
텔레매틱스 단말기를 위한 교통 정보를 활용한 최적 경로 탐색 기법

김진덕*

An Optimal Path Search Method based on Traffic Information for Telematics Terminals

Jin-Deog Kim*

이 논문은 2005년도 동의대학교 교내연구비 지원에 의해 작성되었음

요 약

최근 모바일 단말기의 위치정보를 활용하는 주요 응용 중의 하나인 최적 경로 탐색 시스템은 출발지와 목적지간의 거리뿐만 아니라 탐색 되어지는 구간에 존재하는 많은 교통 상황들을 파악하고 이를 경로 탐색에 활용해야 한다. 그러나 기존의 경로 탐색 알고리즘은 교통상황들을 적절히 이용하지 못하고 있다.

이 논문에서는 새로운 최적 경로 알고리즘을 제안한다. 알고리즘은 최적경로를 검색하기 위해 교통상황을 고려하고, 연산비용을 줄이기 위해 도로를 그리드형태로 나누어 각각의 평균속도를 가지고 휴리스틱 가중치를 부여한다. 또한 알고리즘의 전체 수행시간, 노드 접근 횟수, 최적경로의 정확도를 항목으로 하는 실험을 수행하여 기존의 탐색 알고리즘인 Dijkstra 알고리즘과 A*알고리즘과의 성능평가를 실시하였다. 실험 결과 제안한 알고리즘이 타 알고리즘에 대해 좋은 성능을 보여주었다. 제안한 알고리즘은 향상된 응용을 지원하는 텔레매틱스 시스템에 유용하게 사용될 것으로 기대된다.

ABSTRACT

Optimal path search algorithm which is a killer application of mobile device to utilize location information should consider traffic flows of the roads as well as the distance between a departure and destination. The existing path search algorithms, however, are not able to cope efficiently with the change of the traffic flows.

In this paper, we propose a new optimal path search algorithm. The algorithm takes the current flows into consideration in order to reduce the cost to get destination. It decomposes the road networks into Fixed Grid to get variable heuristics. We also carry out the experiments with Dijkstra and A* algorithm in terms of the execution time, the number of node accesses and the accuracy of path. The results obtained from the experimental tests show the proposed algorithm outperforms the others. The algorithm is highly expected to be useful in a advanced telematics systems.

키워드

모바일 단말기, 텔레매틱스, 최적경로탐색, 고정그리드

I. 서 론

최근 PDA, 휴대폰 등 모바일 단말기에서는 운전자에게 목적지에 도착할 수 있는 최단 경로를 제공하는 차량항법 시스템(CNS)이 주요 응용분야 중의 하나이다.

지금까지는 출발지에서 목적지까지 최단거리로 이동 가능한 도로를 찾기 위한 경로 탐색이 존재 하였으나 현재에는 단순히 경로만을 탐색하는 것을 벗어나서 최소한의 비용으로 가장 빠른 시간 내에 목적지를 찾아 갈수 있는가 하는 문제가 경로탐색에서의 핵심으로 떠오르고 있다. 여기서 말하는 비용을 줄이기 위해서는 두 가지 측면을 고려하여야 한다.

첫째, 탐색된 경로에 대하여 이동 거리가 아닌 이동시간이 감소해야 한다.

둘째, 시스템의 성능 측면에서 경로 탐색의 연산시간 및 탐색되는 노드의 수가 감소해야 한다. 클라이언트/서버 시스템에서 서버 중심형 최적 경로 탐색 서비스를 할 경우 서버에서 대부분의 연산이 수행되며, 많은 수의 클라이언트에서 요구하는 질의에 대한 결과를 전송해주기 위해서는 서버의 과부하 현상이 초래된다. 따라서 연산시간과 노드수의 감소는 텔레매틱스 서버의 효율적 운용에 직접적인 영향을 준다. 반대로 클라이언트 중심형 최적 경로 탐색 시스템일 경우에도 PDA와 텔레매틱스 단말기와 같은 모바일 단말기의 낮은 시스템 성능을 감안하면 검색 비용을 낮추는 것이 매우 중요하다고 할 수 있다.

또한 최적 경로 검색을 위해서는 기존의 도로 정보 네트워크 뿐만 아니라 도로위의 실시간 교통정보를 적절히 이용해야 한다.

이 논문에서는 대표적인 경로 탐색 기법인 A*알고리즘 [1,2]을 변형하여 교통정보에 적응적인 가변 휴리스틱 기반 최적경로 탐색 기법을 제안한다. 구체적으로 이동거리의 증가로 인한 검색범위의 증가를 효과적인 휴리스틱 가중치와 고정 그리드의 검색범위 분할을 적용하여 최적경로를 검색함과 동시에 연산범위를 최대한 줄여 빠른 시간 안에 검색할 수 있는 알고리즘을 제안한다. 그리고 제안된 알고리즘을 구현하여 기존의 경로 탐색 기법인 Dijkstra 알고리즘 [3]과 A*알고리즘과의 성능 비교를 수행하여 고정 그리드 기반 최적경로 탐색 알고리즘이 타 알고리즘에 비하여 효과적인지를 밝힌다.

이 논문의 구성은 다음과 같다. 2장에서 기존의 경로 탐색 알고리즘들의 특징 및 장단점을 알아보고, 3장에서

는 본 논문에서 제시한 가변 휴리스틱 기반 최적경로 탐색 알고리즘에 대해 자세히 살펴보고, 4장에서는 실제로 데이터를 바탕으로 실험한 성능을 평가한다. 마지막으로 5장에서 결론을 맺는다.

II. 관련연구

이 장에서는 경로 탐색의 정의와 경로 탐색을 위한 기존 연구인 Dijkstra 알고리즘 [3]과 A*알고리즘 [1,2,4]에 대해 자세히 기술하고, 아울러 현재 상용 서비스 중인 최적 경로 탐색 기법에 대해 살펴보고자 한다.

2.1. 최적 경로 탐색의 정의

경로 탐색은 정해진 목적지까지 도착하기 위하여 이동할 수 있는 경로를 찾는 것을 말한다. 그러나 단순히 짧은 거리만을 고려한다면 정체구간이나 사고가 발생한 구간이 탐색된 경로에 포함이 되어 있을 경우 그 경로는 최단 경로이지만 이동시간은 최악의 경로가 될 수 있는 가능성이 높다.

따라서 최적경로는 이동하는 거리뿐만 아니라 이동하는 구간의 속도 및 도로 상황들과 같은 종합적인 교통 정보들을 모두 고려하여 탐색되어야 한다. 교통 정보들을 대상으로 연산을 하는 경우 단순히 최단경로를 찾아내는 기법에 비하여 실시간 연산 처리 능력이 필요하다. 따라서 경로 탐색의 연산 시간 최소화화와 경로 정확도의 향상이 최적경로 탐색에 필수적이 요소들이다 [5, 6].

2.2. 최적 경로 탐색 방법론

Dijkstra 알고리즘은 최단 경로 탐색을 위해 사용되는 알고리즘으로 출발지로부터 인접해있는 모든 노드들을 차례로 방문하여 목적지까지의 비용이 가장 작은 경로를 찾아내는 알고리즘이다 [3]. 이 알고리즘은 경로 탐색 정확도는 뛰어나지만, 연산비용이 매우 크다는 단점이 갖고 있다 [7].

A*알고리즘은 최단 거리 찾기에서 효과적으로 동작하는 알고리즘으로 연산 비용을 줄이기 위해 목표에 얼마나 근접한 것인지를 평가하는데 휴리스틱 함수를 사용한다 [1]. 휴리스틱에 의해 먼저 가장 바람직한 방향을 탐색하게 된다. 그 방향이 실패하면 다른 경로를 찾게 된다. A*알고리즘은 탐색의 시작 노드에서부터 바깥쪽으로 균일

하게 진행되지 않고 출발지점에서 목표지점까지 가장 비용이 낮은 경로를 찾는다[6,8].

A* 알고리즘은 가중치의 범위에 따라 목적지 방향으로의 직진성의 차이가 있으므로 정확한 계산을 통한 가중치의 부여가 주요 관건이다[4,8]. 즉, 가중치를 강하게 부여를 하면 검색 범위가 출발지에서 목적지 방향으로 직선에 가까운 형태를 가지므로 탐색 범위와 접근 노드 수 검색 시간은 모두 줄어드는 반면, 탐색된 경로가 최적일 확률은 떨어진다. 반대로 목적지 방향으로 약하게 부여할 경우 검색 범위는 증가하며 접근 노드 수와 검색 시간은 증가한다[4,8].

이 밖에도 이와 관련한 연구[11,12,13,14,15]가 진행되고 있지만 주로 최단 경로 탐색에 관한 연구로 아직까지는 위의 두 가지 기능에서 크게 벗어나지 않고 있다.

이 논문에서는 수집된 교통정보에 대한 통계 데이터를 근거로 A* 알고리즘에서 휴리스틱 가중치를 가변적으로 적용하여 경로 탐색 시간 및 경로 정확도를 최적화 할 수 있는 기법을 제안하고, 구현하여 성능을 평가하고자 한다.

III. 가변 휴리스틱 기반 최적경로 탐색 기법

일반적인 A* 알고리즘은 주로 게임 등에서의 경로 찾기에 많이 활용되고 있다. 이와 같은 경우 비트맵 기반의 화소 단위[6]로 가중치를 부여하여 최적의 경로를 탐색한다.

이러한 A* 알고리즘을 도로 교통 네트워크에 적용하고자 할 경우 도로 교통 데이터가 비트맵 형식의 격자로 되어 있지 않으며, 매우 많은 수의 링크가 존재하여 각 링크마다 휴리스틱 가중치를 달리 부여하는 것이 어렵기 때문에 휴리스틱 가중치는 일정한 값으로 고정한다. 따라서 전술한 바와 같이 고정된 휴리스틱 가중치는 연산속도와 경로 정확도 성능을 동시에 높이는 어렵다. 그리고 시시각각으로 변하는 교통상황에 맞추어 각 구간별로 휴리스틱 가중치를 오퍼레이터의 실시간으로 결정한다는 것도 불가능하다.

그러므로 최적경로 탐색 정확도를 높임과 동시에 경로 탐색 시간을 줄이기 위해서는 실시간적으로 휴리스틱 가중치를 가변화할 수 있는 효과적인 방법이 필요하다.

3.1. 고정 그리드 기반 가변적 휴리스틱

자주 변하는 교통 정보를 고려하여 가변적으로 휴리스틱을 적용하기 위해서 본 논문에서는 고정 그리드를 사용하는 방법을 제안한다. 그림 1에 나타난 바와 같이 고정 그리드는 도로위에 일정한 크기의 격자로 분할하며 여러 개의 노드가 하나의 그리드 안에 들어가게 하는 기법이다.

이 기법은 A* 알고리즘의 화소단위 가중치 부여와 유사하지만, 전술한 바와 같이 도로 교통 네트워크를 A* 알고리즘에 적용하여 경로 탐색을 할 경우 고정된 휴리스틱 가중치를 사용해야만 한다. 반면에 이 논문에서 제안한 그리드 기반 가변 휴리스틱 기법은 그리드 별로 차량의 이동 속도를 근거로 휴리스틱 가중치가 다르게 지정되기 때문에 가변적인 교통상황들을 반영할 수 있다[16].

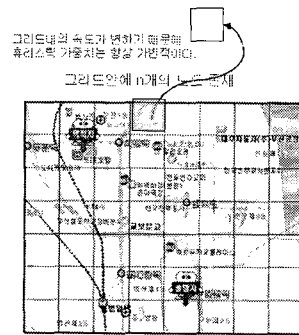


Fig. 1. Fixed Grid in Road Network

제안한 그리드 기반 가변 휴리스틱 기법을 사용하여 탐색을 하는 경우 도로의 속도가 빠른 부분에서는 휴리스틱 가중치를 낮춰 탐색 정확도를 높이고, 속도가 느린 구간에서는 휴리스틱 가중치를 높여 직진성을 강화함으로써 탐색 시간을 줄이는 방법으로 택하여 탐색 정확도 및 탐색 시간 성능을 극대화한 최적 경로 탐색 기법이 된다.

3.2. 교통정보 적응적 탐색 기법

이 논문에서 제안하는 교통환경 적응적 최적경로 탐색 알고리즘은 다음과 같다.

Optimal_Path_Search Algorithm

```
{
Input : Start & Goal Node
Data : Heuristic Value of Each Grid
```

Output : Optimal_Path(List of Node)

```

-----
CurrentNode = Start;
ClosedNode.Add(CurrentNode);
While( CurrentNode != GoalNode)
{
    Get Nodes connected CurrentNode;
    // exclude the nodes of ClosedNode
    OpenNode.Add(Nodes)
    ForEach( Ni in OpenNode )
    {
        Calc. Cost of each Node
        Cost : F(Ni) = G(Ni) + H(Ni) 단,
        G(Ni) : Cost_From_Start(Ni)
                지금까지 계산된, 출발지부터 노드Ni까지의 최소 비
                용
        H(Ni) : Cost_To_Goal(Ni)
                노드Ni부터 목적지까지의 추정 비용으로서 Ni와 목
                적지 사이의 직선경로에서 각 그리드의 휴리스틱 값
                과 그리드 내의 직선 거리의 곱의 합
    }
    Select Nj which has min F(n);
    CurrentNode = Nj;
    ClosedNode.Add(CurrentNode);
}
Optimal_Path : ClosedNode
}

```

위 알고리즘은 의사결정 과정 정보인 휴리스틱(heuristic)에 따라 목표까지의 비용이 가장 적은 경로 상에 있다고 판단되는 노드를 우선적으로 방문하여 가능해진다. 탐색 시 다음 단계에 확장할 노드를 결정하는 휴리스틱 평가함수 $f(n)$ 이 있을 때 $f(n)$ 이 가장 작은 노드를 확장하고 확장할 노드가 목표 노드이면 탐색을 종료 하는 것이다. $f(n)$ 은 다음과 같이 $f(n) = g(n) + h(n)$ 로 나타내어 질 수 있다. 여기서 $g(n)$ 은 시작노드로부터 노드 n 까지 오는 데 드는 이미 계산된 비용이다. $h(n)$ 은 휴리스틱(heuristic) 평가 값으로, 노드 n 에서 목표까지 가는데 드는 '추정된' 비용이다. 알고리즘의 OpenNode는 비교 대상이 되는 후보 노드들이며, ClosedNode는 이미 방문한 노드들이다.

Cost_To_Goal Algorithm

```

{
    Input : Ni, GoalNode
    Data : Heuristic Value of Each Grid
    Output : Estimated Cost to Goal Node
}

```

```

-----
Draw a straight line from Ni to Goal
Get Grids overlapped with the line
Cost = 0;
ForEach( Gi in Grids )
{
    Cost += Heu(Gi) * LineLength(Gi)
}
return Cost
}

```

위의 알고리즘에서 기존의 A* 알고리즘과 가장 큰 차이 점은 추정비용 H값을 구하는 방식이다. H(Ni)값을 보다 빨리, 그리고 정확히 구하기 위해 이 논문에서 제안한 고정 그리드 기반 가변 휴리스틱을 사용한다. 각 그리드의 휴리스틱 가중치는 각 그리드에 포함된 도로 링크의 최고 속도를 근거로 하였다.

각 그리드마다 그 속도가 다르므로, 현재 도로 상황을 고려한 가변 휴리스틱이 적용됨을 알 수 있다. 예를 들어, 한 그리드의 최고 속도가 매우 낮은 경우라면 H(Ni)값이 과대평가(OverEstimated)되는 특성을 보인다. 이는 휴리스틱 가중치가 매우 크다는 것을 의미하며, 탐색의 직진성이 강해 탐색 속도를 높인다. 반면, 한 그리드의 최고 속도가 매우 높은 경우에는 H(Ni)값이 과소평가(UnderEstimated)되며, 휴리스틱 가중치가 낮아 탐색 범위를 넓히는 효과가 있다.

위와 같은 가변 휴리스틱은 도시 도로의 특성을 반영하는 것으로서, 교통 체증지역일 경우 가중치를 높여 탐색 속도를 높이고, 교통 흐름이 원활한 도로는 가중치를 낮춰 탐색 범위를 넓힘으로써 최적의 경로를 찾을 수 있다.

IV. 성능 평가

제시한 알고리즘을 대상으로 시스템을 구현하여 성능을 평가하였다. 개발환경으로서 개발 언어는 C#을 사용하였고, 평가 대상은 Dijkstra 알고리즘, A* 알고리즘, 가변적 휴리스틱을 적용한 고정 그리드 기반 최적경로 탐색 알고리즘(이하 그리드 기반 알고리즘)이다.

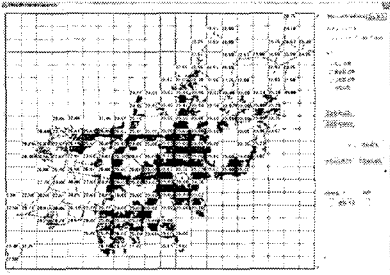


그림 4. 최적 경로 탐색 시스템
Fig. 4. Optimal Path Search System

그림 4는 시스템을 구현한 최종 프로그램의 화면이다. 성능 평가를 위한 데이터로 부산지역의 도로 교통 네트워크 및 실시간 각 노선의 평균 이동 차량 속도를 이용하였다. 따라서 각 그리드의 평균 속도 및 최고 속도를 추출하였다. 성능 평가를 위해 여러 가지의 출발지와 목적지를 임의로 설정하여 검색할 수 있도록 구성하였다. 출발지와 목적지를 정하여 경로 탐색을 시작하면 화면에 탐색된 경로가 빨간 선으로 표시되며 연산시간과 접근 노드 횟수, 평균 속도, 이동 시간의 정보를 확인 할 수 있다.

실험은 1200번의 경로 탐색을 통하여 평균적인 데이터를 산출하여 결과 값에 반영하였다. A*알고리즘의 경우는 휴리스틱 가중치를 한 범위만 지정 할 수 없기 때문에 최고 속도를 20, 40, 60, 80, 100으로 나누어 실험하였다 [16]. 모든 알고리즘은 연산 속도와 탐색 노드 수 그리고 최적경로 정확도를 대상으로 하여 결과 값을 추출 하였다.

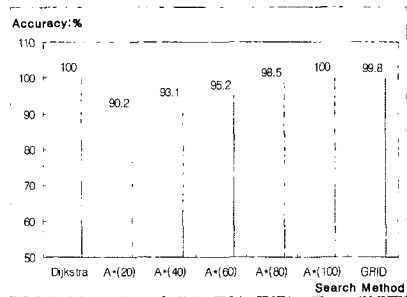
4.1. 이동 거리별 탐색 경로 정확도

경로 탐색 알고리즘에서 정확도란 최적 경로에 얼마나 접근하는 가를 나타내는 척도이다. 따라서 탐색 경로의 정확도가 높다는 것은 질의 결과에 대해 신뢰할 수 있음을 의미한다. 그림 5는 Dijkstra, A*(5개의 휴리스틱별), 그리드 기반 경로 탐색기법에 대해 각각 출발지와 목적지사이의 이동 거리 별로 탐색된 결과의 정확도를 백분율로 나타낸 것이다.

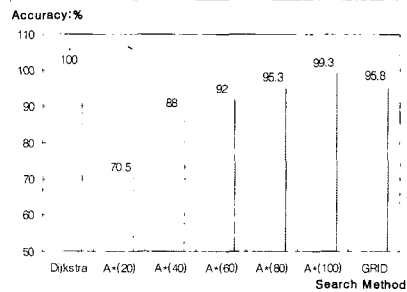
그림 5에서 A*알고리즘과 그리드 기반 알고리즘을 비교해 보면 A* 알고리즘의 휴리스틱 가중치가 매우 큰 경우만 A*기법이 약간 더 좋은 성능을 보일 뿐 나머지 경우에는 이 논문의 탐색 기법이 훨씬 좋은 성능을 보임을 알 수 있다. 또한 A*알고리즘에서 휴리스틱 가중치가 높을 경우 향후 설명할 연산시간과 노드접근 횟수가 매우 증가

하기 때문에 정확도와 연산시간은 상호상쇄효과(Trade Off)가 있음을 그림 6, 7에서 보인다.

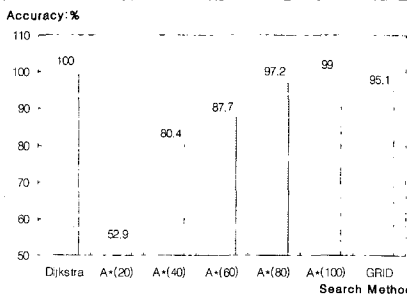
이동 거리 별로 보면 단거리(그림 5(a)) 일 경우 A*와 그리드 기반 탐색 기법이 장거리(그림 5(c))에 비해 Dijkstra의 성능에 더 접근하는 좋은 성능을 보이고 있다. 한편, 장거리일수록 A* 기법은 정확도가 떨어지며 특히 휴리스틱 가중치가 작은 경우 정확도가 50%를 약간 상회하는 저조한 성능을 보이는 반면, 이 논문의 적용적 탐색 기법은 거리에 관계없이 항상 95%이상의 우수한 성능을 보임을 알 수 있다.



(a) Short Distance (10Km 미만)



(b) Middle Distance (10~40Km)



(c) Long Distance (40 Km 이상)

그림 5. 탐색 기법별 경로정확도
Fig. 5. Accuracy of Searched Path

4.2. 이동 거리별 연산 수행 시간

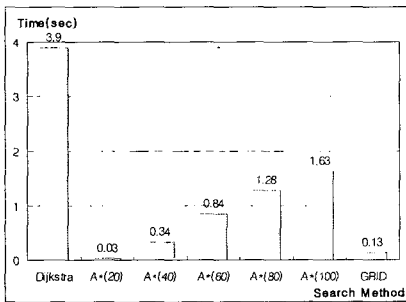
연산 속도의 단축은 시스템의 성능에 직접적인 영향을 주기 때문에 중요한 부분이다. 그림 6은 각 탐색 기법에 대해 각각 출발지와 목적지사이의 이동 거리 별로 경로 탐색으로 위해 소요되는 연산 시간을 나타낸 것이다. 평균적으로 Dijkstra 알고리즘의 연산속도가 가장 큰 값을 가지는 것을 볼 수 있다. 이것은 Dijkstra 알고리즘의 특징인 모든 노드 대상의 경로 검색 때문이라고 할 수 있다. 그리고 A*알고리즘을 보면 휴리스틱 가중치별로 연산 속도의 차이가 크게 나타난다. 특히 휴리스틱 가중치가 적을 경우 직진성이 강하기 때문에 아주 적은 연산시간을 보이

지만, 경로 정확도는 떨어질 것으로 예상되었고, 앞에서 설명한 그림 5에서 이러한 예상이 실험결과로 나타남을 알 수 있었다. 반면, 이 논문의 탐색 기법은 Dijkstra 기법에 근접하는 좋은 성능을 보임을 알 수 있다.

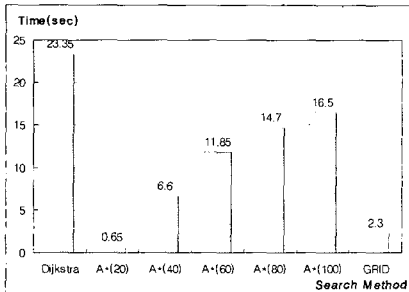
이동 거리 별로 살펴보면 단거리(그림 6(a))일수록 A*와 적응적 탐색 기법이 좋은 성능을 보이는 가운데, 장거리(그림 6(c)) 일 경우 A*알고리즘에 비해 적응적 탐색 기법이 상대적으로 좋은 성능을 보이고 있다.

4.3. 이동 거리별 노드 접근 횟수

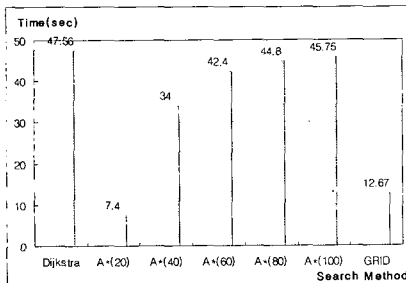
연산 속도와 마찬가지로 접근 노드 수 또한 마찬가지로 시스템의 성능에 영향을 주는 요인이다. 그림 7은 각 탐색에 대해 이동 거리별로 노드 접근 횟수의 평균값을 나타낸 것이다. 전체적으로 그림 6의 연산수행시간과 거의 유사한 형태의 그래프임을 알 수 있다. 그림 7에서도 낮은 휴리스틱 가중치를 적용한 경우에만 A*알고리즘이 가장 좋은 성능을 보이고 있지만, 그림 5에서 설명한 탐색 정확도에서 상당히 저조한 성능을 보이고 있기 때문에 현실적으로 사용이 불가능한 휴리스틱임을 알 수 있다.



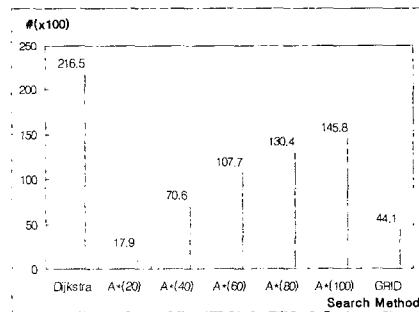
(a) Short Distance (10Km 미만)



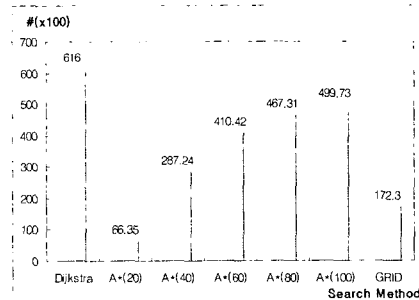
(b) Middle Distance (10~40Km)



(c) Long Distance (40 Km 이상)

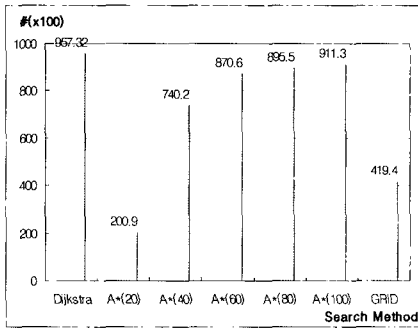


(a) Short Distance (10Km 미만)



(b) Middle Distance (10~40Km)

그림 6. 탐색 기법별 연산시간
Fig. 6. Operation Time



(c) Long Distance (40 Km 이상)

그림 7. 탐색 기법별 노드접근횟수
Fig. 7. Number of Node Access

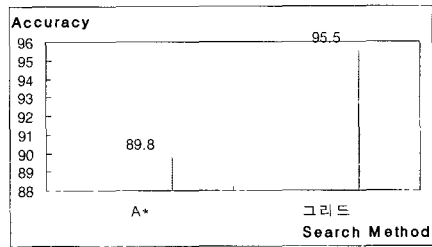
4.4. 각 탐색 기법의 특징 비교

표 1은 각 탐색 기법의 특징을 정리한 것으로서 Dijkstra 기법을 100으로 보았을 때 각 평가 항목을 비교 평가한 것이다. 표1에서 나타났듯이 모든 결과를 토대로 볼 때 경로의 정확도는 Dijkstra가 뛰어나지만, 연산시간 및 노드 접근횟수에서 매우 성능이 떨어진다. A* 기법은 휴리스틱 가중치에 따라 상이한 결과를 보이는 가운데, 연산 속도와 경로 정확도가 상호 상쇄효과로 두 가지 성능을 둘 다 높일 수 있는 방안은 없다. 따라서 도로의 이동 속도의 거의 균등한 특수한 경우에는 좋은 성능을 보이겠지만 현실적으로 그러한 경우는 없으므로 적용 불가능하다. 또한 휴리스틱 가중치를 시스템이 적절하게 설정할 수 없다.

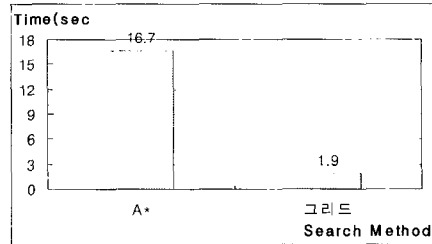
한편, 이 논문의 그리드 기반 탐색 기법은 표 1에서 나타난 바와 같이 경로 정확도가 Dijkstra에 접근함과 동시에 연산 시간과 탐색 노드 수는 Dijkstra의 10~20%에 불과하므로 빠른 시간 내에 최적 경로를 찾아내는 최적의 방법이라 볼 수 있다. 특히, 그리드 기반 탐색 기법은 휴리스틱 가중치를 가변적으로 시스템이 설정하므로 실시간으로 변하는 교통환경에 유동적으로 대처하여 항상 최적 경로를 탐색해 주는 장점이 있다.

표 1. 알고리즘의 특성
Table 1. Characteristics of each Algorithm

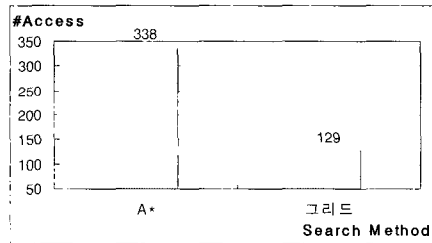
	Dijkstra	A*(가중치별)	그리드 기법
평균경로 정확도	100	52~95	96
평균연산시간	100	5~90	12
평균탐색노드수	100	10~85	25
장점	경로정확도	이동속도가 일정한 도로	가변적 교통상황



(a) Accuracy with Same Operation Time



(b) Operation Time with Same Accuracy



(c) No. of Node Access with Same Accuracy

그림 8. A*와 그리드 기법의 성능 비교
Fig. Performance Comparison(A* vs. Grid)

그림 8은 동일한 조건에서 A*와 본 논문의 그리드 기반 탐색 기법의 성능을 비교 평가한 것이다. 그림 8(a)는 1200 번의 실험 중 같은 출발지와 목적지에 대해 거의 동일한 탐색 비용일 경우의 정확도를 평균한 값으로서 그리드 기반 기법이 훨씬 좋은 정확도를 보이고 있다. 그림 8(b)는 동일한 최적 경로 정확도일 때 평균 연산 시간을 나타낸 것으로서 그리드 기반 기법이 10분의 1 정도의 연산 시간 만으로도 동일한 정확도를 보여 주었다. 또한 그림 8(c)는 동일한 정확도일 때 접근 노드수를 나타낸 것으로서 또한 본 논문에서 제안한 기법이 훨씬 좋은 성능을 보임을 알 수 있다.

이상의 실험 결과를 종합해 볼 때 이 논문에서 제안한 기법은 두 가지 측면에서 명확한 장점이 있음을 알 수 있다. 첫째, 연산 비용과 경로 정확도 측면에서 성능이 뛰어나다는 것이다. 둘째, 오퍼레이터의 개입없이 시스템에

의해 휴리스틱을 가변적으로 설정할 수 있으므로, 실시간으로 변화하는 교통정보를 최대한 반영하여 항상 좋은 성능을 보인다는 것이다.

이와 같은 두 가지 장점은 실시간으로 변화하는 교통정보를 적극적으로 활용해야 하는 텔레매틱스 서버 시스템과 낮은 성능을 가진 텔레매틱스 단말기에서 유용하게 활용될 수 있다.

V. 결 론

이 논문에서 제안한 도로 교통정보에 적응적인 그리드 기반 최적 경로 탐색 기법은 교통 정보를 이용하여 탐색 범위를 최소화 시켜 연산시간 및 경로 정확도를 높임으로써 시스템의 부담을 덜어 줄 수 있다. 각 탐색 기법에 대해 실험으로 성능 평가하여 연산 시간과 노드 탐색 횟수 및 최적 경로 정확도를 확인한 결과 A* 알고리즘의 경우 운영자가 지정해야 하는 휴리스틱 가중치에 의하여 모든 성능이 좌우되며, 연산시간과 경로 정확도의 상쇄효과가 있었다. 반면, 그리드 기반 최적경로 탐색 기법은 수시로 변화하는 교통 상황을 연산에 최대한 반영을 하기 때문에 연산비용의 효율과 최적경로 탐색의 정확도가 모두 뛰어난 성능을 가지고 있다. 연산시간은 Dijkstra 알고리즘의 12% 정도의 시간이 소요되었고, 탐색 노드 수 또한 약 25% 수준에 머물렀다. 최적경로 정확도 또한 95% 이상 접근하는 뛰어난 성능의 향상이 있었다.

이 논문의 결과는 다량의 클라이언트로부터 최적 경로 탐색 질의를 받아 처리해야 하는 텔레매틱스 시스템일 경우 서버에 대한 부하를 대폭 줄일 수 있고, 독자적으로 수행되는 클라이언트 기반 텔레매틱스 기반 시스템일 경우에도 연산시간을 줄일 수 있어 효과적인 운용이 가능하다. 특히 한 번에 최적경로 탐색으로 끝나는 것이 아니라, 탐색된 경로로 운행 중인 경우에도 교통상황의 변화가 발생할 때마다 수시로 최적경로 탐색을 해야 하므로 본 논문에서 제안한 기법은 텔레매틱스 응용 서비스의 품질을 대폭 향상시킬 수 있을 것으로 기대 된다.

참고문헌

[1] J. Mostow, A. Prieditis, "Discovering Admissible

Heuristics by Abstracting and Optimizing: A Transformational Approach," International Conference on Artificial Intelligence, pp.701-707, 1989.

- [2] Peter E. Hart, Nils J.Nilsson, Bertram Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths in Graphs" IEEE Trans. on Systems Science and Cybernetics, Vol. SSC-4, No. 2, pp 100-107, 1968
- [3] E.W.Dijkstra, "A note on two problems in connection with graphs", Numerische Mathematik, Vol. 1, pp. 260-271, 1959
- [4] <http://theory.stanford.edu/~amitp/Game Programming>
- [5] Stephan Winter, "Modeling Costs of Trunks in Route Planning", GeoInformatica, Vol. 6, No. 4, pp. 345-360, 2002
- [6] 이세일, "타일맵에서 A* 알고리즘을 이용한 유닛들의 길 찾기 방법 제안" 한국컴퓨터정보학회 논문지, Vol. 9 No.3, pp. 71 ~ 77
- [7] 이현섭, 안준환, 김진덕, "교통정보 기반 최적 알고리즘", 한국 해양정보통신학회 2004 추계 종합학술대회, Vol.8, No. 2, pp. 425-428, 2004
- [8] H. Kaindl, G. Kainz, "Bidirectional Heuristic Search Reconsidered", Journal of Artificial Intelligence Research, Vol.7, pp.283-317, 1997
- [9] <http://qnavi.bizemeka.com>
- [10] <http://drive.nate.com>
- [11] <http://inavi.co.kr>
- [12] <http://visionjoy.co.kr>
- [13] Stefano Pallottino, Maria Grazia Scutella, "Shortest Path Algorithms in Transportation Models : Classical and Innovative Aspects", TR, Univ. of Pisa, 1998
- [14] Woo Young Kwon, Sanghoon Lee, Il Hong Suh, "A reinforcement learning approach involving a shortest path finding algorithm" Proc. of Int. Conf. Intelligent Robots and Systems 2003, Vol. 7, 2003
- [15] Chang Wook Ahn, R. S. Ramakrishnan, "A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations", IEEE Transaction on Evolutionary Computation, Vol.6, pp. 566-579, 2002
- [16] 이현섭, 김진덕, "고정 그리드 기반 가변 휴리스틱을 이용한 최적경로 탐색", 한국해양정보통신학회 2005 추계 종합학술대회, Vol.9, No.2, pp.137-141, 2005

저자소개

김진덕(Jin-Deog Kim)



1993년 부산대 컴퓨터공학과(공학사)

1995년 부산대 대학원 컴퓨터공학과
(공학석사)

2000년 부산대 대학원 컴퓨터공학과
(공학박사)

1998.3~2001.2 부산정보대학 정보통신계열 전임강사

2001.3~ 현재 동의대학교 컴퓨터공학과 조교수

※관심분야: 객체 지향DB, 지리정보시스템, 공간 질의,
공간 색인, 모바일 데이터베이스, 텔레매틱스