

최고속력 동적휴리스틱을 활용한 경로탐색 알고리즘의 구현

문대진, 조대수

동서대학교

Implementation of Path-Finding Algorithm with Maximum Speed Dynamic Heuristic

Dae-Jin Moon, Dae-Soo Cho

Dongseo University

E-mail : wizardyk@nate.com, dscho@dongseo.ac.kr

요약

동적 휴리스틱이란 고정된 정보가 아닌 서버로부터 실시간 도로정보를 전송받아 동적으로 변경되는 휴리스틱이다. 이 논문에서는 최고속력 동적휴리스틱을 이용하는 경로탐색 기법을 제안하고, 성능 실험을 하였다. 최고속력 동적 휴리스틱이란 지도를 일정크기로 나눈 그리드내의 도로 중 최고속력을 동적 휴리스틱으로 활용하며, 최고속력이 높은 곳을 우선 탐색함으로써 거리에 따른 정보뿐만 아니라 실시간 정보를 활용하여 경로를 탐색한다. 성능 실험 결과 A* 알고리즘에 비해 경로의 질은 향상되었으나 탐색비용이 조금 늘었다.

ABSTRACT

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

경로탐색, 동적 휴리스틱, A*

1. 서론

일반적으로 단말기 기반의 경로탐색 시스템의 경우 실시간 도로정보를 이용하여 경로탐색을 하기 힘들다. 데이터를 전송받을 마땅한 장치가 없으며, 데이터 전송을 받을 수 있다 하더라도 센터 기반의 시스템과 달리 한정된 하드웨어 자원으로 인해 원활한 경로 탐색이 힘들기 때문이다. 따라서 단말기 기반의 시스템은 주로 내장된 데이터를 통해 경로를 탐색한다. 그러나 최근 TPEG[1]과 같은 방송채널이 생기면서 단말기 기반의 시스템에서도 실시간 정보를 활용할 수 있게 되었다.

실시간 정보를 활용할 수 있다 하더라도 모든 도로에 대해 지속적으로 도로정보를 갱신할 경

우 하드웨어 자원이 한정적인 단말기 기반의 시스템에서 오히려 탐색비용이 더 증가할 수 있다. 이전 연구에서 이를 해결하기 위해 동적 휴리스틱이 제안되었다[2,3]. 제안되었던 동적 휴리스틱은 모든 도로의 정보를 갱신하는 것이 아니라 지도를 일정크기로 나눈 그리드의 정보만을 갱신한다. 그리드는 해당 구연내의 도로들 중 최저속력 또는 최고속력만을 정보로 가진다.

이 논문에서는 한정된 하드웨어 자원에서도 실시간 정보를 활용하여 경로를 탐색할 수 있는 기법에 대해 제안한다. 제안하는 방법은 지도내의 그리드에서 최고속력 정보만을 가지며, 이를 통해 2단계의 노드 탐색을 거치며 최종경로를 탐색한다.

제안하는 방법으로 실험한 결과 A* 알고리즘

[4,5]에 비해 경로의 질은 향상하였으나, 탐색비용은 조금 증가하였다. 반면, 모든 도로의 정보를 갱신하는 경로탐색방법에 비해서는 경로의 질은 떨어지지만, 탐색비용을 크게 줄어들었다.

II. 관련 연구

동적 휴리스틱은 경로를 탐색할 때 마다 고정되지 않고 변하는 휴리스틱을 말한다. 모든 도로의 실시간 교통정보를 지속적으로 변경하여 최적경로[6,7]를 탐색하는 센터 기반의 경로탐색 시스템도 동적 휴리스틱의 범주에 속한다. 그러나 단말기 기반의 시스템의 경우 같은 방법으로 사용하기 어렵다. 모든 도로의 정보를 전송받을 경우 데이터 전송 비용과 더불어 이를 휴리스틱으로 활용하여 경로탐색을 할 경우 추가 계산 비용이 더 들기 때문이다.

동적 휴리스틱을 단말기 기반의 시스템에서 사용하기 위해서는 전송 비용 및 탐색 비용을 최소화 시켜야 한다. 이를 위해 기존연구에서는 최저속력 동적휴리스틱(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.

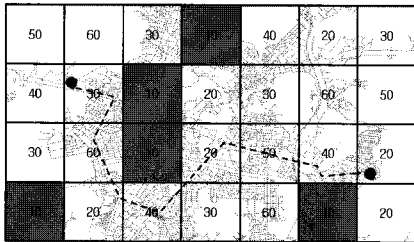


그림 1. 최저속력 동적 휴리스틱 적용 경로탐색의 예

그림 1은 최저속력 동적 휴리스틱을 적용한 경로탐색의 예로 그리드의 최저속력이 10km 이하일 경우 탐색에서 제외한다. 그림에서 조금 돌아가는 경로를 탐색하지만 도로의 속력이 높기 때문에 실제 이동비용은 A* 알고리즘으로 탐색할 경우에 비해 낮다. 그러나 돌아가는 경로로 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. 최고 속력 동적 휴리스틱

최고속력 동적 휴리스틱(DH-MAX)은 그리드 내의 최고속력을 휴리스틱으로 사용한다. 각각의 그리드는 해당 구역 내의 도로 중 최고속력을 휴리스틱 정보로 가지며, 이를 경로탐색에 활용한다.

DH-MAX를 이용하는 경로탐색은 노드탐색을 2단계로 진행한다. 1차 노드탐색은 A* 알고리즘에서 열린 목록을 생성하는 것과 동일하며, 2차 노드탐색은 1차 탐색으로 생성된 목록에서 그리드 최고속력이 가장 높은 노드를 선택한다.

```

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;
}
    
```

알고리즘 1. 제안하는 알고리즘 (2차 노드 탐색)

알고리즘 1은 제안하는 경로탐색에서 2차 노드탐색이며, GetNumberFavorFitness 함수는 2차 노드 탐색 개수를 반환한다. 2차 노드의 탐색범위는 이 함수의 반환 개수에 의해 결정된다. 따라서 이 함수의 정의 따라 알고리즘의 성능이 크게 좌우 될 수 있지만, 이 논문의 성능평가에서는 목록의 최상위 노드의 F값의 1%를 더 가지는 노드까지의 개수를 반환하며, 이는 그림 2와 같다. 이 범위 내에서 그리드 최고속력이 가장 높은 노드를 우선탐색하게 된다.

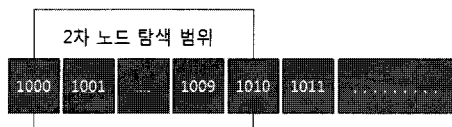


그림 2. 2차 노드 탐색 범위

그림 2는 2차 노드 탐색 범위를 도식화 한 것으로 그림에서 사각형은 하나의 노드를 나타내며, 사각형 내부의 수치는 F값을 나타낸다. 그림에서 2차 노드탐색의 범위는 F값이 1000인 노드부터 그의 1% 상위 값인 1010의 F값을 갖는 노드를 탐색한다.

이 논문에서는 2차 노드 탐색범위를 위와 같이 고정하며, 자세히 다루지 않는다.

IV. 성능실험 및 비교분석

이 논문에서 제안하는 알고리즘으로 성능실험을 하였다. 실험에서 데이터는 부산시내 도로를 기반으로 하는 데이터이며, 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 진입도어들 또한 정체가 생길 가능성이 높기 때문에 도로의 속력은 대체로 주변 도로와 함께 비슷할 수 있기 때문이다.

표 1은 성능실험에 사용된 데이터로 최고 속력 우선 노드 탐색은 이 논문에서 제안하는 방법이다. A* 고정 휴리스틱 Weight 60(A* W60)은 비교대상으로 모든 도로의 속력정보를 이용하여 경로를 탐색한다. 우선탐색 평가공식은 A* 알고리즘으로 노드를 탐색할 때 평가 기준이 되는 값인 F를 계산하는 공식이다. 제안하는 알고리즘에서는 노드평가를 2단계로 거치며, 1차 탐색에서는 거리정보를 기준으로 탐색하며, 2차

색은 최고속력 동적 휴리스틱인 그리드의 최고 속력 정보를 이용한다. 반면 A* W60은 도로의 거리와 도로의 속력정보를 이용하여 이동시간을 계산하고 이를 평가 기준으로 사용한다.

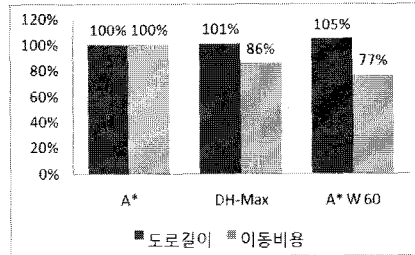


그림 3. 각 알고리즘별 경로의 질

그림 3은 데이터 셋의 알고리즘으로 경로를 탐색한 결과를 나타낸 수치다. 그림에서 도로 길이는 탐색된 경로의 실제 길이를 나타낸다. 이동비용은 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 이동비용은 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.

표 1. DH-MAX 성능실험 데이터 셋

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

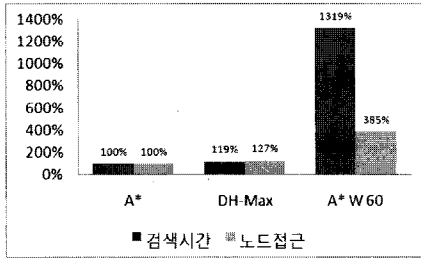
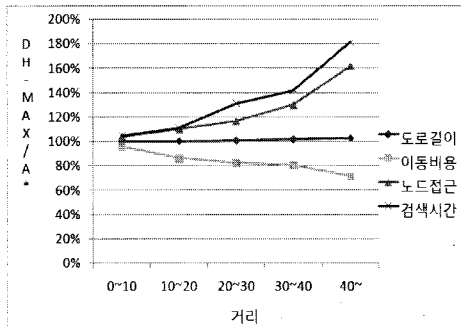


그림 4. 각 알고리즘별 탐색 비용

그림 4는 각 알고리즘별 탐색비용을 그래프로 나타낸 것으로 A* 알고리즘을 기준으로 그에 대한 비율을 수치로 나타낸 것이다. 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 검색시간이 약 12%, 노드접근이 약 17%가 증가하였다.



(단, 그래프에서 거리는 산출된 경로의 거리가 아닌 시작점과 목적지의 직선거리임)

그림 5. 구간별 경로탐색 결과

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, 거리가 멀어질수록 경로의 질은 좋아지고, 탐색비용은 증가하고 있다.

V. 결론 및 향후계획

이 논문에서는 최고속력 동적 휴리스틱을 이용한 경로탐색 기법인 최고속력 우선 탐색 대해 제안하였다. 최고속력 우선 탐색은 경로탐색시 노드를 2단계로 탐색하며, 고정 휴리스틱인 거리 정보를 통해 1차 탐색으로 정렬된 노드목록에서 우선순위가 높은 노드들을 그리드의 최고속력을

통해 2차 탐색을 한다. 고정 휴리스틱만을 사용하는 A* 알고리즘과의 성능평가에서 경로의 질이 약 14%가량 향상되었고 탐색비용은 약 20%가량 늘었다.

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

참고문헌

- [1] <http://en.wikipedia.org/wiki/TPEG>
- [2] 문대진, 조대수, "최저 속력 동적 휴리스틱을 이용한 경로탐색," 한국공간정보시스템학회 논문지, Vol.10, No.2, pp.35-48, 2008
- [3] 문대진, 조대수, "실시간 실시간 도로 정보를 이용한 최고속력 동적 휴리스틱의 설계," 한국해양정보통신학회 2008 춘계 종합학술대회, Vol.12, No.1, pp.827-830, 2008
- [4] 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
- [5] R. Jacob, M.V. Marathe, and K. Nigal, "A computational study of routing algorithms realistic transportation networks," presented at the Second Workshop on Algorithmic Engineering, NJ, 1998
- [6] E. W. Dijkstra, "A note on two problems in connection with graphs," Numerische Mathematik, Vol. 1, pp.260-271, 1959
- [7] R. Agrawal and H.V. Jagadish, "Materialization and Incremental Update of Path Information," Proc. Fifth Int'l Conf. Data Eng., pp.374-383, 1989.