

공간 네트워크상의 이동객체를 위한 궤적기반 색인구조 및 질의 처리 알고리즘의 설계

*엄정호⁰, *장재우, **양재동
 *전북대학교 컴퓨터 공학과, ** (주) 케이테크
 {jhum⁰, jwchang}@dblab.chonbuk.ac.kr

Design of Trajectory-based Index Structure and Query Processing Algorithm for Moving Object on Spatial Network

*Jungho Um⁰, *Jaewoo Chang, **Jae-Dong Yang

*Dept. of Computer Engineering, Chonbuk National University, **K-Tech

요 약

유클리디언(Euclidean)공간을 가정한 이동객체를 위한 색인구조 연구가 많이 진행되어 왔다. 그러나 LBS(Location Based System) 및 텔레매틱스 응용이 활발해짐에 따라, 이를 효과적으로 지원하기 위해서 도로, 철도와 같은 공간 네트워크를 고려한 색인구조 연구가 요구된다. 따라서 본 논문에서는 공간 네트워크상에서 이동객체를 효율적으로 색인할 수 있는 궤적기반 색인구조를 설계한다. 이를 위해 공간 데이터와 시간 데이터를 분리하여 색인하고, 실제계에서 공간네트워크상의 이동객체에 대한 질의를 시공간+궤적 질의, k-최근접 질의, 시간+궤적 질의로 분류하고, 이를 위한 질의 처리 알고리즘을 설계한다.

1. 서 론

공간 데이터베이스상의 이동객체를 위한 색인구조 연구 (3DR-Tree[1], TB-Tree, STR-Tree[2], Mv3R-Tree [3])가 많이 진행되어 왔다. 이러한 연구는 유클리디언 (Euclidean)공간을 고려한다. 그러나 LBS(Location Based System) 및 텔레매틱스 응용이 활발해짐에 따라, 도로, 철도와 같은 공간 네트워크를 고려한 색인구조 연구가 요구된다. 따라서 본 논문에서는 공간 네트워크상에서 이동객체를 색인할 수 있는 궤적기반 색인구조를 설계한다. 이동객체는 시간에 따라 이동객체의 궤적이 쌓여 가는 특성이 있다. 이러한 특성을 고려하여 저장할 데이터를 크게 두 가지로 분류한다. 첫째, 한 번 삽입하면 갱신이 거의 없는 공간 데이터와 둘째, 삽입이 빈번히 발생하는 시간 데이터(이동객체 데이터의 궤적)이다. 이러한 분류에 근거하여, 공간 데이터 즉, 공간 네트워크를 구성하는 에지를 색인하는 구조를 설계하고, 시간 데이터는 각각의 에지를 지나는 이동객체의 시간에 해당하는 데이터를 색인하는 구조를 설계한다.

논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개하고, 3장에서는 본 논문에서 설계한 공간 네트워크상의 이동객체를 위한 색인구조를 제시한다. 마지막으로 4장에서는 결론을 제시한다.

2. 관련 연구

공간 네트워크를 위한 연구는 크게 공간 네트워크상의 이동 객체를 처리하기 위한 연구[4,6]와 공간 네트워크상의 공간질의(영역 질의, k-최근접 질의, 조인 질의 등)를 처리하기 위한 연구[5]로 진행되어 왔다. 첫째, 아테

네 국립기술대학의 연구에서는 고정 공간 네트워크상에서의 이동객체를 처리할 수 있는 색인구조(Fixed Network R-tree (FNR-tree))를 제안하였다.[4]

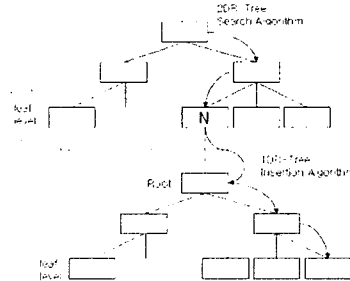


그림 1. FNR-트리 구조

FNR-트리의 구조는 (그림 1)과 같다. FNR-트리는 공간 네트워크를 2DR-트리로 색인하고, 이동객체는 2DR-트리의 리프노드에 1DR-트리를 두어 색인한다. 공간과 시간의 분리로 3DR-트리보다 시공간 영역질의에서 좋은 성능을 보이지만 이동객체의 삽입시 두 개의 트리를 검색해야 하는 단점이 있다.

둘째, 홍콩과학원의 연구에서는 도로나 철도와 같은 공간 네트워크 연결에 따른 클러스터를 형성하여 노드들의 연결 리스트로 저장한다 [5]. 에지들을 polyline 으로 저장하며, 이들에 대한 공간 질의를 위해 R-트리를 사용하여 색인한다. 아울러 이 연구에서는 공간네트워크상의 영역질의, k-최근접 질의, 조인 질의 등을 제안하였다. 셋째, 독일 하겐 대학에서는 FNR-트리를 확장하여 MON-트리(Moving Object in Network-tree)를 제안하였

다.[6] MON-트리의 구조는 (그림 2)와 같다. MON-트리는 FNR-트리의 단점인 삽입시 두 개의 트리를 검색하는 것을 줄이기 위해서 해시(hash) 테이블을 사용하여 2DR-트리를 검색하는 오버헤드를 줄이고, 바로 1DR-트리에 삽입한다. 이와 함께 경로 모델(Route model)을 제안하였다. 이는 실제계에서는 도로의 번호와 같은 경로 개념을 사용하여 공간 네트워크를 구성하기 때문이다.

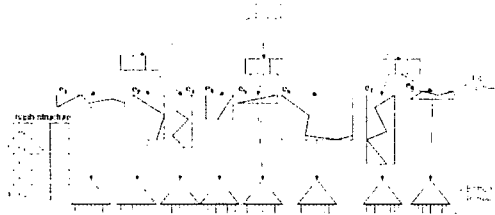


그림 2. MON-트리 구조

3. 궤적기반 색인구조의 설계

공간 네트워크상의 이동객체를 위한 궤적기반 색인구조의 설계 목적은 이동객체를 이용하여 실제계에서 LBS 또는 텔레매틱스와 같은 서비스를 제공할 때 사용자 질의를 효율적으로 처리할 수 있는 색인구조를 설계하는 것이다. 이를 위하여 공간과 시간의 특성이 다르므로 공간과 시간에 대한 색인구조를 따로 두어 설계하며, 사용자 질의를 <표 1>과 같이 분류한다.

표 1. 질의 타입 분류

질의 타입	예제
시공간+ 궤적	유조차에 대한 운행 정보 시스템이 있을 때, 어느 지역에서 유해 물질이 누출된 것을 발견하였다. 이 때 어떤 차량이 유해 물질을 누출을 했고 얼마만큼 오염지역이 퍼졌는지를 알기 위하여 그 날 오염된 지역을 지나간 차량과 그 차량에 대한 궤적을 찾는 질의
k-최근접	도로 운행 중에 주유를 하고자 하여 가장 가까운 주유소 세 개를 찾는 질의
시간+궤적	현재 위치에서 시청까지 가고자 할 때 과거에 지나간 차량들이 이 시간대에 시청까지 얼마만큼 시간이 걸렸는지 또는 평균 속도를 알고 싶을 때 그 시간대와 시청까지의 궤적을 지나간 차량을 찾는 질의

3.1 전체 구조

공간 네트워크상의 이동객체를 위한 궤적기반 색인구조는 (그림 3)과 같다. 공간데이터와 시간데이터의 속성이 다르기 때문에 이를 2장에서 설명한 MON-트리[6]와 같이 공간에 대한 색인구조와 시간에 대한 색인구조로 분리하여 색인한다. 공간에 대한 색인은 R-트리를 사용

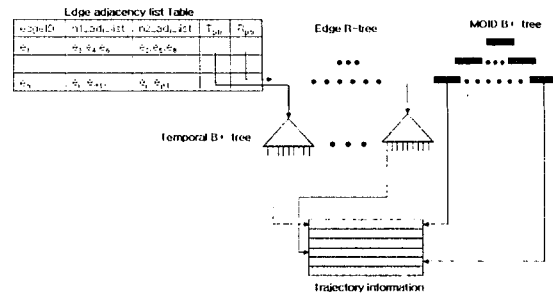


그림 3. 공간 네트워크상의 이동객체를 위한 색인구조

한다. 시간에 대한 색인구조는 MON-트리[6]와 달리 B+-트리를 사용한다. 이는 이동객체에 대한 궤적이 일정영역을 가지는 세그먼트 단위가 아닌 타임스탬프마다 이동객체의 위치를 얻어오며, 아울러 궤적에 대한 정보를 따로 저장하기 때문이다. 본 논문에서 제시한 구조는 앞에서 언급하였던 세 가지 타입의 질의 중 궤적에 관련된 질의를 지원하기 위해 궤적을 이동객체마다 유지한다. k-최근접 질의를 위하여 에지에 대한 연결리스트와 R-트리에 연결된 포인터와 B+-트리로 연결된 포인터 정보를 테이블에 저장한다. 에지 테이블은 고정길이의 배열로 구성된다. 아울러, 객체의 삽입시 테이블의 포인터 정보를 이용하여 적합한 시간 B+-트리를 선택한다. 각각의 색인구조는 다음과 같다.

(1). R-트리 노드 구조

R-트리의 리프 노드의 구조는 다음과 같다.

Rleaf=<MBR, Tptr, Edge_Info>

-MBR: 에지를 포함하는 MBR

-Tptr: Temporal B+-트리 포인터

-Edge_Info=<point_set, poi_position>

-point_set: 에지를 보간한 점들의 집합

-poi_position: 에지상의 POI 위치

(2) 시간 B+-트리(Temporal B+-tree)

시간 B+-트리의 리프노드는 다음과 같이 구성한다.

Bleaf=<time, offset, trajectory_ptr>

-time: 이동객체가 삽입된 타임스탬프

-offset: 실제 궤적상의 위치

-trajectory_ptr: 실제 궤적의 포인터

(3) MOID B+-트리

MOID B+-트리는 이동객체의 아이디를 색인한다. 이동객체의 수가 많아질수록 삽입을 해야할 공간을 빨리 찾기 위함이다.

(4) 에지 연결리스트 테이블

에지를 구성하는 테이블의 구성은 다음과 같다.

Edge=<EdgeID, N1_adj_list, N2_adj_list, Tptr, Rptr>

-EdgeID: Edge ID 정보

-N1_adj_list: 에지를 이루는 두 개의 노드 중 N1에 연결된 에지 리스트

-N2_adj_list: 에지를 이루는 두 개의 노드 중 N2에 연결된 에지 리스트

-Tptr: 시간 B+-트리 포인터

-Rptr: R-트리 포인터

(5) 궤적 파일

궤적 파일의 구성은 다음과 같다.

Traj=<MOID,trajectory_num,segment>

-MOID: 이동객체의 아이디

-trajectory_num: 궤적 세그먼트의 개수

-segment=<t1, e1, x, y, direction>

-t1: 궤적 세그먼트가 삽입된 시간

-e1: 궤적 세그먼트가 지나는 에지

-x,y: 이동객체의 현재 위치

-direction: 이동객체가 에지상에서 이동하는 방향

3.2 삽입 알고리즘

이동객체의 삽입은 이동객체의 아이디, 현재 좌표와 시간, 이동객체가 지나는 에지의 아이디, 이동객체가 이동하는 방향성을 인자로 가져와 (그림 4)와 같은 삽입 알고리즘을 통하여 본 논문에서 제시한 구조에 삽입한다.

1. 이동객체의 아이디를 B+-트리에서 검색
2. 만일 이동객체의 아이디가 검색되면
 - 2.1. 이동객체의 실제 궤적 레코드를 얻어옴
3. 이동객체의 아이디가 존재하지 않을 때
 - 3.1 이동객체의 아이디를 B+-트리에 추가
 - 3.2 이동객체의 아이디를 B+-트리에 추가
4. 에지 테이블을 통해 삽입할 시간 B+-트리를 찾음
5. 시간 B+-트리에 이동객체의 시간 삽입
6. 궤적 레코드와 시간 B+-트리의 포인터를 연결

그림 4. 이동객체의 삽입 알고리즘

3.3 질의 처리 알고리즘

사용자 질의에 대해 시공간+궤적 질의, k-최근접 질의, 시간+궤적 질의로 분류하였다. 시간+궤적 질의 처리 알고리즘은 시공간+궤적 질의 처리 알고리즘과 유사하므로 본 절에서는 시공간+궤적 질의 처리 알고리즘과 K-최근접 질의 처리 알고리즘을 논한다. 시공간+궤적 질의 처리 알고리즘은 시공간 영역에 존재하였던 이동객체의 특정한 시간영역을 지난 궤적을 찾는 것으로 알고리즘은 (그림 5)와 같다.

1. 주어진 공간영역으로 R-트리를 탐색
2. 공간영역에 해당하는 시간 B+-트리가 존재하는지 확인
 - 2.1 만일 존재하지 않으면 다음 B+-트리를 검색
 - 2.2 존재하면 B+-트리를 주어진 시간영역으로 검색
3. 검색된 리프노드에서 궤적 레코드의 포인터를 얻음
4. 궤적 레코드의 포인터를 통해 궤적 레코드를 얻어옴
5. 궤적 레코드에서 시간영역에 해당하는 궤적 세그먼트를 반환

그림 5. 시공간 영역질의+궤적 질의 처리 알고리즘

k-최근접 질의는 이동객체의 현시점에서 가장 가까운 k개의 POI를 찾는 질의이며, 이를 처리하는 알고리즘은

HKUST[5]에서 제시한 INE 기법에 근거하여 이를 확장한다. 이동 객체는 현재 움직이고 있는 방향성이 있으므로 방향성에 따른 K개의 최근접 POI를 찾는다. K-최근접 질의 처리 알고리즘은 (그림 6)과 같다.

1. 현재 객체의 위치 즉 에지를 에지 테이블을 검색
2. 객체의 이동방향에 따라 가장 가까운 에지를 확장
3. 확장한 에지에 POI가 있는지 검색
4. 확장한 에지셋들 가운데 객체의 진행 방향으로 네트워크 거리가 가까운 순서대로 에지를 확장
5. 확장한 에지에 POI가 있으면 결과셋에 삽입
6. POI의 결과셋이 k개가 될 때까지 객체의 진행방향으로 에지를 확장

그림 6. k-최근접 질의 처리 알고리즘

4. 결론

본 논문에서는 공간 네트워크상의 이동객체를 효과적으로 색인하는 색인구조를 제안하였다. 먼저, 공간 데이터와 시간데이터를 구분하여 각각의 색인구조를 따로 구성하고, 질의를 시공간+궤적질의, k-최근접 질의, 시간+궤적질의로 분류하여 이를 만족하는 궤적기반 색인구조를 설계하였다. 궤적 질의를 만족하기 위해 시간데이터를 색인하는 구조와 궤적을 연결하고, k-최근접 질의를 위해 에지 연결리스트 테이블을 구성하였다. 향후 연구로는 설계된 색인구조의 효율성을 보이기 위하여, 궤적기반 색인구조에 대한 성능평가를 수행하는 것이다.

Acknowledgment

본 연구는 중소기업청에서 시행한 산학연 공동기술개발 컨소시엄의 지원에 의한 연구개발 결과임.

참고 문헌

- [1]Theodoridis, Y., Vazirgiannis, M., and Sellis, T. "Spatio-temporal Indexing for Large Multimedia Applications." Proc. of the IEEE Conference on Multimedia Computing and Systems6(4), pp284-298, 1998.
- [2]D. Pfoser, C.S. Jensen, and Y. Theodoridis, "Novel Approach to the Indexing of Moving Object Trajectories." Proc. of VLDB, pp395-406, 2000.
- [3]Tao, Y., and Papadias, D. "Mv3R-tree: a spatiotemporal access method for timestamp and interval queries." Proc. of VLDB, pp431-440, 2001.
- [4]E. Frentzos "Indexing Objects moving on fixed networks" in Proc. of the 8th Intl. Symp. on Spatial and Temporal Database(SSTD), pp289-305, 2003.
- [5]D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao, "Query Processing in Spatial Network Databases." Proc. of VLDB, pp802-813, 2003.
- [6]Victor Teixeira de Almeida, Ralf Hartmut Güting. "Indexing the Trajectories of Moving Objects in Networks." Geoinformatica 9(1): pp33-60, 2005.