

# 컨텐츠 저장 및 질의 성능 향상 방안

## 공간 질의에 관한 연구

지리 정보 시스템(Geographic Information System)은 2차원 공간상의 점, 선, 면 등의 객체들과 같은 지리적인 데이터를 효과적으로 저장, 갱신, 조정하여 지도의 형태로 데이터를 표현한다.

공간 객체를 표현하는 대표적인 색인 구조는 R-tree 계열이 있다. R-tree 계열에는 대표적으로 R+-tree와 R\*-tree등이 많이 쓰이고 있다.

R-tree는 B-tree 계열의 균형 트리이다. 모든 단말 노드는 같은 레벨에 존재한다. 트리의 중간 노드는 하위 노드들의 영역을 포함하는 최소 경계 사각형(Minimum Bounding Rectangle)으로 구성된다. 단말 노드에는 공간 객체가 저장되고 비 단말 노드에는 자식 노드에 대한 포인터와 자식 노드들의 영역을 포함하는 MBR이 저장된다. 비 단말 노드의 형태는 (pointer, MBR)로 나타낸다. 단말 노드의 형태는 공간 객체에 대한 포인터와 공간 객체의 MBR의 형태로 나타내고 (oid, MBR)로 표현한다. oid는 공간 객체가 저장된 디스크 페이지에 대한 포

인터이다. 그림 55는 R-tree의 예를 보여준다. 그림 1에서 R1 ~ R7은 R-tree의 중간 노드이며 하위 노드들의 영역을 포함하는 MBR을 나타내고 있다. R8 ~ R19는 공간 객체를 저장한 MBR이다.

R-tree에서는 비 단말 노드 MBR의 겹침이 생긴다. 따라서 R16에 대한 공간 질의를 수행할 시 R16에 겹침이 발생한 모든 노드에 대한 방문이 이루어져야 한다. 때문에 R-tree에서 검색 수행 시 접근해야 하는 노드의 수는 MBR의 겹침이 많을수록 증가하게 된다. 접근하는 노드가 많을수록 검색 효율은 떨어진다. 이러한 겹침으로 인해 증가된 검색 비용을 줄이기 위해 R\*-tree, R+-tree등의 색인 구조가 제안되었다.

R+-tree는 R-tree의 비 단말노드의 MBR 겹침으로 인한 성능 저하 문제를 해결하려고 했다. R+-tree에서는 단말 노드를 제외한 모든 비 단말 노드들의 MBR은 겹침이 발생하지 않는다. 같은 레벨에서의 겹침이 없으므로 점질의 시 트리의 레벨만큼의 노드만 접근하면 된다. 하지만 하나의 객체가

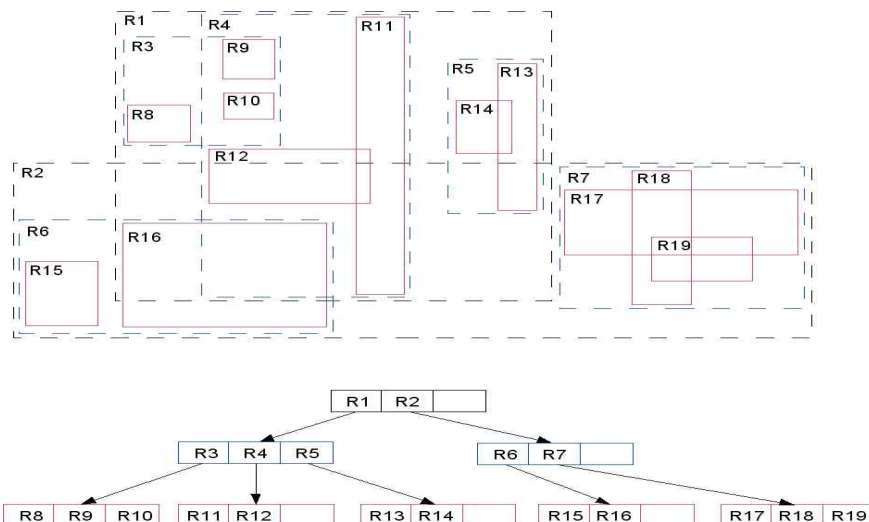


그림 1. R-Tree의 예

여러 노드에 중복 저장될 수 있고, 새로운 객체의 삽입으로 확장된 자식 레벨의 MBR을 부모 레벨의 MBR에 반영할 때 겹침 없이 MBR을 확장하지 못할 수도 있다. R+-tree는 겹침을 없애는 대신 트리의 높이를 증가시켜 공간 사용률을 증가시켰다. R+-tree는 tree 구성 시 MBR의 겹침을 제거함으로써 R-tree 보다 검색 효율을 높였다.

R-tree의 또 다른 변형인 R\*-tree는 R-tree와 기본적인 구조와 연산은 거의 동일하다. R-tree는 색인 구조를 최적화하기 위해 MBR의 넓이만을 최소화 하는 최적화 기준을 제시한 반면 R\*-tree는 다음과 같은

4가지의 최적화 기준을 적용한다.

- 1) 중간 노드의 MBR간에 겹치는 영역의 크기를 최소화 한다.
- 2) MBR의 확장 시 넓이를 최소화 한다.
- 3) MBR의 둘레의 합을 최소화 한다.
- 4) 기억 장소 이용률을 최적화 한다.

R\*-tree는 R-tree에 비해 검색 효율이 높기 때문에 검색 속도가 빠르지만 객체의 삽입이나 삭제 시 MBR 노드의 재구성이 이루어지기 때문에 업데이트 속도는 느리다.