

이동 객체 환경에서 이웃 노드 정보를 활용한 2계층 공간 색인 기법

이진주^{○†}, 박용훈[†], 서동민^{††}, 유재수[†]

[†] 충북대학교 정보통신공학과, ^{††} 한국과학기술정보연구원 정보기술연구실
 {jinjulee83, yhpark}@chungbuk.ac.kr, dmseo@kisti.re.kr, yjs@chungbuk.ac.kr

A Two Layer Spatial Index Structure Using Neighbor Node Information in Moving Object Environments

Jinju Lee^{○†}, Yonghun Park[†], Dongmin Seo^{††}, Jaesoo Yoo[†][†] Dept. Information and Communication Engineering, Chungbuk National University, Korea^{††} Dept. Information Technology research, Korea Institute of Science and Technology Information, Korea

1. 서 론

위치 인식 기술과 휴대 기기의 발달로 인해 이동하는 객체를 기반으로 하는 위치 기반 서비스(Location Based Service, LBS)의 관심이 점점 증가하고 있다. 효과적인 위치 기반 서비스를 제공하기 위한 다양한 질의 처리 기법이 연구되었지만 성능 향상에 한계를 가진다. 그러므로 이를 적절히 지원하는 색인 구조가 제안되어야 한다. 이동 객체 환경을 고려한 많은 색인 기법들이 연구되어 왔다. 대부분의 색인 기법들은 대표적인 공간색인 기법인 R-tree[1] 구조를 응용하였지만 갱신 비용이 크다는 단점이 있다. 최근에는 그리드 기반의 색인 구조들이 제안되었다. 하지만 그리드 기반의 색인 구조는 저장 비용을 많이 소모하고 질의 처리 비용이 R-tree에 비해서 크다는 단점을 가진다. 본 논문에서는 두 색인 구조의 장점을 수용하여 이동 객체 환경에 적합한 효율적인 색인 구조를 제안한다.

2. 제안하는 기법

본 논문에서는 효율적인 이동 객체 색인을 위해 그리드 색인과 kd-tree의 분할 특성을 활용한 새로운 2계층 색인 구조를 제안한다. 제안하는 색인 구조는 상위 계층과 하위 계층으로 구분할 수 있으며 그 특징은 다음과 같다. 하위계층은 객체들을 관리하기 위해 전체 공간을 kd-tree의 공간 분할 기법을 이용하여 노드를 분할한다. 그리고 각 노드는 B⁺-tree의 단말 노드처럼 이웃 노드들의 정보를 유지한다. 이는 범위 질의 또는 k-최근접 질의 처리시 주어진 범위 내에 또는 특정 좌표로부터 주변의 노드를 탐색하는데 사용되어 트리 구조를 갖지 않고도 질의 처리를 가능하게 한다. 상위계층은 직접 접근을 지원하는 그리드 색인 구조를 이용하여 kd-tree의 공간 분할 기법으로 분할된 단말 노드를 색인한다. 이러한 색인 구조는 단 한번의 탐색으로 질의와 관련된 단말 노드 접근을 가능하게 한다. 그리고 B⁺-tree에서의 질의 처리 기법과 마찬가지로 단말 노드에 기록된 이웃 노드들을 탐색하면서 질의를 처리하기 때문에 트리 구조에서의 중간 노드 탐색 비용을 절감한다.

그림 1에서는 제안하는 색인 구조의 구성을 나타낸다. 제안하는 색인 구조는 객체 정보를 포함하는 단말 노드들과 그 노드들에 직접 접근을 하기 위한 그리드 색인으로 구성된다. 그림 1에서 가장 좌측의 그림은 제안하는 색인 구조의 상위계층인 그리드 색인 구조를 나타낸다. 중간 그림은 전체 그리드 영역과 겹치거나 포함되는 단말 노드들을 나타낸다. 가장 우측의 그림은 한 단말 노드에 포함된 객체들을 나타낸다.

2.1 하위계층 색인

하위계층은 객체 데이터를 포함하는 단말 노드들의 집합이다. 단말 노드들은 kd-tree의 분할기법을 이용하여 구성된다. 객체의 이동으로 인해 한 노드에 객체의 개수가 증가하여 오버플로우가 발생하면 kd-tree의 분할기법을 이용하여 분할 축을 결정하고 객체의 개수를 기준으로 노드를 분할한다. 분할은 노드의 분할 레벨에 따라 레벨이 홀수이면 세로축을 기준으로 오른쪽과 왼쪽으로, 짝수이면 가로축을 기준으로 분할한다. 노드들의 분할 레벨과 다른 노드들과의 상대적 위치를 표현하기 위해 비트로 표현된 식별자를 할당한다. 그림 2는 이러한 분할 과정을 보여준다.

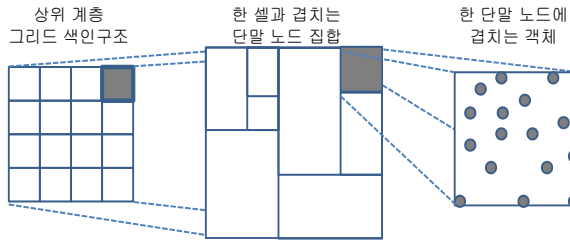


그림 1. 제안하는 색인 구조의 구성

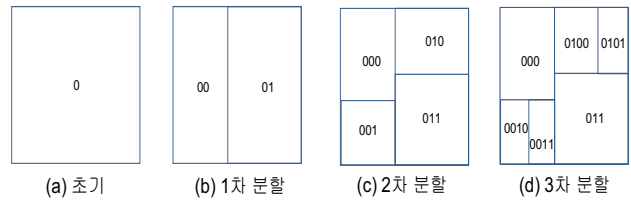


그림 2. 노드 분할 및 식별자 할당

분할 전 노드의 식별자가 '01'인 노드가 분할되면 노드가 2개로 나누어 진다. 이때 세로축 분할인 경우 왼쪽 노드의 식별자는 '010'이되고, 오른쪽 노드의 식별자는 '011'이 된다. 객체의 이동에 의한 갱신으로 노드의 분할이 지속될 경우 노드의 객체 점유율이 낮아져 질의 처리시 노드 접근 횟수가 증가하게 된다. 이를 방지하기 위해서 오버플로우가 발생하면 그 노드와 이전에 분할되었던 노드의 객체 개수를 검사하여 두 노드의 객체 개수가 임의의 임계값 이하이면 분할축을 조정하여 두 노드에 객체가 균등하게 분포시킨다. 언더플로우가 발생할 경우에는 이전에 분할되었던 노드의 객체 개수를 검사하여 두 노드의 객체 개수가 한 노드의 팬아웃 이하이면 병합을 수행한다. 만약 객체 개수가 팬아웃 이상이면 분할축을 조정한다.

하위계층의 단말 노드는 이웃하는 단말 노드들의 정보를 함께 유지하여 일단 질의 점을 포함하는 단말 노드만 찾으면 더 이상 다른 중간 노드의 탐색 없이 그 단말 노드를 중심으로 다른 단말 노드들에 접근하여 질의를 처리한다. 그래서 B⁺-tree 처럼 중간 노드의 탐색 없이 단말 노드들의 탐색이 가능하다. 그러나 kd-tree의 공간 분할 기법에 의해 노드가 분할되어 새로운 노드가 생성되면 노드들은 이웃 노드의 식별자를 변경하거나 노드 정보를 갱신해야 한다. 따라서 한 노드가 분할 또는 병합을 수행하면 자신의 이웃 노드에게 이웃 정보 갱신을 요청하고 자신의 이웃 정보도 갱신한다.

2.2 상위계층 색인

상위계층의 그리드 색인의 역할은 단말 노드를 한번에 탐색하게 하는 것이다. 일단 질의 점을 포함하는 단말 노드를 찾으면 남은 질의 처리는 그 노드의 이웃 노드 정보를 통해 이루어 진다. 때문에 질의와 연관된 노드들을 탐색하기 위해 그 노드들을 포함하는 셀들을 탐색할 필요가 없다. 셀에 기록되는 정보는 셀 영역과 겹치는 단말 노드의 정보이다. 그리드 색인에서는 주어진 질의 점을 포함하는 노드를 탐색하기 위해 각 셀에서는 그 셀의 영역과 겹치는 단말 노드 정보를 유지한다. 단말 노드의 정보는 단말 노드의 식별자와 그 노드의 영역 정보로 구성된다. 그리드 색인은 적절한 해상도를 고려해야 하지만 제안하는 기법에서의 그리드 색인은 단지 질의 점을 포함한 단말 노드를 알아내기 위한 것이고 질의 처리는 단말 노드에서 이웃 노드 정보를 통해 이루어지기 때문에 해상도가 커지더라도 저장 비용 외에는 성능에 큰 영향을 주지 않는다.

3. 결론 및 향후 연구

본 논문에서는 이동객체 환경에서 효과적인 질의 처리를 지원하는 새로운 2계층 색인구조를 제안했다. 제안하는 기법의 우수성을 평가하기 위해 대표적인 공간 색인 R-tree 그리고 최신의 그리드 구조를 이용한 공간 색인 ST²B-tree[2]와 비교를 수행하였다. 평가 결과 제안하는 색인 기법이 다른 색인 구조에 비해서 향상된 성능을 보였다. 특히 연속 질의에서는 가장 향상된 성능이 나타났고, R-tree에 비해 약 2배, ST²B-tree에 비해 약 3배의 성능 향상이 있음을 확인하였다. 향후 스카이라인 질의, Reverse k-최근접 질의와 같은 다양한 질의 처리 기법을 적용하여 성능 평가를 수행 수행할 것이다.

참고문헌

- [1] A. Gutmann, "R-trees: A dynamic index structure for spatial searching," *In Proc. ACM SIGMOD intl. conf. Management of Data*, 1984.
- [2] S. Chen, B. C. Ooi, K. L. Tan, M. A. Nascimento, "ST²B-tree: A Self-Tunable Spatio-Temporal B⁺-tree Index for Moving Objects," *In Proc. Intl. Conf. SIGMOD*, 2008.