

## 실시간 모바일 GIS를 위한 최단 경로 관리자 설계 및 구현

허남은<sup>o</sup> 김경창  
 홍익대학교 컴퓨터 공학과  
 {neheo<sup>o</sup>, kckim}@cs.hongik.ac.kr

### Design and Implementation of Shortest Path Manager for Real-time Mobile GIS

Nameun Heo<sup>o</sup>, Kyungchang Kim  
 Dept. of Computer Engineering, Hongik University

#### 요 약

최근 모바일 기기의 성능이 발전함에 따라 다양한 상용 서비스들이 제공되고 있는데 특히 GIS를 이용한 검색 서비스나 차량용 네비게이션 시스템이 대표적이다. 본 논문에서는 GIS서버의 전반적인 시스템 구조와 최단 거리 관리자의 설계 및 구현, 서버 및 PDA기반의 클라이언트 사용자 인터페이스 등에 대해서 소개를 하고자 한다. 최단 경로 관리자는 클라이언트에서 질의 시 목적지까지 최단 거리를 구하여 질의 관리자에 결과를 넘겨주는 역할을 한다. 클라이언트 사용자 인터페이스 부분은 사용자가 쉽게 질의를 작성할 수 있도록 구현 하였다.

#### 1. 서 론

현재 모바일 폰이나 PDA 등과 같은 다양한 모바일 기기가 등장하고 프로세싱 및 커뮤니케이션 성능이 급속도로 발전하고 있다. 그러한 환경에 발 맞춰 다양한 상용 서비스들이 제공되고 있으며 그 중 GIS 구축을 통한 지도 검색 서비스나 차량용 네비게이션 시스템 등이 각광을 받고 있다. 이러한 시스템에서 제공하는 서비스 중 대표적인 것이 목적지 까지 최단 거리를 나타내는 최단 경로 서비스라고 할 수 있다. 본 논문에서는 지리 데이터의 보다 빠른 검색을 위해 제안된 주기억 데이터 베이스 시스템 기반의 GIS를 기반으로 해서 최단 경로 구현에 필요한 최단 경로 관리자의 설계 및 구현에 대해서 소개를 하고자 한다.

GIS를 구축하는데 있어서 사용되는 데이터는 크게 이미지 데이터와 벡터 데이터가 있다. 현재 차량용 네비게이션이나 ZEUS[2]와 같은 지리정보 시스템의 경우에 지도를 검색하거나 최단 경로를 나타내는 작업을 위해서 벡터 데이터와 이미지 데이터를 같이 사용해서 작업을 수행한다. 일반적으로 이미지 데이터는 지리 데이터가 시각화된 상태를 의미하며 데이터의 갱신이 번거롭고 확대 및 축소 시에 이미지가 깨져 보일 수 있다는 단점이 있다. 반면에 벡터 데이터는 지리 데이터의 특정 좌표계에서의 좌표들로 구성이 되기 때문에 시각화를 위한 프로세싱 과정이 필요하지만 데이터의 업데이트가 편하고 보다 다양한 질의를 통해서 데이터를 선택할 수 있다는 장점이 있다. 따라서 본 시스템에서는 이미지 데이터를 사용하지 않고 벡터 데이터만을 사용해서 최단 경로를 구현 하였다.

최단 경로를 구할 경우 교통량, 사고, 공사 등 실시간으로 변하는 도로의 교통 상황이 중요한 변수가 될 수 있는데 벡터 데이터를 사용하는 본 시스템에서는 이러한

정보를 반영할 수 없기 때문에 지리 데이터의 좌표 상에 거리만을 이용해서 최단경로를 구했다. 최단 경로를 구하는 알고리즘으로는 다익스트라(Dijkstra)의 최단경로 알고리즘을 사용해서 구현 하였다.

본 시스템은 주 기억 데이터베이스 시스템을 기반으로 설계 되었고 주 기억 환경에 적합한 MR-tree[3]와 T-tree[4]의 인덱스 기법을 사용하였는데 새로 추가된 도로 정보 또한 빠른 검색속도를 보장하기 위해서 MR-tree를 이용해서 저장하도록 설계 하였다.

본 논문에서는 먼저 2장에서 GIS 서버의 전반적인 구성 및 최단경로 관리자에 대해서 소개 한다. 3장에서는 최단경로 관리자를 위한 데이터 준비에 대해서 소개 하고 이어 4장에서는 최단 경로 구현 환경에 대해서 소개를 하고 5장에서 결론으로 마무리 하고자 한다.

#### 2. GIS Server 구성 및 최단 경로 관리자

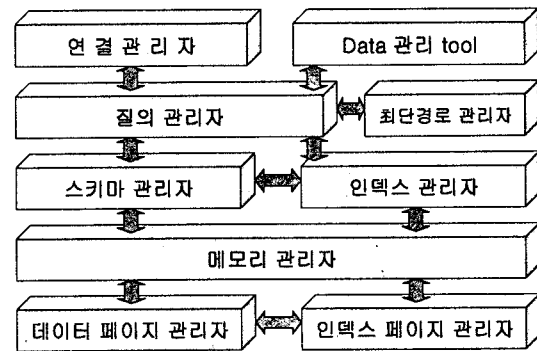


그림 1 GIS 서버 구조

그림 1과 같이 GIS서버의 구조는 크게 질의 관리자와 스키마 관리자, 인덱스 관리자, 메모리 관리자, 데이터

페이지 관리자, 인덱스 페이지 관리자로 구성된 구조에 최단 경로 관리자를 추가 하였다.

최단경로 관리자의 구조는 다음과 같다.

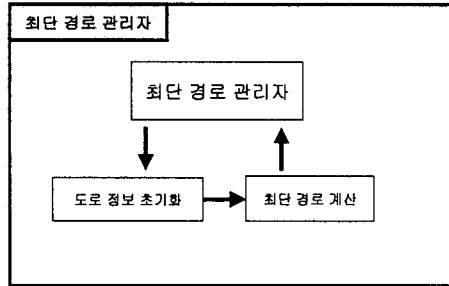


그림 2. 최단 경로 관리자 구조

최단 경로 관리자는 도로정보 초기화 루틴과 최단경로 계산 루틴이 있다. 최단 경로 관리자는 질의 관리자와 연결 되어 사용자가 클라이언트를 통해서 원하는 목적지의 최단 경로를 질의할 경우 질의 관리자가 스키마 관리자를 통해서 먼저 목적지의 위치를 찾고 현재의 위치와 목적지의 위치를 (현재위치, 목적지 위치)형태로 최단 경로 관리자에게 넘겨준다. 그 후에 최단 경로 관리자는 먼저 도로 정보 초기화 루틴에서 DB에 저장된 도로 정보를 모두 읽어 다익스트라(Dijkstra)의 알고리즘을 적용할 수 있도록 도로 정보를 1차원 배열과 2차원 배열의 형태로 초기화 시킨다. 이렇게 초기화된 도로 정보를 사용해서 최단 경로 계산 루틴에서 다익스트라(Dijkstra)의 알고리즘을 이용해서 최단 경로를 구하고 나온 결과 값을 다시 질의 관리자에게 넘겨주게 된다.

3. 최단 경로 관리자를 위한 데이터 준비

국내에서 사용되는 지리 벡터 데이터는 CAD 파일 형식으로 배포되고 있고, 시스템의 특성에 따라서 중간 형태의 파일을 해당 시스템에서 각각 다른 형태로 변환을 해서 저장하고 있다. 기존에 보급된 지리 벡터 데이터는 지리 벡터 데이터에서 도로 정보를 파악해서 최단 경로를 나타내기 위한 정보가 부족하기 때문에 이미지 데이터를 필요시에 같이 사용 한다. 즉 지리벡터 데이터를 통해서 도로의 정보를 파악하고 필요시에 이미지 데이터를 중첩시켜 최단 경로를 나타내게 된다.

하지만 앞장에서도 언급했듯이 이미지 데이터의 경우에 데이터의 갯수의 어려움 및 확대, 축소 시에 이미지의 훼손 등의 단점이 있기 때문에 본 시스템에서는 이미지 데이터를 사용하지 않고 도로 정보를 인식시키기 위해서 기존의 지리 벡터 데이터에 도로 정보를 추가 하였다. 본 시스템에서 사용하는 벡터 데이터 파일은 그림 3에서와 같이 4개의 Layer(Layer0~Layer3)로 이루어져 있는데 최단 경로를 계산하기 위해서 Layer4를 추가 한 것을 보여주고 있다. 도로 정보 추가 시 골목길과 같은 좁은 도로까지 정보에 포함시키면 복잡도가 증가하기 때문에 자동차가 다닐 수 있는 2차원 도로 이상의 큰길은 중심으로 해서 도로 정보를 추가했다.

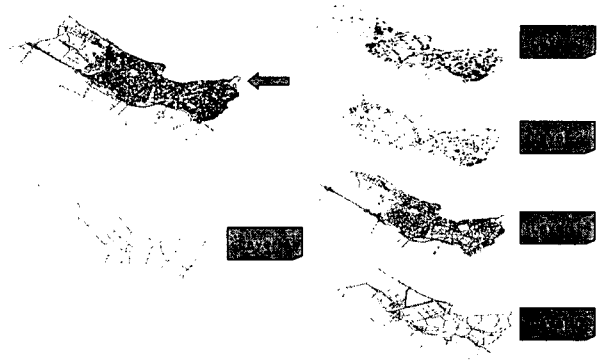
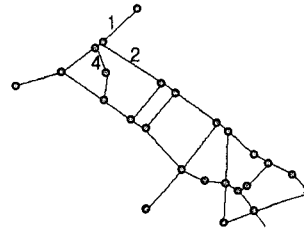


그림 3 도로 정보

3.1 최단경로 계산



ID	LAYER	SHAPETYPE	POINTS	COLOR	DISTANCE
1	4	3	192753 040474 450583 792928 182911 820286 450415 01233	0	53655
2	4	3	193073 978512 450543 266268 183157 574845 450576 16862	0	22078
3	4	3	193157 574845 450576 16862 183158 035251 450545 264524	0	30471
4	4	3	193158 035251 450545 264524 183148 04275 450516 247841	0	7566
5	4	3	193148 04275 450516 247841 192854 648237 451310 485703	0	120100
6	4	3	192854 648237 451310 485703 192534 776846 450265 911802	0	280458
7	4	3	192534 776846 450265 911802 192743 826474 449261 207828	0	138648

그림 4. 도로 정보

그림 4는 경로 계산을 설명하기 위해서 그림 3의 Layer4를 일부분 이용 하였다. 다익스트라(Dijkstra)의 최단 경로 알고리즘에서 최단 경로를 구할 때 두 정점과 정점간의 거리를 2차원의 배열 형태로 표현을 해서 최단 경로를 계산한다. 위에 그림은 추가된 도로 정보의 일부분 이고 밑에 그림은 각 선 별로 DB에 저장된 그림이다. 각 선 하나당 DB에 시작점(X,Y)과 끝점(X,Y), 거리 순으로 저장 되어 있는데 시작점과 끝점을 정점으로 보고 도로의 거리를 두 정점 사이의 거리로 해서 계산한다. 이때 다익스트라(Dijkstra) 알고리즘을 적용하기 위해 도로 정보를 2차원 배열로 표현해야 하는데 그림 5에서와 같이 도로의 한점(X,Y)을 가진 1차원 배열과 거리 정보를 가진 2차원 배열 형식으로 표현 하였다.

4. 최단 경로 구현 환경

그림 6은 모바일 기기에서 실행되는 클라이언트 프로그램의 메인 화면으로 메뉴는 화면 좌측부터 종료, 확대,

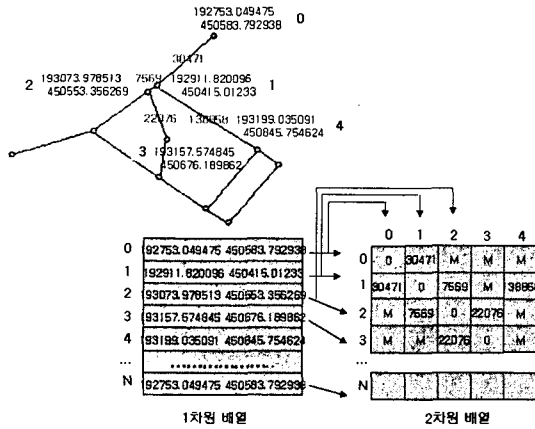


그림 5 도로정보 배열

축소, 찾기, 질의 등으로 구성되어 있다. 최단 경로로 목적지를 검색하기 위해서 먼저 찾기 버튼을 누르면 그림 7과 그림 8의 화면이 팝업 된다.



그림 6 클라이언트 메인

그림 7은 지리 데이터의 이름으로 목적지를 검색하는 인터페이스로 목적지만 찾고 싶으면 찾기 버튼을 누르고 최단 경로를 찾고 싶으면 최단 경로 찾기 버튼을 누르면 최단 경로를 검색하게 된다. 그림 8은 현재 위치에서 가장 가까운 특정 종류의 지리 데이터를 검색하는 인터페이스[5]로 검색하는 방법은 이름으로 찾기에서와 마찬가지로 수행 한다.

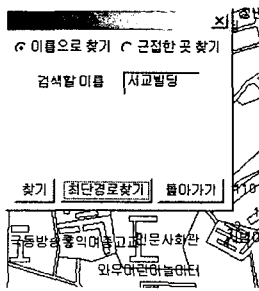


그림 7 이름으로 최단경로 찾기

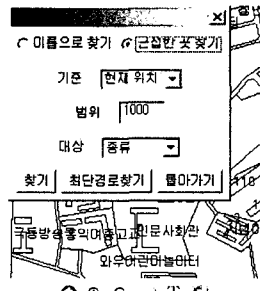


그림 8 근접한곳 최단경로 찾기

그림 9은 최단 경로 검색을 실행했을 때 나타나는 결과 화면이다. 전체 도로 데이터 중에서 필요한 도로 정보만 나타내도록 구현 하였다.



그림 9 최단경로 검색 결과

### 5. 결론

본 시스템은 벡터 기반의 지리 데이터를 주기억 데이터 베이스 시스템을 기반으로 구축해서 모바일 클라이언트가 실시간으로 지리 데이터를 검색할 수 있도록 구현한 시스템에 최단 경로 관리자를 추가적으로 구현 하였다. 전체 시스템에서 최단 경로 관리자가 어떻게 동작하는지 설명 하고 최단 경로를 표현하기 위해서 도로 정보에서 거리 계산을 어떻게 하는지 설명을 하였다. 클라이언트에서는 사용자가 편하게 원하는 질의를 할 수 있도록 구현을 하였다. 본 시스템은 메인 메모리 기반이므로 최단 경로를 다익스트라(Dijkstra)알고리즘으로 계산할 때 모든 도로 정보를 2차원 배열로 저장해서 계산하기 때문에 도로 정보가 증가하는 만큼 메모리의 용량을 차지하므로 이것을 개선하기 위한 추가적인 연구가 필요하다고 볼 수 있다.

### 참고 문헌

- [1] B. R. Badrinath, Shirish Hemant Phatak, 'A Database Architecture for Handling Mobile Clients'
- [2] www.ktdata.co.kr
- [3] Sukwoo Yun, Kyungchang Kim, 'Making Cache-Conscious R-trees for the Main Memory Spatial Indexing'
- [4] Hongjun Lu, Yuet Yeung, Ng Zengping, "T-Tree or B-Tree : Main Memory Database Index Structure Revisited", Australasian Database Conference, 2000
- [5] Nick Roussopoulos, Stephen Kelley, Frederic Vincent, 'Nearest Neighbor Queries'
- [6] Thomas Brinkhoff, 'A Portable SVG-Viewer on Mobile Devices Supporting Geographic Applications'
- [7] Margaret H. Dunham and Abdelsalam helal, 'Mobile Computing and Database: Anything new?', ACM SIGMOD Record 24(4)
- [8] Lauzac, Chrysanthis, 'Personalizing Information Gathering for Mobile Database Clients'