

## A\*를 이용하는 전술적 경로찾기에서 휴리스틱 성능비교

김경혜, 조수진, 설정아, 유건아  
 덕성여자대학교 컴퓨터학과  
 e-mail:kh140@duksung.ac.kr

## Comparison of Heuristics in Tactical path-finding Using A\*

Kyung-Hye Kim, Sujin Cho, Jeong-A Sul, Kyeonah Yu  
 Dept. of Computer Science, Duksung Women's University

## 요 약

전술적 경로찾기에서는 거리나 시간 요소 외에 여러 가지 전술적 요소를 포함한 비용 함수를 사용하여 경로를 탐색한다. 경로찾기에서 가장 많이 이용되는 A\* 알고리즘의 경우, 현재 노드에서 목표까지의 추정값을 의미하는 휴리스틱 함수를 이용하는데 대표적인 허용가능 휴리스틱(admissible heuristic)인 유클리디안 거리(Euclidean distance)를 전술적 경로찾기에서 이용하는 경우, 탐색 성능이 저하되는 단점이 있다. 이는 거리이외에 전술적 요소까지 더해진 실제 비용에 비해 직선 거리만을 고려한 휴리스틱 값이 현저하게 작는데 기인한다. 그러므로 본 논문에서는 A\*를 이용하는 경로찾기에서 탐색의 성능을 향상시킬 수 있는 두 가지 휴리스틱을 제안하고 이들의 허용성을 분석하고 방문 노드수 비교를 통해 탐색 성능을 비교한다.

## 1. 서론

과거의 경로찾기는 장애물을 피하면서 자연스럽게 짧은 경로를 찾는 데 중점을 두어 왔다. 그러나 최근 고급 게임이나 시뮬레이션 환경에서는 적군의 시야로부터 피해야 한다거나 지형적 특징을 이용한다거나 하는 전술적 요소를 고려한 경로찾기에 대한 요구가 늘어나고 있다. 이와 같이 빠른 경로와 그 외의 여러 가지 요소들의 균형을 고려한 경로찾기를 전술적 경로찾기라고 한다 [1]. 전술적 경로찾기는 NPC(non-player character)의 움직임을 더 자연스럽게 해줄 뿐 아니라 게임 사용자에게 좀더 예측 불가능한 지능적 상대를 제공해 준다는 장점이 있다.

경로찾기를 위한 탐색 알고리즘으로 가장 대표적인 것이 인공지능 분야의 A\* 탐색 알고리즘이다. A\* 탐색에서 다음 노드를 선택할 때 이용되는 평가함수는 현재 노드  $n$  까지 소요된 비용함수  $g(n)$ 과 앞으로 남은 거리의 추정값  $h(n)$ 으로 구성되는데 과소추정된(underestimated)  $h(n)$ 을 사용하면 최적의 해를 구할 수 있다는 것이 최대 장점으로 알려져 있다. 경로찾기에서 A\*의 과소추정 휴리스틱으로는 유클리디안 거리나 맨하탄 거리 등이 있고 이들은 최단 거리 해를 찾는 것을 보장한다. 여러 가지 전술적 요소를 고려한 경로찾기에서는 A\* 알고리즘의 비용 함수에 거리 이외의 요소들을 포함하여 탐색에 이용하는 것이 일반적이다. [2]에서는 체스와 같은 보드 게임에서 정의되었던 위치 평가 함수(position evaluation function)를 기존의 비용 함수에 추가하여 탐색에 이용하였으며 [3]에서는 두 노드 사이 링크를 이동하는데 소요되는 비용을 거리와 전술 요소들의 가중치 합으로 나타내어 탐색에 이용하였

다. 이와 같은 비용 함수의 변화로 이동하는데 소요되는 실제 비용은 최단거리 경로찾기에 비해 증가되고 상대적으로 휴리스틱 값은 과다하게 과소추정되어 A\* 탐색의 성능이 나빠지는 것이 문제점으로 지적되고 있다. 즉 경로를 찾기 위해 방문하는 노드의 수가 증가된다는 것이다. 본 논문에서는 기존의 휴리스틱보다 정보가 많은 두 가지 휴리스틱(more informed heuristic)을 고안함으로써 A\*의 탐색 성능을 향상시켜서 빠른 전술적 경로찾기가 가능하도록 한다. 시뮬레이션을 통해 제안된 두 가지 휴리스틱과 기존의 휴리스틱과의 탐색 성능 차이를 비교하고 각 휴리스틱의 장단점을 분석한다.

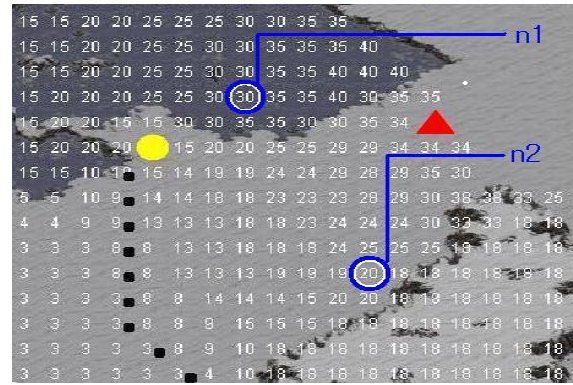
## 2. 전술적 경로찾기를 위한 비용함수

전술적 경로찾기와 기존의 경로찾기와의 차이는 비용 함수에 있다. 기존의 경로찾기가 이동 거리나 시간에만 의존했다면 전술적 경로찾기에서는 지형이나 적으로부터의 가시성 등의 전술적 요소에 대한 비용을 고려해야 한다. 이들을 비용 함수에 반영하기 위하여 위치 평가 함수를 이용한다. 위치 평가 함수는 컴퓨터 체스 등의 게임에서 보드 위치를 평가하기 위해 사용되는 인공지능 기법이다. 체스 보드 위치를 평가하기 위해 말의 중요도, 말의 이동성, 왕의 안전도 등의 여러 가지 특성들의 가중치 합으로 단일 값을 계산하여 리턴한다. 식 (1)은  $m$ 개의 특성( $f_i$ )이 있을 때 노드  $n$ 에서의 위치 평가 함수  $p(n)$ 을 표현한다.

$$p(n) = \sum_{i=1}^m f_i(n) \quad (1)$$

전략적 게임에서도 이와 유사하게 위치 평가 함수를 이

용할 수 있다. 게임에서 전술적 행동을 결정할 때 고려해야 하는 전술 요소들을 특성으로 하고 게임 상태공간 표현에 따라 설정된 노드마다 값들이 할당하여 그들의 합을 위치의 값으로 하면 된다. 본 논문에서 그리드-기반 표현 방식에서 노드는 각 그리드가 노드가 되므로 각 그리드 별로 값들이 할당된다. 전략적 게임에서 적을 피하면서 이동하기 편한 지형과 위치로 가는 경로를 선택하기 위해 고려되는 전술 요소 3가지에 대해 위치 평가 함수가 결정되는 과정을 그림 1에 나타낸다. 지형 정보는 일반 땅에 비해 좌측 상단의 얼음 부분과 우측 하단의 숲은 각각 15, 18점의 높은 값이 할당되었다. 벽 껴안기(wall-hugging)는 벽에 너무 가까운 경우에 이동 경로로서 어색하므로 이를 고려하여 3단계로 점수를 할당하였으며 세 번째 그림은 적(ENEMY)으로부터 가까울수록 높은 값을 할당하였다.



↑ 3가지 정보의 합으로 표시된 위치평가 값

(그림 1) 위치 평가 함수

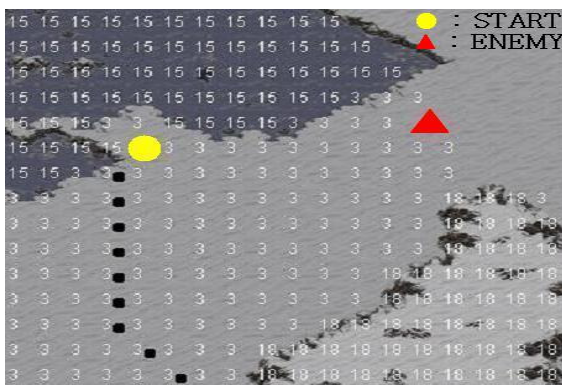
위치 평가 함수가 그림 1과 같이 결정되면 경로찾기 문제에서 A\* 탐색을 위한 비용 함수는 식 (2)와 같이 이동 거리와 이동 선상에 있는 노드들의 위치 평가 함수값의 합으로 계산되게 된다. 여기서  $g(n)$ 은 노드  $n$ 까지의 비용 함수,  $D$ 는 노드  $n$ 까지의 이동 거리를 나타낸다. 본 논문에서는 그리드-기반 배경을 사용하므로 이동 거리는 경로 위에 있는 그리드 수와 같다.

$$g(n) = D + \sum_{i=0}^n p(n) \quad (2)$$

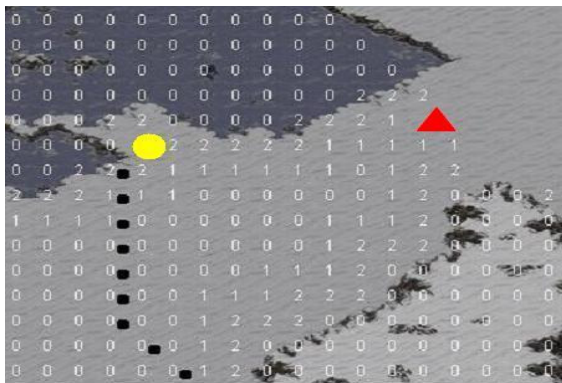
그림 1의 예에서는 시작점으로부터  $n1$ 와  $n2$  노드로 이동한 비용 함수는 각각  $g(n1)=38+75=113$ ,  $g(n2)=90+125=215$ 가 된다. 각 전술 요소에 대한 점수 할당의 적절성에 대해서는 본 논문에서는 다루지 않는다.

### 3. 탐색 성능 향상을 위한 휴리스틱 함수의 개선

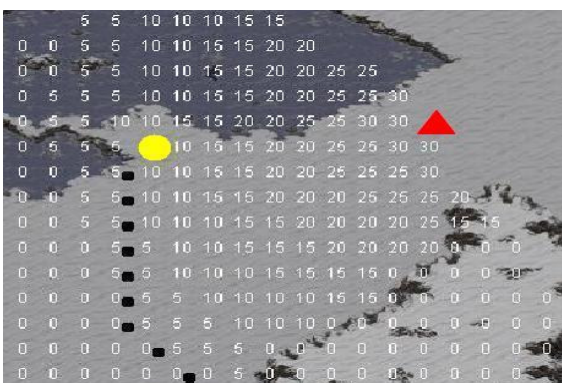
A\* 알고리즘의 휴리스틱  $h(n)$ 은 노드  $n$ 으로부터 목표까지의 추정 비용으로 실제 드는 비용보다 적게 추정되어야 하며 이 조건을 만족할 때 허용 가능 휴리스틱(admissible heuristic)이라고 한다. A\*는 허용 가능 휴리스틱을 이용하여 탐색을 이끌어 최적 해를 구하도록 하는데 이것이 A\* 알고리즘의 최대 장점이다. 그러나 과소 추정 휴리스틱은 허용성(admissibility)을 보장하는 대신에 A\* 탐색의 비효율성을 야기하기도 하는데 이는 휴리스틱이 적게 추정될수록 정보를 적게 활용하는 것이 되기 때문이다. 즉,  $0 \leq h_1(n) \leq h_2(n) \leq h^*(n)$ 의 경우( $h^*(n)$ 이 현재 노드에서 목표 노드까지의 실제 비용),  $h_1(n)$ 과  $h_2(n)$ 는 모두 허용 가능 휴리스틱이지만  $h_2(n)$ 가  $h_1(n)$ 보다 정보를 많이 가진 휴리스틱이며 따라서  $h_1(n)$ 에 의한 방문 노드수가  $h_2(n)$ 에 의한 것보다 많음이 증명되어 있다. '0'은 최악의 허용 가능 휴리스틱이며 전혀 정보를 이용하지 않는 너비-우선 탐색 알고리즘에 해당하여 이를 이용해서 해결할 수 있는 문제는 극히 제한되어 있음이 알려져 있다 [4].



↑ 지형정보



↑ 벽 껴안기 정보



↑ 위협 요소로부터의 가시 정보

전술적 경로찾기에서는 기존의 경로찾기에서 이동 거리를 고려하던 것을 다른 여러 요소들에 대한 이동 비용도 포함하여 비용함수의 증가를 초래한다. 따라서 경로찾기에서 일반적으로 사용되던 직선 거리 휴리스틱은 실제로는 비용보다 상대적으로 더 작게 추정되게 된다. 이와 같은 이유로 전술적 경로찾기를 구현하거나 다른 게임과 문헌들에서 전술적 경로찾기의 문제점으로 탐색 효율의 저하를 지적하고 있다. 이에 대한 해결책으로서 NPC의 동작 영역을 제한하거나 계층적 탐색을 통해 탐색 상태 공간을 줄이려는 시도를 하였고 과소 추정 휴리스틱의 본질을 이용하여 탐색 효율을 개선하고자 하지 않았다. 본 논문에서는 직선 거리 휴리스틱,  $h(n) = dist(n, g)$ 보다 큰 휴리스틱,  $h_1(n)$ 과  $h_2(n)$ , 두 가지를 소개하고 각 휴리스틱의 최적성과 탐색 효율에 대해 논한다.  $f_i(n)$ 은 노드  $n$ 의  $i$ 번째 특성을 나타내며  $m$ 개의 특성이 있다고 가정한다.  $N$ 은 전체 노드 집합을 나타낸다.

$$h_1(n) = dist(n, g) + dist(n, g) \cdot \sum_{i=0}^m \min_{n_k \in N} \{f_i(n_k)\}$$

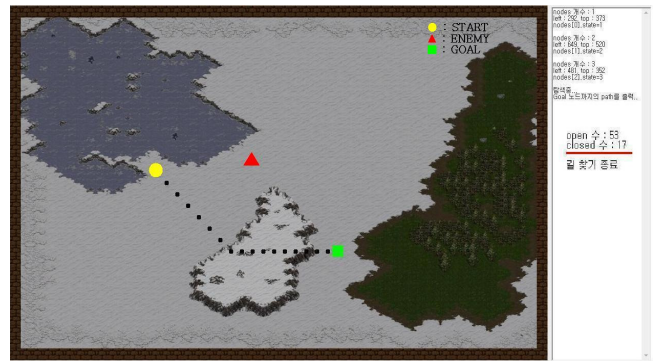
$$h_2(n) = dist(n, g) + dist(n, g) \cdot \sum_{i=0}^m \min_{n_k \in N} \{f_i(n_k) | s.t. f_i(n_k) \neq 0\}$$

$h_1(n)$ 과  $h_2(n)$ 는 각 특성별로 최소값을 취하여 남은 거리의 최소 추정값을 곱하여 얻어지는데 단,  $h_2(n)$ 의 경우는 0이 아닌 최소값을 취하는 것이 차이점이다.  $h_1(n)$ 의 경우에는 목표까지의 최단 거리상에 있는 모든 그리드의 위치 평가 함수가 최소 특성값이므로  $h_1(n)$ 은 과소 추정 휴리스틱이다. 그러나 최소 특성값이 0인 경우,  $h(n)$  대비 증가분이 없다는 의미이므로 탐색 성능의 큰 향상을 기대할 수 없다는 단점이 있다. 반면에  $h_2(n)$ 는 0이 아닌 최소값으로 한정하여 기존의  $h(n)$ 에 비해 증가분을 확실하게 확보하지만 허용 가능 휴리스틱을 보장하지 못한다. 즉,  $h_2(n)$ 을 이용하여 탐색한 결과는 최적 경로가 아닐 수 있다는 것이다. 이 두 가지 휴리스틱에 대한 장단점을 시뮬레이션을 통해 분석한다.

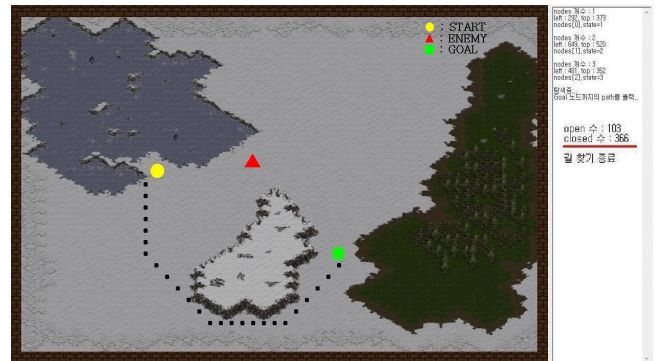
4. 시뮬레이션 결과

RPG(Role Playing Game) 게임에서 지도 제작을 위한 편집기들은 대부분 그리드-기반 혹은 셀-기반이다. 본 논문에서는 시뮬레이션을 위해 이들 편집기와 유사하게 그리드-기반 환경을 자바로 구현하였다. 편집기에서는 게임 배경을 만들고 수정할 수 있으며 장애물과 위협 요소의 위치에 따라 위치 평가 함수가 계산되고 휴리스틱을 선택하여 경로를 계획할 수 있다.

거리 이외의 특성으로는 위협으로부터의 거리, 지형 종류, 벽 겨안기(wall-hugging) 성질 등, 3가지를 고려하였으며 이동 거리를 고려한 비용함수와 비례하기 위해 작은 값을 선호하도록 값을 할당하였다. 자세한 점수 할당은 그림 1에 소개한 바와 같다.



최단거리만 고려 (방문 노드 수 : 17개)



위치평가함수 고려 (방문 노드 수 : 366개)

(그림 2) 최단 거리 함수와 위치 평가 함수에 의한 탐색 결과의 비교

그림 2는 기존 최단 거리 경로찾기의 결과와 위치 평가 함수를 이용한 경로찾기를 비교한 것이다. 위치 평가 함수를 이용하면 위협 요소를 피하는 경로를 얻지만 탐색에 의해 방문한 노드수가 증가했음을 알 수 있다. 위치 평가 함수 사용의 효율성을 위해 제안한 두 가지 휴리스틱  $h_1(n)$ 과  $h_2(n)$ 에 의해 개선된 정도를 <표 1>에 정리하였다. 지도 편집기를 이용하여 배경을 제작하고 그 위에 시작점, 목표점, 위협 요소 등의 위치를 변경하며 5회 실험하여 방문한 노드 수를 평균하여 휴리스틱  $h(n)$ 을 이용한 경우와 비교한 결과이다.  $0 \leq h(n) \leq h_1(n) \leq h_2(n)$ 의 관계를 갖고 있기 때문에 예측대로 결과가 나왔음을 볼 수 있다.

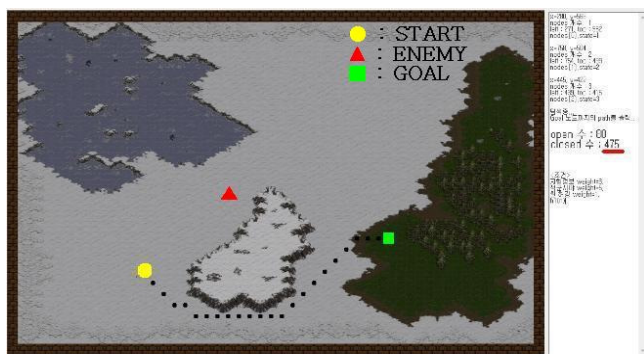
<표 1> 휴리스틱 사이의 성능 비교

	h(n)	h1(n)	h2(n)
case 1	790	393	61
case 2	650	475	51
case 3	1215	914	47
case 4	436	217	17
case 5	1133	694	217
average	844.8	538.6	78.6

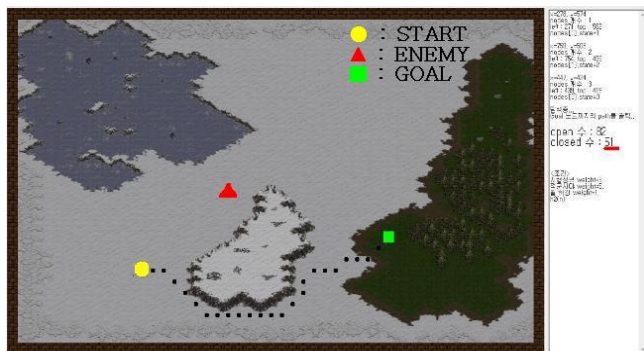
그림 3은 3가지 휴리스틱에 의해 찾은 경로를 나타낸다.  $h(n)$ 과  $h_1(n)$ 은 과소 추정 휴리스틱이기 때문에 최적 경로를 찾는 반면에  $h_2(n)$ 의 경우에는 항상 과소 추정을 보장하는 것이 아님을 알 수 있다. 그러나 거리 이외에 여러 가지 전술 요소를 고려하는 전술적 경로찾기에 있어서는 여러 요소의 합으로 나타내어지므로 결과에 보듯이 경로 자체의 큰 차이는 없었다.



$h(n)$ 에 의한 경로 이동 (방문한 노드 수 : 650개)



$h_1(n)$ 에 의한 경로 이동 (방문한 노드 수 : 475개)



$h_2(n)$ 에 의한 경로 이동 (방문한 노드 수 : 51개)

(그림 3) 휴리스틱에 따른 A\* 탐색 결과

### 5. 맺음말

본 논문에서는 위치 평가 함수를 이용하여 여러 가지 전술 요소를 고려한 경로를 찾을 수 있음을 확인하였으며 이때의 문제점인 탐색 성능 저하를 휴리스틱의 보완으로 개선할 수 있음을 보여주었다.

휴리스틱이 과소 추정이 아닌 경우에는 방문한 노드수

를 크게 줄일 수 있다는 장점이 있는 반면, A\* 탐색의 장점인 최적성을 보장하지 못하는 것이 단점이다. 향후에는 과대 추정 휴리스틱을 고안할 때, 최적 휴리스틱에 비해 허용할 수 있는 상한(upper bound) 설정에 대한 연구가 필요하다. 또한 게임 배경을 표현하는 방법으로는 그리드 기반 이외에 웨이포인트 그래프나 내비게이션 메쉬 표현 방식이 많이 사용되는데 [5] 이들 표현 방식에서도 위치 평가 함수를 경로 찾기에 적용할 수 있도록 확장하는 방법에 대한 연구와 각 특성별 자료 크기의 범위가 다른 문제를 해결하기 위한 연구도 필요하다.

### 참고문헌

[1] W. van der Sterren, "Tactical Path-Finding with A\*", Game Programming Gems 3, Charles River Media, pp294-306, 2002  
 [2] R. Straatman, W. van der Sterren., and A. Beij, "Killzone's AI: dynamic procedural combat tactics", In Proceedings of Game Development Conference, 2005  
 [3] I. Millington, "Tactical and Strategic AI," Artificial Intelligence for Games, Morgan Kaufmann, pp473-562, 2006  
 [4] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach (2nd Edition), Prentice Hall, 2003  
 [5] P. Tozour, "Search Space Representations" In: S. Rabin. (eds.): AI Game Programming Wisdom 2, Charles Rive Media, pp85-102, 2004