

R-tree 기반의 효율적인 하이브리드 공간 인덱스 구조[†]

강홍구*, 김정준*, 한기준*
 *건국대학교 컴퓨터·정보통신공학과
 e-mail:hkkang@db.konkuk.ac.kr

An Efficient Hybrid Spatial Index Structure based on the R-tree[†]

Hong-Koo Kang*, Joung-Joon Kim*, Ki-Joon Han*

*Dept of Computer & Information Communication Engineering, Konkuk University

요 약

최근 대표적인 공간 인덱스 구조인 R-tree를 기반으로 KD-tree나 Quad-tree와 같은 공간 분할 특성을 이용하여 인덱싱 성능을 향상시키기 위한 연구가 활발하다. 본 논문에서는 기존에 제시된 R-tree 기반 인덱스 구조인 SQR-tree와 PMR-tree의 특성을 결합하여 대용량 공간 데이터를 보다 효율적으로 처리하는 인덱스 구조인 MSQR-tree(Mapping-based SQR-tree)를 제시한다. SQR-tree는 Quad-tree를 확장한 SQ-tree와 각 SQ-tree 리프 노드마다 실제로 공간 객체를 저장하는 R-tree가 연계되어 있는 인덱스 구조이고, PMR-tree는 R-tree에 R-tree 리프 노드를 직접 접근할 수 있는 매핑 트리를 적용한 인덱스 구조이다. 본 논문에서 제시하는 MSQR-tree는 SQR-tree를 기본 구조로 가지고 R-tree마다 매핑 트리가 적용된 구조를 갖는다. 따라서, MSQR-tree에서는 SQR-tree와 같이 질의가 여러 R-tree에서 분산 처리되고, PMR-tree와 같이 매핑 트리를 통해 R-tree 리프 노드를 빠르게 접근할 수 있다. 마지막으로 성능 실험을 통해 MSQR-tree의 우수성을 입증하였다.

1. 서론

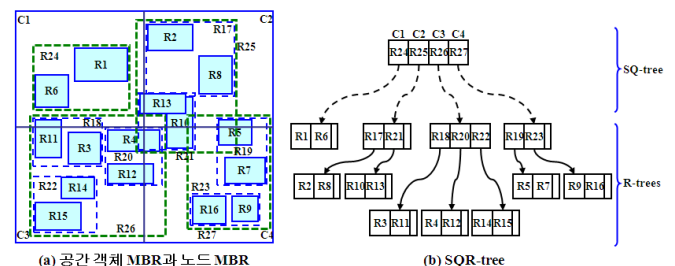
최근 R-tree[1]를 기반으로 KD-tree나 Quad-tree와 같은 공간 분할 특성을 이용하여 인덱싱 성능을 향상시키는 연구가 활발히 진행되고 있다[2]. 기존에 제시된 인덱스 구조로서 SQR-tree[3], PMR-tree[4]가 있다. SQR-tree는 Quad-tree를 확장한 SQ-tree와 각 SQ-tree 리프 노드마다 실제로 공간 객체를 저장하는 R-tree가 연계되어 있는 인덱스 구조로서 R-tree보다 삽입, 삭제 성능을 높였지만 PMR-tree에 비해 검색 성능이 크게 떨어지는 단점이 있다. 반면에 PMR-tree는 R-tree에 R-tree 리프 노드를 직접 접근할 수 있는 매핑 트리를 적용한 인덱스 구조로서 검색 성능이 우수하다.

본 논문에서는 기존에 제시된 PMR-tree와 SQR-tree의 특성을 결합하여 대용량 공간 데이터를 보다 효율적으로 처리하는 인덱스 구조인 MSQR-tree(Mapping-based SQ R-tree)를 제시한다. MSQR-tree는 SQR-tree를 기본 구조로 가지고 R-tree마다 매핑 트리가 적용된 인덱스 구조를 갖는다. 따라서, MSQR-tree는 SQR-tree와 같이 분할된 데이터 공간마다 R-tree가 존재하고 공간 객체를 여러 R-tree에 분산 저장하고 질의 영역에 해당하는 R-tree만

을 접근하면 되기 때문에 질의 처리 비용을 줄일 수 있다. 그리고 PMR-tree와 같이 R-tree 리프 노드를 직접 접근하므로 검색 성능을 크게 향상된다. 마지막으로 성능 실험을 통해 MSQR-tree의 성능이 우수함을 입증하였다.

2. 관련 연구

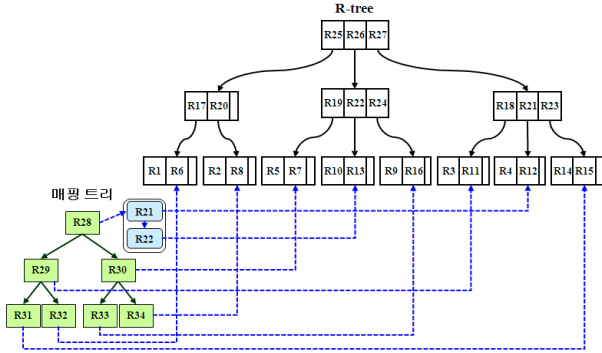
SQR-tree는 SQ-tree와 R-tree를 결합한 인덱스 구조로서, 여기서 SQ-tree는 Quad-tree 구조와 유사하지만 노드마다 자신에게 해당되는 모든 공간 객체를 포함하는 MB R, 즉 노드 MBR을 갖는다[3]. 새로운 공간 객체가 저장될 SQ-tree 노드는 중심점(Centroid)에 따라 결정되며, 해당 노드에서는 공간 객체가 포함되도록 노드 MBR이 확장된다. SQR-tree는 SQ-tree의 리프 노드마다 R-tree를 연계된 구조를 가진다. (그림 1)은 SQR-tree의 예를 보여준다.



(그림 1) SQR-tree 예

[†] 본 연구는 국토해양부 첨단도시기술개발사업 - 지능형국토정보기술혁신 사업과제의 연구비지원(07국토정보C05)에 의해 수행되었습니다.

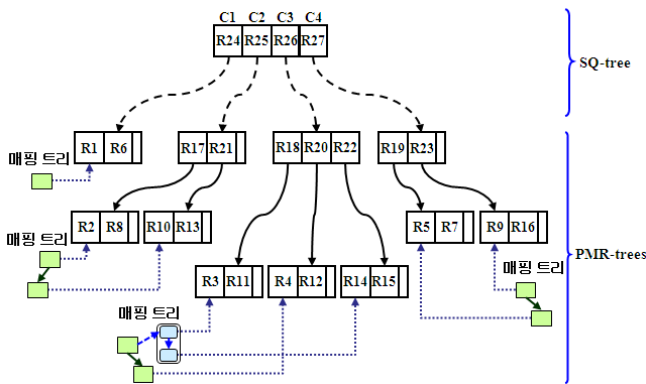
PMR-tree는 R-tree에서 트리 검색 없이 R-tree 리프 노드를 직접 접근할 수 있는 매핑 트리를 이용함으로써 검색 성능을 향상시킨 인덱스 구조이다[4]. 매핑 트리는 데이터 영역을 차원에 따라 반복적으로 분할한 파티션과 각 파티션과 대응되는 R-tree 리프 노드의 MBR과 포인터로 구성된다. (그림 2)는 PMR-tree의 예를 보여준다.



(그림 2) PMR-tree 예

3. MSQR-tree

MSQR-tree는 SQR-tree와 PMR-tree의 특성을 결합한 인덱스 구조로서 SQR-tree의 R-tree에 대해서 매핑 트리를 적용하여 대용량 공간 데이터를 보다 효율적으로 처리하는 하이브리드 인덱스 구조이다. (그림 3)은 MSQR-tree의 전체 구조를 보여준다.



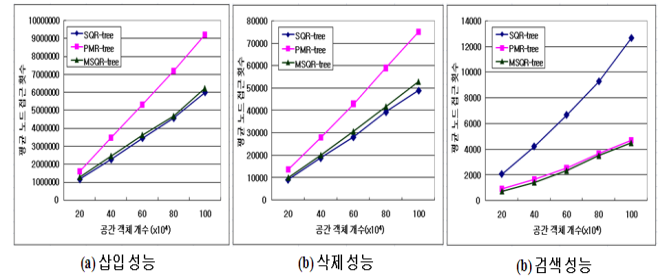
(그림 3) MSQR-tree 전체 구조

(그림 3)과 같이 MSQR-tree의 상위 구조는 SQ-tree로 구성되고, 하위 구조는 각 SQ-tree의 리프 노드마다 연계된 PMR-tree로 구성된다. MSQR-tree는 SQ-tree를 통해 데이터 공간이 하위 공간들로 분할되고, 각 하위 공간에 포함되는 공간 객체가 해당 PMR-tree에 저장된다.

MSQR-tree에서는 검색시 SQ-tree와 매핑 트리를 통해 2단계 필터링이 적용된다. 먼저 단순히 데이터 공간을 분할하는 SQ-tree에서 1단계 필터링이 수행되고, R-tree의 리프 노드를 직접 접근하게 해주는 매핑 트리에서 2단계 필터링이 수행된다. 또한, MSQR-tree에서는 질의 처리 성능 향상을 위해 비교적 단순한 구조인 SQ-tree와 매핑 트리가 메인 메모리에 구성되고, 실제로 대용량 공간 객체를 저장하는 R-tree는 디스크에 구성된다.

4. 성능 평가

실험에서는 Visual C++ 6.0으로 SQR-tree, PMR-tree, MSQR-tree를 구현하였고, 공간 객체로는 균등 분포를 가지며 한 변의 길이가 전체 데이터 영역의 한 변의 길이의 0.01%가 되는 정사각형 20만개에서 100만개를 이용하였다. 성능 비교를 위해 1,000번의 질의를 수행하는 동안 평균 노드 접근 횟수를 비교하였다. (그림 4)는 질의 처리 성능 결과를 보여준다.



(그림 4) 질의 처리 성능 결과

(그림 4)와 같이 삽입과 삭제 성능에서는 MSQR-tree가 SQR-tree 보다 평균 5% 저하되는 것으로 나타났다. 이는 MSQR-tree가 매핑 트리를 적용하므로 삽입과 삭제시 매핑 트리의 갱신 비용이 추가로 발생되기 때문이다. 반면에 검색 성능에서는 MSQR-tree가 가장 우수하였으며, SQR-tree보다 평균 250% 향상되는 것으로 나타났다.

5. 결론

본 논문에서는 기존의 R-tree 기반 인덱스 구조인 SQR-tree와 PMR-tree의 장점을 결합하여 대용량 공간 데이터의 효율적인 질의 처리를 제공하는 MSQR-tree를 제안하였다. MSQR-tree는 공간 객체가 여러 R-tree에 분산 저장되고 질의 처리시 해당되는 R-tree만을 접근하면 되기 때문에 질의 처리 성능이 향상된다. 또한, R-tree 리프 노드를 직접 접근하게 해주는 매핑 트리를 이용함으로써 검색 성능을 크게 향상시켰다. 향후 연구에서는 공간 객체가 비균등한 분포인 환경에서도 질의 처리 성능을 향상시킬 수 있도록 구조를 개선하는 것이 필요하다.

참고문헌

[1] Guttman, A., "R-trees: A Dynamic Index Structure for Spatial Searching," Proc. of the ACM SIGMOD Int. Conf. on Management of Data, 1984, pp.47-57.
 [2] Bo, H., and Qiang, W., "A Spatial Indexing Approach for High Performance Location Based Services," The Journal of Navigation, Vol.60, No.1, 2007, pp.83-93.
 [3] 강홍구, 김정준, 한기준, "SQR-tree: 대용량 공간 데이터를 위한 하이브리드 인덱스 구조," 한국공간정보시스템학회 추계학술대회, 2009, pp.85-92.
 [4] 강홍구, 김정준, 한기준, "신속한 공간 검색을 위한 파티션 매핑 기반 R-tree," GIS공동추계학술대회, 2009, pp.37-44.