

미지의 공간상의 실시간 최단 경로 탐색 알고리즘에 대한 분석

최은미*, 김인철**

*경기대학교 전자계산학과

e-mail: sogmi, kic@kyonggi.ac.kr

Analysis of Real-time, Shortest-Path Finding Algorithms for Unknown Environments

Eunmi Choi*, Incheol Kim**

*Department of Computer Science, Kyonggi University

요 약

본 논문은 미지의 공간 그래프에 대한 실시간 최단 경로 탐색 알고리즘인 RTA* 알고리즘과 PHA* 알고리즘을 소개하고, 예제 그래프를 통해 이들의 특성을 설명한다. 또 3차원 온라인 게임 환경인 언리얼 토너먼트(Unreal Tournament) 게임과 Gamebots 시스템을 이용하여 두 알고리즘 간의 탐색 최적성과 효율성을 비교하였다. 이 실험을 통하여 PHA* 알고리즘은 언제나 최선의 경로를 탐색하지만 이동경로가 많고, 시간이 많이 걸릴 수 있으며, RTA*는 최적의 경로 혹은 차선 경로를 빠른 시간 내에 탐색함을 확인 하였다. 따라서 이러한 특성을 바탕으로 주어진 탐색 환경의 조건에 따라 두 알고리즘을 선택적으로 적용함으로써 보다 높은 효과를 얻을 수 것으로 기대한다.

1. 서 론

탐색(search)은 다양한 분야에서 사용 되고 있으며 인공지능 분야에서 많은 탐색 방법이 연구 되어 왔다. 탐색하고자 하는 상태공간은 일반적으로 그래프로 표현 한다. 그래프는 맵, 게임, 통신 네트워크 등 다양한 환경을 표현 할 수 있기 때문이다.

방문을 통하지 않고도 그래프 전체의 연결 정보를 미리 알고 있는 공간에서의 최단 경로 탐색을 위해서는 전통적으로 잘 알려져 있는 A* 알고리즘 등을 이용 할 수 있다. 하지만 최근에 급속히 발전하고 있는 로봇 혹은 온라인 게임 등에서는 미지의 공간에 놓인 에이전트가 자신의 센서만을 통해서 국지적인 공간 정보를 획득할 수 있다. 이러한 환경에서는 기존의 최단 경로 탐색 알고리즘에 따른 사전 경로 계획은 불가능하며, 따라서 에이전트의 실제적인 방문을 통해서만 그래프의 연결 정보를 얻을 수 있는 새로운 실시간 최단 경로 탐색 알고리즘을 요구 하게 되었다. 대표적인 실시간 최단 경로 탐색 알고리즘들로는 RTA*, PHA*, D* 등이 있다.

RTA*(Real-Time A*) 알고리즘은 실시간 경로 탐색 알고리즘으로 현재 노드에서 이웃한 노드 중에서 휴리스틱 값이 적은 노드를 선택하고 이동하는 행위를 반복하여 시작노드에서 목표노드까지의 경로를 탐색한다. 현재 노드의 이웃한 노드만을 이용한 국지(local)적인 견지에서 다음 노드를 선택하게 되어 최선의 경로를 보장 하지 못하는 경우도 있지만 빠른 시간 내에 최선 혹은 차선의 경로를 탐색하기 위한 알고리즘 이다.

PHA* (Performing A*)알고리즘은 A* 알고리즘 처럼 그래프 전체(global)의 OPEN-LIST 중에서 $f(x)$ 값이 우수한 노드를 선택하는 방식을 취한다. 그래프 전체 노드를 대상으로 다음 방문할 노드를 결정하기 때문에 다음 방문할 노드가 현재 위치에서 멀리 떨어지는 경우도 발생하여 노드를 중복하여 노드를 방문하기도 한다. 하지만 A* 알고리즘이 보장 하듯 언제나 최적의 경로 탐색을 보장한다.

D* (Dynamic A*) 알고리즘은 환경 정보를 부분적으로 알고 있거나, 완벽하게 알고 있는 상황에 적

용 가능한 알고리즘이다. 백포인터(backpointer)를 이용하여 경로를 계획하고, 이동을 통하여 새로운 정보를 습득하여 경로를 재 계획 한다. 이러한 과정을 반복하여 최적의 경로를 탐색 하게 때문에 탐색 도중 장애물을 만나더라도 최적의 경로를 보장해 주는 효율적인 알고리즘이다. 백포인터를 사용하여 탐색을 전개하기 때문에 패스의 양방향성이 보장 되지 않는 유한 그래프를 사용하는 3차원 게임 환경과 실제 물리환경에서는 적합하지 못하다.

본 논문은 게임 환경과 같이 실시간이며 미지의 상태공간에서 제한된 정보를 이용하여 탐색하는 알고리즘을 설명하고 차이점을 비교 분석 한다. 2장에서는 실시간 최단 경로 탐색 알고리즘을 설명하고 탐색 방법을 소개한다. 3장은 게임 환경을 이용하여 탐색 알고리즘을 구현하고 실험을 통해 비교 분석 한다.

2. 실시간 최단 경로 탐색 알고리즘

2.1 RTA* 알고리즘

RTA* 알고리즘은 실시간 경로 탐색 알고리즘으로 현재 노드에서 이웃한 노드 중에서 휴리스틱 값이 적은 노드를 선택하고 이동하는 행위를 반복하여 시작노드에서 목표노드까지의 경로를 탐색 한다.

목표에 도착할 때 까지 다음의 3가지 단계를 반복하여 경로를 탐색한다.

$$1. f(x') = h(x') + k(x, x')$$

현재 위치 x 의 이웃한 노드 x' 각각의 x' 와 목표까지의 거리를 휴리스틱으로 계산한 $h(x')$ 와 각각의 x' 와 현재노드 x 사이의 거리 $k(x, x')$ 를 합하여 각각 x' 의 휴리스틱 $f(x')$ 를 계산한다.

2. 이웃한 노드 중에서 $f(x')$ 가 가장 작은 노드를 선택하여 이동한다.

3. 이웃한 노드 중에서 두 번째로 $f(x')$ 가 작은 값을 선택하여 $h(x)$ 을 갱신한다.

현재 노드의 이웃한 노드만을 이용한 국지(local)적인 견지에서 다음 노드를 선택하기 때문에 최선의 경로를 보장 할 수는 없지만 빠른 시간 내에 최적 혹은 차선의 경로를 탐색 한다.

기존의 RTA* 알고리즘을 사용하게 되면 복잡한 그래프에서 이동경로에 사이클(cycle)이 발생하게 되어 결론에 도달 하지 못하는 경우가 발생한다. 이를 극복하기 위해 3단계의 $f(x')$ 를 $h(x)$ 에 갱신하여 또다시 x 를 방문하였을 때 이미 선택되었던 x' 로 이용하지 않고 두 번째로 $f(x)$ 값이 작은 x' 노드를 선택 하게 하

는 부분이다. 이때 이동한 경로를 히스토리(history)로 저장하여 이미 이동한 경로를 다시 선택하여 이동하지 못하도록 해야 한다.

2.2 PHA* 알고리즘

PHA* 알고리즘은 A* 알고리즘처럼 그래프 전체(global) 노드 중에서 $f(x)$ 값이 우수한 노드를 선택하는 방식을 취한다. 그래프 전체 노드를 상대로 다음 방문할 노드를 결정하고 현재 위치에서 결정된 위치까지 이동하여 연결정보를 획득하고 또다시 그래프 전체 노드를 상대로 다음 방문할 노드를 선택하는 과정을 반복한다. 이러한 방법을 사용하기 때문에 중복하여 노드를 방문하는 경우가 발생하기도 하지만 A* 알고리즘이 보장하듯 언제나 최적의 경로를 탐색 할 수 있다.

```
upper-level(open-list){
  while(open list is not empty){
    target = best node from open-list
    if unexplored(target){
      explore(target) by lower level
    }
  }
}
```

상위 레벨(upper level)은 확장을 위해 다음 노드를 선택하는 부분으로 일반적인 A* 알고리즘과 같이 노드 주변의 노드의 연결을 확인하지 못한 OPEN-LIST 각각의 $f(x)$ 값을 계산하여 다음 확장할 노드를 선택하는 방법과 현재 위치와 가까운 노드를 선택하여 이동경로를 줄이기 위하여 미리 정의된 윈도우 사이즈만큼 현재 노드와 거리가 가까운 노드를 선별하고 선별된 노드 중에서 $f(x)$ 값을 계산하여 확장 할 노드를 선택하는 winA* 방법이 있다.

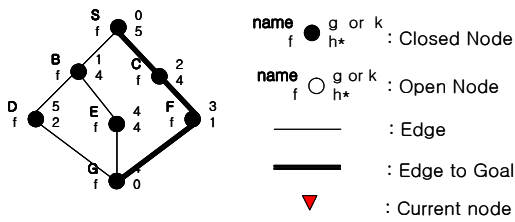
하위 레벨(lower level)은 현재 위치에서 upper level에서 선택한 노드로 이동하는 탐색(navigation) 부분이다. 하위 레벨은 간단 탐색 알고리즘(simple navigation algorithm), 깊이 우선 탐색을 기반으로 하는 탐색(DFS-based navigation algorithm), 개선된 깊이 우선 탐색(Improved A*DFS)으로 크게 나눌 수 있다.

우선 간단 탐색 알고리즘은 물리환경을 무시하고 공중으로 이동하거나, 최적 경로를 알고 있는 경우이다. 깊이 우선 탐색을 기반으로 하는 탐색은 위치를 이용한 DFS, 방향을 이용한 DFS, 일반적인 A* 알고리즘으로 이루어 졌다. 마지막으로 개선된

깊이 우선 탐색 방법은 이동 중에도 새로운 정보를 획득하여 추후 탐색에 도움이 되고자 하는 알고리즘이다.

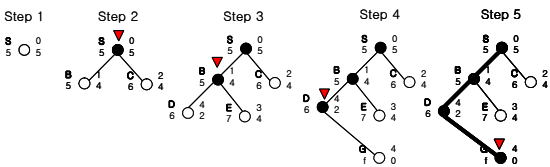
기존 PHA* 알고리즘만을 사용하게 되면 복잡한 환경의 그래프에서 이동경로에 사이클이 발생하게 되어 결론에 도달 하지 못하는 경우가 발생한다. 이를 보완하기 위하여 이미 방문한 경로를 히스토리로 저장하여 다시 사용할 수 없도록 하면 사이클의 발생을 막을 수 있다.

2.3 RTA* 알고리즘과 PHA* 알고리즘 비교

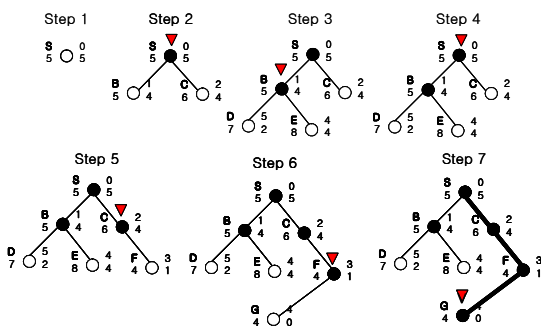


[그림 1] 예제 그래프

그림1은 시작노드 S, 목표노드 G를 가진 전체 그래프이다. 노드의 첨자로 표현된 $g(x)$ 는 시작 노드에서 x 까지의 실제 이동거리, $k(x)$ 는 현재 노드(삼각형)와 x 노드 사이의 거리, $h^*(x)$ 는 x 와 목적노드의 거리를 휴리스틱으로 계산한 값을 나타낸다.



[그림 2] RTA* 알고리즘을 이용한 탐색 과정



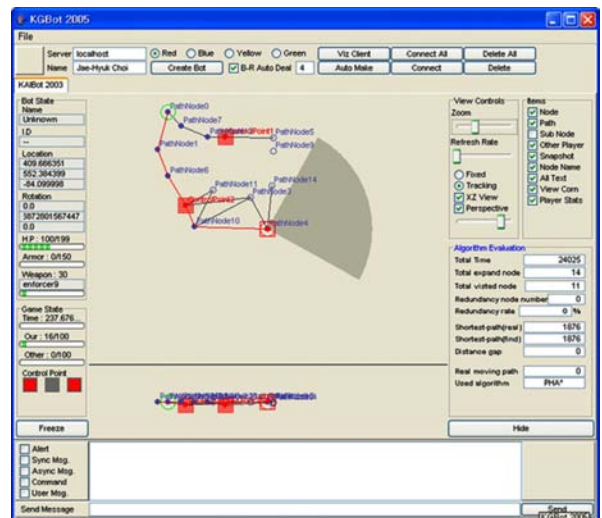
[그림 3] PHA* 알고리즘을 이용한 탐색 과정

시작 노드 S에서 G를 목표로 하여 두 알고리즘으로 탐색하면 RTA* 알고리즘은 S-B-D-G 경로로 이동하여 이동경로비용이 7이다. 최적의 경로는 아니지만 PHA* 알고리즘에 비하여 적은 이동 경로

비용을 가진다. PHA* 알고리즘은 S-B-S-C-F-G 경로로 이동하여 이동 경로 비용이 6이고, S-C-F-G 라는 최적의 경로를 탐색하는데 성공 하였다. RTA* 알고리즘은 최적은 아니지만 빠른 시간 내에 결론에 도달 할 수 있기 때문에 최적의 경로는 아니지만 차선의 경로라도 빠른 시간에 탐색하고자 하는 분야에 적합하고, PHA* 알고리즘은 항상 최적의 경로를 탐색하지만 중복 방문으로 이동 경로 비용이 늘어나고 시간이 걸리더라도 최선의 경로를 탐색해야 하는 분야에 적합하다.

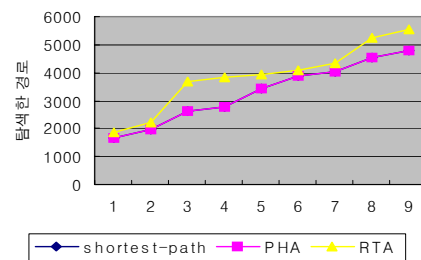
3. 구현 및 실험

경로 탐색을 위한 실험 환경으로는 일인칭 슈팅 게임(FPS)인 언리얼 토너먼트에 기초한 Gamebots 시스템[5]을 이용하였으며, 실험용 에이전트로는 경기대학교 인공지능 연구실에서 개발한 지능형 게임 에이전트인 KGBot[6]을 이용하였다.

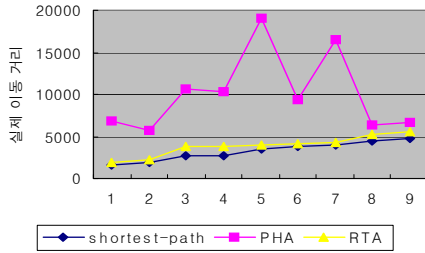


[그림 4] 게임 에이전트 KGBot를 이용한 최단 경로 탐색

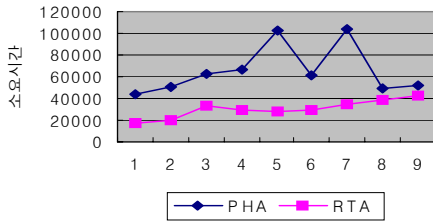
실험에서는 노드의 수가 다양한 9개의 맵을 사용하였고 반복적으로 실험 하였다.



[그림 5] 탐색한 최적 경로



[그림 6] 탐색에 소요된 이동 거리



[그림 7] 탐색에 소요된 총 소요시간

그림 5는 두 노드 사이의 경로를 PHA* 알고리즘과 RTA* 알고리즘을 이용하여 탐색한 결과이다. PHA* 알고리즘은 언제나 최적의 경로를 탐색하는데 비하여 RTA* 알고리즘은 최적의 탐색이 아닌 경우 있으며 맵에 따라서는 큰 차이를 보이기도 했다. 그림 6은 탐색완료까지의 소요시간이다. RTA* 알고리즘이 PHA* 알고리즘보다 빠르게 탐색함을 보여준다. 그림 7은 탐색을 위해 실제로 이동한 거리이다. RTA* 알고리즘은 언제나 최적의 경로를 탐색하지만 많은 이동으로 인하여 실제 이동거리가 많으며 RTA* 알고리즘은 실제 이동거리가 탐색한 경로와 일치하는 경우가 많았다.

4. 결 론

본 논문은 미지의 상태공간을 탐색하여 연결 경로를 알 수 없는 두 노드의 최적의 연결 경로를 찾는 RTA* 알고리즘과 PHA* 알고리즘을 설명하고, 임의의 그래프에 대해 사이클이 발생하지 않도록 부분적인 개선책을 제시하였다. 또한 본 논문에서는 3차원 온라인 게임 환경을 이용하여 두 알고리즘의 최적성과 효율성을 비교하였다. 이 실험을 통하여 PHA* 알고리즘은 언제나 최선의 경로를 탐색하지만 이동경로가 많고, 시간이 많이 걸릴 수 있으며, RTA*는 최적의 경로 혹은 차선 경로를 빠른 시간 내에 탐색함을 확인 하였다. 향후 이러한 알고리즘들을 확장하여 다수의 에이전트들이 함께 협조적으

로 실시간 최단 경로 탐색을 전개하는 다중 에이전트 최단 경로 탐색 알고리즘을 개발하고자 한다.

참고문헌

- [1] 이만재, "게임에서의 인공지능 기술", 정보처리학회지, 제9권 제3호, p69-76, 2002
- [2] Makoto Yokoo and Yasuhiko Kitamura. "Multiagent Real-time A* with Selection: Introducing Competition in Cooperative Search.", Proc. International Conference on Multi-Agent Systems (ICMAS-96), p409-416, 1996.
- [3] Anthony Stentz, "Optimal and efficient path planning for partially-known environments.", In Proceedings of the IEEE International Conference on Robotics and Automation, p3310-3317, 1994
- [4] A. Felner, R. Stern, A. Ben-Yair, S. Kraus and N. Netanyahu, "PHA*: Finding the Shortest Path with A* in An Unknown Physical Environment", Journal of Artificial Intelligence Research (JAIR) 21:631-670, 2004.
- [5] R. Adobbati et al, "Gamebots : A 3D virtual world test-bed for multi-agent research", Proceedings of Agents-01, 2001
- [6] 김인철, "3차원 가상환경에서 동작하는 지능형 에이전트의 구조와 경로 찾기 행위", 정보처리학회논문지 B, 제10-B권 제1호, 2003