악하여 표 3과 같이 최종 분할 경계를 찾을 수 있다. 공간 상의 최종 분할 경계를 표시하면 그림 8과 같다.

다. x, y축 상의 최종 분할 경계 교차 - 단계3

그림 9(a)는 분할 알고리즘 단계3에 해당되며 x축과 y축 상의 각 분할 경계를 교차시켜 최종적인 2차원 분할 영역을 구한 것이다. 그림 9(b)는 (a)의 최종 분할 영역을 통해 얻어진 요약 데이터를 나타낸 것이다.

Ⅳ. 질의 결과 크기 추정

본 장에서는 공간 분할로 얻어진 요약 데이터로부터 질의 결과 크기 추정값을 계산하는 방법을 기술한다. 임의의 질의가 주어질 때 해당 질의에 대한 질의 결과 크기 추정값은 각 분할 영역의 요약 데이터 중 질의 영역과 겹치는 부분에 해당하는 데이터 개수의 합으로 나타낼 수 있다. 이때 질의 영역과 겹치는 부분의 데이터 개수는 분할 영역이 유지하고 있는 요약 데이터의 개수를 질의 영역에 겹치는 비율로 곱하여 나타낸다. 이는 공간 분할을 통해 얻어진 요약 데이터의 개수가 분할 영역 내에 균일하게 분포한다는 가정에 기초한 계산 방법이다. 질의 결과 크기 추정 공식을 정리하면 다음과 같다.

$$S(Q) = \sum_{i=1}^{k} n_i \times r_{i_{Overlap}} \quad (3)$$

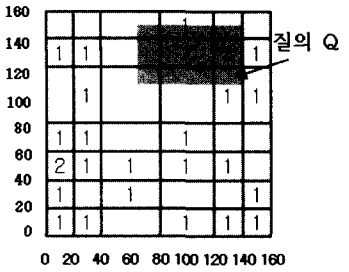


그림 10. 질의 결과 추정.
Fig. 10. Estimation of Query Result Size.

여기서 $s(Q)$는 질의 Q에 대한 질의 결과 크기 추정치이며, n은 질의 Q와 i번째 겹치는 분할 영역 내 요약 데이터의 개수를 의미한다. $r_{i_{Overlap}}$은 분할 영역 내 질의 Q와 겹치는 면적의 비율이다.

그림 10은 제안한 방법에 의해 구해진 요약 데이터이다. 질의 Q의 영역은 x좌표가 70에서 140이고, y좌표는 110에서 150이다. 질의 Q와 교차하는 분할 영역은 4개이며, 질의와 겹치는 분할 영역의 면적을 고려하여 질의 결과 크기 추정치를 구하면, S(Q) = 1*1/2 + 1*1 + 3*1 + 1*1/3 = 4.83개가 된다.

V. 실 험

본 장에서는 다양한 공간 분할 기법의 질의 결과 크기 추정 성능을 알아본다. 실험은 주어진 실제 데이터와 인위 데이터를 기준으로 수행하였으며 각각의 공간 분할 방법을 통해 생성된 요약 데이터의 정확성을 평가한다. 다양한 공간 분할 방법의 성능을 비교하기 위하여 다음의 상대 오차율 공식을 사용한다.

$$E(Q) = \frac{r-e}{r} \times 100 \quad (4)$$

E(Q) : 질의 Q 에 대한 상대 오차율 (%)
r : 실제 질의 결과 크기
e : 추정된 질의 결과 크기

실험은 성능 평가의 공평성을 위해 모든 실험은 동일한 조건에서 이루어진다. 모든 분할 방법에 대하여 버켓이 생성되면, 평균 10,000개의 영역 질의에 대한 질의 결과 크기 추정치를 계산하고 상대 오차율을 이용하여 성능을 평가한다.

실험에 쓰인 데이터는 실제 데이터와 인위 데이터이며, 실제 데이터는 공간 연산에 대한 성능 평가를 위해 일반적으로 사용되는 Long Beach Data를 사용하였다[12]. 실험은 모두 5가지를 수행하였다. 질의 크기에 따른 실험과 최대 면적 차이의 임계치, 버켓 수에 의한 실험은 실제 데이터를 대상으로 이루어졌다. 데이터 크기에 의한 실험은 모두 40,000개의 인위 데이터를 생성하여 실험하였고, 데이터 개수에 의한 실험은 5,000개에서부터 시작하여 70,000개까지의 인위 데이터를 사용하였다. 질의 크기는 전체 공간의 너비와 높이의 2% ~ 30% (전체 공간 면적의 0.04% ~ 9%)로, 데이터 크기는 2% ~ 25% (전체 공간 면적의 0.04% ~ 6.25%)까지 변화를 주며 실험하였다.

1. 질의 크기에 따른 성능 평가

본 실험은 질의 크기의 변화에 따른 각 공간 분할 방법의 성능을 보여준다. 그림 11은 Long Beach 데이터를 100개의 버켓으로 고정시켜 분할한 후 질의 크기 변화에 대한 상대 오차율을 구한 결과이다. 사용된 질의 크기는 전체 영역의 너비와 높이의 2% ~ 30% 정도의 크기이다. 질의 크기가 5%에서 최대 면적 차이 분할 방법이 면적 균등 분할 방법보다 15%의 성능 향상을 보이고 개수 균등 분할, R*-트리 분할 보다 10%의 성능 향상을 보인다. 개수 균등 분할이나 R*-트리 분할은 유사한 오차율을 보였다. 그림에서 질의 크기가 커짐에 따라 오차율이 줄어드는 경향을 보인다. 그 이유는 오차는 질의 영역에 완전히 포함되는 버켓에서는 발생하지 않으며 질의 영역에 부분적으로 겹치는 버켓에서만 발생하게 된다. 따라서 질의 크기가 클수록 질의 영역과 부분적으로 겹치는 버켓의 수가 감소하기 때문에 오차율 감소를 가져오게 된다. 그림에서 면적 균등 분할 방법은 AE, 개수 균등 분할 방법은 CE, R*-트리 분할 방법은 RS, 최대 면적 차이 공간 분할 방법은 MD로 표기하였다.

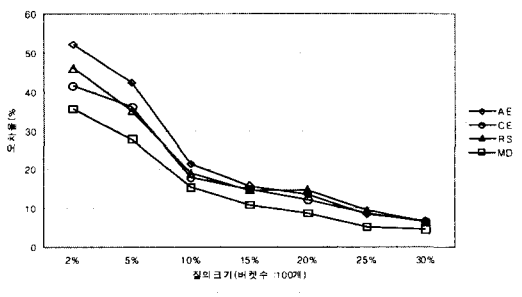


그림 11. 질의 크기에 따른 오차율.
Fig. 11. Estimation Errors by Query Size.

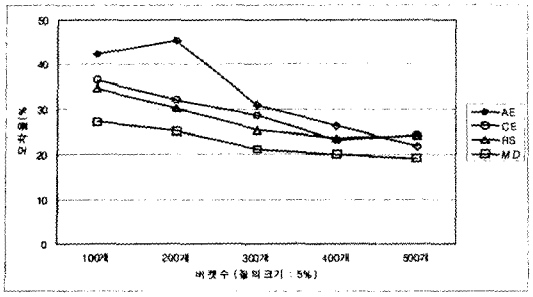


그림 12. 버켓 수에 따른 오차율.
Fig. 12. Estimation Errors by Bucket Numbers.

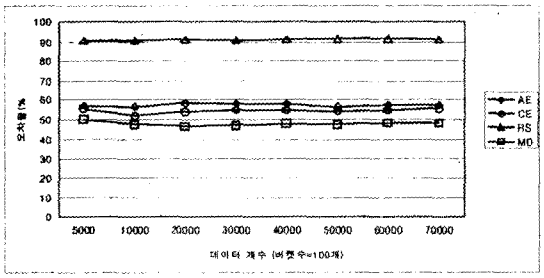


그림 13. 데이터 개수에 따른 오차율.
Fig. 13. Estimation Errors by Data Numbers.

2. 버켓 수에 따른 성능 평가

본 실험은 버켓 수의 변화에 따른 각 공간 분할 방법의 성능을 보여준다. 그림 12는 버켓수를 100에서 500으로 변화시킬 때 평균 상대 오차율을 구한 그래프이다. 데이터의 요약을 위해 더 많은 수의 버켓을 사용한다는 것은 그만큼 데이터의 특성을 잘 반영하는 것이 되므로 대체로 버켓 수가 많아짐에 따라 오차율이 줄어드는 경향을 보인다. 최대 면적 차이 방법이 R*-트리 방법보다 5%에서 8%, 개수 균등보다 10%, 면적 균등보다 5%에서 20%의 성능 향상을 보였다. R*-트리 분할 방법에서 버켓의 수를 조정하는 어려움이 있다. 이 문제는 주어진 버켓 수를 초과하지 않으면서 원하는 버켓 수에 근접하도록 트리의 자식 노드 수를 조정하였다.

3. 데이터 개수에 따른 성능 평가

데이터 개수에 따른 실험에서는 버켓 수를 100개로 고정하고 임의 데이터를 5,000개부터 70,000개까지 생성하여 각 그룹별로 오차율을 측정하였다. 그림 13에서 모든 분할 방법은 데이터 개수에 따라 성능상의 큰 변화를 보이지 않음을 알 수 있다. 최대 면적 차이 분할 방법이 가장 좋은 성능을 보여주지만 R*-트리 방법이 최악의 성능을 보이고 있다. R*-트리는 분할 영역간의 겹치는 영역이 지나치게 많이 발생하여 이 같은 성능을

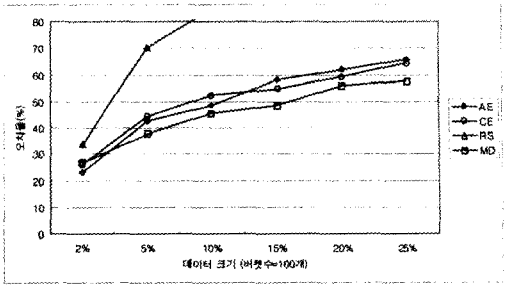


그림 14. 데이터 크기에 따른 오차율.
Fig. 14. Estimation Errors by Data Size.

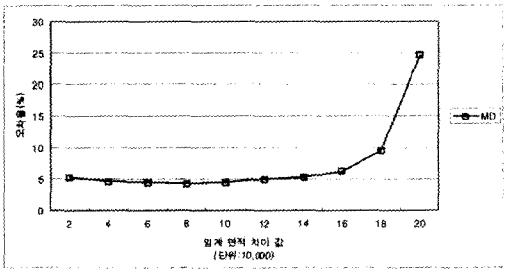


그림 15. 임계 면적 차이 값에 따른 오차율.
Fig. 15. Estimation Errors by Threshold of Area Difference.

보여주고 있다.

4. 데이터 크기에 따른 성능 평가

본 실험은 데이터 크기의 변화에 따른 각 공간 분할 방법의 성능을 보여준다. 실험에서 버켓 수를 100개로 하였고 40,000개의 임의 데이터를 5%~ 25%로 크기를 변화해 가며 성능을 측정하였다. 최대 면적 차이 분할 방법이 면적 균등 분할이나 개수 균등 분할 보다 약 10%의 성능 향상을 보였다. 데이터 크기가 커질수록 오차율이 커지는 것은 여러 분할 영역에 걸쳐 데이터가 표현되므로 실제보다 더 많은 데이터 개수가 카운트되기 때문이다.

5. 임계 면적 차이 값에 따른 성능 평가

최대 면적 차이 분할 방법에서 분할 결정 인자인 임계 면적 차이 값을 변환시킬 때 최소의 오차율을 보이는 임계치를 찾는 실험이다.

그림 15에서 임계 면적 차이 값을 20,000에서 140,000까지로 했을 때에는 성능상의 변화가 거의 나타나지 않지만, 14,000이상인 160,000부터 서서히 증가하다가 18,000부터 오차율이 급격히 증가함을 알 수 있다. 이는 임계 면적 차이 값을 크게 정하면 분할 영역의 개수가 줄어들게 되어 데이터 분포를 정확하게 반영하지 못하

기 때문이다. 만약 저장 공간의 오버 헤드를 줄이고자 한다면, 임계 면적 차이 값을 140,000이나 160,000로 정하는 것이 효과적이라고 할 수 있다.

VI. 결 론

본 논문은 질의 결과 크기를 얻기 위한 공간 분할 방법으로 데이터의 범위와 빈도수를 동시에 고려한 최대 면적 차이 공간 분할 방법을 제안하였다. 이 방법은 1차원 데이터베이스상의 질의 결과 크기 추정의 성능이 가장 우수하다고 보고된 바 있는 최대 면적 차이 분할 방법을 2차원 공간상에 적용하였다. 먼저 1차원의 분할을 구하기 위해서 x축 상의 분할과 y축 상의 분할을 시도한 후, 임계 면적 차이 값을 기준으로 분할된 영역을 교차하여 최종적인 2차원 공간 분할 영역을 구하였다.

제안한 방법과 면적 균등, 개수 균등, 공간 인덱스에 기초한 분할 방법 등 세 가지 분할 방법을 실제 데이터와 인위 데이터를 사용하여 질의 크기, 버켓수, 데이터 크기의 변화에 대하여 질의 결과 추정에 대한 정확도를 비교해 본 결과, 제안한 방법이 가장 좋은 성능을 보였다. 이것은 데이터 범위를 일정하게 분할하는 방법과 한 버켓내의 빈도수를 일정하게 하는 기존의 방법을 극복한 방법으로 두 가지의 영향을 모두 고려하여 성능의 향상을 가져왔다.

참 고 문 헌

[1] Guting, R. H., "An Introduction to Spatial Database Systems," The VLDB Journal, Vol. 3, No. 4, pp. 357-400, October 1994.

[2] ARC/INFO, "Understaning GIS - the ARC/INFO Method," ARC/INFO, 1993.

[3] Ubell, M., "The Mantage Extensible Datablade Architecture," Proc. SIGMOD Intl. Conf. on Management of Data, 1994.

[4] Selinger, P., M.M. Astrahan, D.D. Chamberin, R.A. Lorie, T.G. Price, "Access Path Selection in a Relational Database Mangement System," Proc. SIGMOD Intl. Conf. on Management of Data, pp. 23-34, 1979.

[5] Poosala, V., Y. Ioannidis, P. Haas, and E. Shekida, "Improved Histogram for Selectivity Estimation of Range Predicates", Proc. SIGMOD Intl. Conf. on Management of Data, pp. 294-305, 1996.

[6] Lipton, R. J., J. F. Naughton, and D. A. Schneider, "Practical Selectivity Estimation through Adaptive Sampling," Proc. SIGMOD Intl. Conf. on Management of Data, pp. 1-11, 1990.

[7] Chen, C. M., and N. Roussopoulos, "Adaptive Selectivity Estimation using Query Feedback," Proc. SIGMOD Intl. Conf. on Management of Data, pp. 161-172, 1994.

[8] Acharya, S., V. Poosala, and S. Ramaswamy. "Selectivity Estimation in Spatial Databases", Proc. SIGMOD Intl. Conf. on Management of Data, 1999.

[9] Poosala, V., Y. Ioannidis, P. Haas, and E. Shekita, "Improved Histograms for Selectivity Estimation of Range Predicates," Proc. SIGMOD Intl. Conf. on Management of Data, 1996.

[10] Guttman, A, "R-trees: A Dynamic Index Structure for Spatial Indexing," Proc. SIGMOD Intl. Conf. on Management of Data, 1985.

[11] Beckman, N., H-P Kriegel, R. Schneider, and B. Seeger, "The R*-Trees: An Efficient and Robust Access Method for Points and Rectangles," Proc. SIGMOD Intl. Conf. on Management of Data, pp. 322-331, 1990.

[12] Tiger/line files(tm), 1992 Technical Documentation, Technical Report, U. S. Bureau of the Census, 1992.

저 자 소 개

황 환 규(정회원)

제35권 C편 제 3호 참조