

효율적인 위치 기반 서비스를 위한 다중 버전 기반의 시공간 색인 기법

(A Multiversion-Based Spatiotemporal Indexing Mechanism for the Efficient Location-based Services)

박 현 규 * 손 진 현 ** 김 명 호 ***

(Hyun Kyoo Park) (Jin Hyun Son) (Myoung Ho Kim)

요 약 시간에 따라 변화하는 공간 속성을 다루는 시공간 데이터베이스는 이동 통신 환경에서 위치 기반 서비스를 제공하기 위한 중요한 연구 분야이다. 이동 통신, 교통 관리 그리고 군사 목적의 지휘 통제 시스템 등의 다양한 영역에서 시공간 데이터의 효율적인 관리가 요구되고 있지만, 지속적인 위치 변화로 인한 객체들의 정확한 위치 정보를 데이터베이스에 유지하는 것은 어렵다.

본 논문에서는 다중 버전 B-트리 기반의 Bst-트리 색인 구조를 제안 한다. Bst-트리는 이동 객체들의 궤적에 대한 효율적인 이력 및 미래 영역 질의를 지원할 수 있다. 또한 동적 버전 관리 알고리즘을 통하여 항상 일정한 질의 처리 성능이 유지될 수 있도록 객체들의 이동 형태에 따라 버전을 유지한다. 그리고 다양한 실험을 통하여 본 논문에서 제안된 색인 방법의 유용성을 보인다.

키워드 : 시공간 데이터베이스, 다중 버전, 색인, 위치 기반 서비스, 이동 객체

Abstract The spatiotemporal database concerns about the time-varying spatial attributes. One of the important research areas is related to the support of various location-based services in mobile communication environments. It is known that database systems may be difficult to manage the accurate geometric locations of moving objects due to their continual changes of locations. However, this requirement is necessary in various spatiotemporal applications including mobile communications, traffic control and military command and control (C2) systems.

In this paper we propose the Bst-tree that utilizes the concept of multi-version B-trees. It provides an indexing method for the historical and future range query processing on moving object's trajectories. Also we present a dynamic version management algorithm that determines the appropriate version evolution induced by the mobility patterns to keep the query performance. With experiments we show that our indexing approach is a viable alternative in this area.

Key words : Spatiotemporal Database, Multi-version Structure, Indexing, Location-Based Service, Moving Object

1. 서 론

이동 객체에 대한 효율적인 위치 정보 관리는 시공간

데이터베이스 시스템에서 가장 중요하게 다루어지는 분야로서, 응용 분야로는 이동 통신 시스템, 교통 관리, 군용 지휘통제 시스템 등이 있다[1]. 최근의 이동 컴퓨팅 및 GPS와 같은 위치 정보 관련 기술의 발전은 이동 통신을 이용한 위치 기반 서비스를 가능하게 하고 있으며, 보다 효율적인 위치 기반 서비스를 위해서는 이동 객체들의 위치 정보를 빠르게 검색할 수 있는 색인 기법의 지원이 필수적이다.

이동 객체들의 위치는 지속적으로 변화하기 때문에 정확한 위치 정보를 유지하기 위해서는 매우 잦은 갱신

* 본 연구는 한국과학재단 특정기초연구(과제번호 R01-1999-00244) 지원으로 수행되었음.

* 종신회원 : 한국과학기술원 전자전산학과
hypark@dbserver.kaist.ac.kr

** 종신회원 : 한양대학교 컴퓨터공학과 교수
jhson@cse.hanyang.ac.kr

*** 종신회원 : 한국과학기술원 전산학과 교수
mhkim@dbserver.kaist.ac.kr

논문접수 : 2002년 4월 18일

심사완료 : 2002년 9월 27일

이 요구되지만, 데이터베이스에 의해서 관리되는 정보는 빈번한 갱신을 반영하기 어렵다. 그러므로 데이터베이스에 저장된 위치 정보는 이동 객체들의 현재 위치 상태를 반영하지 못하는 경우가 발생한다.

데이터베이스의 위치 정보의 정확성을 적절히 보장하면서 갱신 빈도를 줄일 수 있도록 고려할 수 있는 방법은 객체의 위치 정보를 시간에 따른 위치 좌표로 표현하지 않고 함수로 표현하는 것이다. 이러한 방법은 함수의 매개변수가 일정 범위 이상 변화할 때 갱신이 이루어지도록 함으로써 갱신 주기를 현저히 감소시킬 수 있다[1, 2].

위치 기반 서비스의 중요한 응용 영역 중의 하나는 특정 지역에 위치한 객체를 검색하는 영역 질의의 지원이다. 영역 질의 처리 문제는 이전의 연구에서 다양하게 다루어져 왔던 부분이지만, 본 논문에서는 기존 연구에서 주로 관심을 가진 과거 자료에 대한 시간-공간 영역 질의뿐만 아니라 미래의 객체 위치에 대한 영역 질의를 지원할 수 있는 색인 구조를 제시하고자 한다.

다루고자 하는 객체의 궤적은 기존 연구 방법과 유사한 일정한 시간 단위로 최초 위치와 속도 함수로 이루어지는 선형 함수로 표현된다고 가정한다. 이와 같이 선형 함수로 표현된 위치 정보는 쌍대(Dual) 변환과 SFC(Space Filling Curve)[3]를 사용하여 색인한다. 또한 과거 자료를 유지 하는 색인 구조에서 일정 수준 이상의 영역 질의 성능을 보장할 수 있도록 동적 관리 알고리즘(Dynamic Management Algorithm)을 바탕으로 다중 버전 색인 구조를 제안한다.

논문의 구성은 다음과 같다. 2장에서는 문제 정의 및 연구 동기를 기술하고, 3장은 관련 연구, 4장은 색인 구조와 질의 처리 방법, 5장에서는 확률 기반 동적 관리 알고리즘, 6장은 실험결과, 그리고 7장은 결론 및 향후 연구 과제를 다루었다.

2. 문제 정의 및 연구 동기

이동 객체는 3차원 공간에서 다양한 곡선 또는 직선으로 표현되는 궤적을 가진 시공간(Spatiotemporal) 객체이며, 위치 기반 서비스는 이동 통신 환경에서 중요한 시공간 응용 영역으로 주목받고 있다. 위치 기반 서비스를 위하여 이동 객체를 다루는 응용 소프트웨어는 위치 정보의 잦은 갱신을 지원해야 하는 문제가 있음에도 불구하고 관계형 데이터베이스 시스템에 추가(Add-on) 형태로 개발되고 있다.

본 논문에서는 위치 기반 서비스에서 요구되는 객체의 궤적(Trajectory)에 대한 현재 질의와 미래(Future)

질의를 다루고자 한다. 기존의 연구는 주로 저장된 궤적에 대한 과거 질의를 중심으로 이루어져 왔으나, 위치 기반 서비스는 과거 질의뿐만 아니라 미래 질의가 중요하게 다루어 진다. 대표적인 미래 질의의 예는 “앞으로 30분 이내 주어진 공간 영역에 존재하는 순찰차를 검색”과 같으며, 시공간 데이터에 대한 일반적인 위치 기반 질의의 형태는 다음과 같다:

질의 유형 1. “시간 t 에서 특정 영역에 존재하는 객체를 검색”

질의 유형 2. “시간 구간 $[t_s, t_e]$ 에서 특정 영역에 존재하는 객체를 검색”

여기서 질의 유형 1과 질의 유형 2는 각각 시간 단위(Time-sliced) 질의와 시간 구간(Time-window) 질의로 불리워진다.

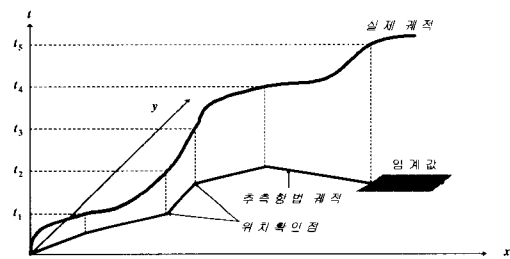


그림 1 이동 객체의 궤적에 대한 3차원 시공간 표현

본 논문에서 고려하는 이동 객체들의 위치 정보는 이동 통신 시스템을 통하여 얻는다고 가정하며, 획득한 위치 정보로부터 실제 궤적은 객체의 미래 위치를 예측하는 추측 항법(Dead-Reckoning)을 통하여 선형 함수로 근사시킬 수 있다. 따라서 그림 1과 같이 궤적은 위치 정보가 획득된 위치 확인점(Waypoint)으로부터 일정 기간 동안 선으로 표현된다.

선 정보는 그림 2에서처럼 R-트리 [1, 6]와 같은 공간 접근 기법(Spatial Access Method: SAM)을 이용하여 색인하는 방법이 있지만 3차원 이동 궤적을 색인하는 방법으로 R-트리는 다음과 같은 몇가지 문제점을 가지고 있다.

- (i) MBR(Minimum Bounding Rectangle)을 이용한 색인은 대상 객체인 선분보다 대단히 큰 영역을 할당한다.
- (ii) 근본적으로 모든 궤적은 갱신이 이루어질 때까지 무한히 확장되기 때문에 선형 직선에 대응되는 MBR을 만들 수 없는 경우가 발생한다.

그러므로 그림 2에서와 같이 궤적은 (p_0, p_1, p_2) 를 포

합하는 하나의 MBR을 부여하거나 또는 선형 직선(p_0 , p_1)과 (p_1 , p_2)로 구분된 2개의 MBR을 이용하여 색인하는 방법 등을 고려할 수 있다. 그러나 이러한 경우에도 p_2 이후 궤적의 선형성이 언제까지 유지될 수 있는지를 예측하는 것은 곤란하다.

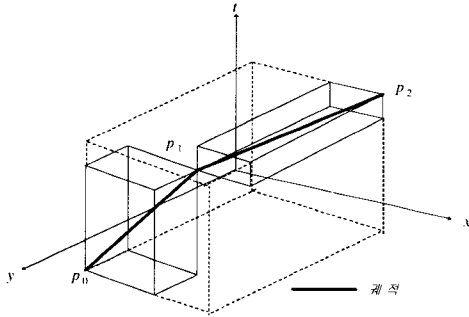


그림 2 MBR을 이용한 시공간 색인

그러므로 3차원 궤적에 대한 효율적인 색인 및 질의 처리 방법의 개발이 요구된다. 또한 색인 구조는 갱신이 발생할 때 과거 자료를 계속 유지해야 하므로 각각의 색인된 정보는 유효 시간(Life-time)을 가진다. 여기서 유효 시간은 마지막 위치 확인점으로부터 정해진 임계값(Threshold)를 벗어나는 편차(Deviation)가 발생하기까지의 시간이며, 색인 구조는 이러한 갱신된 정보를 보존(Persistent) 형태로 저장할 수 있어야 한다.

3. 관련 연구

3.1 관련 연구

기존의 공간 및 시간 데이터베이스 연구는 각각 개별적으로 수행되어 왔으며, 계산 기하학(Computational Geometry)에서는 이동 객체에 의하여 제기되는 다양한 기하 문제(예를 들어 Convex hull, Delaunay triangulation 등)를 해결하고자 하는 노력이 이루어졌다 [3]. 관련 연구로서 [7]에서는 이동 궤적을 다루는 자료구조로서 KDS(Kinetic Data Structure)를 제시하였으며, [8]은 질의 유형 1에 대하여 저장공간 복잡도를 이용한 I/O 복잡도를 가지는 자료 구조를 제시하였다.

시공간 데이터베이스에 관련된 조사 연구는 [1, 6, 9] 등에서 제시되고 있다. 특히 [2]는 이동 객체에 대한 데이터 모델을 제시하였으며, 이를 객체 추적(Tracking)과 같은 분야에 적용한 MOST(Moving Object Spatio-Temporal) 프레임워크와 FTL(Future Temporal Logic)을 제시하였다. 많은 시공간 데이터베이스 연구는

[2]에서 제시된 데이터 모델을 효율적으로 지원하는 색인구조를 다루고 있다. TPR-트리 [1, 6]는 시공간 특성인 객체의 이동 속도에 따른 MBR 구성 휴리스틱을 제시하여 R⁺-트리의 성능을 개선한 색인 구조로서 미래 시간 질의를 처리할 수 있다. TPR-트리는 R⁺-트리의 확장이지만 미래 시간 색인에 대한 구체적인 사례로서 객체의 속도에 의하여 색인 구성을 결정하는 개념을 제시하였다. [9]에서는 미래 질의에 대하여 기존의 kd B-트리 및 R-트리 보다 우수한 색인 방법으로 쌍대 변환을 이용하였으나, 공간과 이동 속도를 고정된 값으로 제한하고 있으며 과거 질의를 배제하고 있다.

객체의 지속적인 갱신을 지원하기 위하여 본 논문에서는 [10, 11, 12]의 다중 버전 구조(Multi-version Structure)를 사용한다. 시공간 색인에서는 이동 형태(Pattern)를 적절히 예측함으로써 지속적인 갱신 문제를 보다 용이하게 해결하며, 따라서 다중 버전 구조를 활용하여 질의 처리 성능을 저하시키지 않는 방법을 찾을 수 있다. 그러나 지금까지 이러한 분야의 연구는 많이 이루어지지 않고 있다.

위치 정보의 갱신 빈도는 이동 통신의 위치 관리 프로토콜에 의하여 영향을 받는다. 이동 통신에 적용되는 위치 관리 정책에 대하여는 [4, 5, 13]에서 제시되고 있으며, 위치 정보를 획득하는 거리 기반(Distance-based) 위치 갱신 프로토콜은 논문에서 가정하는 통신 환경과 동일하다. 이에 대한 기술적 조사 및 상세한 연구는 [4, 5]에서 제시되고 있다.

3.2 쌍대 변환 및 Space Filling Curve

이동 통신 환경에서 이동 객체의 위치 정보는 튜플($t, x, y, F_x(t), F_y(t)$)의 형태로 표현된다. 이 튜플에서 t 는 튜플 정보가 전송된 시간이며, (x, y) 는 위치 좌표 정보이고, $(F_x(t), F_y(t))$ 는 각각의 방향에 대한 속도 벡터이다. 그러므로 주(Primal) 공간에서의 이동 정보에 대한 궤적은 다음과 같이 쌍대 공간으로 일대일 변환되어 표현될 수 있다.

다차원 d 공간 R^d 에서 함수 $xS = v_{d-1}x_{d-1} + v_d$ 로 표현되는 Hyperplane은 쌍대 공간에서 점 (v_1, \dots, v_d) 로 표현되며, 주 공간에서 점은 쌍대 공간에서 $x_d = -v_1x_1 - \dots - v_{d-1}x_{d-1} - v_d$ 를 따르는 Hyperplane으로 표현된다 [8, 9, 15].

그림 3은 주 공간과 쌍대 공간에서 동일한 궤적을 표현한 것으로서, 주 공간 xt 에서 궤적은 쌍대 공간 $x'v$ 에서 점으로 표현된다. 또한 질의 영역 R 은 두개의 Hyperplane으로 이루어지는 스트립(Strip)으로 변환된다. 예를 들어 그림 3에서 주 공간에서의 객체 $o_i(i =$

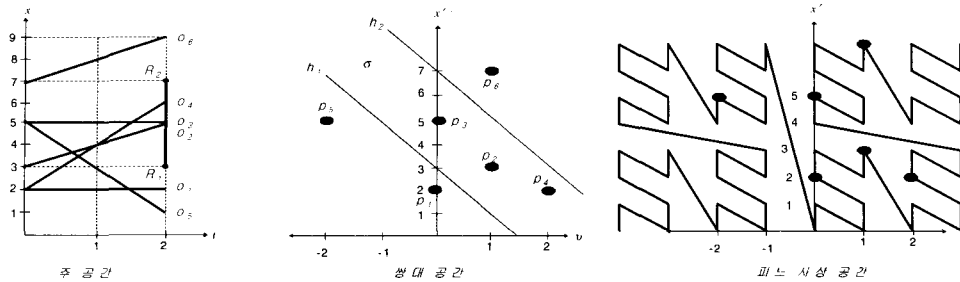


그림 3 주 공간과 쌍대 공간에서 궤적 표현

1.6)의 궤적은 쌍대 공간에서 $p_i(i=1..6)$ 의 점으로 표현된다. 그리고 주 공간에서의 질의 영역 R 의 꼭지점 R_1 과 R_2 는 각각 쌍대 공간에서 Hyperplanes h_1 과 h_2 로서 표현된다. 따라서 주 공간에서 유형 1 또는 2와 같이 x 축과 평행한 질의 영역 R 과 교차하는 궤적에 대한 질의 처리는 쌍대 공간에서 두 Hyperplane 사이의 영역인 σ 내에 존재하는 점 객체를 검색하는 것과 같다.

본 논문에서는 다차원 쌍대 공간에서 점으로 표현되는 이동 객체들의 궤적을 색인하기 위하여 SFC [3]와 일차원 점 접근 방법(Point Access Method)를 사용한다. 특히 피노 곡선(Peano curve) [3, 14]를 이용하여 쌍대 공간에 존재하는 모든 점 객체들에게 다차원 공간의 좌하단부터 선형 순서(Ordering)를 부여한다. SFC(x, v)를 2차원 점 (x, v)에 대하여 피노 곡선에 의한 고유한 값을 부여하는 함수라고 하면, 쌍대 공간에서 각 점들은 SFC(x, v)를 키값으로 색인을 구성한다. 또한 쌍대 공간에서 피노 곡선은 x' 축을 중심으로 좌우 공간을 대칭으로 분할할 수 있는 장점이 있어 다른 SFC보다 효율적인 질의 처리를 가능하게 한다.

4. 이동 객체에 대한 색인 구조

본 논문에서 제안하는 Bⁿ-트리는 [10, 11]의 다중 버전 트리와 B-트리의 특징을 이용한 시공간 색인 구조이다. 이는 전체 객체의 궤적을 구간 단위 선형 함수로 색인하며, 과거 질의 및 미래 질의를 모두 지원하기 위해서 각 구간 단위 유효 시간을 가지는 다중 버전 구조로 구성된다.

동 객체의 궤적은 3차원 (x, y, t)의 시공간 객체이지만 본 논문에서는 y 방향의 변화가 없는 2차원 (x, t)으로 한정하여 색인 방법을 다루고자 하며, 이는 궁극적으로 3차원 시공간으로 확장이 가능하다. 그림 4-(a)에서는 xt 공간에서 3개의 이동 객체 궤적을 표시한다. O_2 는 시간 t_2 와 t_1 에서, O_3 는 매 시간 이동 함수의 변화가 발

생하며, O_1 은 전체 시간 구간에서 동일한 이동 함수를 유지하는 경우이다.

본 논문에서는 색인을 구성하거나 갱신하기 위한 위치 정보는 이동 통신 시스템을 이용하여 얻으며, 거리 기반 갱신 프로토콜에 의하여 이동 객체는 주기적으로 자신의 위치 정보를 확인하는 것을 가정한다.

위치 확인 및 전송을 위하여 단위 시간으로 Δt 를 설정하며, 단위 시간 Δt 는 편의상 1로 정규화하여 사용한다. 그러나 객체는 매 단위 시간별로 자신의 위치 정보를 확인하는 것이 불가능하므로 일정한 시간 간격 $\tau = n\Delta t$ 를 설정하여 위치 정보를 확인한다. 그림 4-(b)는 객체 O_2 의 정보 갱신과 시간을 표시한 것으로 t_2 와 t_1 에서 위치 정보 갱신이 수행된다. 다음에서는 두 가지의 서로 다른 위치 정보 갱신 방법을 정의한다.

정의 1. IU(Instantaneous Update)

IU는 개별 객체의 이동 함수 변화가 발생하여 이동 객체의 요구에 의하여 색인을 갱신하는 것을 말한다. □

정의 2. AU(Adaptive Update)

AU는 전체 객체에 대한 색인을 현재 위치 정보로 일괄 갱신하여 새로운 색인 버전을 구성하는 것을 말한다. □

두개의 연속된 AU 사이의 시간 간격은 그림 4(b)에

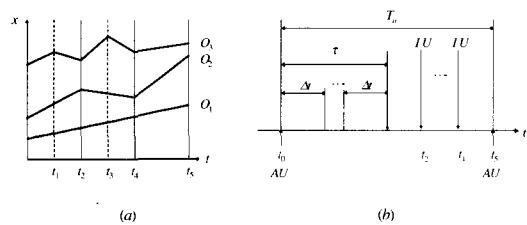


그림 4 시간에 따른 궤적 표현과 갱신 관련 변수

서 T_u 로 표시하였으며, 이를 결정하는 방법은 5장에 설

명한다. T_c 동안 객체의 이동 궤적이 변경되어 이동 함수에 변화가 발생할 수 있다. 이 때 이동 함수의 변화를 데이터베이스에 반영하기 위해서 각 객체는 새로운 위치 정보를 데이터베이스로 전송하는 IU를 수행한다.

4.1 색인 구조 개요

다중 버전 색인 구조는 현재 및 과거 위치 정보를 모두 유지할 수 있는 장점이 있는 반면에 많은 갱신이 일어나면 심각한 오버헤드(Overhead)가 발생한다. 따라서 이러한 경우를 방지하기 위한 적절한 방법이 필요하며 B^{st} -트리는 이러한 목적으로 그림 5와 같이 Branch 노드와 Version 노드로 구분한다. 루트(Root)로부터 Branch 노드는 단일 레벨 색인 구조를 가지며, 각각의 Version 노드는 다중 버전 B-트리 기반 구조를 가진다. AU가 수행되면 수행 시간 별로 새로운 Branch 노드를 생성하며 해당 Version 노드에 대한 포인터를 가진다. 예를 들어 AU가 시간 0과 5에 수행된다고 하면 두개의 Branch 노드 $b1(0)$ 과 $b2(5)$ 가 그림 5와 같이 만들어 진다.

정의 3. 레코드 구조

각 노드의 레코드는 (Key, TS, Pointer)로 구성되며, 각각은 키값, 저장(또는 갱신) 시간 그리고 다음 레벨(중간 노드) 또는 실제 데이터 위치(단말 노드)를 가리키는 포인터이다. □

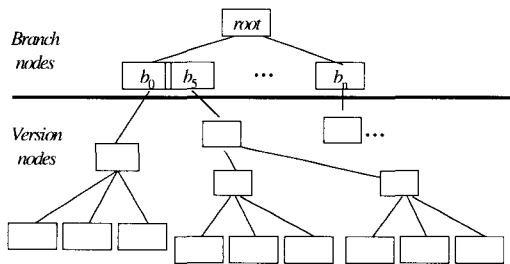


그림 5 B^{st} -트리 색인 구조

Version 노드에서 키값은 3.2절에서 설명된 쌍대 공간에서 $SFC(x, v)$ 에 의하여 생성되는 고유 값이며, TS는 정보 획득 시간과 이동 정보의 갱신이 일어나기 까지 시간으로 구성된 (t_s, t_e) 로 위치 정보의 유효 시간을 의미한다. 현 시간에서 유효한 TS는 $(t_s, *)$ 로 그리고 갱신된 과거 데이터에 대한 TS는 (t_s, t_e) 로 표시된다.

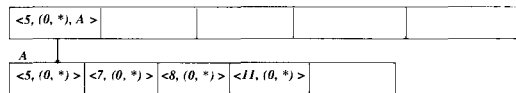
4.2 색인 구조 및 관리

B^{st} 트리에서 삽입과 갱신은 현재 버전의 색인 구조에 대해서 이루어진다. 그러므로 레코드를 삽입하거나 갱신하기 위해서는 현재 Branch 노드로부터 레코드의

키값에 의하여 해당되는 단말 노드를 탐색한다.

보존하는 구조인 B^{st} -트리에서 갱신은 B-트리의 갱신과 다르며, 본 논문에서는 레코드 삽입과 구별하기 위하여 갱신된 레코드를 Variant라고 한다. 색인을 갱신하기 위해서는 Variant를 키값에 의하여 해당되는 노드에 삽입하며, 기존의 레코드는 삭제하지 않고 $TS(t_s, t_e)$ 의 t_e 값을 현재 시간으로 변경한다. 따라서 레코드 갱신 연산은 삭제를 하는 대신에 해당 객체의 유효 시간을 마감하고 그 객체는 새로운 유효 시간으로 시작하도록 삽입한다.

시간(TS) 0에서 색인 구조



시간(TS) 2에서 색인 구조

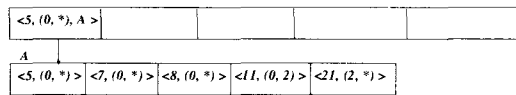


그림 6 Version 노드에서 구조 변화가 없는 IU

그림 6은 갱신 예제로서 Version 노드에서 키값 11을 가지는 레코드를 시간 2에서 이동 함수의 변화로 인하여 키값을 21로 변경하는 경우이다. 갱신은 새로운 정보를 노드 A에 삽입하고, 기존 레코드의 TS값을 $\langle 0, * \rangle$ 에서 $\langle 0, 2 \rangle$ 로 변경한다.

정의 4. 노드 오버플로우 및 언더플로우 조건

P_{max} 와 P_{min} 은 오버플로우에 의한 노드 분할이 발생할 때 분할 방법을 결정하거나, 언더플로우를 결정하는 변수이다. 노드의 크기가 B이고, 유효 레코드의 개수를 N_v 라고 할 때 $B \geq P_{max} \geq P_{min}$ 의 값을 가지며, $P_{min} \geq N_v$ 은 언더플로우 조건이 된다. □

갱신이 그림 6과 달리 색인 구조 변경이 발생하는 경우, 오버플로우가 일어난 노드의 유효한 레코드를 새로 할당된 노드에 모두 복사한 후 기존의 노드에 존재하는 유효한 레코드의 TS를 변경하여 오버플로우가 일어난 노드는 모두 비유효한 레코드로 이루어진 노드로 설정한다. 이 때 노드 분할은 다음 두가지 형태로 나누어 진다.

Case 1 : $N_v > P_{max}$; 2개의 노드를 할당하여 오버플로우가 발생한 노드의 유효 레코드를 키값에 의하여 2개의 노드로 삽입한다.

Case 2 : $P_{min} \leq N_v \leq P_{max}$; 1개의 노드를 할당하여 오버플로우가 발생한 노드의 유효 레코드를 모두 새로운 노드로 삽입한다.

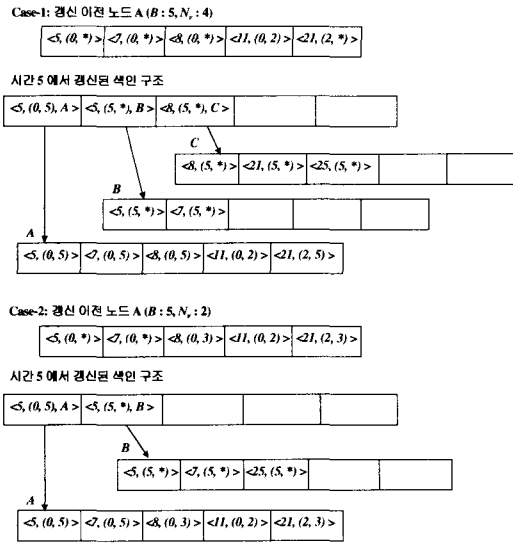


그림 7 구조 변화를 포함하는 갱신 과정

그림 7에서는 위치 정보 $\langle 25, (s, *) \rangle$ 를 입력할 때 Case-1과 Case-2의 경우를 각각 보여주고 있으며, 각 노드의 크기는 5, P_{max} 와 P_{min} 은 각각 3과 1로 설정하였다. Case-1은 노드 A의 유효 레코드 (키값 {5, 7, 8, 21})을 가지는 레코드를 새로 할당된 노드 B와 C에 삽입한다. 삽입된 레코드는 갱신 시간을 TS의 t_s 값으로 설정하여 (5, *)를 가지며, 노드 A의 갱신된 기존 레코드들에 대한 TS의 t_e 값을 5로 바꾼다. 그리고 키값 25를 가지는 Variant를 노드 C에 삽입한다. Case-2의 경우는 2개의 유효 레코드 (키값 {5, 7})를 새로 할당된 노드로 삽입한 후, 키값 25를 가지는 Variant를 삽입하는 Case-1의 과정을 수행한다.

P_{min} 값이 1보다 큰 경우 노드의 유효 레코드의 개수가 P_{min} 값 보다 작을 경우가 발생하게 되며, 이 때에는

언더플로우 상태가 된다. 언더플로우 상태에서는 이웃 노드와 병합(Merge)하는 과정을 통하여 각 노드의 유효 레코드를 $P_{min} \leq N_v \leq P_{max}$ 범위에 존재하도록 하며, 결과적으로 Bst-트리는 저장 공간 복잡도 $O(N/B \log_B N)$ 와 I/O 복잡도를 가진다.

4.3 쌍대 공간에서의 영역 질의 처리

쌍대 변환에 의한 점객체와 Hyperplane은 주 공간과 쌍대 공간에서 위치의 상하관계를 유지한다. 그러나 주 공간에서 영역 질의는 쌍대 공간에서 그림 3과 같이 무한히 확장하는 스트립으로 표현되기 때문에 검색 영역이 대단히 커지는 문제가 발생한다. 따라서 질의 처리를 위해서는 쌍대 공간에서 검색 영역의 크기를 줄이는 절단(Pruning) 과정이 필요하다.

절단 과정은 다음과 같이 수행된다. 대상으로 하는 이동 객체의 최대 속도가 주어졌을 때 최대 속도는 주 공간에서 기울기로 표현된다. 그러므로 그림 8에서 임의의 객체가 최대 속도로 이동한다고 가정했을 때 t_2 의 질의 영역 $R(R_1, R_2)$ 에 속할 수 있는 궤적은 T_2 와 T_3 이다. 그러므로 그림 8의 주 공간에서의 검색 영역은 어렵게 칠해진 사다리꼴 형태 내에 존재하는 궤적으로 제한된다. 한편 주 공간에서 T_1 과 T_4 는 정지 객체의 궤적을 나타낸다. 이러한 주 공간에서의 제한된 검색 영역은 쌍대 공간에 대응된다 (즉, 주 공간에서 T_1, T_2, T_3, T_4 는 쌍대 공간에서 $T^*_1, T^*_2, T^*_3, T^*_4$ 로 변환 된다). 결국 쌍대 공간에서는 $T^*_1, T^*_2, T^*_3, T^*_4$ 로 구성된 4개의 점은 질의 처리를 위한 검색 영역인 스트립을 그림 8과 같이 x' 를 기준으로 음영 영역으로 이루어지는 두개의 직사각형 형태의 영역이 된다.

질의 유형 2와 같은 시간 구간 영역 질의의 경우에도 주 공간에서 시간 t_e 에서 형성되는 사다리꼴 영역을 설정하면 구간 $[t_s, t_e]$ 로 이루어지는 모든 가능한 검색 영역을 포함하므로 쌍대 공간에서 시간 조건 t_e 에 해당하

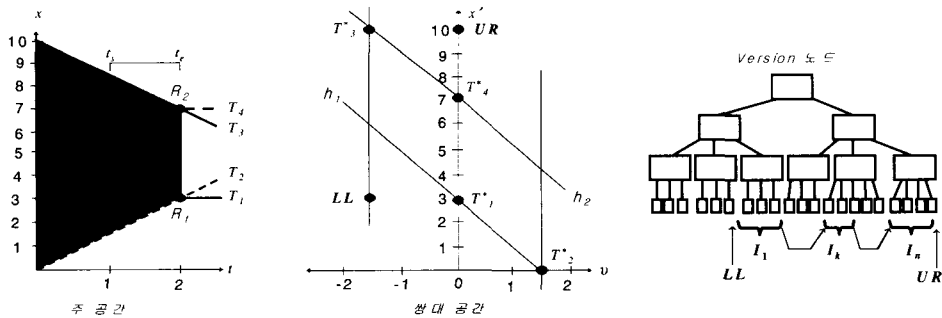


그림 8 영역 질의를 위한 주-쌍대 공간의 표현과 Bst-트리 검색

는 시간 단위 질의 형태가 된다.

그러므로 검색 영역에서 가장 좌하단의 값이 되는 LL (Lower Left)를 찾을 수 있으며, 검색 영역의 가장 큰 값인 UR (Upper Right) 또한 구할 수 있다. 그러나 피노 값에 의한 결과 집합은 오름차 순으로 정렬된 $[LL, UR]$ 구간의 모든 페이지가 해당되지 않는다. 즉 질의를 만족하는 결과 집합은 그림 8의 $Version$ 노드에서와 같이 다수의 페이지 영역 I_n 으로 이루어진다. 각 페이지의 키값이 $[l_i, u_i]$ 일 때 질의 처리 결과는 $LL \leq l_i \leq u_i \leq UR$ 를 만족하는 페이지들이 된다.

관련 페이지를 검색하는 방법은 LL 값을 포함하는 페이지를 탐색하고 계속하여 링크를 따라 UR 보다 큰 l_i 값을 가지는 페이지에 이르기 까지 해당되는 페이지를 검색하는 것이다. $[LL, UR]$ 구간에서 검색 영역에 포함되는 페이지만 검색하는 알고리즘은 [14, 15] 등에서 제시되고 있으므로 여기서는 생략한다

영역 질의를 처리하는 복잡도는 B^s -트리의 높이 (Height)와 질의 영역에 포함되는 단말 노드의 수에 의하여 제한된다. 이는 단말 노드의 레코드로부터 유효 레코드를 추출하는 오버헤드 t 를 포함하는 B 트리 검색 복잡도와 같으므로 $O(\log_B N + t)$ 의 I/O 복잡도를 가진다.

5. 동적 색인 관리

본 논문에서는 동적으로 변화하는 실제 환경에서 객체의 이동을 색인하기 위하여 AU 사이의 주기 (그림 3에서 T_u)를 '적응적(Adaptive)'으로 결정하는 방법을 제시한다. 갱신 주기를 선정하는 것은 객체의 이동 형태, 요구되는 정확도, 기술적 한계 등으로 결정된다. 그러므로 논문에서는 비용 모델(Cost Model)을 설정하여 성능을 평가하고, 성능에 영향을 미치는 매개 변수의 값을 선정한다.

5.1 성능 평가를 위한 비용 모델

N 개의 이동 객체를 색인하는 B^s -트리 성능을 추정하기 위해서는 색인의 크기(Size)와 높이를 검토할 필요가 있다. 이전의 연구에서 [19]는 다중 버전 구조의 비용 모델을 제시하였으나, B^s -트리의 성능 평가 기준이 되는 노드 접근 회수는 객체의 이동 형태에 따라 색인 구조의 변경으로 인하여 변화된다. 따라서 질의 처리 효율을 유형 1 또는 2 형태 별로 추정하기 위한 비용 모델을 설정하고 디스크 저장 공간 크기 및 색인 구성 시간에 관련된 다양한 매개 변수를 도출한다.

정의 5. 임의도(Randomness)

임의도는 이동 형태의 선형성을 결정하며, N_v 을 $Version$ 노드에 존재하는 유효한 레코드의 숫자, N_r 을

$Version$ 노드에서 전체 레코드 개수, v 를 다음 갱신 시간에서 IU 에 의하여 삽입되는 평균 $Variant$ 의 수라고 할 때 이동 형태의 임의도에 의하여 색인에 갱신이 반영되는 임의도는 $r = v / N_r$ 이다. □

B^s -트리는 B-트리의 평균 효율을 따르므로 [13] 특정 시간에서 예상할 수 있는 N_r 는 $\ln 2 \cdot B \cdot P_r$ 이다. 그러므로 B^s -트리의 높이 h 는 식 (1)와 같다.

$$h = 1 + \lceil \log_{N_r}(N/B) \rceil \quad (1)$$

단말 노드는 새로운 노드를 생성하는 노드 분할이 발생하기 전까지 레코드 삽입을 허용한다. 그러므로 $Version$ 노드의 크기는

$$Version \text{ 노드 크기} = \sum_{i=0}^{h-1} \left[\frac{N}{N_r^{i+1}} + \frac{rN(T-1)}{(b-N_r)^{i+1}} \right] \quad (2)$$

이때 T 는 전체 색인이 가지는 시간이며, i 는 노드의 레벨이다.

그리고 미래 질의는 B^s -트리가 일반적인 B-트리와 같이 오직 유효한 레코드만을 가지고 있는 경우에 최적 성능이 보장되므로 $Version$ 노드는 임의도에 의하여 발생하는 노드 분할을 일정 수준이하로 유지할 필요가 있다. 노드 분할이 계속하여 발생하면 $Version$ 노드의 성능은 B-트리와 비교하여 현저히 나쁜 성능을 보이므로 성능 저하점(Degradation Point) 이상의 노드 분할이 일어나지 않도록 해야한다. 이러한 $Version$ 노드에서 성능 저하가 발생하는 임의도는 식 (3)과 같다.

$$성능 \text{ 저하 임의도} \cong \frac{B - N_r}{N_r} = \frac{1 - \ln 2 \cdot (P_{max}/B)}{\ln 2 \cdot (P_{max}/B)} \quad (3)$$

성능 저하 임의도는 조정 매개 변수 (Tuning Parameter)인 P_{max} 의 값에 의하여 크게 영향을 받는다. 또한 B^s -트리의 질의 처리 성능 또한 고려할 필요가 있다. 따라서 식 (4)에서는 질의 조건에 대한 성능 저하 임의도를 제시한다. 즉, 질의에 의한 성능 저하 임의도에 도달하기 까지 AU 구간에서 허용할 수 있는 IU 를 도출할 수 있다.

$$성능 \text{ 저하 임의도} (q_k, q_l) \cong \frac{\frac{N_r}{N_v} q_{l-1}}{\frac{\ln 2 \cdot (P_{max})}{1 - \ln 2 \cdot (P_{max})} (q_l - 1)} \quad (4)$$

여기서 q_k 와 q_l 는 각각 질의 조건에서 키와 시간 조건 범위이며, 성능 저하 임의도는 B^s 트리에서 최적의 Branch 시간 구간 T_u 를 결정하는 값이 된다.

5.2 확률 기반 이동 추적 및 관련 매개 변수

이동 통신 환경에서 객체의 이동 형태 예측에 관련된

연구는 [4, 5, 13] 등에서 제시되고 있으며, 이동 형태가 알려져 있다면 확률적으로 미래의 위치를 예측 가능하다. 본 논문에서는 다음 사항이 데이터베이스 시스템과 이동 객체에 알려져 있고 공유한다고 가정한다.

- 객체의 이동 형태
- 이동 함수의 변화가 마지막 갱신 이후 현재까지 발생하지 않는다
- 마지막 갱신된 위치 정보와 갱신 거리 임계값 그리고 이동 형태는 임의(Random) 이동과 유속(Fluid-Flow) 이동을 모두 표현하는 브라운 운동 과정을 이용한다.

정의 6. 추이(Drift)를 가지는 브라운 운동 (Brownian Motion)

$\{X_t, t \geq 0\}$ 는 다음 조건하에서 추이 계수(Drift coefficient)를 가지는 브라운 운동 과정이다.

- (i) $X_0=0$
 - (ii) $\{X_t, t \geq 0\}$ 는 안정적(Stationary), 독립(Independent), 증가(Increments) 속성을 가진다;
 - (iii) X_t 평균 μt , 분산 t 인 정규 분포이다.[17] □
- 추이를 가지는 브라운 운동은 μ 의 비율을 따라 변화가 발생하는 과정이다. 그러므로 $P_{X_t | X_0}(x_t | x_0)$ 를 최초 $t=0$ 일 때 위치 X_0 에서 시간 t 에서 위치 X_t 에 존재하는 확률 분포라고 하면 $t>0$ 인 미래 위치 x_t 는 마코프(Markov) 과정으로 식 (5)를 따른다.

$$P_{X_t | X_0}(x_t | x_0) = P_{X_t | X_0}(x_t | x_0, x_i) \text{ (for any } i < 0) \tag{5}$$

정의 7. 안정적 과정(Stationary Process)

안정적(Stationary) 과정은 $\{X_t, t \geq 0\}$ 인 확률 변수 X_{t_1}, \dots, X_{t_n} 과 $X_{t_1+s}, \dots, X_{t_n+s}$ 가 모든 n, s, t_1, \dots, t_n 에 대해서 동일한 통합 분포(Joint Distribution)를 따르는 과정이다.[17] □

X_t 가 모든 t_1, \dots, t_n 에 대하여 다항(Multivariate) 정규 분포를 따르면 브라운 운동은 가우시안(Gaussian) 과정이 된다. 그리고 객체의 속도는 정의 6 과 7로부터 안정적 가우스-마코프 과정이며, 자기 상관 함수(Auto-correlation Function)인 식 (6)을 만족한다.

$$p_v(\tau) = \sigma^2 e^{-\lambda_p \tau} \tag{6}$$

이 때 σ^2 는 분산 값이며, $\lambda_p \geq 0$ 는 이동 형태로부터 얻어지는 IU 비율이다. 그러므로 $\alpha = e^{-\lambda_p \tau}$ ($0 \leq \alpha \leq 1$)라고 치환하면, 속도는 이산 시간일 때 식 (7)로 표현된다.

$$v_n = \alpha v_{n-1} + (1-\alpha)\mu + \sqrt{1-\alpha^2} \xi_i \tag{7}$$

식 (7)에서 충분히 큰 n 에 대하여 μ 는 v_n 의 평균이고, σ 는 v_n 의 표준 편차이다. 그리고 ξ_n 은 평균 $\mu_{\xi} = 0$

그리고 표준 편차 $\sigma_{\xi} = \sigma$ 이며 안정적 가우시안 과정을 따르는 독립, 비상관(Uncorrelated) 확률 변수이다. 따라서 일반적으로 v_n 는 최초 속도 v_0 로부터 식 (9)와 같이 표현된다.

$$v_n = \alpha^n v_0 + (1-\alpha)\mu + \sqrt{1-\alpha^2} \sum_{i=0}^{n-1} \alpha^{n-i-1} \xi_i \tag{8}$$

x_k 를 k 번째 위치 정보 확인 시간에 예상 위치, 최초 위치로부터 이격 거리를 $s_{k\tau-1}$ 라고 하면

$$x_k = \sum_{i=k_{\tau}^+}^{k_{\tau}^+ \tau - 1} v_i \tag{9}$$

$$s_{k_{\tau}-1} = \sum_{i=0}^{k-1} x_i \tag{10}$$

이며, x_k 와 s_{k-1} 는 각각 식 (11)와 식 (12)를 평균으로 하는 가우시안 확률 변수이다.

$$\mu_{x_k} = \alpha k \frac{1-\alpha^{\tau}}{1-\alpha} v_0 + \left(\tau - \frac{1-\alpha^{\tau}}{1-\alpha} \right) \mu \tag{11}$$

$$\mu_{s_k} = \sum_{i=0}^{k-1} \mu_{x_i} \tag{12}$$

그러므로 이동 객체는 k 번째 위치 확인 지점에서 조건 $|s_{k\tau-1} - \mu_{s_k}|$ 보다 큰 거리 편차가 발생할 때 IU를 수행한다.

5.3 동적 색인 관리 알고리즘

AU 주기 T_u 는 바로 이전 단계의 T_u 로부터 예상되는 AU의 다음 주기는 식 (13)를 따른다.

$$T_u + (1 - e^{-\lambda_p \tau / \lambda_p}) \tag{13}$$

그러므로 색인의 성능 변화와 저장 공간 효율에 영향을 주는 T_u 의 결정은 높은 임의도를 유발하는 임의 이동하는 객체에 의존하게 되며, 각각의 객체는 자신의 이동 형태를 가지므로 적절한 T_u 선정을 통하여 새로운 Branch를 생성한다.

· 도표 1은 $x_k = x_k - \mu_{x_k}$ 와 $s_k = s_k - \mu_{s_k}$ 일 때 사용되는 확률 변수 및 분포 함수를 정의하였다.

도표 1 갱신 간격 결정을 위한 확률 함수 및 변수 정의

함수	설명
$\bar{P}_k(\hat{s}, \hat{x})$	$\hat{s}_k = \hat{s}, \hat{x}_{k-1} = \hat{x}$ 일 때 $k-1$ 까지 갱신이 일어나지 않을 확률
$P_{k-1}(\hat{s}, \hat{x})$	$\hat{s}_k = \hat{s}, \hat{x}_{k-1} = \hat{x}$ 일 때 k 까지 갱신이 일어나지 않을 확률
$P_k(\hat{s}, \hat{x})$	$\hat{s}_k = \hat{s}, \hat{x}_k = \hat{x}$ 일 때 k 까지 갱신이 일어나지 않을 확률
$\bar{F}(k)$	k 까지 갱신이 일어나지 않을 확률
$f(k)$	두번의 연속 갱신 사이 시간이 k 일 때 확률 분포 함수

마지막 갱신은 객체에 따라 IU이거나 AU가 될수 있

으며, 마지막 갱신 이후 다음 IU가 발생하는 시간까지 확률 분포 함수는 식 (14)을 따른다.

$$p_r(t) = \lambda_r e^{-\lambda_r t} \int_0^t \frac{f(T_u)}{1 - e^{-\lambda_r T_u}} dT_u \quad (14)$$

여기서 시간 t 를 구하는 것은 확률적으로 k 번째의 위치 확인 시에 기대되는 객체의 편이(Displacement)가 거리 임계값 이상이 되는 시간이므로, 식 (9)과 (10)로부터 재귀적으로 k 번째의 위치는 식 (15)와 같다.

$$\hat{s}_k = \hat{s}_{k-1} + x_{k-1} \quad (15)$$

동적 색인 관리는 $k=0$ 일 때,의 확률 분포, 거리 편이 임계값 (으로부터 충분히 작은 를 만족하는 k 값을 구하는 것이다.다. 여기서 요구되는 함수는 식 (16)으로 얻어진다.

$$\bar{F}(k) = (16)$$

각각의 객체에 대한 시간 간격 k 를 계산하면, Branch 시간 간격을 구하는 알고리즘은 그림 9와 같다. 알고리즘은 k 값이 성능 저하점에 해당되는 임의도에 도달할 때까지 반복되며, 새로운 Branch가 생성될 때 현재 시간으로부터 k 까지의 시간을 AU 시간 구간으로 설정한다.

Algorithm determine_interval_of_Adaptive_Update

1. calculate k value of all objects and enqueue them in *queue_k* in a sorted order;
2. dequeue minimum k in *queue_k*;
3. while (true)
4. if (*randomness* > *deg-randomness*) then exit;
5. increase number of *variants* and calculate *randomness*;
6. dequeue the current k in *queue_k*;
7. calculate new k of the current object;
8. enqueue the new k in *queue_k*;

그림 9 AU 갱신 주기 선정을 위한 알고리즘

6. 실험 결과

실험 환경은 256MB 메인 메모리를 갖춘 Sun Ultra-60 시스템을 사용하여 제시된 방법의 효율성과 성능을 실험하였다. 실험을 위하여 사용한 데이터는 [18]의 GSTD 소프트웨어를 이용하여 생성하였으며, Bst-트리 색인 구조는 그림 10과 같이 500,000개의 객체를 대상으로 페이지와 데이터 크기를 변경하면서 수행하였다.

미래 위치 질의를 지원하는 색인에 대한 가장 중요한 요소인 생성 또는 갱신에 필요한 시간은 B^{nl}-트리와 R^{*}-트리를 이용하여 실험하고 그림 10에 나타내었다. 미래 시간 질의를 지원하는 색인 구조는 TPR-트리 [1,6]가 있으나, TPR-트리는 R^{*}-트리를 기반으로 노드내의

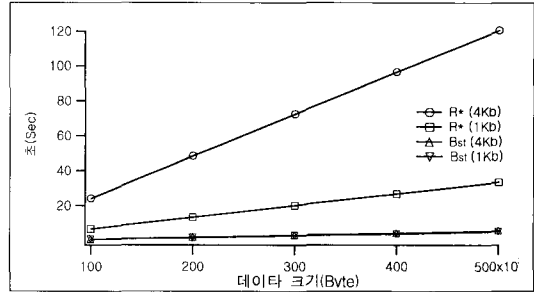


그림 10 삽입에 의한 색인 생성 시간의 성능 비교

최대 속도와 최소 속도를 이용하여 MBR을 구성함으로써 미래 시간 질의를 지원한다. 그러나 지원하는 미래 시간 범위에 따라 TPR-트리의 구성이 변화하므로 B^{nl}-트리의 성능을 R^{*}-트리에서 수행한 결과와 비교하였다. 결과적으로 B^{nl}-트리 키값을 결정하는 피노와 힐버트(Hilbert) 곡선은 모두 결과에 거의 영향을 미치지 않았으며 R^{*}-트리와 비교해서는 대단히 작은 시간이 소요됨을 보인다.

생성된 B^{nl}-트리는 Variant를 성능 저하를 일으키는 임의도 범위 내에서 수용해야 하므로 균일 분포(Uniformly Distributed)된 이동 객체에 대하여 질의 유형 1(시간 단위) 그리고 유형 2(시간 구간) 질의에 대한 임의도 범위를 실험하였다. 그림 11에서는 질의 유형 1 또는 유형 2에 대한 키값의 범위 변화와 키값의 범위가 5%와 10%로 각각 고정했을 때 질의 시간 범위의 변화가 미치는 영향을 보이고 있다. B^{nl}-트리는 질의 유형 1의 경우에는 IU를 허용하지 않는 경우 가장 효율이 좋으며, 질의 조건에서 키 값의 범위와 시간 영역의 범위에 따라 대체로 수용할 수 있는 임의도는 30% 정도 이다.

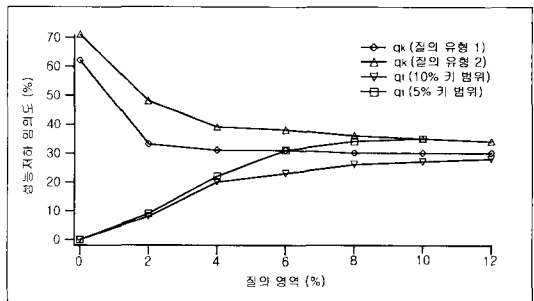


그림 11 임의도와 질의 조건에 따른 성능 저하 평가

또한 그림 12는 20,000 객체 쿼지에 대하여 0% 부터

25%까지의 임의도를 변화시켰을 때 색인 크기의 변화이다. 실험에서 객체의 위치 분포는 균일 또는 정규의 경우 모두 결과에 거의 영향을 미치지 않으며 범위 내에서는 Bst-트리가 R^t-트리에 비하여 훨씬 저장 공간을 작게 차지하는 것을 보인다.

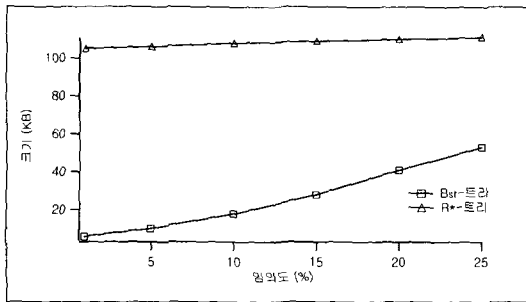


그림 12 Version 노드에서 임의도에 의한 색인 크기 비교

키값의 영역을 5%로 하였을 때 20,000개의 객체 쿼리에 대한 유형 1 질의 처리를 위한 노드 접근 횟수는 그림 13와 같다. 질의 유형 1에서는 임의도의 변화에 직접 영향을 받으므로 0% 임의도에서는 최적값인 B-트리와 같으나 임의도의 증가는 질의 성능을 저하시킨다. 그러나 25% 범위 이내에서 임의도는 성능을 거의 선형으로 감소시키므로 이 범위 이내에서는 색인에 의한 질의 처리 성능을 예측할 수 있다.

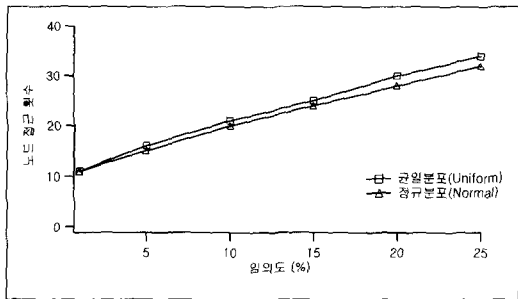


그림 13 Version 노드에서 임의도에 의한 노드 입출력 성능 비교

7. 결론 및 향후 연구 과제

위치 기반 서비스는 현재 및 미래에서 다양하게 요구되는 시공간 데이터베이스의 중요한 응용 분야이다. 본 논문에서는 이동 통신 환경에서 적용 가능하며 위치 기반 서비스에서 요구되는 이동 객체의 추적(Tracking)

및 관리를 지원하는 시공간 색인 구조 Bst-트리를 제안하였다.

본 논문의 기여도는 다음 세가지로 요약할 수 있다.

- (i) 위치 기반 과거 및 미래 질의에 대한 효율적인 색인 방법. 기존의 R-트리에 의한 색인은 색인 구성 및 갱신에 매우 비효율적이나 제시된 색인 방법은 질의 처리 복잡도를 제공하며 색인 구성 및 갱신에 따른 오버헤드를 감소시킨다.
- (ii) 이동 통신 환경에서의 제한 사항을 포함하여 확률 기반 갱신 주기 설정 알고리즘의 제시. 논문에서 제시한 알고리즘은 질의 처리 성능을 보장할 수 있도록 이동 통신 환경에서 사용되는 거리 기반 프로토콜에서 제공되는 위치 정보를 이용하여 예측 기반으로 색인을 관리하는 방법이다.
- (iii) 실험을 통한 Bst-트리의 평가와 최적의 성능 유지를 위한 색인에 영향을 미치는 매개 변수 값을 제시. Bst-트리는 다중 버전 구조로서 시간에 따른 갱신은 색인 구조를 변경시키며 이는 질의 처리 성능에 직접적인 영향을 미친다. 논문에서는 AU와 IU로 인한 색인 구조의 변화가 질의 처리에 미치는 영향을 최소화시키는 확률 기반의 최적값을 제시하였다.

이러한 연구는 아직까지 거의 이루어 지지 않은 분야로 본 논문에서 제시된 Bst-트리와 동적 관리 기법은 위치 기반 서비스를 제공하는 분야에서 직접 적용할 수 있는 색인 구조 및 방법으로 판단된다.

향후 연구과제로서 제시된 색인 구조를 이용한 보다 다양한 실험이 필요하며, 특히 이동 객체의 궤적을 보다 실제 환경을 반영할 수 있는 의미있는 데이터를 생성하여 제시된 방법에 적용하고자 한다. 또한 시간 축 갱신이 일어나는 형태의 히스토그램이 색인에 미치는 영향을 검토하는 것이 필요하며, 보다 질의 처리를 최적화할 수 있는 방법이 요구된다.

참고 문헌

- [1] Saltinis, S., Jensen, C., "Indexing of Moving Objects for Location-Based Services", Proceedings of ICDE, pp. 463-472, 2002.
- [2] Wolfson, O., Sistla, P., Chamberlain, S., Yesha, Y., "Updating and Querying Databases that track Mobile Units", J. of Distributed and Parallel Databases, Vol. 7, pp257-287, 1999.
- [3] Agarwal, P., Erickson, J., "Geometric Range Searching and Its Relatives", Advances in Discrete and Comput. Geom. (B. Chazelle et al.

eds.), American Mathematical Society, Providence, 1998.

[4] Bar-Noy, A., Kessler, L., Sidi, M., "Mobile Users: To Update or not to update", J. of Wireless Networks, 1(2), pp. 175-185, July, 1995.

[5] Wong, V., Leung, V., "Location Management for Next-Generation Personal Communication Networks", IEEE Network, Vol. 14(5), pp. 18-24, 2000.

[6] Saltenis, S., Jensen, C., Leutenegger, S., Lopez, M., "Indexing the Positions of Continuously Moving Objects", Proceedings of SIGMOD, pp. 331-342, 2000.

[7] Basch, J., Guibas, L. Hershberger, J., "Data Structures for Mobile Data", Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms, pp.747-756, 1997.

[8] Agarwal, P., et al., "Efficient Searching with Linear Constraints", J. of Computer and System Sciences 61, pp. 194-216, 2000.

[9] Kollios, G., Gunopulos, D., Tsotras, V., "On Indexing Mobile Objects", Proceedings of PODS 99, pp. 262-272, 1999.

[10] Becker, B., Gschwind, S., Ohler, T., Seeger, B., Widmayer, P., "An Asymptotically Optimal Multiversion B-tree", VLDB Journal 5(4), pp. 264-275, 1996.

[11] Varman, P., Verma, R., "An Efficient Multiversion Access Structure", IEEE TKDE, Vol. 9, No. 3, pp. 391-409, 1997.

[12] Tao, Y., Papadias, D., Zhang, J., "Cost Models for Overlapping and Multi-Version Structures", Proceedings of ICDE, pp. 191-200, 2002.

[13] Liang, B., Haas, Z., "Predictive Distance-Based Mobility Management for PCS Networks", Proceedings of IEEE INFOCOM, Vol. 3, pp. 1377-1384, 1999.

[14] Markl, V., Zirkel, M., Bayer, R., "Processing Operations with Restrictions in RDBMS without External Sorting: The Tetris Algorithm", Proceedings of ICDE, pp. 562-571, 1999.

[15] Bertimas, D., Tsitsiklis, J., Introduction to Linear Optimization, Athena Scientific, 1997.

[16] Ramsak, F. et al., "Integrating the UB-Trec into a Database System Kernel", Proceedings of VLDB, pp.263-272, 2000.

[17] Ross, S., Stochastic Processes, 2nd Ed., John Wiley & Sons, 1996.

[18] Theodoridis, Y., Nascimento, M., "Generating Spatiotemporal Datasets on the WWW", SIGMOD Record 29(3), pp. 39-43, 2000.



박 현 규

1987년 육군사관학교 전산학과 학사. 1992년 Naval Postgraduate School 전산학과 석사. 1999년 ~ 현재 한국과학기술원 전자전산학과 박사과정. 1987년 ~ 현재 육군 전산장교. 관심분야는 시공간 데이터베이스, 분산데이터베이스, C4I 시스템, 시뮬레이션 모델링



손 진 현

1996년 서강대학교 전산학과 학사. 1998년 한국과학기술원 전산학과 석사. 2001년 한국과학기술원 전자전산학과 박사. 2001년 9월 ~ 2002년 8월 한국과학기술원 전자전산학과 박사후 연구원. 2002년 9월 ~ 현재 한양대학교 컴퓨터공학과 전임강사. 관심분야는 분산데이터베이스, 미들웨어, 워크플로우, 컴포넌트, CORBA, 객체지향모델링



김 명 호

1982년 서울대학교 컴퓨터 공학과 학사. 1984년 서울대학교 컴퓨터 공학과 석사. 1989년 MICHIGAN 주립대 전산학과 박사. 1989년 MICHIGAN 주립대 연구원. 1989년 ~ 1993년 한국과학기술원 조교수. 1993년 ~ 1999년 한국과학기술원 부교수. 1999년 ~ 현재 한국과학기술원 교수. 1992년 ~ 1993년 개방형 컴퓨터 통신 연구회(OSIA) 분산 트랜잭션 처리 분과위(TG-TP) 의장. 1996년 ~ 1998년 한국정보과학회 DB연구회지 책임편집위원. 관심분야는 데이터베이스, 분산트랜잭션, 분산시스템, 워크플로우