

# 수명을 가지는 공간 객체를 R-tree 인덱스에서 효율적으로 갱신하기 위한 방법

김병준<sup>○</sup> 송인철 김명호

KAIST 전산학과

[bjkim@dbserver.kaist.ac.kr](mailto:bjkim@dbserver.kaist.ac.kr) [icsong@dbserver.kaist.ac.kr](mailto:icsong@dbserver.kaist.ac.kr) [mhkim@dbserver.kaist.ac.kr](mailto:mhkim@dbserver.kaist.ac.kr)

## An Efficient R-tree Update Method for Spatial Objects with Lifetime Constraints

Byeong Jun Kim<sup>○</sup> In Chul Song Myoung Ho Kim

Division of Computer Science, KAIST

### 1. 서론

최근 LBS시스템 중 수명을 가지는 공간 객체를 적용시킨 LBS 시스템이 주목을 받고 있다. 특히 스마트 폰의 대중화로 시스템 사용자가 늘어남에 따라 시스템 내 수명을 가지는 공간 객체 수가 증가하는데 이것은 공간 객체를 다루는 R-tree[1] 인덱스에서 빈번한 업데이트를 요청한다. 업데이트 시 큰 비용이 발생하는 R-tree는 이러한 빈번한 업데이트를 효율적으로 처리해야 하는데 이와 관련한 기존 연구[2]는 이동객체 환경에서의 빈번한 업데이트가 발생하는 환경, 단순히 빈번한 업데이트가 발생하는 환경을 가정하였다. 본 연구에서는 수명을 가진 공간객체 다수가 존재하는 환경에서 요청되는 R-tree에서의 빈번한 업데이트를 수명을 가진 객체라는 시맨틱을 적극 활용하여 효율적으로 처리하는 방법에 대해 논의한다. 이를 위해 deferred RR-tree라는 새로운 색인 구조를 제안하고 기존 연구와의 비교를 통해 그 효용성을 검증한다.

### 2. 제안하는 DRR-tree

본 논문에서 제안하는 DRR-tree(deferred RR-tree)의 기본적인 구조는 다음과 같다.

1) 디스크 R-tree와 메모리 R-tree를 유지한다. 2) 디스크 R-tree와 메모리 R-tree의 리프노드 엔트리에는 객체의 ID, 공간 좌표 값 이외에 만료시각 정보가 포함된다. 3) 메모리 버퍼가 꽉 찼을 경우 메모리 R-tree의 연산이 디스크 R-tree로 업데이트 된다. 4) 메모리에는 메모리 R-tree 이외에 삭제 시점 결정을 위한 디스크 R-tree 객체의 통계정보가 유지된다.

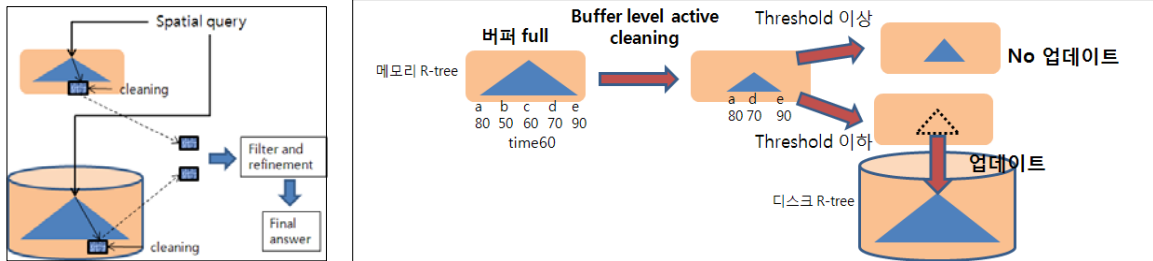
DRR-tree 에서 삭제 과정은 삭제 연산이 R-tree에 반영되는 과정을 의미하는 것으로 이를 클리닝이라고 이른다. DRR-tree의 클리닝은 미래 업데이트 성능의 향상 또는 미래 질의 응답시간 성능의 향상 등의 기준에 따라 질의가 요청되었을 때 (질의 기반 수동 클리닝), 메모리 버퍼가 꽉 찼을 때(버퍼 레벨 능동 클리닝), 디스크 R-tree에 존재하는 만료시각이 지난 객체가 많을 때(디스크 레벨 능동 클리닝), 이렇게 세 가지로 나뉜다.

질의 기반 수동 클리닝은 질의가 요청되었을 때 삭제 연산 후보를 반환된 결과에 한정해 클리닝하는 과정이다. 질의가 요청되었을 때 RR-tree와 마찬가지로 공간 질의는 디스크 R-tree, 메모리 R-tree에 요청되고 디스크 R-tree로부터 반환된 결과 값은 메모리 R-tree에 의해 반환된 결과값에 의해 필터링 및 정제된다. 그 후에 디스크 R-tree로부터 얻어진 결과 값과 메모리 R-tree로부터 얻어진 결과 값에 한정하여 만료시각이 지난 객체에 대한 삭제 연산을 수행한다.

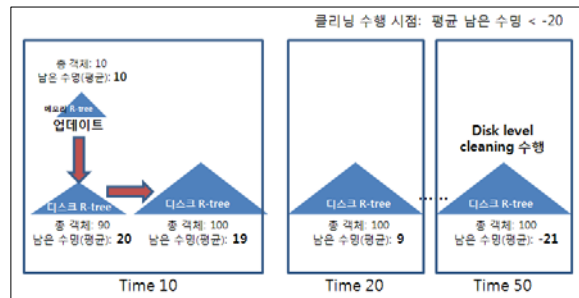
짧은 수명을 가진 객체들이 많이 존재하는 환경에서 DRR-tree는 RR-tree보다 업데이트 성능에 있어 나쁜 효율을 보여줄 수 있다. RR-tree[2]에서는 연산간의 상쇄(annihilation)가 발생하여 가용 메모리를 증가시켜주는데 반해 DRR-tree는 그렇지 않아 디스크 R-tree로의 업데이트 연산이 더 빈번해 질 수 있기 때문이다. 버퍼 레벨 능동 클리닝은 이러한 문제점을 극복하기 위해 고안된 것으로 메모리 버퍼가 꽉 차 디스크R-tree로의 업데이트가 발생할 때 메모리 R-tree의 모든 객체를 삭제 후보로 두고 클리닝을 수행하는 과정을 말한다.

본 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업의 결과임.(No. 2009-0089128).

디스크 R-tree에 존재하는 만료시각이 지난 객체들은 질의 기반 수동 클리닝 과정으로 삭제되지 않는다면 불필요한 디스크 공간을 차지해 미래 업데이트 성능의 저하 느린 질의 응답시간을 발생시킬 수 있다. 디스크 레벨 능동 클리닝은 이러한 문제점을 극복하기 위해 고안된 것으로 디스크 R-tree에 존재하는 모든 객체의 통계정보를 바탕으로 클리닝 시점을 판단하여 모든 객체를 삭제 후보로 두고 클리닝하는 과정을 말한다.



< 질의 기반 수동 클리닝의 개념도 > < 버퍼 레벨 능동 클리닝과 디스크 R-tree로의 업데이트 발생의 개념도 >



< 디스크 레벨 능동 클리닝을 위한 통계 정보 유지의 실시 예 >

### 3. 성능 평가 및 결론

표 1은 실험 환경의 초기 값이다. 버퍼 레벨 능동 클리닝의 남은 객체 수 threshold는 메모리 전체 크기로 삭제되는 객체가 없는 경우에만 디스크 R-tree로의 업데이트가 일어남을 의미한다. 표 2는, 표 1의 실험 환경에 각 분포에 따라 생성된 데이터 집합을 RR-tree와 DRR-tree에 적용시켰을 때 발생하는 한 연산당 발생하는 디스크 R-tree로의 업데이트 횟수를 측정한 결과이다.

deferred RR-tree의 기대할 수 있는 효과는 다음과 같다. 첫째, 삭제 연산을 생략해 전체 연산수를 줄임으로써 메모리 버퍼의 operation R-tree의 이용률을 높인다. 둘째, 디스크 R-tree의 삭제 연산을 최대한 뒤로 미루어 한꺼번에 처리함으로써 디스크 I/O 비용을 감소시킨다.

Parameter	Value
데이터 스페이스	100.0 * 100.0
전체 시간	0~100
객체 수	100000
Buffer level active cleaning의 남은 객체 수 threshold	메모리 크기

<표 1. 실험 환경의 초기 값>

메모리 크기	균등분포		짧은 수명이 많은 분포		
	RR-tree	Deferred RR-tree	RR-tree	Deferred RR-tree	
100	0.004945	0.002495	0.004955	0.00251	
200	0.00247	0.001245	0.002475	0.00125	
400	0.001235	0.00062	0.001235	0.000625	
800	0.000615	0.000305	0.000615	0.000305	
메모리 크기	긴 수명이 많은 분포		정규 분포		
	RR-tree	Deferred RR-tree	RR-tree	Deferred RR-tree	
	100	0.00502	0.00251	0.004995	0.002495
	200	0.00251	0.001255	0.002495	0.001245
	400	0.001255	0.000625	0.001245	0.00062
800	0.000625	0.00031	0.00062	0.00031	

(value: 디스크 R-tree로의 업데이트 / operation)

<표 2. 다양한 데이터 분포에서의 RR-tree와 DRR-tree의 성능 측정 >

### 참고문헌

1. A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 47-57, 1984.
2. Biveinis, S. Saltenis, and C.S. Jensen, "Main-Memory Operation Buffering for Efficient R-Tree Update," Proc. Int'l Conf. Very Large Data Bases (VLDB '07), pp. 591-602, 2007.