

실시간 모바일 GIS를 위한 효율적인 경로탐색 기법

이형석^o 김경창
 홍익대학교 컴퓨터공학과
 {hslee^o, kckim}@cs.hongik.ac.kr

Efficient Path Search Method in Real-Time Mobile GIS

Hyeongseok Lee^o, Kyungchang Kim
 Dept. Computer Engineering, Hongik University

요 약

본 논문에서는, 비교적 제약이 심한 모바일 클라이언트를 위해 구현한 실시간 모바일 GIS에서, 최적의 경로탐색을 위한 최단 경로 탐색 관리자를 추가하였다. CAD파일로부터 추출한 벡터기반의 지리데이터를 사용하여 최단경로 검색의 시간비용을 줄이고자 필요한 정보를 추가하고, 처리비용에 불필요한 부담이 되는 데이터를 제외하는 처리를 한 후, 질의의 빈도가 각각 다른 MBR들간의 특성을 고려하여 상황에 맞는 최단 경로 질의 처리 방법을 선택 수행하도록 하여 서버의 부담을 줄이고 처리 속도를 향상시키고자 하였다.

1. 서 론

최근 mobile phone, PDA 등과 같은 모바일 기기들의 꾸준한 사용량 증가와 함께 모바일 서비스의 다양성이 증가하고 있다. 그 중에서도 GIS를 기반으로 한 지도 검색 서비스나 CNS(Car Navigation System) 등이 각광 받고 있고, 이러한 시스템에서 최적의 경로를 알려주는 최단 경로 서비스는 질의 빈도가 높은 것 중 하나이다. 본 논문에서는, 지리 데이터의 빠른 검색을 위해 제안된 주기억 데이터베이스 시스템 기반의 GIS를 바탕으로, 경로 탐색 관리자를 추가하여 효율적인 경로 탐색을 위한 방법을 소개하고자 한다.

GIS 구축에 있어 사용되어 지는 지도 데이터는 크게 래스터데이터와 벡터데이터로 나뉜다. 래스터데이터는 데이터의 갱신이 어렵고, 확대 및 축소 시에 이미지가 깨져 보일 수 있다는 단점을 가진다. 반면, 벡터 데이터는 지리 데이터의 특정 좌표계에서의 좌표들로 구성이 되기 때문에 시각화를 위한 프로세싱 과정이 필요하지만 데이터의 업데이트가 용이하고 보다 다양한 질의를 통해서 데이터를 선택할 수 있다는 장점이 있다. 때문에, 많은 지도 데이터들은 벡터기반의 데이터로 제공되어 지고, 본 시스템에서도 벡터데이터를 기반으로 GIS시스템을 구현하였고, 이들 데이터를 최단 경로 탐색 관리자를 구현하는데 기본 데이터로 이용하도록 하였다. 효율적인 경로 탐색을 위해 원 데이터(shp format)에 필요한 정보를 추가하고 불필요한 데이터를 제외 시키는 작업을 거쳐 경로 탐색 데이터를 형성하고, 각 MBR에 요청되는 최단 경로 질의의 빈도수와 객체 노드들의 비율을 고려하여 상황에 따라 더 빠르게 수행되어 질 수 있는 최단 경로 질의 처리 방법을

제안한다.

도로의 크기, 방향성과 같은 도로의 속성과 교통량, 사고, 공사 등 실시간으로 변하는 도로의 교통 상황이 최단 경로 탐색에 있어 중요한 변수들이 될 수 있으나, 본 시스템에서는 각 도로 데이터의 위치 정보를 이용한 유클리드 거리를 사용하여 알고리즘을 구현하였다.

본 시스템은 MR-Tree[1]와 T-Tree[2]의 인덱스 기법을 사용하여 구현한 주기억 데이터베이스 시스템을 기반으로 설계 되었고, 추가된 도로 정보 또한 빠른 검색속도를 보장하기 위해서 MR-Tree를 사용하여 저장하도록 하였다.

본 논문에서는 먼저, 2장에서 본 시스템의 기반이 되는 GIS Server의 구성을 소개하고, 3장에서는 추출한 지리 데이터를 앞서 최단 경로 탐색 질의에 효율적으로 사용되어 질 수 있도록 데이터 처리하는 방법을 설명하고, 이어 4장에 각 MBR의 질의 빈도 특성을 고려한 질의 처리 방식을 소개하며, 5장에 결론으로 마무리 하고자 한다.

2. GIS Server 구성 및 최단 경로 관리자

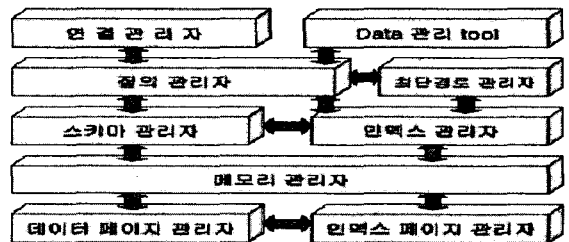


그림 1. GIS 서버 구조

본 시스템은 CPU, 메모리 등의 자원 제약이 심한 모바일 기기들을 클라이언트로 하고 대부분의 질의를 서버에서 처리하여 최소한의 결과만을 클라이언트에 전송하도록 하는 환경에서의 GIS를 고려하여 구현하였다. 그림 1과 같이 GIS서버의 구조는 기본적으로 질의 관리자, 스키마 관리자, 인덱스 관리자, 메모리 관리자, 데이터 관리자, 페이지 관리자로 구성되어 있고, 여기에 경로 탐색을 위한 최단 경로 관리자 모듈을 추가 하였다.

3. 경로 탐색을 위한 데이터 처리

가장 널리 사용되는 지리 벡터 데이터는 CAD파일 포맷으로 배포되고 있고, 많은 GIS시스템에서는 이 데이터를 ESRI사의 Shape 파일 포맷으로 변환하여 사용한다[3].

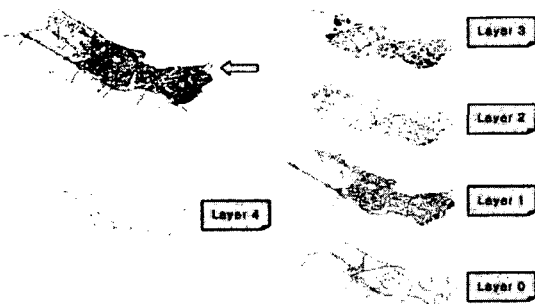


그림 2. 지도 데이터의 Layer 구성

본 시스템에서도 CAD파일 포맷의 지도 데이터를 불러와 각각의 벡터 데이터의 point, polygon 등의 속성에 따라 그림 2와 같이 각 layer0, 1, 2, 3을 구성하여 GIS서버에 데이터베이스를 구성하여 사용한다. 여기에는 특별히 도로 정보를 판단할 layer가 없기 때문에, 도로 정보를 polyline형태로 따로 저장한 Layer 4를 추가하였다.

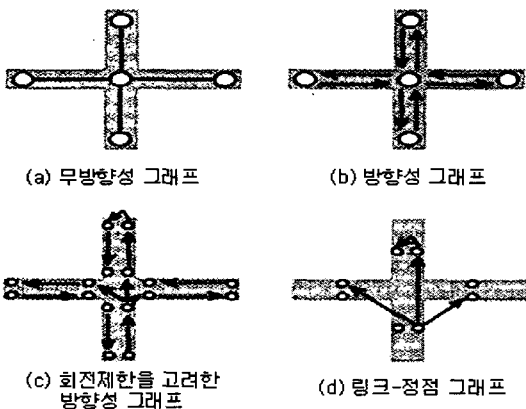


그림 3. 도로 그래프 모델

도로 정보의 데이터를 통해 경로 탐색을 수행하기 위해서는 각 도로의 주요 지점을 노드(node)화 시켜야 하는데, 서론에서 언급했듯이 도로의 제반 속성들은 고려하지 않았기 때문에 그림 3(a)과 같이 방향성을 고려하지 않은 그래프 모델[4]을 선택하였고, 다음과 추가 처리 과정을 수행하였다.

3.1 도로 데이터 준비

CAD파일에서 추출하여 생성한 도로 데이터는 시작점과 끝점만을 가지는 line 벡터의 특성상 그림 4(a)의 도로 R5-R6와 같이 교차점에서 다음 노드로 방향이 꺾이지 않으면 도로 R1-R2와 같이 도로와 도로간의 교차점 정보가 없게 된다. line segment들간의 intersection을 구하는 데 $O((n+k)\log n)$ 의 시간비용을 가지는 Plane Sweep algorithm[5]을 본 도로 벡터데이터에 적용시켜 도로 데이터들간의 교차지점을 구하고 R2는 그림 4(b)와 같이 노드 I가 추가된 도로 R7, R8로 나누어진다(k는 도로간의 교차점의 수에 해당한다).

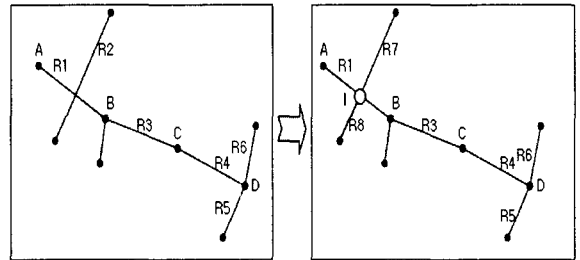


그림 4.

(a) 원시의 shp 파일의 정보 (b) 교차점 처리 후의 정보

또한 각 노드에 연결된 도로의 수를 표 1과 같이 구분하여 연결된 도로가 2개 이하인 경우에 있어 최단 경로 탐색 질의시 처리를 간소화 한다.

표 1. 연결된 도로 수에 따른 처리

연결된 도로 수	노드	해당 노드 예
1 개	터미널 노드	노드 A
2 개	제외 노드	노드 C
3 개 이상	분기 노드	노드 B

노드 C와 같이 연결된 도로가 2개인 경우, 노드 B-D간의 부분 경로에서 노드 C는 최단 경로 탐색 경우의 수에 관여하지 않는 노드로, 실제 경로 탐색을 위해서는 B-C-D노드를 B-D노드만 탐색할 수 있도록, C노드들과 같이 불필요한 노드들을 제거하고 하나의 로드 데이터로 남겨 최단거리 연산을 더 빠르게 수행할 수 있도록 하였다.

3.2 경로탐색을 위한 노드 테이블 형성

경로 탐색을 위해 사전 처리를 거친 도로 정보 노드들 각각을 1차원 배열로 형성하여 위치 정보를 저장하고, 이

들 배열의 인덱스를 이용하여 각 노드들 간의 거리를 그림 5와 같이 2차원 테이블로 형성하여 경로 탐색에 사용한다.

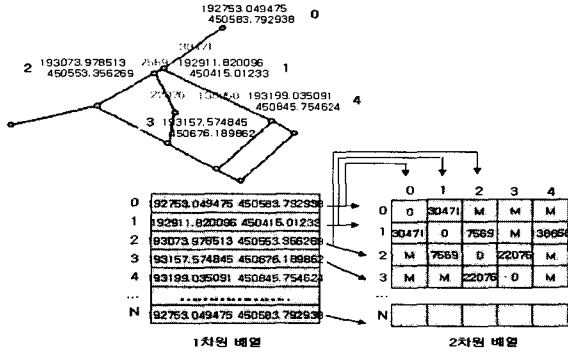


그림 5. 도로정보 노드 테이블

4. 경로 탐색 질의

클라이언트에서 최단 경로 탐색 질의가 발생하면, 서버로 해당 서비스를 요청하게 되고 이들은 서버의 도로데이터와 함께 처리된 데이터 노드들을 이용하여 실질적인 경로 탐색을 수행하여 클라이언트로 결과를 전송해 준다.

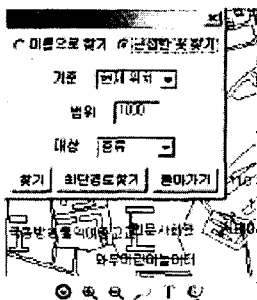


그림 6. 클라이언트의 GUI

일반적인 최단 경로 질의는 Dijkstra algorithm을 통해 결과값과 최단 경로를 전송하나, 본 시스템은 클라이언트에서 데이터베이스를 자체적으로 구축하고 경로탐색을 수행하는 CNS 등과는 달리, mobile phone과 같이 기기의 자원 제약이 비교적 심한 클라이언트를 가지는 환경에서, 서버가 경로 탐색 질의등의 전반적인 것을 담당하여 결과만을 클라이언트에 전송하도록 구현하였기 때문에 많은 클라이언트들이 동시에 최단경로 검색 질의를 요청할 경우 부하가 많이 걸리게 된다.

$O(n^2)$ 의 시간 복잡도를 가지는 다익스트라 알고리즘의 경우에, 위와 같이 많은 질의가 빈번하게 발생하여 n번 이상 반복되면 플로이드 알고리즘의 시간 복잡도인 $O(n^3)$ 을 초과하게 된다. 다익스트라 알고리즘의 경우 계산과정 자체의 복잡성 때문에, 이 알고리즘의 n번 수행한 결과보다 오히려 플로이드 알고리즘이 더 빠르게 된다.

이런 두 알고리즘의 특성을 이용하여, 특정 MBR영역에 요청되는 경로탐색 질의를 카운트 하여 기록하고, 일

정 주기 동안 요청되는 경로 탐색 요청 질의의 평균이 해당 MBR의 노드들 보다 많아질 경우 Floyd algorithm을 이용하여 노드들 간의 최단 경로 테이블을 미리 계산하여 경로와 거리값을 저장하고, 이를 이용하여 서버의 부담을 줄일 수 있도록 구현하였다.

5. 결론

본 시스템은 벡터 기반의 지리 데이터를 주기억 데이터 베이스 시스템을 기반으로 구축하여 모바일 클라이언트가 실시간으로 지리 데이터를 검색할 수 있도록 구현한 시스템에 최단 경로 관리자를 추가적으로 구현 하였다. 추출한 데이터를 일련의 과정을 거쳐 경로 탐색 질의에서 더 적은 공간으로 더 빠르게 수행할 수 있도록 하였다.

경로 탐색 질의에 있어 빈번한 질의를 효율적으로 처리하고자 노드들간의 경로를 사전에 계산하여 가지고 있는데, 만약 도로 정보가 변경되어 노드가 추가되거나 삭제 된다면, 저장된 최단 경로 테이블의 값들 일부는 수정 되어야 한다. 전체를 재계산하는 것은 비효율적임을 고려할 때 기존의 테이블을 이용하여 변화된 도로 정보를 부분적으로 적용하여 계산할 수 있는 알고리즘 구현이 필요할 것이다.

참고 문헌

- [1] Sukwoo Yun, Kyungchang Kim, 'Making Cache-Conscious R-trees for the Main Memory Spatial Indexing'
- [2] Hongjun Lu, Yuet Yeung, Ng Zengping, 'T-Tree or B-Tree : Main Memory Database Index Structure Revisited', Australasian Database Conference, 2000
- [3] ESRI, 'ESRI Shapefile Technical Description', an ESRI White Paper, March 1998
- [4] 홍원철, 명선영, 'ITS를 위한 차량 항법 시스템 개발', Telecommunications review, 9권, 4호, P551-565, 1999
- [5] M.de Berg, M.van Kreveld, M.Overmars, O.Schwarzkopf, "Computational Geometry", Springer, 2000
- [6] B. R. Badrinath, Shirish Hemant Phatak, 'A Database Architecture for Handling Mobile Clients'
- [7] Lauzac, Chrysanthis, 'Personalizing Information Gathering for Mobile Database Clients'
- [8] Margaret H. Dunham and Abdelsalam helal, 'Mobile Computing and Database: Anything new?', ACM SIGMOD Record 24(4)
- [9] Nick Roussopoulos, Stephen Kelley, Frederic Vincent, 'Nearest Neighbor Queries'
- [10] Sheffi, Y., "Urban transportation networks," Prentice-Hall, 1985