공간 네트워크 데이터베이스를 위한 저장 및 색인 구조의 구현 및 성능 평가

엄정호* 장재우*
전북대학교 컴퓨터공학과
{jhum^{O}, jwchang}@dblab.chonbuk.ac.kr

Implementation & Performance evaluation of Storage and Index Structures for Spatial Network Databases

Jungho Um^{0} Jaewoo Chang
Dept. of Computer Engineering, Chonbuk National University

요 약

최근 LBS(Location Based Service) 및 텔레매틱스 응용의 효과적인 지원을 위해, 도로, 철도와 같은 공간 네트워크 데이터베이스에 관한 연구가 활발히 수행 중에 있다. 한편 공간 네트워크 데이터베이스에 대한 효율적인 접근을 위해, 공간 네트워크 자체의 데이터, POI 데이터, 이동 객체 데이터를 효율적으로 저장 및 색인하는 구조가 설계되었다[1]. 따라서 본 논문에서는 설계된 공간 네트워크 데이터베이스를 위한 저장/색인 구조를 구현하고, 이를 공간 네트워크 자체의 데이터 검색, POI 데이터 검색, 이동 객체 궤적 검색 관점에서 성능평가를 수행한다. 아울러 기존에 연구되었던 접근 기법과 검색 성능을 비교한다.

1. 서 론

최근 LBS(Location Based Service) 및 텔레매틱스 응용의 효과적인 지원을 위해, 도로, 철도와 같은 공간 네트워크를 고려한 연구가 활발히 수행 중에 있다 [1,2,3]. 한편 공간 네트워크 데이터베이스에 대한 효율적인 접근을 위해, 공간 네트워크 자체의 데이터를 효율적으로 저장 및 색인하는 구조의 연구가 필수적이다. 일반적으로 공간 네트워크 자체의 데이터는 그래프로 변환하기가 용이하므로 그래프 형태로 변환한 후 저장하는 연구들이 수행되었다[1,2,3]. 첫째, 미네소타 대학에서는 공간 네트워크를 그래프 형태로 변환하여 Z-order로 표현하고, 연결에 따른 클러스터를 형성하여 CCAM기법에 따라 저장하는 연구를 수행하였다[2]. 둘째, HKUST에서의 연구는, 노드를 연결에 따른 클러스터로 형성하지만, 지역적 클러스터 효과를 높이기 위해 Hilbert-order 를 사용한다 [3]. 에지는 polyline 형태로 저장하며, 이들의 공간질의를 위해 R-tree를 사용한다. 마지막으로, 전북대에서는 이들의 장점을 결합하여, 공간 네트워크상에 존재하는 모든 정보, 즉 노드, 노드/에지 관계성 정보, POI(point of interest) 데이터, 이동 객체 데이터를 효율적으로 저장 및 색인하는 구조를 설계하였다[1].

본 논문에서는 [1]에서 설계된 공간네트워크 데이터베이스를 위한 효율적인 저장/색인 구조를 구현하고, 이를 공간 네트워크 자체의 데이터 검색, POI 데이터 검색, 이동 객체 궤적 검색 관점에서 성능평가를 수행한다. 아울러 기존에 연구되었던 접근 기법과 검색 성능을 비교한다. 논문의 구성은 다음과 같다. 2장에서는 구현한 공간 네트워크 데이터베이스를 위한 저장 및 색인 구조를 소개하고, 3장에서는 구현된 저장 및 색인 구조의 성능 평가를 제시하고, 기존의 방법과 성능을 비교한다. 마지막으로 4장에서는 결론을 제시한다.

본 연구는 대학 IT 연구센터 육성지원사업의 연구결과로 수행되었음

2. 설계된 저장 및 색인구조

공간 네트워크상에 존재하는 데이터는 공간 네트워크 자체의 데이터(노드, 에지) POI 데이터, 이동객체 데이터 등이 존재한다. 이들을 위한 설계된 저장 및 색인구조는 (그림 1)과 같다.

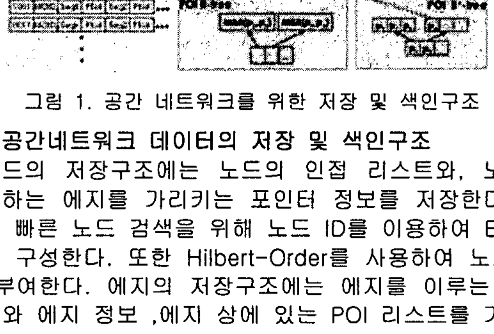


그림 1. 공간 네트워크를 위한 저장 및 색인구조

2.1 공간네트워크 데이터의 저장 및 색인구조

노드의 저장구조에는 노드의 인접 리스트와, 노드가 구성하는 에지를 가리키는 포인터 정보를 저장한다. 아울러 빠른 노드 검색을 위해 노드 ID를 이용하여 B+-트리를 구성한다. 또한 Hilbert-Order를 사용하여 노드 ID를 부여한다. 에지의 저장구조에는 에지를 이루는 노드 정보와 에지 정보 ,에지 상에 있는 POI 리스트를 가리키는 포인터 정보를 저장한다. 또한 공간 질의 처리를 위해 R-트리를 구성한다.

2.2 POI 저장 및 색인 구조

POI(Point of Interest)는 주유소, 식당과 같은 관심 대상이 되는 장소를 의미한다. 이들은 이동객체와 달리 정적인 특성을 지니기 때문에, 독립적인 저장/색인 구조를 구성한다. POI를 위한 저장/색인 구조는 POI 정보, POI가 위치한 에지정보를 저장한다. 아울러 POI에 대해 효과적인 접근을 위한 B+-트리를 구성하고, POI의 공간 질의 처리를 위한 R-트리를 구성한다.

2.3 이동객체 궤적 저장 및 색인구조

공간 네트워크상에서 움직이는 이동 객체는 공간 네트워크에 의존적이다. 그러나 기존의 이동 객체의 저장 및 색인을 위한 TB-트리는, 이상적인 공간을 가정하기 때문에 공간 네트워크를 위한 궤적 색인 구조로는 적합하지가 않다[4]. 따라서 공간 네트워크를 위한 이동 객체의 궤적 색인 방법으로 공간 네트워크의 에지에 기반을 둔 시그니처를 사용한다. 시그니처에 기반한 궤적 저장/색인 구조는 (그림2)와 같다.

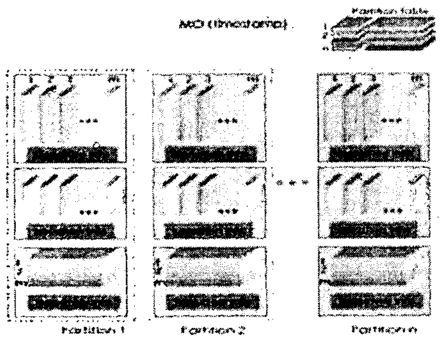


그림 2. 이동객체를 위한 저장 및 색인구조(시그니쳐)

이동 객체 궤적을 저장하기 위해 타임스템프(time stamp)에 따라 파티션(partition)에 저장한다. 한 파티션은 Signature Info, Location Info, Trajectory Info로 구성된다. Trajectory Info는 이동 객체의 궤적을 저장하고, Signature Info는 이동 객체 궤적에 대한 시그니쳐를 저장하며, Location Info는 궤적정보의 파일 상에서의 위치를 저장한다.

3. 성능평가

공간 네트워크 데이터베이스를 위한 저장/색인 구조는 Intel Xeon Dual CPU 2.4GHz 및 memory 2GB를 사용하는 Windows Server 2003에서 C++로 구현하였다. 성능평가를 수행하기 위해 사용된 공간 네트워크 데이타는 17만개의 노드와 22만개의 에지로 이루어진 샌프란시스코 만의 공간 네트워크 데이터를 사용하였다[5]. 성능평가를 위한 POI 및 이동 객체 데이터를 위해서는, 공간 네트워크 이동객체 생성 알고리즘에 근거하여 10,000개의 POI와 5,000개의 이동객체가 100초 동안 이동한 궤적 세그먼트를 생성하였다[6]. 성능평가를 위해 공간 네트워크 데이터베이스 검색을 위한 질의 타입을 (표 1)과 같이 정의하였다.

표 1. 공간 네트워크 데이터베이스 검색을 위한 질의 타입

질의 타입	설명
공간 네트워크 질의 (spatial network query)	공간 자체 데이터(노드, 에지)를 검색하는 질의
공간 질의 (spatial query)	특정 점 및 지역을 질의조건으로 하여, 에지나 POI를 검색하는 질의
궤적 질의 (trajectory query)	이동 객체의 부분 궤적 또는 궤적을 검색하는 질의

먼저, 공간 네트워크 질의와 공간 질의는 기존의 공간 네트워크 데이터베이스를 위한 저장구조로 잘 알려진 CCAM[1]과 비교를 하였다. 아울러, 궤적 질의는 기존의 궤적 색인 구조로 잘 알려진 TB-트리[4]와 성능 비교를 수행하였다. 성능은 질의처리를 위해 검색하는데 걸리는 전체 시간 즉, CPU 및 디스크 I/O 시간을 합한 시간을 의미하며, 논문에서 구현한 저장/색인 구조는 Our AO(Our Access Organization)라 명명한다.

3.1 공간 네트워크 질의 성능비교

공간 네트워크 질의는 노드의 경우에는 노드 아이디를 주고, 하나의 노드를 찾는 평균 시간을 측정하였고, 에지의 경우에는 노드 아이디를 주고 노드에 연결된 에지를 찾는 평균 시간을 비교하였다. 성능평가 결과는 (표2)와 같다.

표 2. 공간 네트워크 질의 성능비교(단위: ms)

	Our AO	CCAM
노드 탐색	0.0643	0.0702
에지 탐색	0.1249	0.1288

공간 네트워크의 노드 탐색의 경우, Our AO가 0.064ms, 기존 CCAM 이 0.07ms 로써, Our AO 방법이 검색 성능이 우수하다. 그 이유는 CCAM은 연결성(connectivity)에 근거하여 클러스터링을 하는 반면, 본 논문에서 구현한 Our AO 구조는 Hilbert-Order에 의하여 지역적 클러스터링을 수행하며, 이를 통하여 질의에 따른 보다 적은 디스크 탐색을 수행하기 때문이다. 에지 탐색의 경우, Our AO가 0.124ms, 기존 CCAM 이 0.128ms 로써, Our AO 방법이 에지 검색 성능에서 약간 우수하다.

3.2 공간 질의 성능비교

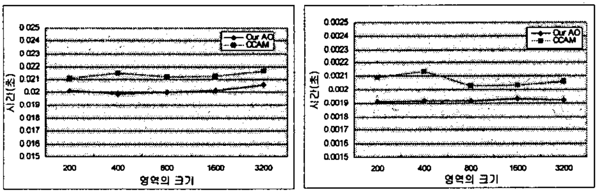
공간 질의는 점(point)을 중심으로 에지나 POI를 검색하는 경우 및 영역을 중심으로 검색하는 경우의 두 가지가 존재한다. 점을 중심으로 검색하는 질의는 에지 탐색의 경우, 랜덤하게 POI를 선택하여 그 POI를 포함한 에지를 검색하는 시간을 측정하였고, POI 탐색의 경우, 랜덤하게 선택한 POI를 가진 에지 상에 있는 모든 POI를 검색하는 시간을 측정하였다. (표 3) 은 점을 통한 공간 질의 성능 결과를 나타낸다. Our AO가 26ms, 기존 CCAM 이 25.9ms 로써, Our AO 방법이 검색 성능이 약간 저하된다. 이는 CCAM은 에지를 순차적으로 검색을 하고,Our AO는 R-tree를 사용하여 검색한다. 따라서 질의 점이 R-tree의 MBR영역이 중복되는 지점이 될 경우 순차탐색보다 오히려 검색 성능이 저하되기 때문이

다. 하지만 POI 탐색에 대해서는, Our AO가 1.951ms, 기존 CCAM 이 1.953ms 로써, Our AO 방법이 에지 검색 성능에서 약간 우수하다. 이는 Our AO 가 에지에서 POI에 접근할 수 있는 구조를 가지고 있지만 CCAM은 이와 같은 구조가 있지 않으므로 성능의 차이가 나고 있다. 일반적으로 에지를 통해 POI를 검색하는 질의가 더 많으므로, Our AO 구조가 비록 에지 탐색 시간이 더 느리지만, POI에서 더 빠른 검색 시간을 보이므로 에지 탐색에 대한 단점을 보완할 수 있다.

표 3. 점으로 검색한 공간 질의 성능비교 (단위: ms)

	Our AO	CCAM
에지 탐색	26.0058	25.9550
POI 탐색	1.9511	1.9531

영역을 중심으로 검색하는 경우는 영역을 200, 400, 800, 1600, 3200의 다섯 구간으로 나누어 성능비교를 수행하였다. 이와 같이 수행한 이유는 에지의 평균 길이를 측정한 결과, 약1000의 길이를 가지기 때문에 이를 기준으로 구간을 나누었다. 여기서 1에 해당하는 길이는 실제 공간상에서 1.5cm에 해당한다. (그림 3)의 (a)는 주어진 영역을 중심으로 에지를 찾는 평균 시간을 측정한 결과이며, (그림 3)의 (b)는 POI를 주어진 영역으로 찾는 평균 시간이다.



(a) 영역으로 에지 검색 (b) 영역으로 POI 검색

그림 3. 영역 공간 질의

성능 결과에서 보는 것과 같이, 에지 검색의 경우, Our AO가 약 20ms, 기존 CCAM 이 21ms 로써, Our AO 방법이 에지 검색 성능에서 우수하다. 아울러 POI 검색의 경우, Our AO가 약 1.9ms, 기존 CCAM 이 약 2.1ms 로써, Our AO 방법이 보다 우수하다. 이는 CCAM은 에지나 POI에 색인 구조 없이 순차 탐색을 하는 반면, Our AO 가 영역질의에 효율적인 R-tree를 에지 및 POI의 색인 구조로 가지기 때문에 좀 더 검색 성능이 향상되었다. 아울러 영역 구간사이에서는 영역에 상관없이 일정한 탐색 시간을 보이고 있다. 이와 같은 이유는 성능에 영향을 줄 정도로 영역 구간이 넓지 않고, 하나의 에지 내의 범위로 영역 구간을 정했기 때문이다.

3.3 궤적 질의 성능비교

이동 객체의 궤적 질의는 질의 세그먼트의 수에 따라 해당 이동 객체의 궤적을 검색하는 시간을 측정하였다. 한편 궤적 질의의 유형을 살펴보면, 이동 객체의 모든 궤적을 찾는 것보다, 특정 시간에 이동하는 객체의 부분 궤적을 찾는 질의가 더 많다. 따라서 이에 근거하여 궤적 길이를 2에서부터 20까지 변화시키면서, 부분 궤적 질의의 성능 평가를 수행하였다. (그림 4)는 이동 객체의 궤적 질의 검색 시간을 측정한 결과를 보여준다.

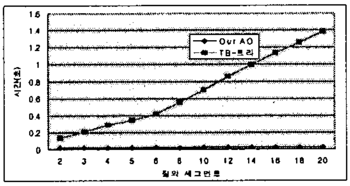


그림 4. 궤적 질의

(그림 4)에서 나타난 것처럼 세그먼트가 2 인 경우, Our AO 가 약 22ms, TB-트리로 140ms, Our AO 가 약 700% 성능 향상을 보인다. 아울러 세그먼트가 20 인 경우, Our AO 가 약 24ms, TB-트리는 1380ms 로써, Our AO 가 약 50배 성능 향상을 보인다. 그 이유는 Our AO 는 질의궤적이 다수의 세그먼트를 가지고 있어도, 이것을 하나의 질의 시그니쳐로 구성하여 탐색을 수행하므로 질의궤적 세그먼트 개수에 상관없이 일정한 검색 성능을 보인다. 반면, TB-트리는 각 세그먼트마다 세그먼트의 영역 질의 후, 각 영역을 통과하는 모든 이동객체를 탐색하기 때문에 세그먼트의 수가 증가할수록 비례적으로 검색 시간이 증가한다.

4. 결론

본 논문에서는 [1]에서 설계된 공간 네트워크를 위한 효율적인 저장/색인 구조를 공간 네트워크 자체의 데이터 검색, POI 데이터 검색, 이동 객체 궤적 검색 관점에서 성능평가를 수행하였다. 아울러 이를 기존의 공간 네트워크의 저장 방법인 CCAM 및 궤적 색인 구조인 TB-트리와 비교하여 성능이 우수함을 보였다. 향후 연구로는 공간 네트워크를 위해 구현된 저장/색인구조가 이동 객체 궤적 질의뿐 아니라 공간 질의도 효율적으로 수행할 수 있도록 확장하고, 아울러 확장된 저장/색인 구조의 성능평가를 수행하는 것이다.

참고문헌

[1] 강홍민, 장재우 "공간 네트워크 데이터베이스를 위한 저장 및 색인 구조의 설계" 한국정보과학회 가을 학술발표 논문집 제31권 제2호 pp133-136,2004

[2] S. Shekhar and D.-R. Liu,"CCAM: A Connectivity-Clustered Access Method for Networks and Network Computations." IEEE Tran. on Knowledge and Data Engineering, Vol. 9, No. 1, pp 102-119, 1997.

[3] D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao, "Query Processing in Spatial Network Databases." Proc. of VLDB, pp, 802-813, 2003.

[4] D. Pfoser, C.S. Jensen, and Y. Theodoridis, "Novel Approach to the Indexing of Moving Object Trajectories." Proc. of VLDB, pp 395-406, 2000.

[5] http://www.maproom.psu.edu/dcw/

[6] T. Brinkhoff, "A Framework for Generating Network -Based Moving Objects." GeoInformatica, Vol. 6, No. 2, pp 153-180, 2002.