

3차원 R-트리를 이용한 이동체 색인에 관한 연구

전 봉 기*

A Study on Indexing Moving Objects using the 3D R-tree

Bong-Gi Jun *

요 약

이동체 데이터베이스는 연속적으로 이동하는 이동체의 위치와 궤적을 검색하기 위한 데이터베이스 질의를 효과적으로 처리해야 한다. 질의 처리의 성능 향상을 위하여, 연속적으로 이동하는 이동체를 위한 효과적인 색인 기법이 필요하다. 3D R-tree와 TB-tree는 현재 위치에 대한 영역 질의를 처리할 수 없다. 현재와 과거 위치에 대한 영역 질의를 처리하기 위하여, 기존의 3D R-tree를 태그 now를 가지도록 수정하였다. 대부분의 시공간 색인 구조들은 이동체의 과거 위치에 대한 영역 질의를 효과적으로 처리할 수 없다. 이와 같은 쟁점을 설명하기 위하여 기존의 3D R-tree를 기반으로 하는 TA3DR-tree이라 불리는 색인 방법을 제안한다. TA3DR-tree는 기존의 3D R-tree와 TB-tree와의 성능 평가의 우수한 성능 결과를 보였다.

Abstract

Moving-objects databases should efficiently support database queries that refer to the trajectories and positions of continuously moving objects. To improve the performance of these queries, an efficient indexing scheme for continuously moving objects is required. To my knowledge, range queries on current positions cannot be handled by the 3D R-tree and the TB-tree. In order to handle range queries on current and past positions, I modified the original 3D R-tree to keep the now tags. Most of spatio-temporal index structures suffer from the fact that they cannot efficiently process range queries past positions of moving objects. To address this issue, we propose an access method, called the Tagged Adaptive 3DR-tree (or just TA3DR-tree), which is based on the original 3D R-tree method. The results of our extensive experiments show that the Tagged Adaptive 3DR-tree outperforms the original 3D R-tree and the TB-tree typically by a big margin.

▶ Keyword : 이동체 데이터베이스(Moving Objects Databases), 지리정보시스템(Geographic Information System), 시공간 색인(Spatio-temporal Indexing)

• 제1저자 : 전봉기
• 접수일 : 2005.06.03, 심사완료일 : 2005.08.10
* 신라대학교 컴퓨터공학부 전임강사

I. 서론

위치 기반 서비스(LBS, Location-Based Service)는 무선 통신 서비스의 핵심 응용으로 대두되고 있다. 유비쿼터스(ubiquitous) 네트워크 시대의 도래와 함께, 장치의 이동성을 기반으로 하는 다양한 응용 서비스 개발 및 위치 기반 서비스 플랫폼 개발이 더욱 필요하게 되었다[1]. 연속적으로 이동하는 이동체의 위치를 검색하는 것은 위치 기반 서비스에서는 중요한 응용 중의 하나이다. LBS 응용의 개발을 위해서는 시간의 흐름에 따라 증가하는 이동체의 위치 정보를 효과적으로 관리하고, 빠른 검색을 제공하는 이동체 색인의 개발이 필요하다.

3D R-tree[2]는 이동체의 보고 위치만을 저장하기 때문에 이동체의 현재 위치를 검색할 수 없다[3]. 이동체의 이동 경로인 궤적은 선분으로 모델링되며, 이 선분은 2개의 점으로 구성된다. 시각 t_i 에 이전 위치 (x_i, y_i) 가 보고되었고, 시각 t_j ($t_i < t_j$)에 새로운 위치 (x_j, y_j) 가 보고되었다. 시각 t_j 에 선분 $\langle (x_i, y_i, t_i), (x_j, y_j, t_j) \rangle$ 는 3D R-tree에 삽입된다. 현재 위치인 (x_j, y_j, t_j) 는 새로운 위치 보고가 되기 전까지 UC(Until Changed) 상태이다. 이 논문에서는 이동체의 현재 위치를 이동체가 가장 최근에 보고한 위치로 가정한다. 시간 간격 $[t_i, \text{now}]$ 동안에 데이터베이스에서는 이동체의 이동 위치를 알 수 없다. 이 논문에서는 3D R-tree[2], TB-tree[4] 등과 같은 색인에서 now를 포함하는 시공간 질의를 처리할 수 없는 문제점을 해결하였다.

기존의 이동체의 색인 구조 연구에서는 다음과 같은 이동체의 이동 특징을 고려하고 있지 않는 문제가 있다. 첫째, 이동체들의 궤적은 겹침이 심하고, 사각 공간(dead space)이 크다는 문제가 있다. 색인에서 겹침은 다중 경로를 유발하여 검색 성능을 저하시킨다. 둘째, 선형적으로 증가하는 시간 도메인에서 데이터의 삽입 순서를 고려하고 있지 않다. 이동체의 위치 정보는 현재 시각을 기준으로 보고하기 때문에, 저장된 데이터 보다 큰 시간 값을 가진다. 시간 축으로 분할된 과거 노드는 추가 삽입이 없다. 셋째, 이동체의 분포는 시간에 따라 동적으로 변한다. 이동체 위치 데이터의 삽입 패턴은 시간에 따라 변하기 때문에, 동적인 색인 구조가 필요하다.

이 논문에서 제안하는 TA3DR-tree(Tagged Adaptive 3DR-tree)는 이동체의 이동 특징을 고려하여 다음과 같은 정책을 이용하여 질의 성능을 개선 하였다. 첫째, 이동체들의 색인은 노드간의 겹침을 최소화하게 하는 구조를 가져야 한다. 노드간의 겹침을 조사하여 겹침이 심한 노드들을 심한 겹침(High Overlap) 노드로 규정하고, 강제 합병 후 재분할(re-split)을 수행하여 겹침을 최소화 하였다. 둘째, 노드간의 심한 겹침을 유발하는 선분을 긴 선분(Big Line Segments)으로 규정하여 절단(Clipping)하여 겹침을 최소화 하였다. 셋째, 이동체의 현재 및 과거 위치 검색을 위하여 지식 노드가 UC(Until Changed) 페이지인지를 표시하는 태그 now를 유지하도록 기존의 3D R-tree의 자료구조를 수정하였다.

실험 평가를 위해 이동체 궤적의 영역 질의 성능이 우수한 3D R-tree[2]와 TB-tree[4]를 C++로 구현하여 실험하였다. 제안하는 TA3DR-tree는 강제 합병 정책과 긴 선분 절단 정책으로 인하여 단말 노드간의 겹침이 작기 때문에 3D R-tree와 TB-tree 보다 영역 질의의 성능이 뛰어나다. 실험 데이터에 따라 차이가 있으나 TA3DR-tree는 3D R-tree 보다 10~30% 정도의 성능 향상이 있었다. TB-tree와의 성능 평가에서도 City Simulator로 생성된 데이터의 20% 영역 질의를 제외하면 20~30% 정도의 성능 향상이 있었다.

이 논문의 구성은 다음과 같다. 2장에서는 관련 연구들을 소개하고, 3장에서는 TA3DR-tree의 위치 모델링 방법과 색인의 자료구조를 소개한다. 4장에서는 강제 합병 정책과 긴 선분의 절단 정책을 소개하고 분석한다. 5장에서는 실험을 수행하여 TA3DR-tree의 성능을 비교하였다. 마지막으로 6장에서 결론 및 향후 연구를 기술한다.

II. 관련 연구

이동체의 모델링은 이동체의 위치가 시간에 따라 연속적으로 변하기 때문에 시공간 모델링에서 이를 연속적 모델과 이산적 모델로 구분한다[6, 7]. 연속적 모델은 이동체를 무한개의 점 집합으로 표현한다. 이를 3차원의 연속적 곡선으로 표현할 수 있으며 벡터를 기반으로 한다. 반면 이산적 모델은 이동체를 유한개의 점 집합으로 표현하는 것으로 3

차원 도메인에서 다중선(polyline)으로 표현한다. 다중선은 선분들의 집합이다[11,13]. 이 논문은 이산적 모델로 이동체의 위치 정보를 모델링하고, 이동체의 궤적(trajjectory)을 3차원 선분들의 집합으로 정의한다.

이동체 데이터베이스의 질의 종류는 크게 영역 질의, 타임슬라이스(time-slice) 질의, 궤적 질의로 나누어진다[4]. 영역 질의는 주어진 시간 간격 동안에 공간 윈도우(spatial window)에 속하는 이동체들을 검색하는 3차원 질의이다. 타임슬라이스 질의는 특정 시간에 주어진 공간 윈도우에 속하는 이동체들을 검색하는 질의이다. 궤적 질의는 특정한 이동체의 궤적을 검색하는 질의이다. 이전 연구에서 모든 이동체 질의에서 뛰어난 성능을 보인 색인은 없었다.

이 논문은 가장 일반적인 질의인 영역 질의의 성능 향상을 목적으로 한다. 이동체의 위치 데이터에 대한 영역 질의 성능을 실험한 대표적인 색인에는 3D R-tree[2], HR-tree[6], TB-tree[4]가 있다.

3D R-tree[2]는 공간색인인 R-tree에서 한 축에 시간 도메인을 추가함으로써 시공간 색인으로의 확장이다. 3D R-tree는 기존의 다차원 색인인 R-tree의 3차원 버전으로서, 현재 또는UC(until changed)에 대해 처리할 수 없는 문제가 있기 때문에 객체의 모든 이동을 시간에 대하여 닫힌(closed)되어 있다고 가정한다[3]. 즉 닫힌-선(closed-line)들만이 삽입 될 수 있다. 특정 시간과 영역에 대한 질의가 가능하지만, 기존의 R-tree의 단점인 노드 간의 겹침과 사각 영역이 너무 크기 때문에 검색 성능이 비효율적이라는 문제가 있다. 하지만, 현재까지 3D R-tree는 영역질의 성능이 우수한 색인으로 분류된다.

HR-tree[6]는 R-tree에 거래시간 개념을 추가하여 이력 정보를 표현할 수 있으며 연속적인 상태를 표현하기 위하여 R-tree에 중첩(overlapping) 개념을 추가한 것이다. HR-tree는 이동체들의 이동이 빈번하지 않은 경우에는 효율적이지만, 이동이 많이 발생하는 경우 단말 노드 및 비단말 노드를 새로 생성해야 하고, 특히 영역 질의의 성능이 저하되는 문제를 가진다.

TB-tree[4]는 궤적 질의 및 복합 질의의 성능 향상을 위하여 극단적인 궤적 보존 정책을 사용하였다. 하나의 단말 노드는 오직 동일한 궤적에 속하는 선분들만 저장할 수 있다. 공간적으로 근접한 선분들이 같은 궤적이 아니면, 서로 다른 노드에 저장된다. 이러한 이유에서 TB-Tree는 궤적에 대한 질의 성능은 좋은 반면 시간 간격에 대한 질의나 공간 간격에 대한 질의에는 좋지 않은 성능을 보인다.

III. 이동체의 위치 모델링과 자료구조

이 장에서는 이동체의 현재 위치와 과거 위치를 함께 색인화하는 TA3DR-tree의 색인 구조를 제안하고, 기존의 시공간 색인의 구조와 비교한다.

3.1 이동체의 위치 모델링

이동체 O_i 가 가장 최근에 보고한 시각을 t_k 라 할 때, 시간 간격 $[t_k, \text{now})$ 동안에는 이동체 데이터베이스에서 이동체 O_i 의 위치 변경은 없다. 이동체의 최근 위치 보고 시각과 현재 시각 사이의 이동체의 위치 정보가 존재하지 않기 때문에, 기존의 3D R-tree에서는 시간 간격 $[t_k, \text{now})$ 을 포함하는 영역 질의와 현재 위치 검색을 처리할 수 없다.

이동체의 위치 정보는 가장 최근에 보고한 시각 이후에도 데이터베이스에 존재하여야 한다. 시각 t_k 에서 이동체 O_i 의 보고 위치가 (x_i, y_i) 이라 할 때, 시각 t_j 에 새로운 위치 (x_j, y_j) 를 보고하면 새로운 선분 $\langle(x_i, y_i, t_k), (x_j, y_j, t_j)\rangle$ 가 색인에 삽입되어야 한다. 이때 (x_j, y_j, t_j) 를 UC 점(Until Changed point)이라 하며, 이동체의 가장 최근에 보고한 위치를 의미하며 데이터베이스에서 아직 변경되지 않은 이동체의 위치로 현재 위치로 볼 수 있다.

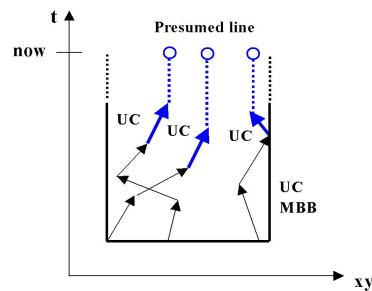


그림 1. TA3DR-tree에서 UC 점과 추정 선분, UC MBB의 예
Fig 1. An example of UC points, Presumed Lines and a UC MBB

[정의1] 추정 선분 (Presumed Line)

추정 선분은 UC 점을 시작점으로 가지면, 공간 좌표 값의 변경 없이 시간 축으로 성장하는 열린-선분이다. 이 동체 Q_i 의 UC 점을 UC_i 라 하고, UC_i 의 좌표가 (x_i, y_i, t_i) 이라 하면, Q_i 의 추정 선분은 $\langle(x_i, y_i, t_i), (x_i, y_i, now)\rangle$ 이다.

이 논문에서는 선분을 2가지로 구분한다.

첫째, $\langle(x_i, y_i, t_i), (x_j, y_j, t_j)\rangle$ 와 같이 닫힌-선분(closed line)이다. 이는 이동체의 궤적을 의미하며, 서로 다른 시각에 보고된 이동체의 과거 위치인 2개의 점으로 구성되는 고정 크기의 선분이다.

둘째, $\langle(x_i, y_i, t_i), (x_j, y_j, now)\rangle$ 와 같은 열린-선분(open-ended line)이다. 열린-선분의 끝점의 시각인 now는 증가하는 현재 시각을 표현하는 변수이다. 만약 이동체 Q_j 가 시각 t_j 이후인 시각 t_k 에 새로운 위치 (x_k, y_k) 를 보고하면 이동체 Q_j 의 열린-선분은 $\langle(x_k, y_k, t_k), (x_k, y_k, now)\rangle$ 로 변경된다.

현재-포함 시간 간격 질의를 처리하기 위해, 이동체의 최근 보고 위치와 현재 시각으로 구성된 열린-선분(open-ended line)을 정의해야 한다. 열린-선분은 질의가 발생할 때 마다 재계산되어야 하기 때문에 데이터베이스에 저장되지 않는다. 일반적인 선분의 끝점과 이동체의 최근 보고 위치를 구분하기 위하여, 최근 보고 위치를 UC(Until Changed) 점으로 정의한다. UC 점을 P_{uc} 이라 하고, 이전 위치를 P_{prev} 이라 하자. 닫힌-선분은 선분 $\langle P_{prev}, P_{uc}\rangle$ 으로 정의하고, 이동체의 최근 이동으로 표현된다. (그림 1)과 같이 추정 선분(Presumed Line)은 가상 선분 $\langle P_{uc}, now\rangle$ 으로 정의되며, 이동체의 현재 위치를 의미한다. TA3DR-tree에서 추정 선분은 색인에 삽입/저장되지 않지만 검색 시에 고려되는 가상 선분이다. 새로운 위치 보고 시 마다 추정 선분은 재정의되지만 추정 선분은 색인에 저장되지 않은 가상의 선분이기 때문에 재정의에 따른 색인 구조의 변화는 없다.

[정의 2] UC 최소경계박스(UC MBB)

이동체 Q_i 의 UC 점을 UC_i 라 할 때, UC_i 를 하나 이상 포함하는 노드의 최소경계박스를 UC 최소경계박스이라 한다. UC 최소경계박스가 아닌 최소경계박스를 고정 최소경계박스(fixed Minimum Bounding Box)라 하며, UC_i 를 포함하지 않는다.

이 논문에서는 UC 점을 포함하지 않는 일반적인 노드의

MBB와 구분하기 위하여 UC 점을 포함하는 노드의 MBB를 구분한다. 이동체 Q_i 의 UC 점을 UC_i 라 하면, 하나 이상의 UC_i 를 포함하는 노드의 MBB를 UC MBB라 하고, UC_i 를 포함하지 않는 노드의 MBB를 고정 MBB라 한다. UC MBB를 포함하는 부모 노드 또한 UC MBB이다. 선분의 삽입이 진행되고 있는 TA3DR-tree의 루트는 항상 UC MBB이다.

시간 간격 $[t_k, now)$ 을 포함하는 영역 질의와 현재 위치 검색을 처리하기 위하여 UC MBB를 계산하여야 한다. UC MBB의 상위 시간 경계는 시간에 따라 증가하고, 값은 now이다. 고정 MBB의 최소경계박스가 $\langle(x_i, y_i, t_i), (x_j, y_j, t_j)\rangle$ 이라면, UC MBB의 시간 증가 경계 박스는 $\langle(x_i, y_i, t_i), (x_j, y_j, now)\rangle$ 이다. 영역 질의 처리 시에 UC MBB의 now 값은 질의의 발생 시각인 t_q 로 대체된다.

3.2 색인 구조

TA3DR-tree는 UC 점들과 UC MBB들을 구분하기 위하여 태그 now를 R-tree의 자료구조에 추가하였다. 태그 now는 비트 스트림(bit stream)으로 노드의 헤드에 추가하였다. 태그 now는 시간 간격 $[t_k, now)$ 을 포함하는 영역 질의와 현재 위치 검색 질의를 처리하는 위한 과정에서 참조된다. (그림 2)는 TA3DR-tree의 자료구조를 보여주고 있다. 단말 노드에서 태그 now는 UC 점을 가진 엔트리를 표시한다. 비단말 노드에서 태그 now는 UC MBB인 엔트리를 표시한다. (그림 2(b)(c))는 UC MBB가 시간 증가 경계 박스임을 보여준다.

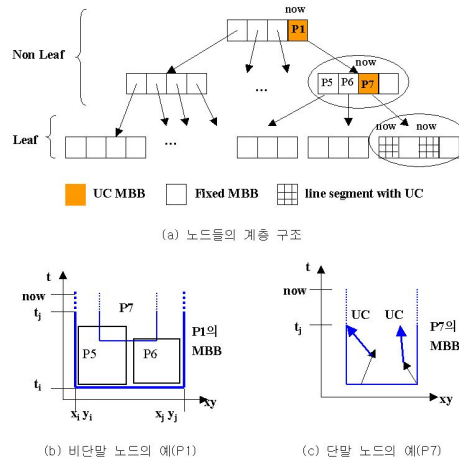


그림 2. TA3DR-tree의 자료 구조
Fig 2. Structure of the TA3DR-tree

(그림 2(c))에서와 같이 UC 점을 포함하는 노드의 MBB는 UC MBB이다. (그림 2(b))는 UC MBB를 포함하는 부모 노드 또한 UC MBB임을 보여준다. 새로운 선분의 삽입으로 더 이상 UC 점이 존재하지 않으면 해당 노드는 고정 MBB가 된다. 해당 노드가 UC MBB에서 고정 MBB로 전환되면 부모 노드로 UC 정보의 변경을 전파한다.

이 장에서 소개하는 TA3DR-tree의 색인 구조를 관련 연구[7]에서 제시하는 기준으로 기존의 시공간 색인 구조와 비교한다.

<표 1>는 7가지 기준으로 대표적인 시공간 색인과 TA3DR-tree를 비교하였다. “우수한 질의 종류” 항목은 각 색인의 성능이 다른 색인에 비해 우수하거나, 색인의 개발에서 성능 향상을 목적으로 하는 질의 종류를 기술하였다. 지금까지 연구에서 기술한 모든 질의 종류에 우수한 색인 구조는 없었다.

표 1. 시공간 색인 구조와의 비교
Table 1. Comparison of spatio-temporal index structures

색인 특징	TA3DR-tree	3D R-tree	HR-tree	MV3R-tree	STR-tree	TB-tree
시간 차원	거래 시간	유효 시간	거래 시간	거래 시간	거래 시간	거래 시간
시간 전개	연속적	연속적 또는 이산적	이산적	이산적	연속적	연속적
현재 시각 표현	가능	불가능	가능	가능	불가능	불가능
우수한 질의 종류	시공간 영역 질의	시공간 영역 질의	타임 슬라이스 질의	타임 슬라이스 & 영역 질의	복합 질의	궤적 질의
삽입 정책	최소 겹침 & 강제합병	최소 확장	최소 확장	최소 겹침	궤적 보존	궤적 보존
노드 분할 정책	시간축 비균등분할 & 절단정책	최소 영역	최소 영역	최소 영역	궤적 연결성	분할 없음
버전 중복	중복 없음	중복 없음	중복	중복	중복 없음	중복 없음

IV. 삽입 정책

이 장에서는 노드 간의 겹침 문제를 해결하기 위한 TA3DR-tree의 삽입 정책들을 소개하고, 기존의 이동체 색인들과 비교 분석한다.

4.1 강제 합병 및 절단 정책

이동 궤적의 삽입은 노드의 최소경계박스의 확장으로 유발하여 노드 간의 심한 겹침(High Overlap) 현상을 발생시킨다. 강제 합병 정책은 심한 겹침이 발생하는 노드들을 합병하여 심한 겹침을 제거한다. 만약 합병된 노드가 오버플로우가 발생하면 재분할을 수행한다. 긴 선분 절단 정책은 재분할 시에 분할된 2개의 노드가 심한 겹침이 발생하지 않도록 한다.

이동체의 이동 궤적은 시간에 따라 연속적으로 증가하므로 근접한 노드 간의 겹침이 시간이 경과함에 따라 더 심해진다. 이동체들의 이동 궤적을 삽입하면 삽입 노드의 최소경계박스는 확장되며, 근접한 노드의 최소경계박스와 겹침이 발생한다. 3D R-tree에서는 근접한 노드 간의 겹침이 증가하여 검색 비용이 증가하는 문제는 고려하지 않는다. TA3DR-tree에서는 최소경계박스의 확장에 의해 발생하는 겹침을 줄이는 정책으로 TR-tree[12]의 강제 합병 정책을 사용한다. TR-tree는 과거 이동 경로만을 색인화하는 TA3DR-tree의 초기 모델이다.

노드 간의 심한 겹침은 긴 선분의 매우 큰 최소경계박스(MBB)가 원인이 된다고 할 수 있다. 기존의 분할 정책들은 이용하더라도 긴 선분에 의한 심한 겹침 문제를 해결할 수 없다. TA3DR-tree는 긴 선분을 결정하고 절단하는 방법을 TR-tree의 긴 선분의 절단 정책을 사용한다.

4.2 강제 합병 및 절단 정책의 분석

이 장에서 소개하는 강제 합병 및 절단 정책은 노드 간의 겹침을 줄이는 정책이다. 또한 노드의 분할 과정에서 절단 정책은 긴 선분을 절단하여 삽입하기 때문에 분할되는 노드의 최소경계박스를 작게 한다. 강제 합병 및 절단 정책의 효과를 분석하기 위하여 GSTD 알고리즘으로 생성된 정

구분포 데이터를 사용하여 TA3DR-tree와 3D R-tree, TB-tree의 단말 노드의 최소경계박스의 크기와 겹침의 크기를 상대적으로 비교하였다.

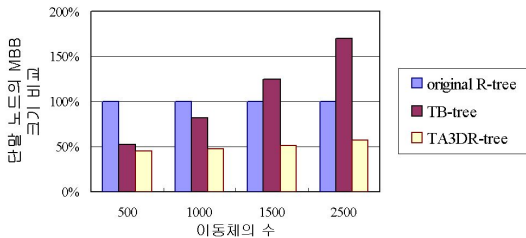


그림 3. 단말 노드의 MBB 크기 비교
Fig 3. Comparing the size of MBB of leaf nodes

(그림 3)은 단말 노드 MBB의 크기를 상대적으로 비교한 것이다. 단말 노드의 MBB는 3차원 박스이기 때문에 부피 값을 구하였다. 비교되는 각 색인에서 모든 단말 노드의 부피를 더하여 합을 구하였다. 이를 비교하기 위하여 (그림 3)에서와 같이 3D R-tree와의 비교치를 그래프로 그렸다. TA3DR-tree는 강제 합병 및 절단 정책의 효과로 단말노드의 MBB 확장을 최대한 억제하여, 단말 노드 MBB의 크기가 50% 감소하였다. TB-tree는 이동체의 수가 증가할수록 단말 노드 MBB들의 크기의 합이 증가하였다. 이는 TB-tree가 공간 근접성을 무시하고, 단말 노드에 같은 이동체의 쿼리만을 저장하기 때문이다. TB-tree에서 단말 노드 MBB의 크기가 동일하다고 가정하면, 색인에서 단말 노드 MBB들의 크기의 합은 이동체 수에 비례하여 증가한다. 이러한 현상은 단말 노드의 간의 겹침이 증가하는 것을 의미한다.

노드 간의 겹침은 영역 질의 성능을 관련된 중요한 요소 중의 하나이다. 관련 연구[8]에서 색인에서의 겹침을 겹침 비율로 정의하였다. 색인에서 단말 노드를 R_i 라 하고, n 개 단말 노드가 있다면, 색인에서 노드간 겹침 비율의 정의는 다음과 같다.

$$Overlap = \frac{\left| \bigcup_{i,j \in \{1..n\}, i \neq j} (R_i \cap R_j) \right|}{\left| \bigcup_{i \in \{1..n\}} R_i \right|} \dots\dots\dots (4.1)$$

수식 (4.1)은 노드 간의 겹침 영역을 노드의 크기로 나눈 값으로 (그림 3)에서와 같이 노드의 크기가 큰 경우에 노드 간의 겹침 비율이 작아 진다. 단말 노드의 겹침을 비교하기 위하여 노드간의 겹침을 각 노드의 평균 겹침 영역으로 정의한다. 평균 겹침 영역은 다음과 같은 수식으로 구하였다.

$$OverlapArea = \frac{\left| \bigcup_{i,j \in \{1..n\}, i \neq j} (R_i \cap R_j) \right|}{n} \dots\dots\dots (4.2)$$

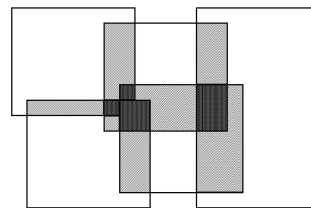


그림 4. 2차원 공간 도메인에서의 겹침과 다중 겹침
Fig 4. Overlaps and Multi-overlaps of 2 dimensional spatial domain

수식 (4.2)에서 색인의 겹침 영역을 총 단말 노드의 수로 나누는 이유는 같은 (그림 4)에서와 같이 다중 겹침의 영향으로 겹침 비율을 가진 노드들로 구성된 2개의 색인에서 노드의 수가 많은 색인이 겹침 영역이 크기 때문이다. 이는 단순한 겹침 영역의 총 합은 다중 겹침 영역을 중복 계산하기 때문이다. TB-tree와 3D R-tree의 경우에서 TB-tree는 노드 수는 3D R-tree 보다 작기 때문에 겹침 영역의 크기만을 비교할 경우에 TB-tree가 겹침이 작은 것으로 판단될 수 있다. 이러한 문제점을 해결하기 위하여 수식 (4.2)와 같이 평균 겹침 영역을 구하여 색인의 겹침을 비교하였다.

(그림 5)는 단말 노드의 평균적인 겹침 영역을 비교한 것이다. TB-tree는 이동체의 수가 증가할수록 3D R-tree와 TA3DR-tree에 비해 단말 노드간의 겹침이 많다. 이는 TB-tree의 삽입 정책이 공간적인 근접성을 무시하기 때문이다. 또한 이동체의 수가 증가할수록 노드간의 겹침도 비례하여 증가한다. TA3DR-tree는 강제 합병 및 절단 정책의 효과로 노드간의 겹침이 3D R-tree에 비해 작다.

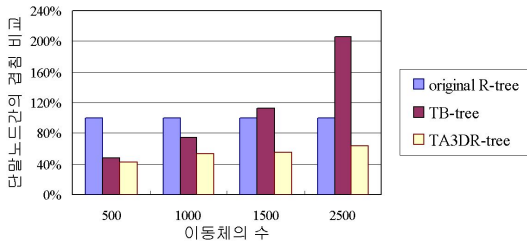


그림 5. 단말 노드간의 겹침 비교
Fig. 5. Comparing overlap of leaf nodes

(그림 3)과 (그림 5)에서 TA3DR-tree는 단말 노드의 MBB 크기와 겹침이 3D R-tree에 비해 50% 정도로 줄었다. 이는 3차원적으로 비교하여 영역 질의에서 각 축으로 20% 정도로 비교 영역이 감소한 것이다. 이러한 이유로 TA3DR-tree는 3D R-tree에 비해 영역 질의 성능 향상을 예상할 수 있다.

V. 실험

이 장에서는 논문에서 제안하는 TA3DR-tree의 성능을 평가하기 위해 이동체의 색인 중에서 영역 질의 성능이 가장 우수한 3D R-tree, TB-tree와 성능을 비교하고자 한다. 색인의 성능 비교의 기준을 제시한 관련 연구[9]을 바탕으로 색인의 크기와 질의 처리 성능을 실험을 통해 비교하였다.

시공간 색인인 HR-tree는 타임 슬라이스 질의는 우수하나 영역 질의 성능이 나쁘다. 궤적 질의 처리를 목적으로 하는 TB-tree는 이동체 수가 작거나, 영역 질의의 크기가 큰 경우에 3D R-tree 보다 성능이 우수하다.

최근 이동체 색인에 대한 성능 평가 테스트 환경에 대해서도 많은 연구[5, 13]가 이루어지고 있다. 이 논문에서는 이러한 연구 중 이동체 생성에 대한 대표적인 도구인 GSTD 생성기[10]와 IBM의 City Simulator를 사용하여 다양한 조건에서 이동체의 위치 데이터를 생성할 수 있다.

표 2. 실험 데이터 집합
Table 2. Data sets for experiments

약어	위치 데이터 분포	위치 데이터 생성 도구	이동 방향	이동 간격
G_Normal	정규 분포	GSTD	임의	보통
G_Skewed	비대칭 분포	GSTD	동북쪽	짧다
City_Person	건물/도로를 기반	City Simulator	임의	이주 짧다

실험에서 사용되는 데이터 집합은 <표 2>와 같다. GSTD 알고리즘을 이용하여 생성된 실험 데이터는 차량의 이동에 비유될 수 있으며 평균적인 이동 거리를 가진다. City Simulator를 사용하여 생성된 위치 데이터는 사람의 이동으로 짧은 이동 거리를 가진다.

5.1 공간 활용도의 비교

높은 공간 활용도는 일반적으로 트리의 높이와 같은 질의 비용을 감소시킨다. R-tree는 증가하는 시간 도메인을 고려하지 않았기 때문에 공간 활용도가 대략 57%이다. R*-tree는 제삼입 연산에 의해 63%의 결과를 얻었다. R*-tree는 공간 활용도가 높았지만, R-tree와 영역 질의 성능을 평가를 해보았으나, 성능 개선이 없어 실험 평가에서 제외하였다. TB-tree는 같은 객체의 궤적을 모아서 데이터 페이지에 저장하기 때문에, 90% 이상의 공간 활용도를 가진다. 하지만, TB-tree는 단말 노드 간의 겹침이 심하기 때문에, 4.2절에서와 같이 영역 질의의 성능이 떨어진다.

(그림 6)은 GSTD 알고리즘으로 생성한 G_Normal 데이터 집합에서 이동체 수와 이동체 보고 횟수의 변화에 따른 공간 활용도 변화를 보여 준다. TA3DR-tree는 시간 축의 비균등 분할의 효과로 77% ~ 68%의 공간 활용도를 가진다.

City Simulator를 사용하여 생성된 데이터 집합(City_Person)은 이동 간격이 작기 때문에 같은 이동체의 이동이 단말 노드 내인 경우가 많이 발생한다. 이것은 색인에서 공간적인 노드 겹침 현상이 작고, 시간 축 분할을 자주 유발시킨다. City_Person 데이터 집합에서 TA3DR-tree는 시간 축의 비균등 분할의 효과로 75% ~ 87%의 공간 활용도를 가진다.

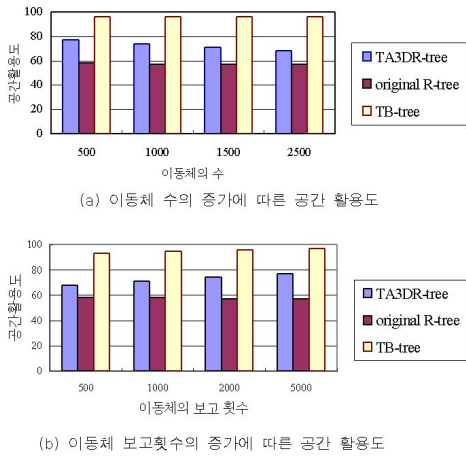


그림 6. 이동체 수와 보고 횟수에 따른 공간 활용도 (G_Normal 데이터)

Fig 6. The space utilization versus the number of moving objects and the number of sampling positions (G_Normal Data)

5.2 현재 위치 검색 질의의 성능 평가

기존의 3D R-tree와 TB-tree는 이동체의 현재 위치에 대한 고려가 없다. 이동체의 현재 위치를 이동체가 가장 최근에 보고한 위치로 가정할 때, 3D R-tree와 TB-tree는 이동체의 최대 허용 샘플링 시간 값을 가정하여 영역 질의를 수행해야 한다. 하지만 응용에 따라 이동체의 위치 보고 간격은 서로 다르며 비동기적이다. 또한 단절 현상과 같은 무선 통신 환경의 제약과 통신 비용으로 인하여 최대 허용 샘플링 시간 값은 평균 보고 시간 간격 보다 큰 값을 가진다.

이 논문에서는 3D R-tree와 TB-tree에서 질의 수행의 정확성을 무시하고, 데이터 생성기의 위치 생성 시간 주기를 최대 허용 샘플링 시간 값으로 하여 TA3DR-tree와 비교하였다. (그림 7)과 같이 TA3DR-tree는 3D R-tree와 TB-tree보다 현재 위치 검색 성능이 우수하다. TB-tree는 단말 노드의 겹침이 크기 때문에 타임슬라이스(time-slice) 질의에서 나쁜 성능을 보인다. 현재 위치 검색은 타임슬라이스 질의의 일부분이다. TA3DR-tree는 now 태그를 이용하여 UC MBB만을 검색하기 때문에 3D R-tree보다 현재 위치 검색 성능이 10~20% 성능 향상이 있었다. 하지만 실제 응용에서는 이동체들의 보고 주기가 다르기 때문에 3D R-tree의 현재 위치 검색 성능은 이보다 나쁠 것이다.

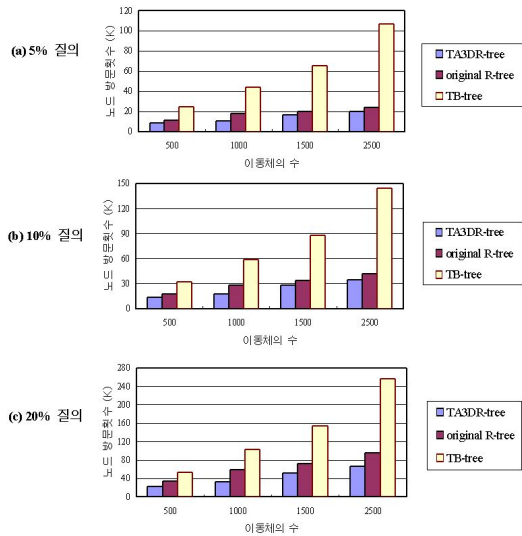


그림 7. 이동체 수에 따른 현재 위치 검색 질의의 성능 비교
Fig 7. Comparison of the query performance for searching the current position versus the number of moving objects

5.3 영역 질의의 성능 평가

공간 도메인의 크기가 같다면 이동체 수가 많을 수록, 이동체의 겹침은 겹침이 심해진다. 겹침의 겹침은 색인에서 노드 간의 겹침을 유발하기 때문에, 이동체의 수가 증가할 수록 TB-tree의 영역 질의 성능은 저하될 것이다.

영역 질의의 크기(RQS)는 각 도메인 크기의 5%, 10%, 20%이다. 영역 질의의 성능은 RQS 값에 따라 3가지 종류의 영역 질의를 각각 1000개 생성하여 수행하였다. 영역 질의는 색인의 전체 도메인을 벗어나지 않는 범위에서 임의 생성하였다. 방문 횟수는 1000번의 영역 질의를 수행하는 동안 방문한 노드의 총 횟수이다.

5.3.1 정규분포 데이터의 영역 질의의 성능평가

(그림 8)의 데이터 집합은 GSTD 알고리즘으로 생성되었고, 보고 횟수가 2000으로 동일하다. (그림 8(a)(b))에서 TB-tree는 노드간의 겹침 현상으로 인하여 3D R-tree에 비해 성능이 나쁘다. TA3DR-tree는 강제 합병 정책과 긴 선분의 절단 정책을 사용하여 노드간의 겹침을 최소화하기 때문에 성능이 우수하다.

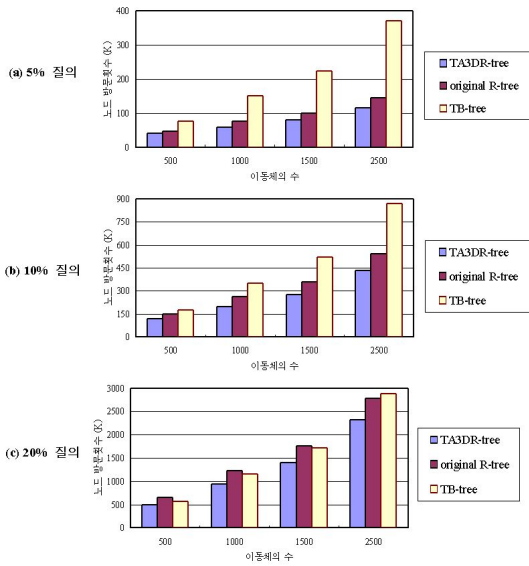


그림 8. 영역 질의 성능 비교(G_Normal 데이터)
Fig 8. Comparison of the range query performance(G_Normal Data)

(그림 8(c))에서 TB-tree가 3D R-tree와 비슷한 영역 질의 성능을 보인 이유는 TB-tree의 공간 활용도가 3D R-tree 보다 월등히 우수하기 때문에 색인의 크기가 작기 때문이다. 특히, 20%와 같이 비교적 질의 영역의 크기가 큰 경우에는 공간 활용도가 영역 질의 성능에 중요한 요소가 된다. RQS=20%의 경우에도 TA3DR-tree는 공간 활용도를 높였기 때문에 영역 질의 성능이 우수하다.

5.3.2 CitySimulator 데이터의 영역 질의 성능 평가

City Simulator는 도심 속의 사람들이 이동하는 위치 데이터를 생성하기 때문에, GSTD 알고리즘으로 생성된 데이터 보다 이동체의 궤적들이 작게 겹친다. 또한 City Simulator는 이동체들이 동일한 시간에 위치 보고를 하는 데이터를 생성한다. 이는 동시적 위치 보고를 의미하며, TB-tree 구조의 특징에 의해 TB-tree 색인의 노드간의 시간 축 겹침을 줄여주는 현상이 발생한다. 이와 같은 이유로 인하여 TB-tree의 성능이 (그림 8)의 실험 결과 보다 좋았다. 하지만 이러한 현상은 현실 세계에서는 발생하기 어려운 일이다.

(그림 9)에서 영역 질의 성능의 결과로 이동체 수가 증가할수록 TA3DR-tree의 성능이 우수하다. 이는 TA3DR-tree가 강제 합병 정책과 긴 선분의 절단 정책을 사용하여 노드간의 겹침을 최소화하기 때문에 겹침이 심한 데이터 집합에서 더욱 성능이 우수하다. (그림 9(b))에서 TB-tree가 3D

R-tree 보다 이동체 수가 2,000개와 4,000개에서 영역 질의 성능이 우수한 이유는 TB-tree의 공간 활용도가 3D R-tree 보다 월등히 우수하기 때문에 색인의 높이가 작기 때문이다. (그림 9(c))에서 TB-tree는 TA3DR-tree와 같은 성능을 보였다.

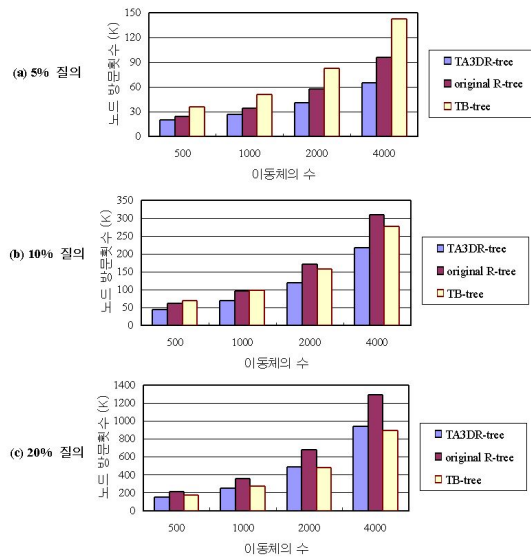


그림 9. 영역 질의 성능 비교(City_Person 데이터)
Fig 9. Comparison of the range query performance(City_Person Data)

5.3.3 비대칭분포 데이터(G_Skewed)의 영역 질의 성능 평가

비대칭 분포 데이터는 모든 이동체가 서남쪽에서 동북쪽으로 같은 방향으로 이동한다. (그림 10)의 데이터 집합은 보고 횟수가 1000으로 동일하다. 이는 모든 이동체가 서남쪽에서 출발하여 1000번의 이동으로 동북쪽에 도착하는 것을 의미하며, 위치 보고 간격 사이의 이동 거리가 짧다.

(그림 10)에서 TA3DR-tree는 모든 경우에서 3D R-tree와 TB-tree에 비해 성능이 우수하다. (그림 13(a))에서와 같이 RQS = 5%에서, TB-tree의 성능이 (그림 8)과 (그림 9)의 실험 결과 보다 나쁜 이유는 한쪽 방향으로만 이동하기 때문에 TB-tree의 단말 노드 내의 이동 반경이 3D R-tree와 TA3DR-tree 보다 크기 때문이다. (그림 10(c))에서와 같이 RQS = 20%에서, TB-tree는 3D R-tree에 비해 성능이 우수하다.

비대칭 분포 데이터에서 TA3DR-tree는 시간축 비균등 분할 효과가 낮았다. 이는 이동체가 동북쪽으로 연속적으로

이동함으로써 단말 노드 내에 선분들이 누적되지 않고, 공간적으로 확장되는 특징을 보였다.

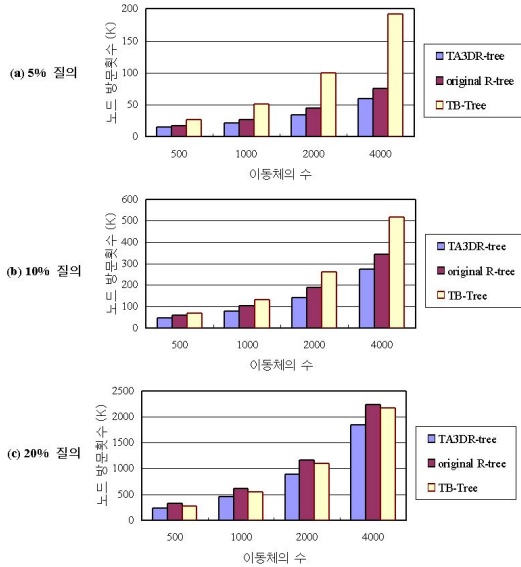


그림 10. 영역 질의 성능 비교(G-Skewed 데이터)
Fig 10. Comparison of the range query performance(G-Skewed Data)

VI. 결론

이 논문에서는 이동체의 현재 위치와 과거 궤적을 하나의 색인으로 저장하는 새로운 색인 구조를 제안하였다. 제안하는 TA3DR-tree는 기존의 3D R-tree와 TB-tree에서 현재 위치를 표현하지 못하는 문제를 추정 선분을 이용하여 해결하였으며, UC 점을 포함하는 UC MBB만을 검색함으로써 현재 위치 검색의 성능이 개선되었다.

TA3DR-tree는 삽입으로 인해 겹침이 심화되는 문제를 강제 합병 후 재분할 정책을 사용하여 해결하였다. 노드 간의 심한 겹침(High Overlap)을 유발하는 최소경계박스를 강제 합병하여 겹침을 줄이는 정책을 사용하였다. 또한 분할 후에도 심한 겹침을 야기하는 긴 선분을 정의하였으며, 절단 정책을 사용하여 제거 함으로써 노드간의 겹침을 최소화 하였다.

TA3DR-tree는 이동체의 이동인 궤적을 3차원 선분으로 저장하며, 연속적인 이동을 표현 한다. 또한 이동체 궤적의 영역 질의 성능 향상을 목적으로 한다. 이와 관련된 이전 연구에서 이동체의 수가 많고 이동 거리가 큰 경우에 3D R-tree가 영역 질의 처리에서 가장 우수하였다. TB-tree는 이동 거리가 작고 이동체의 수가 작은 경우에 3D R-tree 보다 큰 영역 질의 성능이 우수하다. 하지만 TB-tree는 공간 활용도가 좋으나, 단말 노드간의 겹침이 심하여 작은 크기의 영역 질의 성능이 3DR-tree와 TA3DR-tree 보다 나쁘다.

향후 연구로서 TB-tree와 같은 궤적 질의 성능이 우수한 색인을 개발하고자 한다. 이동체 색인에서는 색인 구축 비용이 크기 때문에 영역 질의와 궤적 질의의 성능이 모두 우수한 색인의 개발이 필요하다.

참고문헌

- [1] J. Green, D. Betti, and J. Davison, Mobile Location Services: Market Strategies, Ovum Ltd, 2000.
- [2] Y. Theodoridis, M. Vazirgiannis, and T. K. Sellis, "Spatio-temporal indexing for large multimedia applications," IEEE Int'l Conf. on Multimedia Computing and Systems, pp. 441-448, 1996.
- [3] M. A. Nascimento, J. R. O. Silva, and Y. Theodoridis, "Evaluation of access structures for discretely moving points," Proc. of Int'l Workshop on Spatio-Temporal Database Management, pp. 171-188, 1999.
- [4] D. Pfoser, C. S. Jensen, and Y. Theodoridis, "Novel approaches in query processing for moving objects," Proc. of Int'l Conf. on Very Large Data Bases, pp. 395-406, 2000.
- [5] T. Brinkhoff "A framework for generating network-based moving objects," Proc. of the Int'l Conf. on Scientific and Statistical Database Management, pp. 253-255, 2000.

- [6] M. A. Nascimento and J. R. O. Silva, "Towards historical R-trees," ACM symposium on Applied Computing, pp. 235-240, 1998.
- [7] Y. Theodoridis, T. K. Sellis, A. Papadopoulos, and Y. Manolopoulos, "Specifications for efficient indexing in spatiotemporal databases," Int'l Conf. on Scientific and Statistical Database Management, pp. 123-132, 1998.
- [8] S. BERTHOLD, D. A. KEIM, and H. P. KRIEGL, "The X-tree : An index structure for high-dimensional data," Proc. of Int'l Conf. on Very Large Data Bases, pp. 28-39, 1996.
- [9] J. Zoble, A. Moffat, and K. Ramamohanarao, "Guidelines for presentation and comparison of indexing techniques," ACM SIGMOD Record, vol. 25, no. 3, pp. 10-15, 1996.
- [10] Y. Theodoridis, J. R. O. Silva, and M. A. Nascimento, "On the generation of spatiotemporal datasets," Proc. of Int'l Symposium on Spatial Databases, pp. 147-164, 1999.
- [11] 이기영, 노경택, "공간 데이터 변환 시스템의 설계 및 구현," 한국컴퓨터정보학회 논문지, 제8권, 제4호, 2003.
- [12] 전봉기, 홍봉희, "이동체 데이터베이스를 위한 해쉬 기반의 공간 색인," 한국정보과학회 학술발표 논문집, 제28권, 제2호, pp. 205-207, 2001.
- [13] 이형석, 배상호, "도로의 노선선정을 위한 GIS 연계 기법의 평가," 한국컴퓨터정보학회 논문지, 제8권, 제4호, 2003.

저자 소개



전 봉 기

2003년 8월 부산대학교컴퓨터공학과
공학박사
1998년~1998년 한국통신연구소 전
임연구원
2003년 9월~현재 신라대학교 컴퓨
터정보공학부 전임강사
<관심분야> 데이터베이스, 공간 데
이터베이스, 이동체 데이터
베이스