

가변적 휴리스틱을 적용한 최적경로탐색

이현섭* 안준환** 김진택*

동의대학교 컴퓨터공학과* 부산대 컴퓨터공학과**

Optimal Path Search using Variable Heuristic

Hyoun-sup Lee* Jun-Hwan An** Jin-Doeg Kim*

Donggeui University* Pusan National University**

E-mail : jdk@deu.ac.kr

요 약

계속적인 교통 흐름을 감안한 최적 경로 탐색 시스템은 목적지로 가기 위한 비용을 줄이기 위해 필요하다. 그러나 낮은 계산 능력을 가진 클라이언트에서 최적 경로를 탐색하는 것은 많은 비용을 요구한다. 따라서 적은 비용과 적절한 최적 경로를 탐색이 가능한 방법이 요구된다. 이 논문에서는 연산시간을 줄이기 위해 가변적인 휴리스틱을 이용하는 경로 탐색 기법을 새롭게 제안한다. 이들 휴리스틱은 하나의 사각형 영역인 그리드 내에 존재하는 자동차들의 평균속도의 변화에 의해 결정된다.

ABSTRACT

Optimal path search systems to take continuously changed traffic flows into consideration is necessary in order to reduce the cost to get destination. However, to search optimal path in client terminals with low computing power yields high computational cost. Thus, a method with low cost and near optimal path as well is required. In this paper, we propose a path search method using variable heuristic for the sake of reducing operation time. The heuristic is determined by the change of the average speeds of cars located in grid which means a rectangle region.

키워드

최적 경로, A*알고리즘, 휴리스틱, 그리드

I. 서론

통계청(건설 교통부) 에서 제공하는 데이터를 보면 최근 10년 동안 국내의 차량 보급량은 2배 이상의 증가를 보이고 있다. 그러나 도로망의 증가는 교통량의 증가에 비해 10% 수준에 그치고 있다. 이에 따라 교통상황은 악화되고 상승적으로 체증이 나타나는 도로 또한 증가하여 많은 시간과 비용을 소비하는 결과를 가져온다. 이러한 조건 속에서 가장 적은 비용으로 목적지로 이동하기 위해 최적경로 탐색이 필수적이다.

이 논문에서는 A*알고리즘[1]을 변형하여 최적경로를 탐색하는 시스템을 제안한다. 구체적으로 이동거리의 증가로 인한 검색범위의 증가를 효과적인 휴리스틱 가중치를 적용하여 최적경로를 검색함과 동시에 연산범위를 최대한 줄여 빠른 시간 안에 검색할 수 있는 처리방법을 제안한다. 그리고 실질적인 최적경로에 대한 최상의

모델을 위해 시간 변화에 따른 경로 재검색 메커니즘을 제안한다.

이 논문의 구성은 다음과 같다. 제 2장에서는 기존의 경로 탐색 기법들의 장단점을 알아보고, 3장에서는 A*알고리즘의 구조를 살펴보고 도로 상황에 맞는 휴리스틱을 적용하여 검색효율을 높일 수 있는 시스템을 제안한다. 기존의 A* 알고리즘을 여러 가지 교통상황에 맞게 적용하고 이에 따라 변하는 가변적인 휴리스틱가중치를 제안한다. 그리고 시간 변화에 따른 정보 효율의 저하와 교통상황의 변화가 존재하므로 이를 고려한 현실적인 최적경로 검색방법을 제안한다. 마지막으로 4장에서 결론을 맺는다.

II. 관련 연구

현재 경로 탐색을 위해 사용되고 있는 기법들로 여러 가지 탐색 알고리즘들이 사용되고 있으나 도로상에서 적용하는 최적경로 검색을 위한 알고리즘은 A*알고리즘[1]과 Dijkstra알고리즘[2]을 변형시켜 사용하는 경우가 가장 많다.

* 이 논문은 MOCIE 04' 지역특화기술개발사업 (110015183)에 의하여 지원되었음

먼저 Dijkstra 알고리즘은 최단 경로 탐색을 위해 사용되는 알고리즘으로 출발지로부터 인접해있는 모든 노드들을 차례로 방문하여 목적지까지의 비용이 가장 작은 경로를 찾아내는 알고리즘이다. 모든 노드들을 검색하므로 이동거리가 늘어나면 연산범위가 커지는 단점이 있다.

이에 반해 A*알고리즘은 최단 거리 찾기에서 효과적으로 동작하는 알고리즘이다. Dijkstra 알고리즘이나 Best-First Search (BFS)[3] 알고리즘의 연산속도를 훨씬 앞지르고 있다. 그 이유는 기존의 그래프 탐색 알고리즘들과 비교하여 목표에 얼마나 근접하였는지를 평가하기위해 휴리스틱 함수를 사용한다는 것이다. 그러나 휴리스틱 정보의 가중치가 약할 경우 검색 범위의 증가를 가져오는 반면 가중치를 강하게 부여할 경우 검색 시간은 줄어들지만 탐색되어지는 결과가 정확하게 최적 경로라고 단정 지을 수 없는 문제가 발생한다.

기존의 여러 업체에서는 휴리스틱 가중치를 약하게 부여하여 검색하거나 Backward Seeking을 사용하여 약하게 부여된 가중치를 통해 연산시간의 증가를 줄이는 알고리즘을 채택하여 사용하고 있다. 양방향 탐색의 경우 연산속도의 증가는 가져오지만 A*알고리즘을 사용하여 이 탐색을 진행 할 경우 A*알고리즘의 단점인 휴리스틱 부여 여부에 따라 최적경로를 검색하지 못하는 특징 또한 같이 지니게 된다. 즉 Dijkstra알고리즘을 사용하여 양방향 탐색을 하는 것이 가장 이상적인데 이 또한 전체 이동거리가 늘어나면 검색범위의 증가로 인하여 큰 시간적인 효과를 얻지 못한다.

이 논문에서는 앞서 살펴본 알고리즘의 특징들을 적절하게 사용하여 여러 가지의 경우들이 존재하는 교통상황에 맞는 휴리스틱 가중치를 적용하여 효과적으로 최적경로를 검색하는 방법을 제안한다.

III. 최적경로 검색을 위한 가변적인 휴리스틱

3.1 A* 알고리즘의 구조 및 휴리스틱

A* 알고리즘은 탐색의 시작 노드에서부터 바깥쪽으로 균일하게 진행되지 않고 출발지점에서 목표지점까지 가장 비용이 낮은 경로를 찾는다.

이는 문제의 특성에 대한 의사결정 과정 정보인 휴리스틱(heuristic)에 따라 목표까지의 가장 좋은 경로 상에 있다고 판단되는 노드를 우선적으로 방문하여 가능해 진다. 탐색 시 다음 단계에 확장할 노드를 결정하는 휴리스틱평가함수 $f(n)$ 이 있을 때 $f(n)$ 이 가장 작은 노드를 확장하고 확장할 노드가 목표 노드이면 탐색을 종료하는 것이 기본적인 아이디어이다. $f(n)$ 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 비용이다. 시작노드에서 노드 n 까지 오는 경로들은 여러 개가 있을 수 있는데, 그 경로들 중 특정한 경로의 비용을 의미한다. $h(n)$ 은 휴리스틱(heuristic) 평가 값으로, 노드 n 에서 목표까지 가는데 드는 '추정된' 비용이다. 휴리스틱(heuristic)은 '경험에 기초한 추측'을 뜻한다. 이것이 '추정된' 비용인 이유는 목표까지의 실제 비용을 아직은 알지 못하기 때문이다. 휴리스틱을 계산하는데 여러 가지 방법이 있을 수 있지만, 일반적으로 현재 위치에서 목표까지의 일직선 거리 값을 사용한다. 마지막으로 위 식의 결과인 $f(n)$ 가 의미하는 것은 시작노드에서 노드 n 을 경유해서 목표노드 까지 가는데 드는 '추정된' 비용을 의미한다. 즉, $f(n)$ 가 낮을수록 그 경로가 최단 경로일 가능성이 크다는 의미가 된다.

도로에서 A*알고리즘을 적용하여 사용할 때 검색된 현재 위치에서 목표까지의 일직선 거리를 최대 속도로 이동한다는 휴리스틱을 사용할 경우 이동거리가 증가 하는 경우에 최초로 검색된 경로의 도착 시간 안에 존재하게 되는 경로는 큰 폭으로 증가한다. 그리고 도로 네트워크에서 직선거리를 사용하여 경로를 검색하는 것은 큰 오류를 지니고 있다. 도로의 형태는 거미줄과 같이 산재해 있으며 직선거리 사이에 산이나 이동 불가능한 지형이 있으면 도로는 우회하여 존재하는데 직선거리에 의존하여 탐색 하는 것은 정확한 데이터 값을 산출하기 어렵다. 따라서 A*알고리즘을 그대로 도로에 적용하여 사용하는 것은 우리가 따른다. 3.2절에서는 이런 현상을 최소화 하며 최적경로를 찾아 낼 수 있는 방법을 제안한다.

3.2 순수 A* 알고리즘 휴리스틱 가중치의 한계

도로에는 여러 가지의 상황이 존재한다. 그 중 도로에서 일어나는 대표적인 상황들에 대하여 적절한 휴리스틱 가중치를 적용 하는 것이 중요하다.

출발지와 목적지사이의 거리가 비교적 짧을 때의 경우 휴리스틱 가중치를 강하게 적용할 필요는 없다. 검색 되어지는 노드의 수가 적기 때문에 인접하는 모든 노드들을 검색하며 진행하여도 빠른 시간에 최적경로를 찾아 낼 수 있다.

그러나 출발지와 목적지사이의 거리가 상대적으로 먼 경우에는 앞서 제시한 방법으로 검색하기에는 존재하는 노드의 수도 상당하며 일반적인 도로에서 볼 수 없는 예외적인 상황(고속도로, 외곽도로)들이 존재 하는 것을 볼 수 있다. 즉 이런 경우 휴리스틱 가중치를 조절해야 할 필요가 있다. 휴리스틱 가중치를 일반적으로 현재 노드의 위치에서 도착지까지의 직선거리를 최대속도로 이동하는 비용으로 낮게 설정하면 검색 범위가 넓어질수록 포함되는 노드의 수가 늘어나게 되므로 연산과정은 증가한다. 그리고 휴리스틱 가중치를 강하게 부여 할 경우 연산시간의 효과적인 절감은 나타나지만 최적경로를

탐색하지 못하는 경우가 발생한다.

3.3 도로상황의 그리드 표현

도로의 교통 흐름은 고정되어 있지 않다. 이런 도로상황의 특수성을 경로검색에 반영하여 탐색하기 위해 그리드를 사용한 검색 방법을 이 논문에서 제안한다.

지도위에 존재하는 도로를 일정한 간격의 격자로 나누어 격자안의 속도 변화에 따른 휴리스틱 가중치를 부여하는 방법이다. 격자를 나눌 때 고속도로나 외곽 도로 등의 고속주행 도로는 격자에서 따로 분리하여 속도를 체크해 주어야 한다. 일반도로 내에 고속주행 도로가 포함되어 있으면 전체적인 속도 변화에 큰 영향을 줄 경우가 발생할 수 있으며 고속주행 도로의 입구와 출구는 여러 격자에 분포 되어 있기 때문에 따로 관리하는 것이 바람직하다.

(1) 탐색경로내의 그리드평균속도 저하

그림 1은 일반적인 시내 도로의 교통 흐름을 나타낸 지도이다.

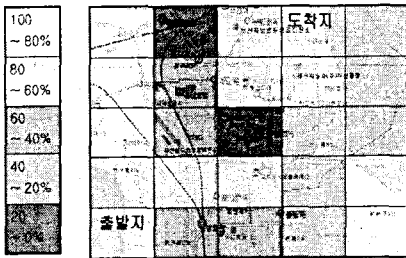


그림 1. 구간별 평균 최대속도에 대한 현재 속도의 비율

출발지와 목적지 사이에 존재하는 격자에 색이 진한 것을 볼 수 있는데 이것은 속도가 저하되고 있는 격자들을 표시하고 있다. 여기서 출발지점과 도착지점 사이에 존재하는 격자안의 평균속도가 떨어지고 있기 때문에 휴리스틱 가중치 또한 같이 낮추어 연산을 한다. A*의 연산 과정을 보면 경로 탐색을 할 때 (출발지점에서 현재 노드까지 이동한 비용 + 현재 노드에서 목적지 까지 예상되는 비용)의 값이 작은 노드들을 대상으로 탐색한다. 즉, 목적지 까지 예상되는 비용 부분이 휴리스틱 가중치라고 할 수 있는데 평균속도가 떨어지고 있는 상황에서 예상비용은 최적화된 값을 가질 필요가 없다. 여기서 말하는 최적화된 값은 현재 노드에서 목표지점까지 직선거리를 최대의 속도로 주행 했을 때의 값을 의미한다. 격자내의 전체 속도가 떨어지고 있으므로 예상비용 즉 휴리스틱 가중치를 강하게 부여함으로써 검색되는 노드들의 수를 감소시킬 수가 있다. 일반적인 교통상황을 볼 때 출근시간과 퇴근시간 및 특별히 정체가 나타나는 시간에 볼 수 있는 현상이다.

(2) 탐색경로내의 그리드평균속도 증가

그림 2에서는 앞에 경우와 반대로 격자내의 속도가 변화가 없거나 증가되고 있음을 볼 수 있다. 전체적인 속도가 증가 하고 있으므로 휴리스틱 가중치를 최적화에 맞추어 연산할 필요가 있다. 그러나 최적화에 맞추어 연산하는 경우 이동거리가 멀어지면 멀어질수록 검색대상이 되는 노드들의 수가 크게 증가하여 연산 속도를 떨어뜨릴 위험이 존재한다. 이 경우 전체적인 속도의 변화가 없거나 증가하고 있으므로 가중치는 최적화에 맞추고 검색 범위를 그림3에서와 같이 최단경로에 맞추어 잘라내는 방법을 사용한다. 일정한 속도로 이동하는 경우 최단경로가 최적 경로에 가깝기 때문에 검색범위를 최단경로에 맞추어 잘라내서 검색한다 하여도 최적경로를 충분히 찾아 낼 수 있다. 24:00 이후의 교통상황 및 출퇴근 시간을 제외한 시간에 볼 수 있는 현상이다.

단순히 노드와 노드간의 경로검색이 아닌 도로 상황에 따른 경로 검색이므로 이 두 가지 경우보다 더욱 많은 경우가 도로에는 존재한다. 하지만 이 방법을 사용하면 도로 상황에 대한 가변적인 정보들을 사용하고 있기 때문에 현재 교통 상황을 고려한 최적경로를 찾아내 줄 수 있다.



그림 2. 평균 속도가 최대 속도에 가까운 도로의 상황

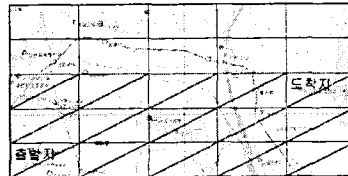


그림 3. 최단거리에 맞춘 검색 범위

3.4 그리드를 이용한 실질적인 최적경로 탐색

현재 업체에서 서비스 되고 있는 Backward Seeking(양방향 탐색)이나 Dijkstra 알고리즘을 통한 경로 탐색, A*의 휴리스틱 가중치를 부여하여 연산한 최적경로 검색, 그리고 본 논문에서 제시한 속도의 변화에 따른 휴리스틱 가중치 부여의 경로 탐색, 이 모든 경로 탐색 기법은 최적 경로 및 최단 경로를 검색하기 위해 사용되고 있는 알고리즘이다. 이와 같은 경로 탐색 기법들은 최초 경로검색 뒤에 경로를 따라 주행하다보면 가변적인 도로 상황 때문에 최적경로가 변할 수 있는 치명적인 약점을 가진다. 즉 검색 당시에는 최적경로였던 도로가 주행하는 도중 진행 경로 위에서 교통사고 및 예기치 못한 문제가

발생하였을 경우 그 경로는 최적경로가 아닌 최악의 경로가 될 수 있다. 따라서 실질적인 최적 경로 검색을 위하여 추가로 하나의 방안을 제시한다. 본 논문에서 제시한 도로를 격자로 나누어 각각의 그리드의 평균 속도를 가지고 휴리스틱 가중치를 부여하는 방법을 개선한 방안이다. 최초 최적경로 검색 후에 도로를 주행할 때 서버에서 최적경로에 포함된 그리드안의 속도를 체크하여 그림 4같이 급속하게 속도의 변화가 나타나는 부분이 발생하면 현재위치에서 목적지까지의 경로 재검색을 통하여 그림 5에서와 같이 악화된 도로 상황에서 벗어날 수 있게 하는 방법이다.

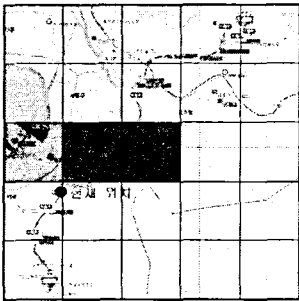


그림 4. 이동중 진행 경로가 포함된 그리드에서 급격한 속도 저하구간 발생

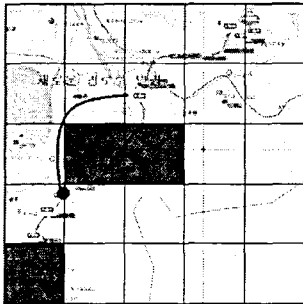


그림 5. 현 위치의 새로운 최적경로 검색

지금까지 도로를 그리드형태로 나누어 경로를 검색하는 방법에 대하여 서술하였다. 두 가지의 방법 모두 이동거리가 늘어남에 따라 많은 그리드를 포함하게 된다. 이 때 그리드의 속도변화가 동일하게 하락하거나 증가할 가능성은 적다. 따라서 경로 검색 대상에서 포함되는 그리드의 전체 평균속도의 변화에 따라 휴리스틱 가중치의 변화 또한 이루어 지야 하며 도로가 복잡한 도심지역에서는 최대한 그리드의 크기를 작게 하여 민감한 속도변화에 대응해야 하고 시외의 도로에서는 속도변화의 폭이 작고 검색되는 노드의 수 또한 적으므로 그리드의 폭을 크게 하여 검색 하는 것이 효과적이다.

일반적인 도로 데이터를 바탕으로 그리드의 크기변화를 임의로 주어 최적의 크기를 찾아내

는 실험을 진행하고 그리드내의 평균속도를 변화시켜 최적경로를 찾아내는지 연산 시간은 얼마나 감소시킬 수 있는지 시뮬레이션을 통해 실험 하고자한다. 그리고 도로위의 실제적인 데이터를 바탕으로 하여 같은 실험을 반복한 후 시뮬레이션 결과와 일치하는지 검증 하고자한다.

IV. 결론

도로에는 수많은 노드들이 존재한다. 더불어 수많은 교통상황들이 산재해 있다. 이러한 정보들을 바탕으로 최소의 비용으로 최적경로를 찾아내기 위해 기존의 탐색 기법을 사용하는 것은 정확한 최적경로 탐색을 위해 비용이 많이 소모된다. 실제로 Dijkstra 알고리즘은 모든 경우의 경로를 검색하기 때문에 최적경로를 찾아 낼 수 있다. 이동 거리가 증가할수록 더욱더 많은 노드간의 경로 검색 때문에 Dijkstra 경로 탐색은 높은 연산시간을 동반한다. A*알고리즘을 사용한 경우 연산시간의 단축은 가져올 수 있으나 교통정보가 개입되지 않은 휴리스틱 가중치를 사용할 경우 최적경로 연산이 어렵다. 따라서 이 논문에서는 도로를 그리드형태로 나누어 각각의 평균 속도 변화에 따른 가변적인 휴리스틱 가중치를 사용하여 경로를 탐색하는 알고리즘에 대하여 서술하였다.

구체적으로 그리드의 속도 저하에 따른 강한 가중치를 부여하여 검색 속도를 증가시키며 그리드의 속도 증가 및 유지에 따른 최적화된 가중치를 부여하고 최단경로에 준하여 검색 범위를 축소시켜 연산속도의 증가를 가져오는 방안을 제안하였다. 교통상황과 도로의 흐름을 파악하여 연산과정에 반영하였기 때문에 최적경로를 검색 해 낼 수 있다. 현재 변형된 알고리즘을 질적인 교통상황을 대입하여(부산시내 도로 및 교통 상황)구체적인 데이터를 계산하기위한 도로데이터 포맷을 구축중이며, 최적화된 그리드의 범위를 산출하기 위한 연구를 계획 중이다.

참고문헌

- [1]en.wikipedia.org/wiki/A%2A_search_algorithm
- [2]en.wikipedia.org/wiki/Dijkstra%27s_algorithm
- [3]en.wikipedia.org/wiki/Breadth-first_search
- [4]Stefano Pallottino, Maria Grazia Scutella, "Shortest Path Algorithms in Transportation Models : Classical and Innovative Aspects", TR, Univ. of Pisa, 1998
- [5]이현섭, 안준환, 김진덕. "교통정보 기반 최적 알고리즘", 해양정보통신학회 추계학술대회, 2004
- [6]theory.stanford.edu/~amitp/GameProgramming
- [7]Woo Young Kwon; Sanghoon Lee; Il Hong Suh, "A reinforcement learning approach involving a shortest path finding algorithm" Intelligent Robots and Systems, Proceedings of IEEE/RSJ International Conference, 2003