

효과적인 차량 검색을 위한 이동 객체 색인 기법

정영진[○], 최현미, 류근호

충북대학교 데이터베이스 연구실
{yjjeong, hmchoi, khryu}@dmlab.chungbuk.ac.kr

A Moving Object Index Technique to Effectively Search Vehicles

Young-Jin Jung[○], Hyon-Mi Choi, Keun-Ho Ryu
Database Laboratory, Chungbuk National University

요 약

현실 세계에서 자동차, 비행기, 태풍의 이동 등과 같이 시간에 따라 연속적으로 위치 및 모양이 변화하는 객체를 이동 객체라 한다. 이와 같은 이동 객체 정보를 다루기 위해서는 연속적으로 변화하는 정보를 저장해야 하며, 이로 인해 대용량 정보를 효과적으로 다루기 위한 색인이 필요하다. 특히, LBS를 제공을 위한 GPS 환경에서의 차량 관리 시스템 등에서는 이동 객체 관리를 위해 빠른 검색을 필요로 한다. 그리고, 기존의 이동 객체 색인은 대부분 R-트리의 구조를 가지기 때문에, 데드 스페이스 및 오버랩 등 R-트리의 문제점을 갖게 되고, 고려하는 초점에 따라 이 문제가 더 커진다. 따라서, 이 논문에서는 PDA를 활용한 차량 추적 시스템을 개발하고, R-트리의 문제점을 해결하기 위한 MP-트리 노드 연산 기법을 제안한다. 제안된 노드 연산 기법은 하위 노드의 경계 정보를 순서대로 저장함으로써, 데드 스페이스 및 오버랩으로 인한 불필요한 검색 비용과 분할 비용을 줄이고, 특정 시점 질의 및 시공간 범위 질의를 효과적으로 처리한다. 기존 이동 객체 색인과 비교한 실험으로부터 이동 객체 질의 처리 및 공간 활용에 대해 노드 연산이 유용하게 쓰임을 확인하였다.

1. 서 론

최근 GPS(Global Positioning System)와 무선 데이터 전송 능력이 있는 휴대용 전화기의 보급 및 이동/무선 컴퓨팅 기술의 발달로 인해 이를 이용한 응용 서비스에 대한 관심이 고조되고 있다. 특히 이동 차량, 비행기, 선박, 휴대용 전화기, 노트북 컴퓨터 등을 이용한 위치기반서비스(LBS: Location Based Services)가 무선 인터넷 시장의 중요한 이슈가 되고 있다[1]. LBS의 주요 응용 대상인 차량, 비행기, 선박, 휴대용 전화기, 노트북 컴퓨터 등은 모두 자유롭게 이동하면서 그 위치를 변경할 수 있는 특징을 가진다. 이와 같이 시간의 흐름에 따라 객체가 이동하면서 그 위치 및 모양을 연속적으로 변경하는 특징을 가지는 데이터를 이동 객체(Moving Object)[2, 3]라 하며, LBS에서 가장 기본적이면서도 중요한 역할이 바로 이동객체의 위치정보관리기법이다. 특히, 이동 객체에 대한 위치 정보는 시간이 흐름에 따라 그 변화량이 방대하게 증가되기 때문에 대용량 데이터를 관리하기 위한 데이터베이스 시스템의 활용이 반드시 필요하게 된다. 또한, 연속적으로 위치 정보가 변경되는 이동 객체에 대하여 데이터베이스를 이용하여 저장 및 관리하기 위해서는 연속적인 객체의 위치 정보 변화를 데이터베이스 내에

표현하고 이를 이용한 질의 및 검색이 가능해야 한다. 따라서 저장된 많은 정보를 효율적으로 검색하는 인덱스 구조가 필요하다.

기존의 이동 객체 색인들은 처리하는 데이터에 따라 크게 두 부분으로 나눈다. 첫째는 이동 객체의 과거 이력정보 및 궤적의 색인이고, 두 번째는 이동 객체의 현재 위치와 가까운 미래 위치의 색인이다[4]. 그리고 이런 응용에 대한 대부분의 이동 객체 색인은 R-트리[5]를 응용한 구조를 가지기 때문에, 데드 스페이스, 오버랩 등 R-트리의 문제를 그대로 가지고 있으며, 응용에 따라 문제점이 더 커지는 경우도 있다.

이를 해결하기 위하여 이 논문에서는 기존의 R-트리를 수정한 MP-트리 노드 연산 기법을 제안한다. 제안된 MP-트리는 3차원 MBB(Minimum Bounded Box)의 각 축에 하위 노드들을 순서대로 정렬하여 저장함으로써, 데드 스페이스를 줄이고 이동 객체의 과거와 현재에 대한 특정 시점 질의 및 시공간 범위 질의를 효과적으로 처리한다.

이 논문의 전체적인 구성은 다음과 같다. 먼저 2장에서는 이동 객체 질의와 이동 객체에 대한 색인들을 소개하고 문제점을 알아본다. 3장에서는 개발된 차량 추적 시스템을 소개하고, 4장에서는 이 논문에서 제안

한 MP-트리의 구조에 대해 설명한다. 5장에서는 노드 연산의 개념 및 그 쓰임에 대해 예를 들어 설명한다. 6장에서는 이동 객체 관련 질의에 대해 MP-트리와 기존의 STR-트리, R-트리와 비교 실험하며, 차량 추적 시스템의 질의 처리 결과를 설명한다. 마지막으로 7장에서는 결론을 맺는다.

2. 관련 연구

이동 점 객체 색인 구조의 관련 연구로 이동 객체와 이동 객체 색인에 대한 기존의 연구들을 소개하고 문제점을 파악한다.

이동 객체는 시간이 흐름에 따라 객체의 위치나 영역 같은 공간 정보가 연속적으로 변화하는 객체를 말하며 크게 이동 점(moving point)과 이동 영역(moving region)으로 나눌 수 있다[6, 7]. 이동 점은 시간에 따라 객체의 위치가 변하는 것으로 이런 이동 점의 예는 택시, 배, 비행기, 사람 등이 있고, 이동 영역은 시간에 따라 객체의 위치뿐만 아니라 모양까지 변하는 것으로 이러한 이동 영역의 예는 적조현상, 바다에서의 기름 유출에 따른 오염, 오존층의 변화, 폭풍의 이동 경로 등이 있다.

기존의 이동 객체 정보를 다루는 색인들은 응용에 따라 크게 두 종류로 나눈다[4]. 그것은 이동 객체의 과거 이력정보 및 궤적의 색인과 이동 객체의 현재 위치와 가까운 미래 위치의 색인이다.

과거 이력 정보 및 궤적의 색인은 Pfoser가 제안한 STR-트리(Spatio-Temporal R-tree)[8], TB-트리(Trajectory Bundle tree)[4] 그리고, 이계영이 제안한 CR-트리(Combined R-tree) [9] 등이 있다. 3차원 MBB 구조를 갖는 R-트리에 이동 객체의 궤적 보호 및 공간 속성을 고려한 색인 구조가 STR-트리이다. STR-트리는 저장되는 객체의 수가 많아지면, 성능이 떨어지고, 궤적에 관련된 질의를 제외하면, R-트리보다 좋지 않은 성능을 보인다. TB-트리도 3차원 MBB 구조를 갖지만, 좌표를 생각지 않고 단지 궤적만을 고려한다. 그리고 연결 리스트를 사용하여 이동 객체의 궤적을 시간 순서대로 연결함으로써 궤적질의를 용이하게 처리한다. 트리 구조는 STR-트리와 같지만 궤적 보호를 특히 중요하게 생각하기 때문에, 시공간 범위 질의에서 STR-트리와 R-트리보다 검색 속도가 느리다. CR-트리는 기존의 R-트리에 TB-트리와 같이 연결 리스트를 사용하여 궤적을 연결한 구조를 갖고 있다. 따라서 시공간 범위 질의에서 R-트리만큼의 성능을 가지며 연결리스트를 사용하여 궤적질의에서도 효과적이다. 그러나 R-트리와 TB-트리를 혼합한 형태이기 때문에 STR-트리나 TB-트리보다 많은 저장 공간

을 가지는 단점이 있다.

현재 위치 및 가까운 미래의 위치를 다루는 색인은 TPR-트리(Time Parameterized R-tree)[10], TPR*-tree[11] 등이 있다. TPR-트리는 이동 객체의 속도와 방향을 고려하여 대략적인 가까운 미래의 위치를 계산해내고, 계산된 위치까지 고려하여 트리를 구성함으로써, 빈번한 업데이트를 줄였고, TPR*-tree는 노드의 분할을 효과적으로 처리하여, 검색 성능을 향상시켰다.

과거 이력정보와 궤적을 다루는 색인에서, 이동 객체의 좌표와 궤적을 둘 다 고려할 경우와 궤적만을 고려하여 노드를 구성하는 R-트리의 응용들은 오히려 데드 스페이스, 오버랩 등의 단점이 더 커지게 되어 효과적인 검색이 어려워진다. 이 논문에서는 이 문제를 해결하기 위하여 MP-트리 노드 연산을 제안하였다. 그리고 이런 이동 객체 색인들은 위치기반 서비스(Location Based Services) 및 전장 정보 분석[12, 13] 등의 응용 분야에서 시간과 공간 속성을 포함한 시공간 질의 처리[14, 15]를 위해 효과적인 데이터 검색을 해준다.

3. PDA 기반 차량 추적 시스템

이 장에서는 MP-트리를 활용한 PDA 기반 차량 추적 시스템에 대해 설명한다.

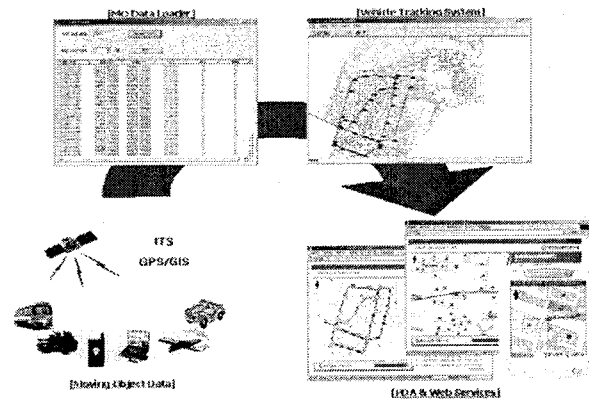


그림 1. 차량 추적 시스템 구성도

그림 1은 차량 추적 시스템의 전체 구성도를 나타낸다. GPS 및 ITS에서 각 차량의 위치 데이터를 PDA를 통해 보내오면, MODataLoader에서 차량 데이터를 이동 객체 데이터베이스에 저장하고, 이를 이용해서, 차량 추적 시스템 엔진에선, 질의 처리를 수행한다. 또, 처리된 결과는 GML을 이용하여 인터넷과 PDA로 결과를 제공할 수 있다.

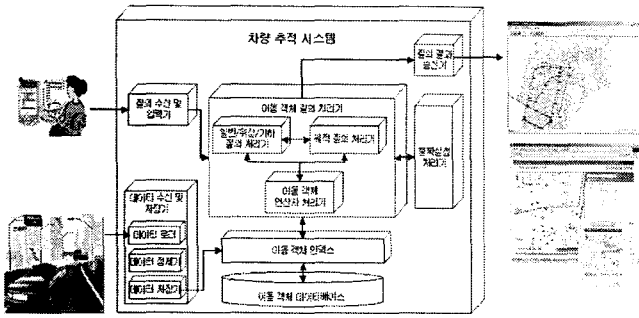


그림 2. PDA 기반 차량 추적 시스템

그림 2의 이동 객체 관리 시스템은 차량 위치 정보를 받아 저장하는 데이터 수신 및 저장기와 이동 객체 데이터를 저장하는 이동 객체 데이터베이스, 사용자의 질의를 받아 질의 처리기에 제공하는 질의 수신 및 입력기, 이동 객체 질의를 처리하는 질의 처리기, 위치 불확실성 처리를 위한 불확실성 처리기, 계산된 결과를 반환하여 인터넷 및 PDA로 결과를 전달하는 질의 결과 송신기로 구성된다. 제안된 MP-트리는 이동 객체 인덱스에 활용되어 특정 시점 질의 및 궤적 질의를 처리하였다. 위 처리 과정은 실제 PDA 유저의 데이터를 바탕으로 구현되었으며, MP-트리가 유용하게 쓰임을 확인하였다.

4. MP-트리의 구조

MP-트리는 기존의 R-트리를 기반으로 노드 연산을 사용하여 특정 시점 질의 및 시공간 범위 질의를 효과적으로 처리하는 이동 객체 색인이다. 그림 3은 R-트리를 사용하여 이동 객체 궤적을 나타낸 것이다. 이동 객체의 이동을 표현하는 라인 세그먼트를 3차원 MBB로 표현하여 이동 객체의 궤적을 저장한다.

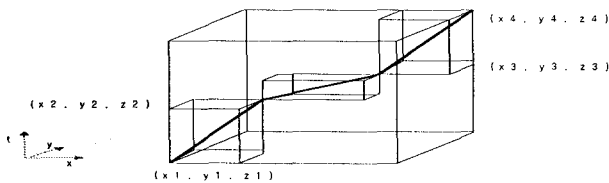


그림 3. R-트리를 사용한 이동 객체 궤적의 표현

하위 노드들로 표현된 라인 세그먼트의 끝점을 서로 연결하면 궤적이 되며, 이동 객체가 지나온 경로를 나타낸다. 이와 같은 접근에서는 궤적이 서로 겹치는 상황 등으로 데드 스페이스 및 오버랩이 많이 발생할 수 있는데, 이로 인한 불필요한 검색을 줄이기 위해 MP-트리는 노드 연산을 활용하여 특정 시점 및 시공간 범위 질의를 효과적으로 처리한다.

MP-트리는 그림 4와 같이 리프 노드를 제외한 노드들은 각 축에 대한 노드 저장소(Projection storage)

를 가진다. 이 노드 저장소는 하위 노드들의 정보를 각 축에 대해 순서대로 저장하여 효과적인 질의 처리를 하도록 도와준다.

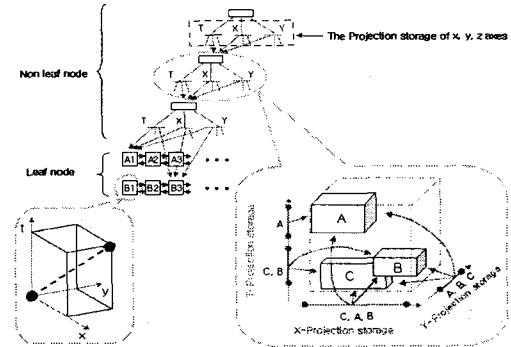


그림 4. MP-트리 구조

년-리프 노드와 리프 노드는 x, y, t 축에 대한 노드 저장소를 통해 하위 노드를 가리킨다. 그리고, 리프 노드 엔트리는 앞, 뒤 시간대에 대해서 연결 리스트로 연결되어 있기 때문에, 이동 객체 궤적에 대한 질의를 처리할 때 유용하다.

5. MP-트리의 노드 연산

제안된 MP-트리에서 하위 노드를 좀 더 효과적으로 검색하기 위해 사용된 노드 연산을 설명한다. MP-트리에 대한 이해를 돕기 위해 노드 연산 및 노드 저장소를 정의한다.

정의 1. 노드 연산(Projection operation)은 한 노드의 각 축마다 하위 노드의 경계를 얻은 후, 서로 교차하는 공통된 경계를 각 축의 노드 저장소에 저장하는 연산이다.

정의 2. 노드 저장소(Projection storage)는 노드 연산을 통해 얻은 하위 노드의 공통된 경계를 저장하는 장소로써, 노드 검색 시에 오버랩 및 데드 스페이스로 인한 불필요한 연산을 줄이는데 쓰인다.

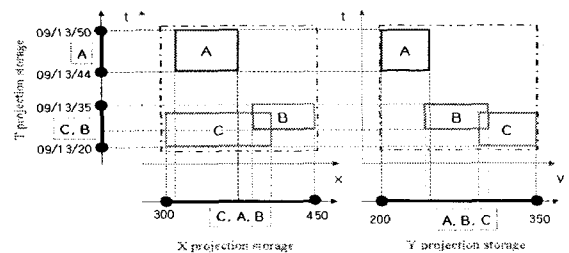


그림 5. MP-트리의 노드 연산

그림 5는 하위 노드의 경계 정보를 얻고 그 중 공통된 경계 정보를 노드 저장소에 저장하는 노드 연산을 나타낸다. 각 축의 노드 저장소는 하위 노드의 정보를 순서대로 정렬하여 저장하기 때문에 특정 시점에 대한 질의나 시공간 범위 질의를 할 때, 질의 범위에 해당하지 않는 노드에 대한 불필요한 검색을 줄여서 검색의 효율을 높일 수 있다.

표 1. MBR과 각 노드들의 경계 정보

	X 축	Y 축	T 축	
			From_time	To_time
MBB	300, 450	200, 350	2003/09/09/13/20	2003/09/09/13/50
A	310, 380	200, 260	2003/09/09/13/44	2003/09/09/13/50
B	390, 450	250, 320	2003/09/09/13/28	2003/09/09/13/35
C	300, 400	300, 350	2003/09/09/13/20	2003/09/09/13/30

그림 5에 있는 각 노드의 경계 정보를 정리하면 표 1과 같다. 기본적인 구조는 R-트리와 같기 때문에 상위 노드 MBB는 하위 노드 A, B, C의 시공간 경계를 모두 포함한다.

표 2. MBB의 각 축에 대한 노드 저장소 정보

Projection 축	범위		객체 pointer
X 축	300	450	C, A, B
Y 축	200	350	A, B, C
T 축	2003/09/09/13/20	2003/09/09/13/35	C, B
	2003/09/09/13/44	2003/09/09/13/50	A

상위 MBB의 노드 저장소 정보를 정리하면 표 2와 같다. 저장소의 구성을 보면, 하위 노드들의 경계 정보를 모아 특정 간격 안에 해당되는 객체를 순서대로 저장한다. 이 작업을 반복해서 하위 노드들을 모두 각 축에 대하여 순서대로 저장한다. 이 정보는 질의할 때, 각 축에서 질의 범위 안에 해당되는 값을 빠르게 찾아주며 데드 스페이스 및 오버랩으로 인한 불필요한 검색을 줄일 수 있다. 데드 스페이스를 최소화하는 것은 검색할 때 알맞은 데이터를 찾지 못하는 확률[16]을 감소시킨다.

이 논문에서는 이동 객체 정보가 시간에 따라 무한정 들어온다고 가정하기 때문에, 공간에 대한 정보보다 시간에 대한 정보를 먼저 처리하는 것이 검색 시간을 줄일 수 있다. 따라서 시공간 범위 질의나 특정 시점에 대한 질의를 처리할 경우 T축-> X축-> Y축 순으로 찾는다. 이를 위해 입력과 분할 알고리즘에서도 앞과 같은 순서를 우선시 하여 처리한다. 만약 MP-트리의 구조를 적용하는 응용분야가 공간에 대해, 또는 특정 축에 대해서 넓은 범위가 존재할 경우, 검색하는 순서를 다르게 하여 검색 효율을 높일 수 있다. 특정 시점에 대한 시공간 질의를 처리할 때 노드 연산이 어떻게 유용하게 쓰이는지를 표 2의 데이터를 활용하여 설명한다.

(질의 1) "2003년 9월 9일 오후 1시 40분에 객체의 위치를 모두 검색하시오"

표 3. 질의 2의 각 축에 대한 정보 체크

검색하는 축	조건에 만족하는 객체
T	

질의 1에 답하기 위해 각 축에 대해 체크된 객체는 표 3과 같다. 우선 T축에 대해서 체크했을 때, 단지 T축의 링크의 경계를 검색하는 것에 의해서 조건에 만

족하는 객체가 없다는 사실을 알 수 있다. 만약 하위 노드 집합의 경계를 저장하는 노드 저장소가 없다면, 검색되는 노드의 하위 노드를 모두 검색한 후에 질의를 마치게 되어 불필요한 검색을 하게 된다.

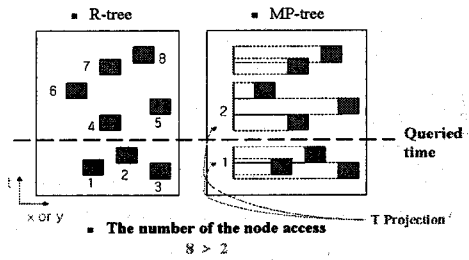


그림 6. 질의 1에 대한 노드 검색 수 비교

질의 1에 대한 검색을 R-트리와 비교하면 그림 6과 같다. 이것은 데드 스페이스에 대한 질의이다. 하위 노드의 공통된 경계를 저장하는 노드저장소가 유용하게 쓰임을 알 수 있다. 또한 정렬된 정보는 데드 스페이스 뿐만 아니라 오버랩 된 부분에 대한 추가적인 검색에서도 위와 같은 이유로 불필요한 검색을 줄인다.

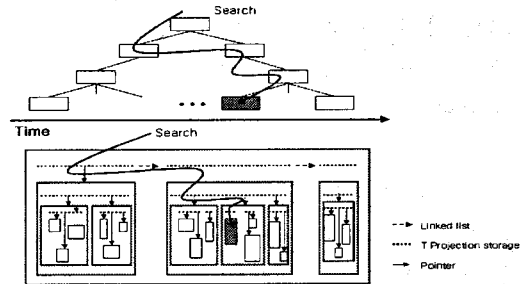


그림 7. MP-트리의 검색 과정

그림 7은 MP-트리의 전체적인 검색 과정을 알기 쉽게 나타낸 것이다. 하위 노드의 경계 정보를 갖고 있는 노드 저장소가 검색시에 유용하게 쓰임을 알 수 있다.

6. 실험 및 평가

이 장에서는 시공간 범위 질의 등과 같은 이동 객체 관련 질의에 대해서, 이 논문에서 구현한 MP-트리의 노드 접근 수를 기존의 STR-트리, R-트리, CR-트리를 가지고, 아래 표와 같이 이동 객체 관련 질의에 대한 노드 접근 수를 비교 및 평가한다. 그리고, 차량 추적 시스템에서의 질의 처리 결과를 넣었다.

표 4. 질의 데이터

질의 종류	매개 변수	실험 데이터 범위					
		10%	25%	50%	75%	90%	
특정 시점 질의	질의 위치	10%	25%	50%	75%	90%	
	공간 범위	10%	25%	50%	75%	100%	
시공간 범위 질의	공간 범위	10%	25%	50%	75%	100%	
	시간 범위	1%	5%	10%	15%	20%	
객체 점의	시간, 공간 범위	시간	1%	5%	10%	15%	20%
		공간	10%	25%	50%	75%	100%
객체 점의	질의 위치	시간 10%			시간 20%		

실험할 질의 종류와 실험 데이터는 표 4에 나타나

있다. 이와 같은 질의에 대해 10만, 25만, 50만, 75만, 100만개의 이동 점 객체 데이터를 입력시키고 노드 접근 수를 체크하여 각 트리의 성능을 비교하고 평가한다. 대량의 실제 데이터는 구하기 어렵기 때문에, 가상의 데이터로 실험하였으며, 논문의 크기 제한으로 인해, 특정 시점 질의 결과만 다루었다.

6.1 특정 시점 질의

특정 시점 질의는 “2003년 8월 13일 오후 4시 32분의 543번 택시의 위치를 구하시오”와 같이 한 시점에 대한 위치 질의이다. 먼저, 검색 비용이 과거의 시점 질의와 현재의 시점 질의를 처리할 때 달라지는지 알아보기 위해, 각 트리에 입력된 데이터의 전체 시간 중 10%, 25%, 50%, 75%, 90% 지점에 대해 질의한다.

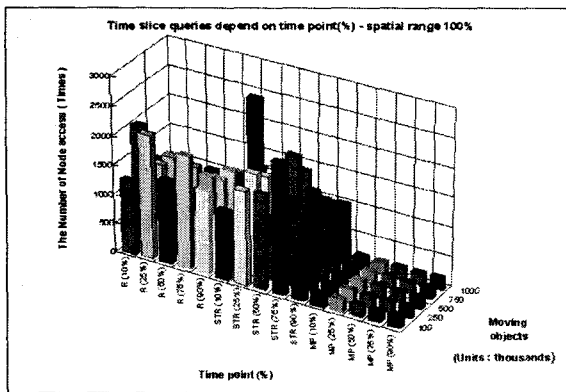


그림 8. 시점에 따른 특정 지점 질의

그림 8은 각 시점에 따른 노드 접근 수를 나타낸다. STR-트리와 R-트리가 시간축의 50%, 90% 지점에 대한 검색에서, 다른 시점보다 더 적은 노드 접근 수를 나타내는 반면, MP-트리는 이보다 적은 노드 접근 수로 어떤 시점에서든 동일한 노드 접근 수를 보였다. 시점에 따른 특정 시점 질의에 대한 노드 접근 수는 $MP < R < STR$ 이다.

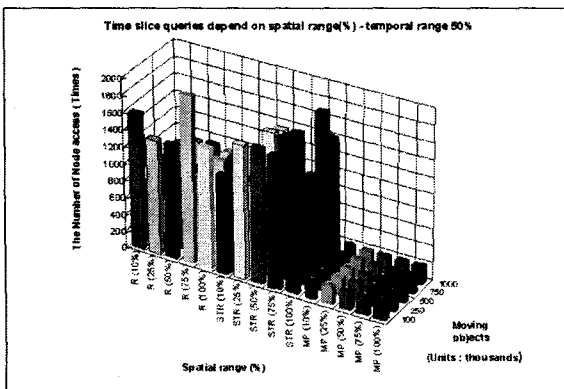


그림 9. 공간 범위에 따른 특정 시점 질의

동일한 시점(시간 50% 지점)에서, 공간 범위를 다르게 했을 때의 결과는 그림 9와 같다. 공간 범위에 따른 특정 시점 질의에 대한 노드 접근 수는 $MP < R$

$< STR$ 이다. 위의 실험 결과에서 MP-트리는 시간에 대해 고르게 분할되어 있으며, t축에 대한 노드가 유용하게 쓰임을 알 수 있다. 앞의 실험을 정리해 보면, 이 논문에서 제안한 MP-트리가 기존의 STR-트리, R-트리보다 특정 시점 질의, 시공간 범위 질의에서 좀 더 좋은 성능을 보였고, 이로 인해 궤적에 대한 결합 질의에서도 기존의 CR-트리보다 빠른 검색 속도를 보였다. 이로써, 노드연산을 활용한 MP-트리는 하위 노드를 각 축에 대해서 순서대로 정렬하여 저장함으로써, 데드 스페이스 및 오버랩으로 인한 불필요한 검색을 감소시키고, 검색 효율을 높였다.

6.2 차량 추적 시스템의 질의 처리

PDA 기반 차량 추적 시스템은 이동 객체 위치 및 궤적 질의를 노드 연산 기법을 활용하여 효과적으로 처리하고 이를 인터넷 및 PDA 로 제공하는 시스템이다.

ID	X	Y	...
21111	21111	44900	...
21112	21112	44900	...
21113	21113	44900	...
21114	21114	44900	...
21115	21115	44900	...
21116	21116	44900	...
21117	21117	44900	...
21118	21118	44900	...
21119	21119	44900	...
21120	21120	44900	...
21121	21121	44900	...
21122	21122	44900	...
21123	21123	44900	...
21124	21124	44900	...
21125	21125	44900	...
21126	21126	44900	...
21127	21127	44900	...
21128	21128	44900	...
21129	21129	44900	...
21130	21130	44900	...

그림 10. 이동 객체 데이터 로더

그림 10의 이동 객체 데이터 로더는 이동 객체 정보를 원하는 시간 간격으로 데이터베이스에 저장하는 입력 모듈이다.

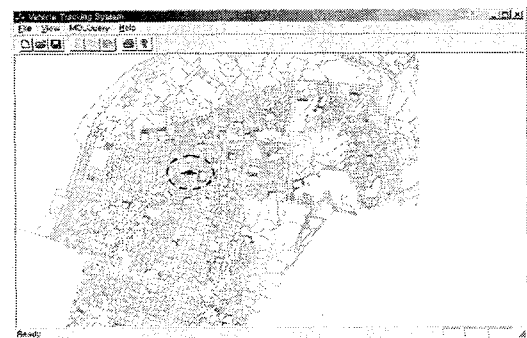


그림 11. 특정 시점 질의 처리 결과

그림 11은 2003년 2월 7일 17시 3분, X축: 211000.752 ~ 215000.601, Y축: 449000.833 ~ 450300.411의 범위 안의 'SL82AH2468' 이동 객체를 검색하는 질의 결과이다.

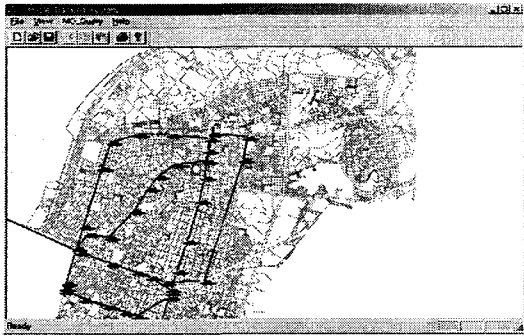


그림 12. 궤적 질의 처리 결과

그림 12은 2003년 2월 7일 16시 32분부터 18시 19분 까지, X축: 211000.752 ~ 215000.601, Y축: 449000.833 ~ 450300.411까지의 범위 안에 포함된 'SL81BA3578' 이동 객체를 검색하는 질의 결과이다.

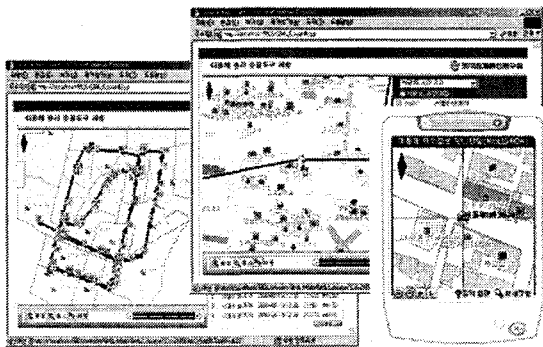


그림 13. 인터넷 및 PDA 서비스

그림 13는 그림 12에서 처리된 결과를 인터넷 및 PDA로 제공한 결과이다. 위 처리 과정은 실제 PDA 유저의 데이터를 바탕으로 구현되었으며, MP-트리가 유용하게 쓰임을 확인하였다.

7. 결 론

기존의 이동 객체 정보를 다루는 색인들은 대부분 R-트리를 응용한 구조를 가지기 때문에, 데드 스페이스 및 오버랩 등 R-트리의 문제를 그대로 가지고 있다. 단지 궤적만을 고려하는 등의 여러 응용에 따라 이와 같은 단점은 더 커지기도 하며, 효과적인 검색을 방해한다.

이와 같은 문제를 해결하기 위해서 이 논문에서는 MP-트리를 제안하였다. 제안된 MP-트리는 노드 연산을 수행하여 각 MBB의 하위 노드들을 각 축에 대해 순서대로 정렬하여 저장함으로써, 이동 점 객체의 과거와 현재에 대해서 특정 시점에 대한 질의 및 시공간 범위 질의를 효과적으로 처리하였다.

기존의 이동 객체 색인과 저장 공간을 비교했을 때, 비교적 적은 공간을 차지함을 알 수 있는데, 이는 각 축에 대한 노드 연산이 분할 알고리즘에서의 비용을 줄일 뿐만 아니라 노드들의 오버랩을 효과적으로 줄여 주기 때문이다. 실험 결과를 종합해 봤을 때, 노드연산

은 모든 축에 적용되는 것보다, 각 응용마다 중요한 개의 축을 선택하여 실행하는 것이 가장 좋은 형태로 생각된다. 이 형태에서는 입력 및 갱신할 때, R-트리보다 한 축의 노드 저장소를 체크하는 시간이 더 필요한 단점이 있지만, 분할 및 질의 처리 그리고, 공간 활용 측면에서 장점을 보인다.

앞으로는 이동 객체 정보의 불확실성 처리와 궤적 질의 및 범위 질의 등, 좀 더 실생활과 가까운 질의 처리를 위한 실험을 수행하며, 적절한 MBR 구조 및 알고리즘을 연구할 예정이다.

[참고 문헌]

- [1] Jeffery H. Reed, Kevin J. Krizman, Brian D. Woerner, and Theodore S. Rappaport, "An Overview of the Challenges and Progress in Meeting the E-911 Requirement for Location Service", IEEE Communication Magazine, pp. 33~37, 1998.
- [2] Martin Erwig, Ralf Hartmut Guting, Markus Schneider and Michalis Vazirgiannis, "Spatio-Temporal Data Types : An Approach to Modeling and Querying Moving Object in Databases," CHOROCHRONOS Technical Report CH-97-08, December, 1997.
- [3] Ralf Hartmut Guting, Michael H. Bohlen, Martin Erwig, Christian S. Jensen, Nikos A. Lorentzos, Markus Schneider, and Michalis Vazirgiannis, "A Foundation for Representing and Querying Moving Objects," ACM Transactions on Database Systems, Vol. 25, No.1, pp. 1-42, March, 2000.
- [4] Dieter Pfoser, Christian S. Jensen, and Yannis Theodoridis, "Novel Approaches in Query Processing for Moving Objects," CHOROCHRONOS Technical Report CH-00-3, February, 2000.
- [5] Guttman, "A.:R-trees: a Dynamic Index Structure for Spatial Searching," In Proceedings of the ACM-SIGMOD Conference on the Management of Data, pp. 47-57, 1984.
- [6] Luca Forlizzi, Ralf Hartmut Guting, Enrico Nardelli, and Markus Schneider, "A Data Model and Data Structures for Moving Objects Databases," Proc. ACM SIGMOD Conf., Dallas, Texas, pp. 319-330, 2000.
- [7] Martin Erwig, and M. Schneider, "Developents in Spatio-Temporal Query Languages," In Proceedings of DEXA Workshop on Spatio-Temporal Data Models and Languages, 1999.
- [8] Dieter Pfoser, Yannis Theodoridis, and Christian S. Jensen, "Indexing Trajectories of Moving Point Objects," CHOROCHRONOS Technical Report CH-99-03, October, 1999.
- [9] 이계영, 권동섭, 송병호, 이석호, "이동 객체의 궤적 검색을 위한 인덱스 구조", 제28회 한국정보과학회 추계학술발표회, pp. 217-219, 2001년 10월.
- [10] S. Saltenis, C. Jensen, S. Leutenegger and M. Lopez, "Indexing the Positions of Continuously Moving Objects," In Proc. of the 19th ACM-SIGMOD Int. Conf. on Management of Data, Dallas, Texas, 2000.
- [11] Yufei Tao, Dimitris Papadias, Jimeng Sun, "The TPR*-tree: An Optimized Spatio-Temporal ACcess Method for Predictive Queries", VLDB2003, 2003
- [12] K. H. Ryu and Y. A. Ahn, "Application of Moving Objects and Spatiotemporal Reasoning", TimeCenter TR-58, 2001.
- [13] Seong Seung Park, Yun Ae Ahn, Keun Ho Ryu, "Moving Objects Spatiotemporal Reasoning Model for Battlefield Analysis" Proceeding on MGA2001, Vol. 33, No. 4, pp. 108~113, April 1, 2001.
- [14] Dong Ho Kim, Keun Ho Ryu, Hong Soo Kim, "A spatiotemporal database model and query language", The Journal of Systems and Software 55, pp. 129-149, 2000
- [15] Dong Ho Kim, Keun Ho Ryu, Chee Hang Park "Design and implementation of spatiotemporal database query processing system", The Journal of Systems and Software 60, pp. 37-49, 2002
- [16] Rasa Bluijute, Christian S. Jensen, Simonas Saltenis, Giedrius Slijvinskas, "R-Tree Based Indexing of Now-Relative Bitemporal Data," VLDB 1998: pp. 345-356, 1998.