

시공간 근접성을 고려한 TB-tree의 동적 삽입 정책

장중우[○] 임덕성 홍봉희
부산대학교
{jwchang[○], dsleem, bhong}@pusan.ac.kr

Dynamic Insertion Policy based on Spatiotemporal Proximity in the TB-tree

Jongwoo Chang[○] Duksung Lim Bonghee Hong
Dept. of Computer Engineering, Pusan National University

요 약

이동체 데이터베이스에서 대용량 궤적 정보를 효율적으로 검색하기 위해서는 색인이 필요하다. 특히 궤적을 선택하는 과정과 선택된 궤적의 일부분을 추출하는 과정으로 이루어진 복합 질의를 효율적으로 처리하기 위해서는 궤적 보존을 지원하는 TB-tree와 같은 색인 구조가 적합하다. 그러나, TB-tree는 공간적인 인접성을 고려하지 않아 비단말 노드의 검침과 사장 영역이 매우 커 영역 질의의 비용이 증가하는 문제가 있다.

이 논문에서는 복합 질의 및 영역 질의를 효율적으로 처리하기 위하여, TB-tree에서 사장 영역을 감소시킬 수 있는 최대 영역 감소 정책을 제시한다. 최대 영역 감소 정책은 공간 활용도를 최대로 유지하면서 사장 영역을 점진적으로 감소시키는 삽입 및 분할 정책으로서 TB-tree의 비단말 노드의 검침을 줄여 영역 질의의 비용을 감소시킨다.

1. 서 론

최근 GPS(Global Positioning System) 기술 및 무선 통신 기술의 발달로 인하여 위치 기반 서비스(LBS : Location Based Service)에 대한 요구가 증가되고 있다. 위치 기반 서비스에서 효과적으로 이동체의 위치를 저장하고 검색하기 위해서는 이동체 데이터베이스가 필요하다.

일정 시간 또는 이동체의 속도나 방향의 변경이 있을 때 마다 이동체의 위치를 저장할 경우 다수의 이동체로부터 획득된 궤적 정보의 양은 시간이 흐름에 따라 증가하므로 대용량의 정보가 된다. 이러한 궤적 데이터에 대하여 특정 시간 동안 주어진 장소에 있었던 이동체를 찾는 영역 질의와 영역 질의 등으로 선택된 이동체의 궤적을 추출하는 복합 질의를 효과적으로 처리할 필요가 있다.

복합질을 효율적으로 처리할 수 있는 색인으로 TB-tree가 있다[1]. TB-tree는 궤적 추출을 효과적으로 하기 위하여 단말 노드에 동일한 이동체의 궤적을 모아두는 궤적 번들 기법을 사용한다. 그러나 궤적 번들 기법으로 인하여 공간적인 특성을 고려하지 못하여 비단말 노드에서의 겹침(overlap)과 사장 영역(dead space)을 크게 만들어 영역 질의시 노드 방문 회수를 증가시키는 문제점이 있다. 반면 영역 질의를 효과적으로 처리할 수 있는 색인인 3D R-tree[2]는 궤적을 보존하지 않아 궤적 추출 비용이 크므로 복합 질의에는 효과적이지 못하다.

이 논문에서는 효율적인 복합 질의와 영역 질의를 위하여 사장 영역을 감소시킬 수 있는 삽입 및 분할 정책인 최대 영역 감소 정책을 제안하여 복합 질의에 효과적인 TB-tree에 이를 적용한다. 이 정책은 TB-tree 비단말 노드의 사장 영역과 겹침을 감소시켜 영역 질의의 비용을 감소시킬 것으로 기대된다.

이 논문의 구성은 다음과 같다. 2장에서 관련 연구를 기술하고, 3장에서는 TB-tree의 특징 및 공간적인 특성을 고려하지 못하여 발생하는 문제점을 기술한다. 4장에서는 사장 영역을 감소시킬 수 있는 최대 영역 감소 정책을 제시하고, 5장에서는 결론 및 향후 연구를 기술한다.

2. 관련연구

이동체 궤적을 색인할 수 있는 방법으로 시공간 색인인 3D R-tree[2], 궤적 색인으로 공간적인 특성에 초점을 둔 SETI[3], 그리고 궤적 보존에 초점을 맞춘 TB-tree[1] 등이 있다.

3D R-tree는 시간과 공간에 대한 색인을 분리하여 유지하지 않고 3차원 MBB(Minimum Bounding Box) 형태로 저장한다. 따라서 시간과 공간에 대한 색인이 각각 존재하는 것보다 3차원 영역 질의에 효과적이다. 그러나 R-tree[4]의 삽입 및 분할 정책은 연대순(chronology)으로 삽입되는 이동체의 특성을 고려하지 않아 공간 활용도가 저하되는 단점이 있다. 따라서 이동체가 많은 경우 색인의 크기가 크고, 동일한 이동체의 궤적이 많은 노드에 분산 저장되어 높은 궤적 추출 비용이 필요하므로 복합 질의에 효율적이지 못하다.

SETI[4]는 공간적으로 잘 나뉘어진 셀(cell)과 각 셀에 시간 색인이 있는 2단계 구조로 되어 있어 하나의 셀 안에 포함되는 작은 영역 질의는 효과적이다. 그러나 한 선분이 여러 개의 셀에 해당되는 경우는 셀 마다 중복 저장하므로 셀을 둘 이상 포함하는 영역 질의에는 질의 영역에 포함되는 셀을 모두 검색한 다음 중복을 제거하는 과정의 추가 비용을 수반한다.

TB-tree는 단말 노드에 동일한 이동체의 궤적만 저장하도록 궤적 번들 기법과 동일한 궤적이 저장된 단말 노드들을 링크로

연결시켜 제적 추출에 효과적이다. TB-tree는 R-tree의 분할 방식이 아닌, 가장 최근에 삽입된 엔트리를 새로운 노드에 두는 방식을 사용하여 높은 공간 활용도를 가진다. 그러나 제적 번들 기법으로 인하여 근접성과 같은 공간적인 특성을 고려하지 못한 문제가 있다.

3. 문제점의

이 장에서는 TB-tree의 특성과 그 특징들로 인하여 공간적인 특성을 고려하지 못하게 되어 발생하는 문제점에 대하여 기술한다.

TB-tree는 이동체의 위치를 이산적으로 추출하고, 추출된 이동체의 연속적인 두 위치를 선분으로 표현하고 연결된 선분들의 집합을 제적으로 표현한다. 또한 TB-tree는 하나의 단말 노드에 동일한 이동체의 제적만을 저장하는 제적 보존 정책을 사용하며, 동일한 제적을 저장하고 있는 단말 노드간의 링크가 존재한다. 그림 1은 TB-tree의 구조이다.

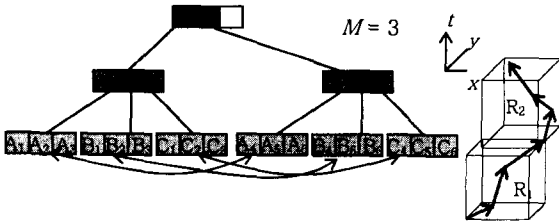


그림 1. TB-tree 구조

R-tree의 최소 영역 확장(Least Area Enlargement) 정책과 달리 TB-tree는 시공간적으로 인접한 두 이동체의 제적은 서로 다른 단말 노드에 저장될 뿐만 아니라 비단말 노드에서도 시간순으로 저장되므로 제적 간의 공간적 근접성을 고려하지 못한다. 그러나 공간 근접성을 고려하기 위하여 단말 노드를 분할하는 정책은 제적 보존 정책에 위배되어 제적 추출 비용을 증가시키게 된다.

그림 2는 비단말 노드의 팬아웃(fanout)이 3인 TB-tree에 시간대가 비슷한 이동체 A, B, C, D의 제적 선분 A₁, B₁, C₁, D₁이 순서대로 삽입되었을 때 공간 도메인만 고려한 그림이다. 그림 2(a)는 공간 근접성에 대한 고려없이 삽입되는 순서대로 색인이 생성되어 비단말 노드 R_a는 A₁, B₁, C₁을 포함한 단말 노드로 구성되고, D₁을 포함한 단말노드가 다른 비단말 노드 R_b를 구성하게 되어, R_b는 R_a에 완전히 포함되어 겹침이 심하고 R_a는 사각 영역이 매우 크다.

반면 그림 2(b)는 삽입 순서가 아닌 공간 근접성을 고려하여 생성된 경우이다. C₁이 가장 근접성이 낮으므로 A₁, B₁, D₁이 비단말 노드 R_c를 구성하게 되고, C₁이 비단말 노드 R_d를 구성하게 되어, 두 노드 R_c와 R_d의 겹침이 없고 R_c의 사각 영역도 그림 2(a)의 R_a에 비하여 작으므로 영역 질의에 효율적이다. 따라서 제적 색인은 그림 2(b)와 같이 공간 근접성을 고려해야 한다.

TB-tree는 공간적인 특성을 고려하지 않는 대신, 시간적인 특성을 고려한다. 이동체가 위치 보고 순서대로 색인에 삽입이 되면, TB-tree의 임의 노드의 각 엔트리들은 보고 시간 순서대로 정렬이 된다. 따라서 타임 슬라이스 질의와 같은 시간 정보 추출에 효과적이다.

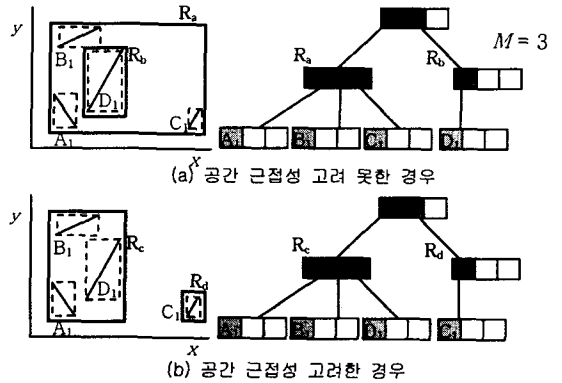


그림 2. 비단말 노드 공간 도메인의 겹침과 사각 영역

4. 삽입/분할 정책

이 장에서는 성격이 서로 다른 도메인간의 연산을 위한 방법과 TB-tree의 장점을 유지하면서 비단말 노드에서의 겹침과 사각 영역 문제를 개선할 수 있는 삽입/분할 정책과 알고리즘을 제시한다.

4.1 최대 영역 감소 정책

최대 영역 감소 정책(Maximum Area Reduction)은 비단말 노드의 사각 영역을 감소시키기 위한 삽입/분할 정책으로 새로운 이동체의 삽입이나 단말 노드의 분할을 처리하는 ChooseNode 알고리즘과 비단말 노드 분할을 처리하는 SplitNonleafNode 알고리즘으로 이루어져 있다.

첫 번째 알고리즘인 ChooseNode는 R-tree의 ChooseLeaf에 대응되는 알고리즘으로 새 엔트리가 삽입될 적당한 노드를 찾는 과정이다. 단말 노드를 레벨 0이라 하고 부모 노드는 자식 노드보다 레벨이 1 높다고 할 때, 루트 노드에서 출발하여 레벨 1에 도착할 때까지 적용하며 각 레벨 마다 새 엔트리를 삽입할 가장 적합한 서브트리(subtree)를 찾아간다. ChooseNode는 어떤 이동체가 처음 삽입되거나 단말 노드가 분할된 경우 호출되어 새로운 단말 노드의 적합한 부모 노드를 찾는다. TB-tree는 제적 보존을 위하여 이미 이동체가 들어있는 단말 노드에는 다른 이동체를 삽입할 수 없으므로, 단말 노드를 제외한 레벨 1까지만 탐색한다.

2차원의 경우 적합한 노드를 찾는 방법은 노드의 각 서브트리에 대하여 주어진 엔트리를 포함하고 대신 서브트리 내의 다른 엔트리가 새노드에 저장되었을 때 가장 큰 영역 감소를 가져올 수 있는 서브트리를 선택한다. 서브트리의 MBB의 공간 중심에 위치할수록 분할 후 영역 감소가 커질 확률이 높다.

그림 3에서와 같이 새로 추가된 엔트리 E에 대하여, 서브트리 A, B, C의 MBB에 대하여 E가 C의 MBB 중심 쪽에 위치하고 있으므로 C를 선택하면 더 큰 영역 감소가 일어날 가능성이 높다

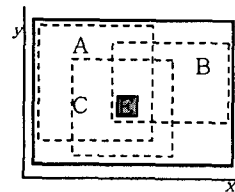


그림 3. ChooseNode의 서브트리 선택

그러나 이동체 색인에서는 연대순으로 삽입되는 이동체의 특성을 고려하여 과거 영역 엔트리가 분할되어서는 안된다. 이를 위하여 과거 영역 여부를 판단할 수 있는 Growing MBB(GM) [5]을 사용한다. 따라서 노드 내의 GM을 포함한 엔트리만 조사한다.

두 번째 알고리즘인 SplitNonleafNode는 비단말 노드 분할 시에 사장 영역을 최대 감소시킬 수 있는 엔트리를 새로운 노드에 둔다. SplitNonleafNode는 팬아웃이 M 인 노드에서 오버플로우가 발생하였을 때, 가장 사장 영역을 크게 만들 수 있는 엔트리를 분할 대상으로 선정하여, 분할 대상만 새로운 노드에 둔다. 따라서 오버플로우가 발생했던 노드에는 M 개의 엔트리가 남아 있으므로 TB-tree와 동일한 공간 활용도를 유지하면서 사장 영역을 줄일 수 있다.

그림 4는 2차원에서의 최대 영역 감소 분할 정책을 나타낸 것으로, 그림 4(a)에서 팬아웃이 3인 노드에서 엔트리 D가 삽입되어 오버플로우된 상태에서, B를 분할 대상으로 선정하여 D 대신 B가 새로운 노드에 삽입되면 오버플로우 되었던 노드의 MBB 영역의 차이가 가장 커진다. 이때 사장 영역이 가장 크게 줄어들기 때문에 그림 4(b)와 같이 B를 분할 대상으로 선정하여 새 노드에 둔다.

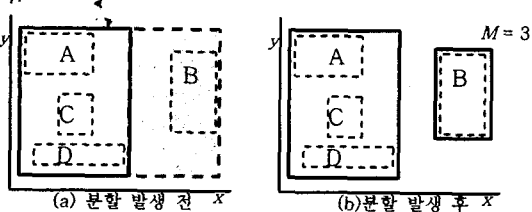


그림 4. 최대 영역 감소 분할 정책

이동체 색인에서는 SplitNonleafNode도 ChooseNode와 마찬가지로 GM을 고려하여 노드 내의 GM을 포함한 엔트리 중에서 분할 후 부피의 폭을 가장 크게 줄일 수 있는 엔트리를 새 노드에 둔다.

또한 3D R-tree는 시간 도메인과 공간 도메인과 동일하게 처리하였으나 현실적으로 시간 도메인과 공간 도메인은 단위와 성격이 다르므로 서로 다르게 취급되어야 한다. 따라서 본 논문에서는 서로 다른 성격의 두 도메인간의 연산을 하기 위하여 노드의 각 도메인을 [0, 1]로 정규화하여 도메인의 특성을 동일하게 만든 뒤 연산한다.

4.2 알고리즘

알고리즘 1은 최대 영역 감소 정책의 알고리즘과 이를 적용하기 위하여 변경된 TB-tree의 알고리즘이다.

```

Algorithm insert(N, E)
INS1 Invoke FindNode(N, E) to select a leaf node N' which has a predecessor of E
INS2 If node N' is not found,
    Create a new leaf node N'
    Invoke ChooseNode(E) to select a non-leaf node(level 1) which is parent node of N'
INS3 If N' has space
    Insert E
    else
    Create a new leaf node N'' for E
    Invoke ChooseNode(E) for choosing parent node of N''
    AdjustTree(N'')
INS4 AdjustTree(N')
    
```

```

Algorithm ChooseNode(E)
CN1 Set N to be the root
CN2 If N is level 1
    return N
    else
    Choose the GM entry by Maximum Area Reduction policy
    Set N to be the childnode pointed to by the child pointer of chosen entry an repeat from CN2
CN3

Algorithm AdjustTree(N)
AT1 If N is the root
    return
AT2 If N is overflow
    Invoke SplitNonleafNode(N)
AT3 Let P be the parent node of N, and let EN be N's entry in P
AT4 Adjust MBB(EN) so that it tightly encloses all entry boxes in N
AT5 Set N = P and repeat from AT1

Algorithm SplitNonleafNode(N)
SNN1 For each GM entry E, calculate d = volume(N) - volume(N-E)
SNN2 Select E' which makes largest d
SNN3 If rightmost node N'' whose level is same as N has space
    Insert E' to N''
    else
    Create a new non-leaf node for E'
SNN4 AdjustTree(N'')
    
```

알고리즘 1. 최대 영역 감소 정책의 알고리즘

5. 결론 및 향후 연구

본 논문에서는 기존의 복합 질의에 효율적인 TB-tree에 비단말 노드에서의 겹침과 사장 영역이 커지는 문제를 사장 영역을 줄일 수 있는 삽입 및 분할 정책을 통하여 해결하고자 한다. 이를 위하여 사장 영역을 줄일 수 있는 최대 영역 감소 정책을 제안하였다. 최대 영역 감소 정책을 통하여 TB-tree의 장점인 높은 공간 활용도를 유지하면서 비단말 노드의 사장 영역 및 겹침을 줄여 영역 질의의 비용 감소가 기대된다.

향후 연구로써 최대 영역 감소를 위하여 서브트리를 선택하는 효율적인 알고리즘에 대한 연구와 실험을 통한 성능 평가가 필요하다.

6. 참고문헌

[1] Dieter Pfoser, Christian S. Jensen, Yannis Theodoridis, "Novel Approaches to the Indexing of Moving Objects.", In Proc. of the 26th VLDB Conf, pp. 395-406, 2000.
 [2] Yannis Theodoridis, Michael Vazirgiannis, Timos Sellis, "Spatio-Temporal Indexing for Large Multimedia Applications", In International Conf. on Multimedia Computing and Systems, pp. 441-448, 1996
 [3] V. Prasad Chakka, Adam C. Everspaugh, Jignesh M. Patel, "Indexing Large Trajectory Data Sets With SETI", Proceedings of the 2003 CIDR Conference, 2003
 [4] Antonm Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching", In Proc. of ACM-SIGMOD Conf. on the Management of Data, pp. 47-57, 1984
 [5] 임택성, 전봉기, 홍봉희, 조대수, "이동체를 위한 레적 색인의 분할 정책", 개방형 지리정보시스템 학회 2002 추계 학술대회는 문집(제5권 2호), pp. 173~176, 2002