
최고 속력 동적 헤리스틱을 이용한 경로탐색

김지수* · 이지완* · 조대수**

Path Finding with Maximum Speed Dynamic Heuristic

Ji-Soo Kim* · Ji-wan Lee* · Dae-Soo Cho**

이 논문은 2007년도 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임

요 약

일반적으로 단말기 기반의 경로탐색 시스템은 자체적으로 내장된 정보만을 사용하기 때문에 센터 기반의 탐색보다 경로의 질이 떨어진다. 최근 TPEG와 같은 기술을 이용하여 단말기 기반의 경로탐색 시스템에서도 실시간 교통 정보를 전송 받을 수 있다. 그러나 단말기의 제한된 성능으로 인해 실시간 교통정보를 모두 활용하여 경로의 질을 높이면 탐색 비용이 급격히 증가하는 문제가 있다. 이 논문에서 단말기 기반 시스템에서 경로탐색의 질을 높이기 위해 동적 헤리스틱을 제안한다. 동적 헤리스틱이란 고정된 정보가 아닌 서버로부터 실시간 도로정보를 전송받아 동적으로 변경되는 헤리스틱이다. 이 논문에서는 최고속력 동적 헤리스틱을 이용하는 경로탐색 기법을 제안하고, 성능 실험을 하였다. 최고속력 동적 헤리스틱이란 지도를 일정크기로 나눈 그리드내의 도로 중 최고속력을 동적 헤리스틱으로 활용하며, 최고속력이 높은 곳을 우선 탐색함으로 인해 거리에 따른 정보뿐만 아니라 실시간 정보를 활용하여 경로를 탐색한다. 성능 실험 결과 A* 알고리즘에 비해 경로의 질은 향상되었으나 탐색비용이 조금 늘었다.

ABSTRACT

Generally, the Terminal Based Navigation System(TBNS) used embedded road data searches a path that has less qualitative than The Center Based Navigation System(CBNS). TBNS has not used real time road data but it is recently able to use it with technique such as TPEG. However, it causes to increase a cost of exploring by using real time road data for improvement quality of a path, because of limited performance. In this paper, we propose a Dynamic Heuristic to improve quality of path in the TBNS. Dynamic Heuristic(DH) is not fixed data and is dynamically modified using transferred real time road data from server. In this paper, we propose path-finding algorithm with Maximum Speed Dynamic Heuristic (DH-MAX) and do an experiment. The DH-MAX is to be used the highest speed as DH, in real map divided by same size. And proposed algorithm searches path using the priority searching only of the fixed data, but also the highest speed with real time information. In the performance test, the quality of path is enhanced but the cost of searching is increased than A* algorithm.

키워드

동적 헤리스틱, 그리드 제거법, A* 알고리즘

* 동서대학교 컴퓨터정보공학부

** 동서대학교 컴퓨터정보공학부 조교수(교신저자)

접수일자 2009. 03. 18

심사완료일자 2009. 03. 30

I. 서 론

최근 차량의 보급과 더불어 텔레매틱스 서비스를 제공하는 네비게이션 단말기 보급이 급격히 증가하고 있다. 네비게이션 단말기는 경로탐색의 주체에 따라 크게 두 가지로 나누어진다. 센터 기반의 경로탐색 시스템은 수집된 도로 교통 정보를 기반으로 한 최적 경로탐색을 실시한다. 사용자는 시작점과 목적지를 서버로 전송하며, 서버는 경로를 탐색한 후 결과를 사용자에게 재전송 한다. 반면, 단말기 기반의 경로탐색 시스템은 단말기 내에 내장된 도로 정보만을 이용하여 자체적으로 경로를 탐색하게 된다.

경로를 탐색하는 데에 있어 하드웨어의 성능은 경로 탐색의 성능과 직결된다. 고성능의 컴퓨터 일수록 한 번에 길을 탐색할 수 있는 범위가 달라지며, 탐색 속도 또한 다르기 때문이다. 센터 기반의 경로탐색 시스템의 경우 고성능의 서버 컴퓨터를 사용하여 경로를 탐색하기 때문에 실시간 도로 데이터를 사용하여 최적 경로 탐색이 가능하며, 사용자의 요구에 즉각 반응할 수 있다. 상대적으로 저사양의 PDA 등의 단말기 기반의 경로탐색 시스템은 내재된 도로 데이터만을 이용하여 경로를 탐색하므로, 일반적으로 최적 경로 탐색이 아닌, 준최적 경로 탐색 알고리즘을 다룬다. 이러한 성능의 차이에도 불구하고 사용자들은 센터기반의 시스템 보다 단말기 기반의 시스템을 더 선호하는 경향이 있다. 왜냐하면, 단말기 구입 후 추가 비용이 없기 때문이다.

일반적으로 단말기 기반의 시스템은 실시간 교통 정보를 활용하여 경로탐색을 하기 힘들다. 교통 정보를 전송 받을 수단도 마땅치 않으며, 데이터 전송을 받을 수 있다 하더라도 센터 기반의 시스템과 달리 한정된 하드웨어 자원으로 인해 원활한 경로 탐색이 힘들기 때문이다. 그러나 최근 네비게이션 단말기는 경로탐색 시스템 뿐만 아니라 DMB 방송 및 다양한 멀티미디어 서비스를 제공한다. DMB 방송을 전파하는 TPEG[1]은 일종의 채널로서 서버와 클라이언트간의 단방향 통신을 사용한다. 만약 TPEG를 이용하여 실시간 교통 정보를 전송 받을 수 있다면, 단말기 기반의 시스템에서도 탐색경로의 질을 높일 수 있다. 그러나 단말기의 하드웨어 자원이 제한적이기 때문에 대량의 데이터를 전송받아 생신할 경우 시간적인 탐색비용이 증가한다. 따라서 이를 개선할 수 있는 새로운 경로탐색 기법이 요구된다.

이 논문에서는 한정된 하드웨어 자원에서도 실시간 정보를 활용하여 경로를 탐색할 수 있는 기법에 대해 제안한다. 제안하는 방법은 지도를 일정한 크기의 그리드로 나누며, 서버로부터 각 그리드에 존재하는 도로들의 속력중에서 최고 속력 정보를 전송 받게 된다. 최고 속력 정보는 휴리스틱을 계산하는데 사용되며, 이를 통해 2단계의 노드 탐색을 거치며 최종경로를 탐색한다.

이 논문의 구성은 다음과 같다. 2장에서는 기존의 경로탐색 알고리즘에 대해 알아보고, 3장에서 제안하는 알고리즘에서 사용되는 동적휴리스틱에 대해 다룬다. 4장에서는 제안하는 알고리즘으로 실행한 실험 결과를 분석하고, 5장에서 결론 및 향후 과제에 대해 설명하겠다.

II. 관련연구

A* 알고리즘[2]은 최적일 가능성성이 높은 곳을 우선 탐색함으로써 경로의 질은 높아면서 탐색 비용은 최대한 줄이는 효율적인 탐색 알고리즈다. 이 알고리즘에서 경로를 탐색할 때 대상을 평가하는 함수로 goal, heuristic, fitness를 사용한다. Goal은 이미 정해진 값으로, 시작점에서 현재 탐색 노드까지 오는데 드는 비용이고, heuristic은 정해지진 않았지만 추정할 수 있는 값으로, 현재 노드에서 목적지까지 드는 비용이며, fitness는 goal과 heuristic을 합한 값이다. 일반적으로 A* 알고리즘으로 경로를 탐색할 때 fitness 값이 가장 최적인 곳을 우선 탐색함으로서 탐색비용을 줄이고 경로의 질을 높인다.

A* 알고리즘을 변형한 여러 알고리즘이 있는데, 그 중 하나는 overdo 휴리스틱을 사용하는 길 찾기 알고리즘[3,4]이다. A* 알고리즘으로 실제 도로와 같이 거대한 도로망에서 경로를 탐색할 경우 탐색 비용이 맵의 크기에 비례해 증가한다. overdo 휴리스틱을 사용한 경로탐색 알고리즘은 A* 알고리즘에서 노드 우선 탐색권을 설정하는 값 중 휴리스틱의 비중을 높임으로써 보다 빠르게 경로를 탐색하는 방법을 제공한다.

도로를 계층적으로 나누어 경로를 탐색하는 계층적 경로탐색 기법[5]도 있다. 이 방법은 모든 도로를 2개 내지 3개의 계층을 두어 하위 계층의 도로에서는 상위 계층으로 빠져 나갈 수 있는 길만 찾고, 상위 계층에서 목

적지 주변까지 경로를 탐색한 뒤 다시 하위 계층에서 길을 찾는 방법이다. 이 경로탐색 기법은 매우 큰 도로망이 라 할지라도 빠르게 경로를 탐색할 수 있는 방법이다.

기존의 경로탐색 연구의 대부분은 고정된 데이터를 변형하여 도로의 가중치로 이용하여 최단 경로를 빠르게 탐색하는 것에 대해 중점을 맞추었다. 그러나 실제 도로 교통망에서 최적의 경로는 고정된 데이터로만으로 최적의 경로를 찾을 수 없다. 차량이 모든 도로에서 항상 같은 속력으로 이동할 수 없으며, 때로는 거리상으로 먼 길이 이동시간이 빠를 가능성이 있다. 따라서 이동 거리 뿐만 아닌 이동시간 까지 줄이기 위해서는 고정된 데이터가 아닌 유동적인 교통정보가 필요하며, 이를 활용하는 휴리스틱이 필요하다.

가변적인 그리드 속력을 이용한 경로탐색 기법[6]은 실시간 교통정보를 활용할 경우를 대비한 경로탐색 기법이다. 이 경로탐색 기법은 경로탐색시 우선권을 가지는 노드를 탐색할 때 고정된 데이터인 거리만으로 평가하지 않고 가변적인 데이터인 각 도로의 속력정보를 고려한 탐색을 한다. 즉, 시간(T)=거리(S)/속력(V)를 이용하여 이동 시간(T)을 노드 평가시 가중치로 사용하였다. 노드를 평가하기 위한 추정 값인 heuristic은 지도를 일정 구역으로 나눈 그리드내의 모든 도로들의 평균속력을 해당구역의 속력 정보로 활용하여 경로를 탐색하였다. 이로 인해 경로를 탐색할 때 시간과 공간적인 탐색비용을 줄이고, 실시간 교통정보를 활용하는 동적 휴리스틱을 사용하였다.

그러나 이 경로탐색 기법에서 제안된 방법인 T 를 이용하여 노드를 평가할 경우, 목적지 까지 탐색되는 모든 도로의 속력을 미리 알아야 한다. 즉, 탐색되는 모든 도로들의 실시간 속력정보를 지속적으로 갱신해야 경로를 탐색할 수 있다. 현재의 단말기 기반의 경로탐색 시스템에서 이 기법을 활용할 경우, 지속적으로 모든 도로의 교통 정보를 갱신 시켜야 하는데, 이 갱신 비용은 하드웨어 자원이 한정적인 단말기 기반의 시스템에서 경로를 찾기 위한 탐색비용에 치명적인 영향을 준다.

동적 휴리스틱을 단말기 기반의 시스템에서 사용하기 위해서는 전송 비용 및 탐색 비용을 최소화 시켜야 하므로, [7]에서는 최저속력 동적 휴리스틱(DH-MIN)을 이용하여 경로탐색에 활용하였다. 최저속력 동적 휴리스틱은 지도를 일정 크기로 나눈 그리드내의 최저속력을

동적 휴리스틱으로 사용한다. 경로탐색시 이 최저속력이 낮은 구역을 탐색범위에서 제외시킴으로 속력이 낮은 구간을 탐색할 경우의 수를 줄임으로서 경로의 질을 크게 향상 시켰다.

그러나 최저속력 동적 휴리스틱을 이용한 경로탐색에서는 제외된 그리드로 인해 탐색범위가 넓어져 탐색비용이 증가하는 단점이 있다. 특히 일부 경로의 경우 탐색에서 제외된 구역이 주요 교차로이거나 강 또는 하천 등을 가로지르는 다리가 있는 지역일 경우 크게 돌아가는 경로를 탐색할 경우가 생겨 탐색비용이 매우 커지는 단점이 있다.

III. 본 론

3.1 동적 휴리스틱

경로를 탐색할 때 같은 시작점과 목적지를 가지는 경로라 할지라도 도로 교통의 상황에 따라 최적의 경로가 변할 수 있다. 실제 거리상으로는 가장 짧은 경로라 할지라도 현재 도로 상황을 고려했을 때 정체 구간이거나 신호를 많이 받는 구간이 포함되어 있을 경우 최적 경로가 최악의 경로가 될 수 있다. 따라서 거리상의 최적 경로가 이동 시간상의 최적 경로가 아님을 뜻한다. 실제 도로에서 최적 경로를 탐색하는 것은 도로의 이동거리 뿐만 아니라 현재 도로 상황을 반영하는 속력에 대한 정보 또한 고려하여 경로를 탐색하여야 한다.

일반적으로 준 최적 경로탐색에서 휴리스틱은 목적지까지 이동하는데 드는 추정 비용으로 거리에 관한 정보를 이용하여 비용을 추출한다. 거리를 이용하는 휴리스틱의 경우 동일한 거리에서 항상 같은 값으로 고정된 휴리스틱이 산출된다. 그러나 단순히 거리에 관한 정보로는 현재 도로 상황에 따른 최적 경로를 탐색할 수 없으므로, 동적 휴리스틱을 적용한 경로탐색 알고리즘이 필요하다. 이 논문에서는 고정되지 않고 도로 상황에 따라서 경로 탐색 때마다 변하는 휴리스틱을 동적 휴리스틱이라 정의한다.

3.2 최고 속력 동적 휴리스틱

제한된 하드웨어 성능을 가진 단말기에서 모든 도로에 대한 속력 정보를 갱신하여 경로를 탐색할 경우 경로를 탐색하는 시간외에 데이터를 전송받는 시간이 추가

된다. 따라서 단말기 성능을 고려하여 최소한의 휴리스틱 정보를 전송 받아 경로 탐색에 이용한다. 전송 받는 데이터는 모든 도로의 속력 정보가 아닌 맵을 일정 크기의 그리드로 나누고 그리드내에 존재하는 도로들 중 최고 속력에 관한 정보만을 전송 받는다. 여기서 그리드는 지도를 일정 크기로 나눈 영역이며, 하나의 그리드는 0개 이상의 도로가 존재한다. 제안하는 동적 휴리스틱은 그리드의 최고 속력 정보를 활용하여 계산하며, 정의1과 같이 정의한다.

방향성을 가지는 간선 e_{xy} 는 두 점 v_x, v_y 을 잇는 간선이며, v_x 가 시작 위치이다. 각각의 그리드는 v 와 e 의 집합으로 구성되며, 하나 이상의 그리드에 걸친 간선 $e_{xy} = \langle v_x, v_y \rangle$ 는 v_x 가 속하는 그리드에 포함된다. 가중치 w 는 간선의 속력이며, 그리드의 최고 속력은 해당 그리드에 속하는 간선(도로)들 중 가장 높은 속력을 가지는 도로의 속력이다.

정의1. 간선과 정점 및 그리드 최고 속력

Definition 1. Edge, Vertex and Maximum Speed of Cell

```

G = < V, E, w >
directed edge exy = < vx, vy >
w(x, y) = exy · speed
Gridn · V =
{ vx | ∀vx ∈ V, Gridn · within(vx) = true }
vx ∈ Gridn · V, GetGrid(vx) = Gridn
Gridn · E = { < vx, vy > | ∀vx ∈ Gridn · V }
Gridn · MaxSpeed =
max( { w(x, y) | ∀exy ∈ Grid · E } )

```

그리드의 속력은 수집된 교통정보를 토대로 서버에서 계산하여 클라이언트로 전송된다. 클라이언트에서는 그리드의 최고 속력 정보만을 개신하고, 이를 동적 휴리스틱에 활용하여 경로를 탐색한다. 그리드의 최고 속력이 높다고 해서 해당 그리드내의 모든 도로가 좋은 것은 아니다. 그러나 원활한 소통이 이루어지는 도로의 주변 도로나 진입도로는 일반적으로 정체구간일 가능성이 적다. 따라서 그리드의 최고 속력이 높으면, 해당 그리드 내의 도로들 또한 속력이 높은 구간일 가능성이 높다.

3.3 최고 속력 동적 휴리스틱을 이용한 경로탐색 알고리즘

준 최적해를 탐색하는 방법은 휴리스틱을 이용하여 경로를 탐색하게 된다. 즉, 경로 탐색에 힌트가 될 수 있는 지식 등을 휴리스틱으로 이용함으로써, 최적 해를 보장하지는 않지만 빠른 시간에 경로를 탐색하는 방법으로서, A*알고리즘이 가장 널리 활용되고 있다. A* 알고리즘은 현재 노드와 연결된 모든 노드에 대해서 fitness값을 계산하고, 열린 목록에 저장한다. 열린 목록에서 우선순위가 가장 높은 노드를 추출하게 되며, 추출된 노드는 현재 노드로 변경된다. 이러한 작업을 반복하여 현재 노드가 목적지가 될 때까지 수행하여 경로를 찾게 된다.

노드간의 Fitness 값의 간격

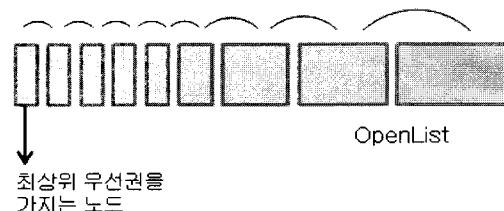


그림 1. 목록에 저장된 노드

Fig 1. Stored Node in list

DH-Max를 이용하는 경로탐색은 A*알고리즘 탐색 방법을 기반으로 하지만, 열린 목록에서 노드를 추출 시, 동적 정보를 활용하여 노드를 추출하게 된다. 일반적으로 경로 탐색 시 저장되는 노드들 중 우선권순위가 높은 노드들은 비용의 차이가 크지 않다. 그럼 1은 A* 알고리즘에서 열린 목록에 저장된 노드들을 도식화 시킨 것으로, 우선권이 높은 노드들 간의 fitness값의 차이가 적다. 거대한 도로망에서의 노드 목록일수록 비슷한 수준의 우선권을 가지는 상위 노드가 많다. 그러나 열린 목록에서 우선권이 가장 높은 노드는 해당 위치에서 추정된 비용이 좋은 것이지 최적 경로 상의 노드가 아닐 수가 있다. 따라서 이 논문에서는 다음과 같은 경로탐색 기법을 제안한다.

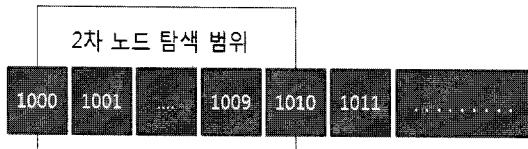


그림 2. 2차 노드 탐색 범위
Fig 2. Boundary of searching in list

제안하는 알고리즘은 2단계의 노드탐색을 통해 최상위 우선순위를 가지는 노드를 선택한다. 1차 노드 탐색은 고정된 데이터인 거리정보를 이용하여 우선순위가 정해진 노드 목록을 생성한다. 이 노드 목록은 그림 2과 같은 형태로 저장되어 있으며, 우선순위가 높은 노드들 중 N개를 2차 노드 탐색을 통해 최우선 순위를 가지는 노드를 뽑는다. 이 N은 GetNumberFavorFitness 함수를 통해 정의되며, 이는 노드 중 fitness 값이 최상위 노드 fitness 값의 1%를 더한 값보다 적은 노드들의 개수를 반환하며, 이는 그림 2와 같다. 그림 2는 2차 노드 탐색 범위를 도식화 한 것으로 그림에서 사각형은 하나의 노드를 나타내며, 사각형 내부의 수치는 F값을 나타낸다. 그림 2에서 2차 노드탐색의 범위는 F값이 1000인 노드부터 그의 1% 상위 값인 1010의 F값을 갖는 노드를 탐색한다. 2차 노드 탐색은 3절에서 제안한 최고 속력 동적 휴리스틱을 적용하여 노드를 탐색한다.

알고리즘 1. 제안하는 알고리즘 (2차 노드 탐색) Algorithm 1. Proposed Algorithm

```

SelectNode_DHMax
{
    input : OpenList
    (거리에 따른 우선탐색권이 결정된 노드 목록)
    output : Favor
    (동적 휴리스틱에 의해 우선 탐색될 노드)

    S_Node, Favor : Node;
    S_Node := OpenList.FirstNode;
    Favor := OpenList.FirstNode;
    N := GetNumberFavorFitness(OpenList);
    for i:=1 to N do
        if GetGrid(Favor).MaxSpeed <
            GetGrid(S_Node).MaxSpeed then
            Favor := S_Node;
            S_Node := S_Node.next;
}

```

즉, 2차 노드 탐색에서는 도로가 속하는 그리드의 최고 속력이 가장 높은 도로가 최우선 순위를 갖게 된다. 알고리즘 1은 이 논문에서 제안하는 알고리즘으로 최상위 우선순위를 가지는 노드를 선택하는 과정을 나타낸 것이다.

V. 구현 및 성능평가

이 논문에서 제안하는 알고리즘으로 성능실험을 하였다. 실험에서 데이터는 부산시내 도로를 기반으로 하는 데이터이며, 정점의 개수는 42,357개이고, 간선의 개수는 106,254개이다. 하나의 간선은 방향성을 가지는 하나의 도로를 뜻하며, 평균 간선의 길이는 66m이고, 최대 길이는 4,432m, 최소 길이는 4m이다. 테스트용 프로그램에서 가로는 약 46km, 세로는 약 40km이다. 그리드의 크기는 0.5km*0.5km이며 지도에서 총 7300여개의 그리드가 존재하지만, 실제 도로를 포함하는 부분은 약 40% 가량이다.

성능 실험에서 도로의 속력은 그리드 단위를 기준으로 임의로 주었으며, 하나의 그리드 내부의 최대와 최소의 도로 속력의 차이는 최대 10km/h로 생성하였다. 즉, 하나의 그리드 내부의 도로들 간의 속력은 크게 차이나지 않는다. 왜냐하면 한 도로가 정체를 일으킬 경우 주변의 진입도로들 또한 정체가 생길 가능성이 높기 때문에 도로의 속력은 대체로 주변 도로와 함께 비슷할 수 있기 때문이다.

표 1은 성능 실험에 사용되는 알고리즘간의 특징을 표로 나타낸 것이다. A* 고정 휴리스틱 Weight 60(A* W60)은 비교대상으로 모든 도로의 속력정보를 이용하여 경로를 탐색한다. 우선탐색 평가공식은 노드를 탐색 할 때 평가 기준이 되는 값인 F를 계산하는 공식이다. 제안하는 알고리즘에서는 노드평가를 2단계로 거치며, 1차 탐색에서는 거리정보를 기준으로 탐색하며, 2차 탐색은 최고속력 동적 휴리스틱인 그리드의 최고속력 정보를 이용한다. 반면 A* W60은 도로의 거리와 도로의 속력정보를 이용하여 이동시간을 계산하고 이를 평가 기준으로 사용한다.

Table 1. Comparisons of A*, DH-Max and A*W60 algorithm

표 1. 각 알고리즘의 특징 비교

비교 대상	A* 알고리즘	최고 속력우선 노드 탐색	A* 고정 휴리스틱 Weight 60
휴리스틱	맨해튼 거리	맨해튼 거리 및 DH-MAX	맨해튼거리 / 60(이동시간)
우선 탐색 평가 공식	$\sum_{i=s}^c E_i.Length + M(E_c, D)$	1차 탐색 : $\sum_{i=s}^c E_i.Length + M(E_c, D)$ 2차 탐색 : $E_c.GridMaxSpeed$	$\sum_{i=s}^c (E_i.Length / E_i.Speed) + M(E_c, D) / 60$
실시간 도로 데이터	사용하지 않음	그리드 단위로 적용	모든 도로에 적용

표 2은 실험에 쓰인 데이터 셋과 질의 셋으로 각 알고리즘은 거리에 따라 각각 100개의 임의의 S와 D를 뽑아 실험을 하였다. 알고리즘별로 700회의 경로탐색을 실시하였으며, 총 2100회의 경로탐색을 하였다.

Table 2. Set of Query and Set of Data
표 2. 질의 셋 및 데이터 셋

질의 셋	데이터 셋(실험 횟수)		
S와 D거리	A* 알고리즘	최고 속력우선 노드 탐색	A* 고정 휴리스틱 Weight 60
5km 미만	100	100	100
5~10km	100	100	100
10~15km	100	100	100
15~20km	100	100	100
20~30km	100	100	100
30~40km	100	100	100
40km 이상	100	100	100

그림 3은 데이터 셋의 알고리즘으로 경로를 탐색한 결과를 나타낸 수치다. 그림에서 도로 길이는 탐색된 경로의 실제 길이를 나타낸다. 이동 비용은 도로 길이에서 실제 각 도로의 평균속력을 이용하여 이동시간인 $t=s/v$ 를 계산한 값이다.

그래프에서 각각의 수치는 최적경로에 대한 정확도를 나타낸 것으로 DH-MAX와 A* W60은 A* 알고리즘으로 탐색했을 때의 결과에 대한 비율을 나타낸 것이다. 도로 길이는 A* 알고리즘에 비해 조금씩 높은 수준으로 나타나지만 이동 비용은 DH-MAX의 경우 평균 14% 가량 향상된 수준을 보이고 있다. 실제 도로는 A*에 비해 약간 돌아가는 경로를 찾지만 빠른 곳을 우선 탐색함으로 인해 이동 비용이 낮아진 것을 볼 수 있다. A* W60의 경우 가장 높은 정확도를 가지는 경로를 탐색한다. 실제 도로는 A* 와 DH-Max 보다 더욱 돌아가는 경로를 찾지만, 이동비용이 가장 낮다. 그러나 이는 모든 도로에 대해 실시간 정보를 모두 반영한 상태에서 경로를 탐색했기 때문에 가능한 것이며, 단말기 기반의 시스템에서는 사용하기 힘든 알고리즘이다.

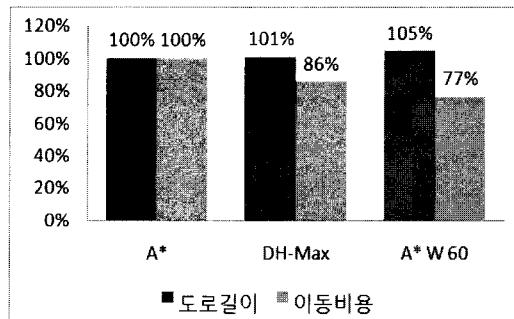


그림 3. 각 알고리즘별 경로의 질

Fig 3. Quality of path Comparison of A*, DH-Max and A*W60 algorithm

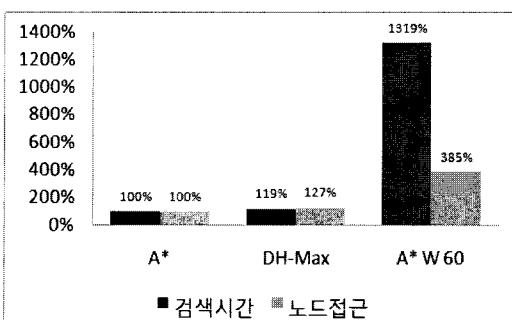


그림 4. 각 알고리즘별 탐색 비용

Fig 4. Cost Comparisons of A*, DH-Max and A*W60 algorithm

그림 4는 각 알고리즘별 탐색비용을 그래프로 나타낸 것으로 A* 알고리즘을 기준으로 그에 대한 비율을 수치로 나타낸 것 1이다. 가장 높은 정확도를 가지는 경로를 탐색하는 A* W60의 경우 검색시간과 노드접근 수가 매우 높게 나오는 것을 볼 수 있다. 반면, 제안하는 방법은 A* 알고리즘에 비해 검색 시간이 약 12%, 노드접근이 약 17%가 증가하였다.

그러나 이 실험은 단순히 경로를 탐색 하는 비용만을 비교한 것이다. 실시간 도로 정보를 전송받는 전송비용은 포함되어 있지 않기 때문에 전체적인 탐색 비용이 증가 할 수 있다. 하지만 A*Weight60의 경우 경로를 탐색하기 위해 모든 도로의 속력 정보를 갱신하여야 하지만 DH-Max의 경우 지도에 존재하는 그리드의 최고 속력 정보만을 갱신하게 된다. 실험에서 사용되는 도로의 개수는 대략 10만개이며, 그리드의 개수는 0.5km 기준으로 약 7천여개이지만, 실제 도로를 포함하고 있는 그리드는 전체 그리드의 40%인 3천여 개에 불과하다. 즉, 전송 받는 데이터의 단순한 수치로 따졌을 때 약 97%의 차이가 보인다. 따라서 제안하는 알고리즘의 경우 모든 도로를 갱신하는 알고리즘에 비해 전송비용이 낮으므로 전체적인 탐색비용은 더욱 차이가 날 것이다.

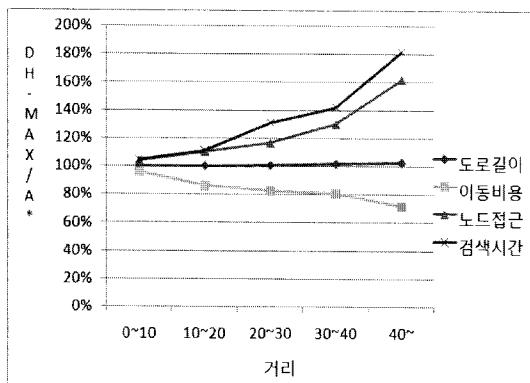


그림 5. 구간별 경로탐색 결과
Fig 5. Cost Comparisons of path on sections

그림 5는 최고속력 우선 노드 탐색 알고리즘의 경로의 질과 탐색비용을 경로의 길이에 따라 세분화 시킨 그래프로 A* 알고리즘으로 탐색할 경우에 대한 비율을 나타낸 것이다. 그림에서와 같이 짧은 거리를 탐색할 때에는 A* 알고리즘과 성능이 큰 차이를 보이지 않지만, 거

리가 멀어질수록 경로의 질은 좋아지고, 탐색비용은 증가하고 있다.

V. 결 론

이 논문에서는 실시간 교통 정보를 효율적으로 이용하여 경로탐색에 적용할 수 있는 모델을 설계 하였다. 제안하는 기법은 주로 단말기 기반의 경로탐색 시스템에서 활용할 수 있는 알고리즘으로, 경로 탐색을 위한 노드의 우선순위를 결정하기 위해 탐색을 2단계로 진행한다. 1차 노드 탐색에서는 거리에 따른 가중치를 적용한 고정 휴리스틱을 사용하며, 2차 노드 탐색에서는 최고 속력 동적 휴리스틱을 적용하여 최우선순위를 가지는 노드가 선택된다.

최고 속력 동적 휴리스틱은 TPEG과 같은 방송용 채널을 통해 데이터를 전송 받아 활용한다. 전송 받는 데이터는 일정크기의 그리드내의 최고 속력 정보이며, 이를 이용하여 제안하는 알고리즘에서 2차 노드 탐색에 사용한다. 고정 휴리스틱만을 사용하는 A* 알고리즘과의 성능평가에서 경로의 질이 약 14%가량 향상되었고 탐색비용은 약 20%가량 늘었다. 또한 출발지와 목적지까지의 거리가 멀어질수록 경로의 질은 좋아 졌으나, 탐색비용이 증가하였다.

실시간 교통정보를 사용하여, 보다 정확한 경로를 탐색하는 A* W60과 비교하면 경로의 질은 A* W60이 약 9%가량 높지만 탐색비용은 10배가량 높았다. 그러나 이 실험에서는 데이터 전송비용은 포함되지 않았기 때문에, 경로의 정확도를 10% 높이기 위해 드는 비용은 더욱 더 차이가 날 수 있다.

향후 서버에서 데이터 전송 비용을 포함한 정확한 실험이 요구되며, 노드 접근 및 검색 시간을 줄이는 기법에 대한 연구가 필요하다.

참고문헌

- [1] <http://en.wikipedia.org/wiki/TPEG>
- [2] Peter E. Hart, Nils J. Nilsson and 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] R. Jacob, M.V. Marathe, and K. Nigel, "A computational study of routing algorithms realistic transportation networks," presented at the Second Workshop on Algorithmic Engineering, NJ, 1998
- [4] Jung, S., and Pramanik, S., "An Efficient Path Computation Model for Hierarchically Structured Topographical Road Maps." IEEE Trans, Knowledge and Data Eng. Vol. 14, No. 5, 2002, pp. 1029-1046
- [5] G. R. Jagadeesh, T. Srikanthan, and K. H. Quek, "Heuristic Techniques for Accelerating Hierarchical Routing on Road Networks," IEEE Trans. Intelligent Transportation Systems, Vol. 3, No 4, pp.301-309, 2002
- [6] 이현섭, 김진덕, "고정 그리드 기반 가변 휴리스틱을 이용한 최적 경로 탐색," 한국해양정보통신학회 2005 추계 종합학술대회, Vol.9, No.2, pp.137-141, 2005
- [7] 문대진, 조대수, "최저 속력 동적 휴리스틱을 이용한 경로탐색", 한국공간정보시스템학회 2008 Vol,19, No2, 2008



조대수(Dae-Soo Cho)

1995 부산대학교 컴퓨터공학과
졸업(공학사)

1997 부산대학교 컴퓨터공학과
졸업(공학석사)

2001 부산대학교 컴퓨터공학과졸업(공학박사)

2001~2004 ETRI 텔레매틱스연구단 선임연구원

2004~현재 동서대학교 컴퓨터정보공학부 조교수

※ 관심분야 : GIS, 공간데이터베이스. LBS, 스트림 데이터처리

저자소개



김지수(Ji-soo Kim)

2002년~동서대학교
멀티미디어공학
※ 관심분야 : 공간 DB, 경로 탐색,
DSMS



이지완(Ji-wan Lee)

2002년~동서대학교
멀티미디어공학
※ 관심분야 : 공간 DB, 경로 탐색