Web GIS를 위한 주기억 장치 기반 공간 색인

김진덕 · 진교홍

동의대학교

Spatial Index based on Main Memory for Web GIS

Jin-deog Kim · Kyo-hong Jin

Dongeui University

E-mail : {jdk,khjin}@dongeui.ac.kr

요 약

최근 메모리 가격의 하락과 함께 주기억 장치 기반 데이터베이스 기술의 필요성이 대두되고 있다. 또한 불특정 다수가 인터넷 환경을 통해 이용하는 Web GIS(Geographical Information System)는 데이터의 변경보다는 분석을 위한 데이터 검색이 많으며 고속의 처리를 요구한다. 그러므로 Web GIS를 위한 데이터 저장 하부구조로서 디스크를 기반으로 하는 것보다 메모리를 기반으로 함이 바람직하다.

이 논문에서는 Web GIS에서 널리 사용되고 있는 다차원 공간 데이터를 주기억 장치에 보다 적은 저장 용량으로 표현할 수 있는 방법으로서 상대 좌표값과 MBR(Minimum Bounding Rectangle)의 크기를 이용한 데이터 표현법을 제안한다. 그리고 점 질의나 영역 질의를 간단한 방법으로 처리하는 메모리 기반 공간 색인 기법을 제안한다. 실험 결과 색인의 크기와 MBR 비교 연산의 횟수 측면에서 불균일 분포 데이터에서도 좋은 성능을 보임을 알 수 있다.

ABSTRACT

The availability of the inexpensive, large main memories coupled with the demand for faster response time are bringing a new perspective to database technology. The Web GIS used by an unspecified number of general public in the internet needs high speed response time and frequent data retrieval for spatial analysis rather than data update. Therefore, it is appropriate to use main memory as a underlying storage structures for the Web GIS data.

In this paper, we propose a data representation method based on relative coordinates and the size of the MBR. The method is able to compress the spatial data widely used in the Web GIS into smaller volume of memory. We also propose a memory resident spatial index with simple mechanism for processing point and region queries. The performance test shows that the index is suitable for managing the skewed data in terms of the size of the index and the number of the MBR intersection check operations.

키워드

GIS, 메인 메모리 데이터베이스, 공간 색인, 공간 질의

Ⅰ. 서 론

지금까지 전통적인 데이터베이스 시스템에서는 데이터가 보조기억장치의 파일 시스템에 존재한다는 가정 하에 보다 빠른 자료 검색을 위해 색인[1] 및 검색 알고리즘 등이 연구되었고, 운용되고 있다. 그렇지만 이와 같은 알고리즘이나 방법론은 보조 기억 장치 자체의 속도 한계 때문에 응답 속도에서 사용자의 요구를 수용하지 못하고 있는 실정이다[2]. 그래서 보조 기억 장치를 대신하여 데이터 및 색인을 메모리에 두고 관리하는 방식을 채택한 데이터베이스 시스템은 날로 증대되는 데이터의 고속 처리에 대한 요구를 적절히 수용할 수 있는 방안으로서, 앞으로의 데이터 베이스 관리 시스템의 주된 방향이 될 것이다.

또한 최근의 데이터 베이스 응용 프로그램의 추세는 주로 Web과 연동되어 운용[2]되고 있다.

데이터의 읽기(Read)가 많고 변경(Update)은 자주 발생하지 않는 Web GIS는 대표적인 응용 사례이다. Web GIS는 불특정 다수가 Web 환경을 통해 연산을 요청하는 경우가 대부분으로서, 클라이언트의 수가 기존의 Lan 환경에 비해 월등히 많아 응답속도 측면에서 기대 이하인 경우가 많다. 그러므로 Web GIS를 위한 데이터 저장 하부 구조로서 디스크를 기반으로 하는 것 보다 메모리를 기반으로 함이 바람직하다.

그러나 지금까지 연구되어온 디스크 기반의 데이터 표현 및 색인 기법은 메모리 기반 데이터베이스 시스템에는 적절하지 않다. 다시 말해, 디스크 기반의 페이지 I/O를 위한 노드 구조와 디스크 탐색 횟수를 줄이기 위한 클러스터링 등은 메모리 기반 시스템에서는 무의미하기 때문이다. 메모리 기반 데이터 베이스 시스템은 처리 속도 향상이라는 긍정적인 측면도 있지만, 공간 효율성(Storage Efficiency)을 극대화해야 하는 난제도 포함하고 있다.

따라서 이 논문에서는 Web GIS에 사용되고 있는 다차원 공간 데이터를 주기억 장치에 최소의 저장 용량으로 표현할 수 있는 데이터 표현법과 보다 빠른 데이터 처리 및 검색을 위한 메모리 기반 공간 색인 기법에 관한 연구를 수행하고자 한다.

이 논문의 구성은 다음과 같다. 2장에서는 메모리 기반 공간 색인에 관한 관련연구를 다루고 3장에서는 메모리내의 데이터 표현법, 4장에서는 메모리 기반 공간 색인구조를 제시한다. 5장에서는 제시한 방법에 대한 성능 분석을 하고, 6장에서 결론을 맺는다.

II. 관련연구

지금까지 수많은 공간 색인에 관한 연구가 진행되어 왔다. 그러나 이들 연구의 거의 대부분의 디스크 기반 공간 색인[3,4,5]이다. 관련 연구 [3]에서는 지금까지 연구된 공간 색인 중 제일 성능이 좋은 것으로 평가되고 있는 R*-tree에 관한 연구를 수행하였다. R*-tree는 클러스터링, 균형트리, 불균일 데이터의 처리와 같은 장점이 있지만 메모리를 기반으로 한 공간 색인일 경우에는 노드 사용률의 저하, 과도한 링크 필드로 인한 자료 밀집도(Data Density)저하, 다단계 색인으로 인한 색인 자체 용량의 증가, MBR간의 겹침 등으로 적절하지 않다.

관련 연구 [4]은 그리드 파일을 제시한 것으로 점질의일 경우 두 번의 디스크 페이지 접근으로 처리가 가능하다. 이와 같이 그리드 파일은 그 구조가 간단하며, 정형적인 구조의 배열을 사용하므로 메모리 구조에 쉽게 적용하기 쉽고 색인의 크기가 작지만, 불균일 데이터 분포를 다루기가 어려운 단점이 있다.

최근에는 메인 메모리 데이터베이스에서 사용할 수 있는 색인[6,7,8,9,10]에 대한 연구도 진행되고 있다. 관련 연구 [7]에서 연구된 T-tree는 메모리에 존재하는 문자 데이터를 검색하기 위한 메모리 기반 1차원 색인이다. T-tree는 AVL 트리와 B-tree의 장점을 취한 색인으로서, 이진 트리의 구조를 유지하면서 각 노드는 다중 값(multi value)을 가져 링크 필드를 최소화하고 데이터 밀집도를 높여 메모리의 공간 효율성을 높였다. 그러나 이 연구는 문자 데이터에 적용되는 색인으로서, 다차원의 공간 데이터에는 적용될 수 없다.

관련 연구 [6]에서는 메모리 기반 색인인 CR-tree를 제시하고 캐쉬를 보다 효율적으로 활용하기 위한 방안을 제시하고 있다. CR-tree는 메모리의 공간 효율성을 높이기 위해 노드 구조를 압축하여 같은 용량의 캐쉬에 보다 많은 노드가 적재되고, 이로 인해 캐쉬의 적중율(hit ratio)을 높일 수 있다는 원리를 이용하고 있다. 그러나 R-tree를 사용하고 있기 때문에 앞에서 살펴본 문제점을 그대로 내포하고 있다.

III. 공간 데이터의 메모리 표현법

GIS 자료가 지닌 특성은 도형자료의 양이 일반적으로 대단히 크다는 것이다. 보조기억장치 보다 상대적으로 크기가 작은 메모리 기반 데이터 베이스 시스템에서는 압축 기법이 필요하다. 이 장에서는 상대 좌표값 및 MBR의 크기를 이용한 데이터 압축 방법을 제시해 공간 색인의 표현시 MBR의 양을 줄이고자 한다.

(1) 상대좌표값을 이용한 데이터의 압축

객체의 MBR 표현을 위해 좌하점과 우상점의 두점을 이용한다. 공간 데이터는 일반적으로 넓은 지역을 대상으로 하기 때문에 공간 객체의 좌표값은 매우 큰 값을 가지므로 MBR은 16byte(각 좌표당 4byte)의 저장공간이 필요하다. 그렇지만 그림 1처럼 기준값을 이용하여 상대 좌표값으로 표현하면 MBR은 8byte로 표현될 수 있다. 따라서 정보 손실은 전혀 발생하지 않으면서 데이터를 압축할 수 있다. 그러나 정보 처리시에 추가 연산이 필요하다는 단점이 있다. 그렇지만 모든 데이터가 메모리에 존재하므로, 복구 연산 시간은 디스크 접근 시간을 상쇄하고도 남음이 있다.

(2) MBR의 크기를 고려한 압축

이 논문에서 제시하는 MBR 크기를 고려한 압축법은 아래 식과 같이 MBR의 좌하점은 절대 좌표계와 같이 표현하되 우상점 대신 MBR의 크기로 표현하는 것이다. 이 방법은 MBR의 정확도는 적절히 유지하면서 데이터의 양을 줄이는 방법이다.

MBR=X좌표(2)+Y좌표(2)+X크기(1)+Y크기(1)

예를 들어 그림 2에서 객체 A는 크기가 X축으로 121이고 Y축으로 102이다. 객체 B는 X축으로 418, Y축으로 379이다. 그러므로 MBR의 크기값을 1byte로 표현하고자 할 경우 객체 A의 MBR

은 (119,121,121,102)이다. 앞의 두값은 좌하점을 나타내는 것이고, 뒤의 두값은 MBR의 X,Y의 크기값이다. 같은 방법으로 객체 B의 MBR은 (400, 173, 418, 379)이다. 그러나 MBR의 크기값을 1byte로 표현해야 하기 때문에 418과 379는 그 범위를 벗어난다. 그래서 크기값을 정량화해서 표현한다. 즉, 데이터 공간을 n 등분한다. 이 때 (데이터 공간/n)=δ가 정량화의 기본단위이다. 그러므로 255-n값이 기준치가 되어 MBR의 크기값이 기준치 이하일 때는 그 값을 그대로 표현하고 기준치 이상일 경우에는 (MBR의 크기값/δ+기준치) 값이 MBR의 크기값으로 저장된다. 따라서 n이 50이라고 가정하면 기준치는 205이고 정량화의 기본단위가 20이 되어 그림 2에서 객체 B의 MBR은 (400, 173, 226, 224)로 표현된다.

이와 같은 차이값 표현 중 정량화 방법은 MBR의 커짐 현상이 발생하고 이는 false hits의 증가를 초래한다. 그러나 대부분의 공간 객체의 크기는 일정 기준치 이하일 경우가 대부분이다. 그와 같은 경우에는 값의 오차가 전혀 없다.

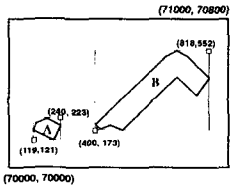

그림1. 상대 좌표값

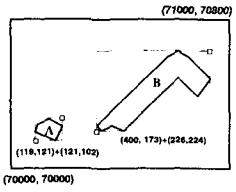

그림2. MBR 크기이용법

IV. 메모리 기반 공간 색인

Web GIS에 적합한 메모리 기반 공간 색인은 데이터의 인접성을 고려할 필요가 없다. 그리고 검색 위주의 데이터 처리에서는 각 노드의 빈영역을 두지 않아도 되며, 이에 따라 공간 효율성을 높일 수 있다. 이 논문에서는 상대적으로 작은 용량의 메모리를 고려하고, 고속처리에 적합한 단순한 공간 색인 구조이며, 손쉬운 대규모 로딩(Easy Bulk Loading)이 가능한 2차원 다중 해쉬에 의한 공간색인구조를 제안한다.

이 색인 구조는 각 객체의 X 값과 Y 값을 기준으로 각각 해쉬하여 주어진 해쉬 버켓에 할당한다. 해쉬 함수는 다음과 같다. 해쉬 테이블은 2차원 구조이며, 오버플로우가 발생했을 경우 2차 해쉬을 수행하여 하위 해쉬 버켓을 결정한다.

$H_{x1}(x) = int((x - X_{min})/(X_{max}-X_{min})*N_x)$

단, N_x는 해쉬 테이블의 X축 버켓의 수

$H_{y1}(y) = int((y - Y_{min})/(Y_{max}-Y_{min})*N_y)$

단, N_y는 해쉬 테이블의 Y축 버켓의 수

그러나 이 방법은 데이터 분포가 불균일 경우에는 오버플로우 현상이 자주 발생하게 된다. 그래서 오버플로우 처리를 위해 다음과 같은 방법을 택하고 있다. 오버플로우가 발생한 버켓에 대해서는 2차 해쉬(H_{x2}, H_{y2})을 수행하여 하나의 버켓에 들어가는 객체의 수를 일정하게 유지하도록 하였다. 그러므로 불균일 분포의 데이터일 경우에도 점 질의와 영역 질의에 큰 손실없이 적절한 응답시간을 보일 것으로 판단된다. 2차 해쉬 함수는 1차 해쉬 함수와 동일하지만 min, max값이 전체 데이터 공간이 아니라 오버플로우가 발생한 버켓의 최저값과 최고값으로 대치된다. 이와 같은 해쉬은 오버플로우가 발생하지 않을 때까지 하위로 계속 진행된다.

그렇지만 공간 데이터의 특성상 심한 불균일 분포(skewed data)일 경우가 많으므로 오버플로우가 그 만큼 자주 발생한다. 그래서 오버플로우 발생 빈도를 줄이기 위해 하나의 오버플로우 버켓의 용량을 유동적으로 변화시키면서 해쉬을 처리한다. 즉, 해쉬 버켓의 기본 용량이 100개의 객체를 담을 수 있다면, 그 용량의 1.5배까지는 오버플로우로 인정하지 않고 처리한다는 것이다.

이것이 가능한 이유는 디스크 기반 공간 색인일 경우에는 페이지 단위로 I/O를 하기 때문에 해쉬 버켓의 용량이 페이지의 크기와 밀접한 관련이 있지만, 메모리 기반 공간 색인은 그러한 것을 무시할 수 있기 때문이다. 그리고 이 논문의 환경인 Web GIS가 읽기 위주의 응용으로서 변경이 자주 발생하지 않기 때문에 주기적인 변경작업을 수행하는 것은 다소의 시간 소요를 허용할 수 있기 때문이다. 그리고 4장에서 제시한 MBR 압축 기법을 공간 색인시에 그대로 적용할 수 있기 때문에 공간 색인의 양이 줄어들게 된다.

이 공간 색인은 점 질의(Point Query)에 대해 아주 간단한 해쉬으로 원하는 데이터를 취할 수 있는 방법이며, 특히 특정 지역을 검색(Region Query)하는 Web GIS에서 Bulk Loading연산이 직관적으로 처리될 수 있는 공간 색인 구조이다. 또한 비용이 매우 큰 공간 조인(Spatial Join)에 대해서도 공간 분할이 정규적으로 이루어지고 단일 조인(Single Join) 현상을 보임으로 검색 영역이 줄어드는 장점이 있다.

V. 분석

(1) 실험 평가 데이터

실험 평가 데이터는 현재 공간 데이터베이스 연산의 벤치마크 데이터로 널리 이용되고 있는 Sequoia 데이터[11]이다. 이 데이터는 GIRAS Land use/ Land cover를 표현한 다각형의 집합이며, 원래의 데이터는 총 38개의 레이어로 구성된다. 공간 데이터는 불균일 분포 상태이다. 표 1에서 Sequoia 데이터의 특성(객체의 숫자, 각 객체의 점의 개수)을 정리하였다.

표 1. Sequoia 데이터의 특성

# of obj.	min(#of points)	max(#of points)
79607	3	8400

(2) MBR 표현법에 따른 공간 색인 크기

일반적으로 공간 데이터의 공간 색인 과정에서 객체의 MBR이 차지하는 비율은 80%이상이다. 따라서 MBR의 단축 표현법은 전체적으로 공간 색인의 크기를 줄일 수 있는 지름길이다. 이 논문에서 제안한 공간 색인으로 Sequoia 벤치마크 데이터를 구축한 결과 표 2와 같았다. 여기서 1차 버켓 사이즈를 50으로, 2차 부터는 버켓 사이즈를 5씩 감소시켰다. 표 2에 나타난 바와 같이 이 논문에서 제안한 MBR 크기 표현법을 사용한 경우 일반적으로 사용하는 절대 좌표계를 사용했을 때의 절반 수준으로 공간 색인의 양이 줄어듦을 알 수 있었다.

표 2. 공간 색인의 크기 비교

표현법	절대좌표	상대좌표	MBR크기이용
색인크기	1522K	948K	797K

(3) R*-tree와의 비교

이 논문에서 제안한 공간 색인과 R*-tree와의 공간 색인의 크기 비교 결과는 다음과 같다. R*-tree는 노드의 크기를 1Kbyte로 하였고, 색인 구축 결과 공간 효율성은 평균 66%였으며 트리의 깊이는 4였다. 그리고 MBR은 크기를 기반으로 한 표현법을 사용하였다. 이때 R*-tree의 크기는 1957K이다. 그러므로 이 논문에서 제안한 공간 색인은 R*-tree에 비해 40%정도의 메모리 공간만을 사용함을 알 수 있다. 이는 R*-tree의 특성상 다단계로 형성이 되며, 균형 트리를 유지하기 위해 공간 효율성이 떨어지기 때문이다.

질의의 검색 성능을 의미하는 여과 단계의 MBR 비교 연산회수는 다음과 같다. 점질의일 경우 R*-tree는 평균 56회이며, 이 논문에서 제안한 공간 색인은 평균 31회로 60% 수준이었다. 결론적으로 불균일 분포를 갖는 데이터를 이용한 실험 결과 이 논문에서 제안한 공간 색인이 R*-tree와의 비교 실험에서는 공간 색인의 양과 MBR 비교 연산 횟수에서 모두 좋은 성능을 보였다.

VI. 결 론

이 논문에서는 현재 많이 응용되고 있는 Web GIS에 사용되고 있는 다차원 공간 데이터를 주기억 장치에 최소의 저장 용량으로 표현할 수 있는 데이터 표현법으로서 상대 좌표값과 MBR의 크기를 이용한 방법을 제안하고, 보다 빠른 데이터 처리 및 검색을 위한 메모리 기반 공간 색인 기법으로서 데이터를 분포상태를 감안한 2차원 다중 해쉬 기반 공간 색인 기법을 제안하였다.

이 논문에서 제안한 MBR의 크기를 이용한 방법은 그 표현량을 줄이면서 공간 데이터의 대부분을 차지하는 크지 않은 객체인 경우에는 오차가 전혀없다는 장점이 있다. 또한 2차원 다중 해쉬 기반 공간 색인 기법은 메모리 구조에 맞게 간단한 구조이면서 특정 지역 검색과 같은 질의를 보다 쉽게 수행할 수 있다. 실험 결과 MBR의 크기를 이용한 데이터 표현법은 50%정도의 데이터 감축효과가 있었고, 제안한 색인 구조는 R*-tree에 비해 색인의 크기 측면에서는 40%, MBR 비교 연산 횟수 측면에서는 60% 정도로 좋은 성능을 보여주었다. 이를 통해 R*-tree가 디스크 기반 공간 색인으로서 페이지 I/O와 클러스터링을 적절히 표현하지만 메모리 기반 시스템에서는 오히려 단점으로 작용함을 알 수 있었다.

앞으로는 현재 서비스 중인 Web GIS에 제안한 데이터 표현법과 공간 색인구조를 도입해 질의 응답 시간, 색인의 크기 등의 정량적인 실험 평가를 하고자 한다. 그리고 보다 작은 메모리 용량을 가진 PDA와 같은 시스템에도 적용해 볼 필요가 있다.

참고문헌

[1] D. Greene, *An Implementation and Performance Analysis of Spatial Data Access* , Proc. of Int. Conf. on Data Engineering, pp. 606-615, 1989

[2] A. Ailamaki, D.J. Dewitt, M.D. Hill, D.A. Wood, *DBMSs on a modern processor : where does time go?* , Proc. of Int. Conf. on VLDB, pp. 510-521, 1999

[3] N. Beckmann, H.P. Kriegel, R. Schneider, B. Seeger, *R*-tree : An Efficient and Robust Access Method for Points and Rectangles* , Proc. of Int. Conf. on ACM SIGMOD, pp. 322-331, 1990

[4] K. Hinrichs, J. Nievergelt, *The Grid File : A Data Structure designed to Support Proximity Queries on Spatial Objects* , Proc. of Int. Conf. on Graph Theoretic Concepts in Computer Science, pp. 100-113, 1983

[5] H. Lu, B.C. Ooi, *Spatial Indexing : Past and Future* , IEEE Data Engineering Bulletin, Vol. 16, No. 3, pp 16-21, 1993

[6] K.H. Kim, S.K. Cha, K.J. Kwon, *Optimizing multidimensional index trees for main memory access* , Proc. of Int. Conf. on ACM SIGMOD, 2001

[7] T.J. Lehman, M.J. Carey, *A Study of index structures for main memory database management system* , Proc. of Int. Conf. on VLDB, pp. 294-303, 1986

[8] J. Rao, K.A. Ross, *Cache conscious indexing for decision-support in main memory* , Proc. of Int. Conf. on VLDB, pp. 78-89, 1999

[9] J. Rao, K.A. Ross, *Making B+-trees cache conscious in main memory* , Proc. of Int. Conf. on ACM SIGMOD, pp. 475-486, 2000

[10] A. Shatdal, C. Kant, J.F. Naughton, *Cache conscious algorithms for relational query processing* , Proc. of Int. Conf. on VLDB, pp. 510-521, 1994

[11] M. Stonebraker, J. Frew, K. Gardels, J. Meredith, *The SEQUOIA 2000 Storage Benchmark* , Proc. of Int. Conf. on ACM SIGMOD, pp. 2-11, 1993