이동체의 현재 및 과거 정보 검색을 위한 시공간 색인 기법에 관한 연구

탁경현° 황진호 손진현
한양대학교 컴퓨터 공학
takkh97°@database.hanyang.ac.kr, jhhwang@cse.hanyang.ac.kr, jhson@hanyang.ac.kr

Spatio-temporal Indexing Method to retrieve current and past information of Moving Objects

Kyung Hyun Tak° Jin Ho Hwang Jin Hyun Son
Department of Computer Science and Engineering, Hanyang University in Ansa

최근 이동체의 위치정보를 이용한 많은 응용분야를 위한 연구가 활발히 이루어지고 있다. 이동체의 위치 정보는 지속적으로 변화하고 그 양 또한 많다. 기존의 정적인 공간 데이터를 위한 색인 구조로는 시공간 데이터인 이동체의 위치정보를 효율적으로 처리하기 어렵다. 따라서 본 논문에서는 이를 관리하기 위한 색인 기법을 제안하고자 한다.

이동체 데이터베이스의 질의는 영역 질의, 타임스탬프 질의, 궤적 질의, 복합 질의로 구분할 수 있다. 또한 색인 구조마다 처리할 수 있는 질의의 종류가 다르다. 표 1은 질의 종류에 따른 색인 구조에 대한 분류이다.

질의 종류	색인 구조
영역 질의	RT-Tree, LUR-Tree, Bottom-up Updates, Hashing, Duality transformation, SV-Model, PSI, Duality Transformation with the Kinetic Data Structure
타임스탬프 질의	3D R-Tree, TR-Tree, MR-Tree, HR-Tree, MV3R-Tree, TIR-Tree
궤적 질의	STR-Tree, TB-Tree, CR-Tree, MOTB-Tree, SETI, SEB-Tree, 2+3 R-Tree, 2-3 TR-Tree
복합 질의	TPR-Tree, PR-Tree, NSI, VCI R-Tree, STAR-Tree, R^{EXP}-Tree, TPR*-Tree

표 1. 질의에 따른 색인 구조

본 논문에서는 이동체의 현재와 과거 데이터를 저장 관리 하여 영역 질의, 타임스탬프 질의 그리고 궤적 질의를 처리할 수 있는 색인 기법을 제안한다. 이동체의 과거 위치 정보를 위한 색인 기법들은 크게 grid file 기반, quad-tree 기반, KDB-tree 기반, R-tree 기반으로 나뉘는데 이 논문에서는 R-tree 기반을 사용하였다.

기존의 가장 많이 알려진 R-tree 기반의 색인으로는 3D R-tree와 MV3R-tree를 들 수 있다.

3D R-tree는 시간을 하나의 공간 차원으로 간주하여 하나의 3차원 R-tree를 구성한 색인이다. 이 색인은 시간이 증가함에 따라 각 MBR(Minimum Bounding Rectangle)에 해당하는 공간적인 영역이 늘어나기 때문에 한 시점만을 고려하는 타임스탬프 질의에 대하여 처리 능력이 현저하게 떨어진다. 다만 영역 질의에서는 좋은 성능을 보이고 시간과 공간에 관한 질의 모두를 처리할 수 있으며 크기가 작다는 장점을 가지고 있다[1].

MV3R-tree는 MVR-tree의 말단 노드에 보조 3D R-tree를 결합한 색인 구조이다. 보조 3D R-tree를 이용하여 타임스탬프 질의를 처리하고 MVR-tree를 이용하여 영역질의를 모두 처리할 수 있다는 장점을 가지고 있다. 그러나 데이터를 입력 시에 MVR-tree와 보조 3D R-tree를 동시에 갱신해 주어야 하기 때문에 갱신에 대한 오버헤드가 3D R-tree보다 두 배 정도 크고 색인의 크기는 Version split으로 인해 중복된 데이터를 발생하여 1.5배정도 크다는 단점을 가지고 있다[2].

이동체에 대하여 색인 구조를 설계 시 고려할 사항으로 색인의 크기, 노드들 간의 중첩과 사장 공간,

update 비용, 처리할 수 있는 질의 종류, 비정규 분포 데이터 처리 등을 들 수 있다. 본 논문에서 제안하는 MVH R-tree는 이러한 사항들을 고려하여 설계 되었다.

종류	노드 및 페이지 형태
중간 노드	$<ptr_{parent}, T_{start}, T_{end}, (ptr_1, MBR_1), \ldots, (ptr_n, MBR_n)>$
말단 노드	$<ptr_{parent}, T_{start}, T_{end}, version\_split, (O\_ID_1, ptr_{page}, MBR_1), \ldots, (O\_ID_n, ptr_{page}, MBR_n)>$
페이지	$<ptr_{leaf}, T_{start\_page}, T_{end\_page}, ptr_{prev}, ptr_{next}, (T_{start}, T_{end}, x_1, y_1), \ldots, (T_{start}, T_{end}, x_n, y_n)>$

표 2. 노드 및 페이지의 형태

표 2는 본 색인 구조의 각 노드 형태를 나타낸다. 모든 노드(Page 포함)는 각 노드가 만들어진 시점, 끝나는 시점, 부모 노드에 대한 포인터와 각 객체에 대한 레코드들을 가지고 있으며, 페이지에는 이전과 이후의 historical data를 갖고 있는 page에 대한 포인터를 갖고 있으며 페이지안의 레코드는 실제 객체가 한자리에 머물렀던 시간정보와 위치 정보를 나타낸다. 또한 중간노드와 말단 노드의 레코드에는 객체의 위치 관리를 위해 MBR이 사용되었으며 말단노드의 레코드에는 객체를 구분하기 위한 객체 구분자(O_ID)가 저장된다.

본 색인 구조는 R-tree 기반으로 R-tree와 유사한 특성들을 따르고 있다. 첫째, 루트를 제외한 모든 노드는 살아있는 m과 M 사이의 레코드를 가지고 있어야 한다. M은 한 노드가 가질 수 있는 최대 살아있는 레코드 수, m은 M의 1/2 보다 작거나 같은 값을 말한다. 둘째, 객체들을 그룹화 시키기 위하여 MBR을 사용하고, 셋째, 모든 말단 노드들은 같은 level 상에 존재한다. 차이가 나는 특성으로는 첫째, 말단 노드의 각 record는 page를 하나씩 가지고 있고, 둘째, key split 뿐만 아니라 version split도 사용한다. 셋째, Hash table을 사용하여 page를 손쉽게 찾도록 한다. page는 객체의 historical data를 저장하는 공간이고, version split이란 과거 시점의 위치정보를 따로 관리하기 위해 특정 시점에 노드를 분할하여 새로운 historical tree에 저장하는 방법을 말한다. 그림 2.은 MVH R-tree의 구조이다.

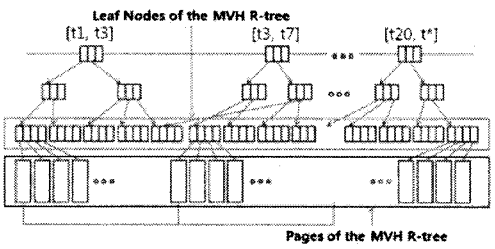

그림 5. MVH R-tree의 구조

본 색인 구조는 가능한 하나의 tree에 historical data를 저장하기 위하여 page를 사용한다. 또한 사라진 객체에 대한 정보 일지라도 version split이 일어나기 전에는 삭제 하지 않는다. 따라서 계산 양을 줄일 수 있다. 또한 노드 간의 중첩이 발생할 경우 Overlap Avoidance Algorithm을 통하여 중첩을 최대한 줄여 질의 처리를 효율적으로 할 수 있도록 하였고, 중첩을 피할 수 없을 경우에 한해서만 version split을 통해 새로운 tree를 생성한다. 비록 새로운 tree를 생성하지만 기존에 변화되지 않은 다른 노드들은 참조를 하여 중복된 데이터를 줄인다. 또한 hash table을 사용하여 빠르게 객체의 해당 page를 찾을 수 있도록 하였기 때문에 update시 시간을 절약 할 수 있다. 또한 version split으로 인하여 분할된 historical data를 저장하고 있는 page 들은 서로 참조를 하도록 하였다. 따라서 궤적 질의에 대하여 처리가 용이 하다.

본 색인은 영역 질의, 타임스템프 질의 그리고 궤적 질의에 대하여 모두 처리 할 수 있다. Key split과 강제 합병의 횟수는 MV3R-tree 보다 자주 일어나지만 version split의 횟수를 현저히 줄임으로 update 비용을 줄였다. 또한 hash table 과 page로 인하여 크기가 증가하지만 MV3R-tree와 비교하여 발생하는 노드의 수가 적기 때문에 오히려 보다 적은 공간을 차지한다.

Reference

[1] Y. Theodoridis, M. Vazirgiannis, and T. Sellis. Spatio-Temporal Indexing for Large Multimedia Applications. In Proc. of the IEEE Conference on Multimedia Computing and Systems, ICMCS, June 1996.

[2] Tao, Y., Papadias, D. The MV3R-tree: A Spatio-Temporal Access Method for Timestamp and Interval Queries. VLDB, 2001