

최적 경로 탐색을 위한 계층 경로 알고리즘의 탐색 영역 결정 기법

이현섭 · 김진덕
동의대학교

A Method to determine Search Space of Hierarchical Path Algorithm for Finding Optimal Path

Hyoun-Sup Lee · Kim Jin-Deog Kim

Dong Eui University

E-mail : lhskmj@naver.com

요 약

최적경로 탐색은 텔레매틱스에서 효용가치가 높은 기술이다. 그렇지만 기존 시스템의 최단 경로가 항상 최적 경로라고 볼 수 없다. 즉 도로위에서는 이동시간이 최소인 경로를 최적경로라고 정의할 수 있다. 이런 최적경로를 탐색하기 위한 여러 가지의 기술 및 알고리즘들이 존재한다. 계층 경로 알고리즘은 로드 네트워크를 주, 부 레이어로 나누어 경로를 탐색한다. 2단계로 나누어 경로를 탐색하기 때문에 경로 탐색 연산시간의 성능은 뛰어나다. 탐색되는 경로 또한 최적 경로에 가까운 결과를 가진다. 2단계로 계층을 분할 할 때, 부 도로를 포함하는 주요도로 영역의 할당 방법은 성능에 큰 영향을 미친다. 본 논문에서는 계층 경로 알고리즘에서의 주요도로 선정에 대한 탐색 영역 결정 기법에 대하여 제안한다. 그리고 제안한 기법을 계층 경로 탐색에 적용하는 방법을 기술한다.

ABSTRACT

To find optimal path is killer application in the telematics system. The shortest path of conventional system, however, isn't always optimal path. That is, the path with minimum travelling time could be defined as optimal path in the road networks. There are techniques and algorithms for finding optimal path. Hierarchical path algorithm categorizes road networks into major layer and minor layer so that the performance of operational time increases. The path searched is accurate as much as optimal path. At above 2 system, a method to allocate minor roads to major road region influences the performance extremely. This paper proposes methods to determine search space for selecting major roads in the hierarchical path algorithm. In addition, methods which apply the proposed methods to hierarchical route algorithm is presented.

키워드

최적경로, 계층 경로 탐색, 단일 노드, 다중 노드, 영역 설정

1. 서 론

현재 경로안내 시스템은 탐색 비용 및 이동 비용의 감소를 위해 최적경로 탐색이 핵심적인 기술로 인식되어 지고 있다. 최적경로는 경로 정확도와 연산 비용이라는 감소 두 가지 요소를 고려한다. 이 두 가지 사항을 고려한 알고리즘들이 많이 제시되어 지고 있으며 대표적으로 Dijkstra, A*, 계층 경로 탐색, 그리드 기반 경로 탐색 알고리즘 등이 있다. 연산 성능 향상을 위한 계층 경로 탐색 알고리즘은 도로 네트워크를 주 영역과 부 영역으로 나누어 두 계

층으로 최적경로를 탐색하는 알고리즘이다.

현재 많은 논문에서 계층 경로 탐색 알고리즘의 효과적인 응용방법과 연산비용의 감소를 주제로 연구되어 지고 있다. 그러나 주/부 영역을 나누는 기준에 대해서는 구체적으로 제시되지 않았다. 그러나 영역의 지정 방법에 따라 이러한 연산비용을 더욱 줄일 수 있으며, 기존의 방법으로는 지정되기 힘든 형태의 도로 네트워크도 존재하므로 영역을 분할하는 기법은 계층 경로 탐색 알고리즘에서 중요한 기술이다. 영역 분할이란 도로 네트워크를 2계층으로 나누는 것을 의미 하는데 주 영역과 부 영역으

로 구성된다. 주 영역의 경우 도로 네트워크에서 고속으로 이동할 수 있는 노드의 집합이며 이러한 집합으로 형성되는 영역을 부 영역이라고 한다.

본 논문에서는 이러한 계층 경로 탐색 알고리즘의 효과적인 2단계 영역 설정 기법에 대하여 제시한다. 기존에 언급되지 않은 단일 주 영역 노드를 중심으로 하는 단일 노드 부 영역 설정에 대하여 제시하고 다중 주 영역 노드를 구성하여 여러 형태의 다각형 모양의 영역을 설정하는 다중 노드 부 영역 설정 기법에 대해 제시한다.

본 논문은 2장에서 기존의 계층 경로 탐색 알고리즘의 특징과 한계에 대하여 언급한다. 그리고 3장에서는 여러 가지 영역 설정 기법을 제시하고 4장에서는 제시된 기법들에 대한 분석을 제공한다. 끝으로 5장 결론에서 정리한다.

II. 관련 연구

계층 경로 탐색 알고리즘은 연산 비용과 최적 경로 접근 정확도를 높이기 위하여 여러 형태의 방법론이 제시되었다.

Karimi[2]논문에서는 출발지 및 목적지를 포위하는 영역을 만들어 검색 노드의 수를 제한하는 휴리스틱 탐색 방법을 제안하였다. 영역은 여러 개의 window에 의하여 설정되며 그 창 모양이나 크기는 네트워크의 크기에 따라 결정된다.

G. R. Jagadeesh[3]이 제시한 탐색 방법은 출발지와 목적지가 지정되면 각각의 지점이 포함되는 부 영역을 대상으로 1차 탐색이 이루어진다. 1차 탐색은 부 영역안의 지정된 노드(출발/목적)에서 그 부 영역을 감싸고 있는 4개의 주 영역 노드를 찾는 탐색이다. 1차 탐색은 출발지 부 영역과 목적지 부 영역에서 이루어지며 총 8개의 주 영역 노드가 선택 된다. 2차 탐색은 선택 되어진 출발지 주 영역 4개의 노드와 목적지 주 영역 4개의 노드를 $\min(n*m)$ 의 형태로 경로 비용을 계산한다. 여기서 n 은 출발지 주 영역에서 선택된 4개의 노드이며 m 은 목적지 주 영역에서 선택되어진 4개의 노드이다. 즉, 16개의 이동 경로를 탐색하여 최소 비용을 가지는 노드를 최종 결과로 가진다.

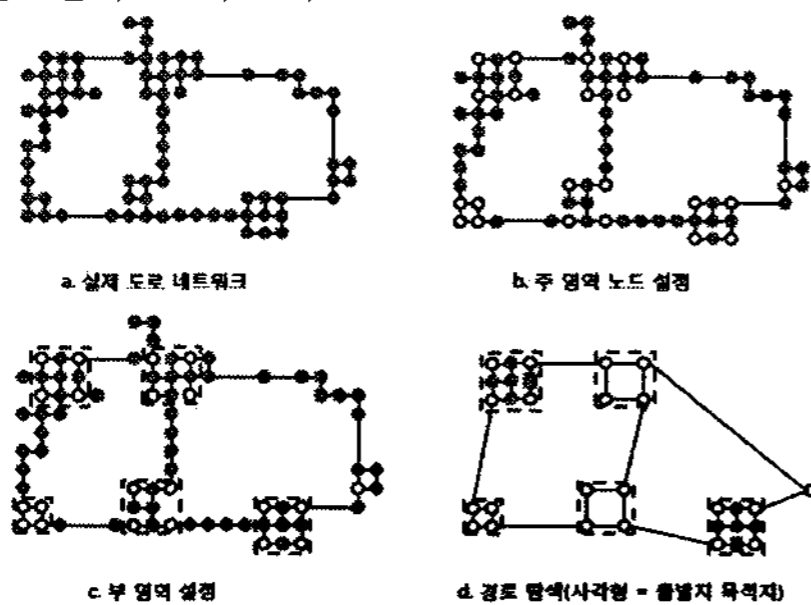


그림 1. 계층 경로 탐색 알고리즘 연산 과정

기존의 계층 경로 탐색 알고리즘은 그림 1의 형태로 그리드를 통하여 주 영역을 설정하는 부분이 도로 네트워크에서 자연스럽게 정의된다. 실제 도로네트워크에서는 자연스럽게 그리드 형태의 주 영역이 설정되기 어려운 부분이 존재하며 1차 탐색 비용과 2차 탐색의 $\min(n*m)$ 의 연산 비용 또한 개선의 여지가 있다. 따라서 이러한 문제점을 해결하기 위하여 효과적인 영역 지정이 필요하다.

III. 영역 설정 기법

3.1 단일 노드 부 영역 설정

계층 경로 탐색 알고리즘은 두 개의 계층으로 구성된다. 주/부 영역으로 이루어지며 기존의 계층 탐색 알고리즘의 경우 4개의 주 영역 노드로 만들어지는 그리드로 부 영역이 지정된다. 앞서 언급한 4개의 노드로 구성을 할 경우 나타나는 두 가지의 문제점을 해결하기 위해 부 영역을 설정할 때 단 하나의 주 영역 노드로 영역을 설정하는 기법에 대하여 제시한다. 주 영역 노드는 도로 네트워크에 존재하는 고속도로의 IC, 도시 고속도로의 진출/입 구간, 그리고 도시의 외곽을 빠르게 이동할 수 있는 외곽 순환도로의 출입구로 지정한다. 즉, 일반 도로에 비하여 빠른 속도로 이동할 수 있는 특정 도로들을 주 영역 노드로 설정한다. 본 논문에서는 주 영역 경로의 경우 항상 부 영역 경로의 이동속도보다 빠르다는 가정을 한다. 그리고 부 영역은 이러한 주 영역 노드에 대하여 인접한 노드들로 구성한다.

구성 방법은 각각의 주 영역 노드를 중심으로 voronoi edge를 통해 부 영역을 지정하는 방법을 사용한다. 이 경우 그림2와 같이 영역이 나누어진다[4]. 즉 하나의 부 영역은 단일 주 영역 노드를 가지는 방법으로 부 영역에서 주 영역을 탐색하는 연산을 최대한으로 줄일 수 있다.

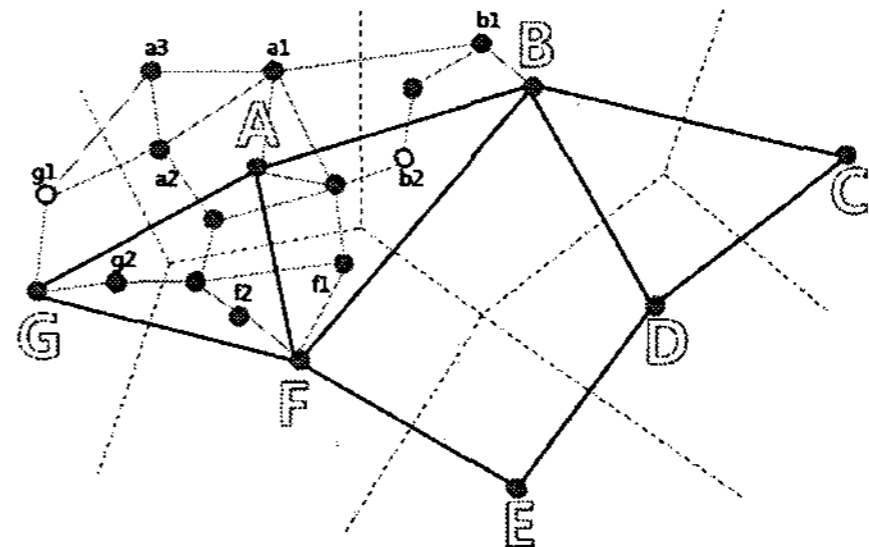


그림 2. 단일 노드 영역 설정

데이터를 저장하는 방법은 데이터베이스에 테이블 형태로 저장하는 방법과 각각의 주 영역의 노드를 상위 트리에 넣고 부 영역에 포함되는 노드들을 하위 트리로 구성하는 방법이 있다[5].

본 논문에서는 데이터베이스에 구성 노드를 저장하여 탐색하는 방법을 대상으로 설명한다. 전체 도로 네트워크의 노드를 저장하고 있는 데이터베이스 테이블에서 주 영역 노드가 선택이 되면 이를 기준으로 하여 그림 3. 과 같이 부 영역이 지정되면 테이블이 생성된다.

테이블은 표1 형태로 존재하며 주 영역 노드와 노드에 포함되는 부 영역 노드들의 정보가 저장되며 출발지와 목적지가 지정되면 생성된 테이블에서 주 영역 노드를 찾아 주 영역 노드의 검색연산을 실행하면 모든 탐색이 종료 된다.

표 1. 단일 노드 부 영역 데이터 테이블

주 영역 노드	영역 포함 노드	인접 부 영역
A	a1, a2, a3 ...	B, F, G
B	b1, b2 ...	A, C, D, F
F	f1, f2 ...	A, B, E, G
G	g1, g2 ...	A, F

인접 부 영역 필드가 존재하는 이유는 현재 부 영역안의 도로 네트워크의 상황이나 목적지의 위치에 따라 반드시 자신의 주 영역 노드 부 영역 노드들이 선택을 하는 것이 최선의 경로를 보장 하지 않기 때문이다. 예를 들어 출발지 b2에서 목적지 g1로 이동할 경우 b2노드는 영역에 존재하는 주 영역 노드인 B를 탐색하며 g1노드는 주 영역 노드인 G를 1차로 탐색한다. 이 때 b2에서 B로 이동해서 목적지를 탐색하는 경로 보다 b2에서 A나 F로 이동하여 목적지를 탐색하는 비용이 더 낮을 수 있다. 도로 네트워크의 이동 속도는 항상 가변적으로 변하기 때문에 이런 경우는 꼭 고려해야 한다.

도로 네트워크의 시간에 따른 속도 변화에 따라서 인접 영역을 탐색할 수 있다. 도로 네트워크의 속도가 심각하게 저하되는 rush hour 시간의 경우는 인접 영역에 대한 탐색을 전부 실행하여 최적에 가장 근접한 경로를 탐색하고 전체 도로 네트워크의 속도가 고속을 유지할 수 있는 시간의 경우는 현재 출발지 노드에서 인접한 주 영역 노드까지의 직선거리를 계산하여 출발지 목적지 직선거리보다 일정 거리 이상 멀어질 경우 탐색 대상에서 제외시키는 방법을 사용한다.

if (all road network average speed < 10km/h)

1. Search All near Major Node
2. Choice Optimum Major Node
3. Path Finding

else if (all road network average speed >= road network limit speed)

1. Start Node -> Destination Node (직선거리 계산)
2. All Near Major Node -> Destination Node (직선거리 계산)
3. 1의 결과와 2의 결과를 비교하여 1보다 일정 길이 이상 길면 경로 탐색 대상에서 제외

여기서 주 영역 탐색 방법으로는 노드의 개수

에 따라 Dijkstra알고리즘이나 A*알고리즘을 선택하여 연산한다.

3.2 다중 노드 부 영역 설정

영역 설정 기법의 다른 방법으로 다중 노드를 통한 부 영역 설정 기법에 대해 설명한다. 다중 노드로 영역을 제안하는 이유는 도로 네트워크의 특성상 단일 노드로 표현하기에는 많은 수의 주 영역 노드가 존재하기 때문이다. 다시 표현하면 주 영역을 구성하는 노드가 많을 경우 단일 노드로 표현을 하려면 많은 양의 데이터를 처리하고 테이블을 만들어야 하는 문제가 발생할 수 있다. 이 경우 여러 개의 다중 노드로 부 영역을 표현할 경우 테이블의 수와 부 영역을 연산하기 위한 비용을 줄일 수 있다.

본 논문에서는 다음과 같은 순서로 다중 노드 부 영역 설정 방법을 제안한다. 먼저 주 영역 노드 위에 동일한 크기의 고정그리드를 적용한다. 이 후에 적용된 그리드 내에 포함되어 있는 주 영역 노드들을 연결한다. 이 결과 3개 이상의 노드가 영역에 있다면 하나의 다각형을 만들어 낼 것이고 하나 또는 두 개의 노드만 존재한다면 포인트나 하나의 선 형태로 존재한다. 이 후 인접한 모든 그리드 내에 있는 노드와 기준이 되는 그리드 내의 노드를 직선으로 모두 연결한다. 노드의 수에 따라 직선의 수는 달라지는데 이 만들어지는 직선이 만약 먼저 이루어진 그리드 내의 영역 간선과 교차되면 직선을 삭제하는 순서의 작업을 진행한다. 나머지 남은 직선을 대상으로 만들어진 직선이 교차되는 부분이 생기면 이 직선도 삭제 한다.

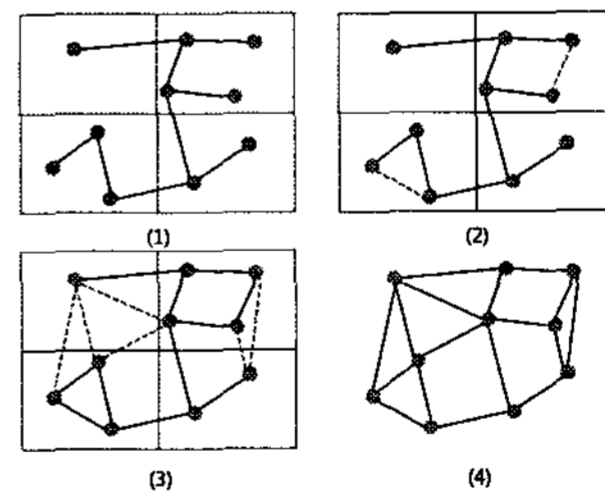


그림 3. 다중 노드 영역 설정

그림 3은 앞서 설명한 영역 지정 순서에 대한 그림이고 아래는 영역설정 기법의 의사코드이다.

```

Region_Determination
{
  if (그리드 내에 3개 이상의 노드가 존재)
  {
    존재하는 노드 모두 연결;
    교차 간선 삭제;
    남은 간선 = 영역의 경계
  }
  인접한 그리드의 모든 노드를 n*m 형태의
  간선으로 연결
  if (선행 작업 시 만들어진 영역의 경
  계나 교차 간선 존재)
    모두 삭제;
  else
    남은 간선 = 영역의 경계
}
    
```

IV. 예외사항 및 발생 문제 해결 방안

지금까지 기존의 방식과는 다른 형태의 영역 지정에 대하여 설명했다. 단일 노드를 중심으로 하는 부 영역 설정 방법과 다중 노드를 중심으로 하는 주 영역 설정 방법을 제안하였다. 이 두 가지의 방법을 이용하면 효과적으로 연산시간을 줄일 수 있을 것으로 추측되며 최적경로의 정확도를 얼마만큼 높일 수 있는가에 대하여 고려하고 영역을 설정할 때 중요한 요인으로 적용시켜야 한다.

현재 두 가지의 알고리즘에서 도로 네트워크 상에 존재하는 장애물에 대한 처리가 문제점으로 발생할 가능성이 있다. 장애물로 인하여 지정된 영역 안의 부 영역 노드에서 주 영역 노드를 탐색하는 것이 불가능할 경우가 발생할 수 있다. 즉, 부 영역의 중심을 호수나 산등의 장애물이 관통하여 그 영역 내의 부 영역 노드들에서 주 영역 까지 이동하려는 경로가 존재 하지 않는 경우가 대표적인 문제이다.

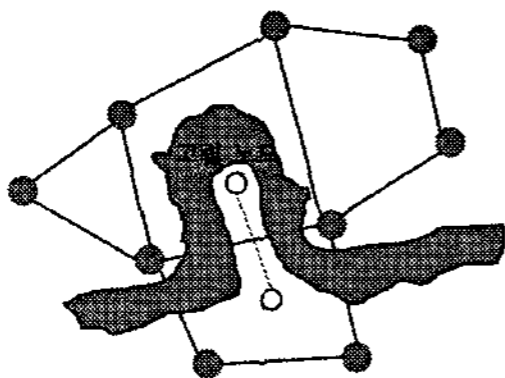


그림 5. 영역 내 고립 노드 발생

이 경우 그림5와 같이 설정 영역 인근의 영역을 통하여 주 노드를 탐색해야 한다. 다중 노드로 영역을 설정할 경우에는 min(n*m)의 모든 경로를 체크하여 연산을 수행하기 때문에 그리드 형태나 그이상의 다각형 형태로 영역이 구성되면 경로 탐색비용이 크게 증가한다.

또한, 주 영역을 이용하지 않고 순수하게 마이너 노드만을 선택하여 이동했을 때의 경로가 최적경로 정확도가 더 높게 나올 경우도 발생한다.

탐색비용의 경우 n*m의 연산을 모두 실행 하지 않고 출발지와 목적지 사이에 가중치가 있는 그리드 영역을 설정하여 추정치가 치명적으로 떨어지는 노드가 나올 경우 이 노드들은 탐색에서 제외하여 연산시간을 감소시킬 수 있다[6].

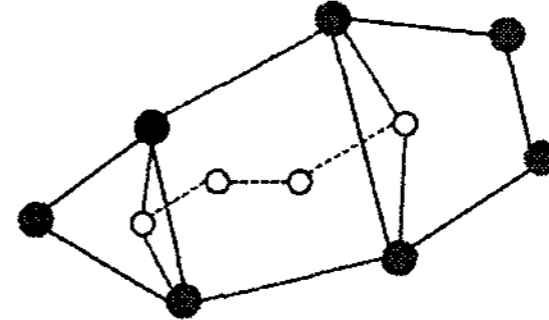


그림 6. 주 영역 이동 비용보다 부 영역 이동 비용이 낮을 경우

그림 6의 경우처럼 주 노드의 이동비용이 부 노드로 이동한 비용보다 크게 나올 경우가 존재한다. 즉 정확도의 문제가 발생하는데 이 경우 출발지와 목적지 사이의 직선거리가 주 영역 경로를 이동했을 경우에 비하여 50% 수준일 경우이면 무조건 부 영역 노드들도 탐색을 해주어야 더욱 정확한 최적경로를 탐색할 수 있다.

V. 결론 및 향후 연구

최적경로를 탐색하기 위해서 고려할 사항 중에 가장 중요한 것이 연산 비용의 감소이다. 최적경로탐색은 부가적으로 고려해야할 도로 정보들이 많기 때문에 연산속도의 증가는 당연한 결과이다. 따라서 이러한 연산 비용을 최소화 하면서 최적경로의 정확도를 유지할 수 있는 탐색 알고리즘이 필요하다.

본 논문에서는 지금까지 도로 네트워크에 효과적으로 적용할 수 있는 계층 경로 탐색 영역을 지정하는 기법에 대하여 제안 하였다. 여러 기법들을 설명하고 방법에 대해서 언급하였으나 정확한 데이터를 산출하기 위해서는 4장에서 언급하였듯이 동일한 데이터를 통하여 각각의 알고리즘들을 비교하는 작업이 필요하다. 따라서 향후 연구는 제시된 알고리즘과 기존의 계층 경로 알고리즘을 구현하여 동일한 경로 데이터를 가지고 테스트를 할 것이다. 또한 제시된 영역 설정 기법들의 장점을 모아 테스트 결과를 토대로 더 효과적인 영역 설정 방법에 대하여 연구할 계획이다.

참고문헌

[1] <http://www.inavi.co.kr>, <http://www.rotis.com>
 [2] H.A.Karimi, "Real-time optimal route computation: a heuristic approach," ITS J., vol. 3, no. 2, pp. 111 - 127, 1996.
 [3] G. R. Jagadeesh, T. Srikanthan, and K. H. Quek, "Heuristic Techniques for Accelerating Hierarchical Routing on Road Networks," IEEE

- Trns. Intelligent Transportation Systems. Vol. 3,
No 4, pp.301-309, 2002
- [4] Improving Worst - Case Optimal Delaunay
Triangulation Algorithms - Geoff Leach
Department of Computer Science Royal
Melbourne Institute of Technology Melbourne,
Australia. June 15, 1992
- [5] Min Woo Lee, Soo Hong Park, "임베디드 시스템
의 객체 관계형 DBMS에 적합한 공간 인덱스 방
법 비교 연구," 한국지리정보학회지, Vol.8, No.1,
Startpage 63, Endpage 74, Totalpage 12
- [6] Hyoun Sup Lee, "Optimal path search based on
fixed grid for telematics applications" 동의대학교
학위논문