한국시뮬레이션학회 논문지
Vol. 19, No. 2, pp. 91-98 (2010. 6)

가중치 학습과 결합된 전술적 경로 찾기의 구현

유견아[1†]

Implementation of Tactical Path-finding Integrated with Weight Learning

Kyeon Ah Yu

ABSTRACT

Conventional path-finding has focused on finding short collision-free paths. However, as computer games become more sophisticated, it is required to take tactical information like ambush points or lines of enemy sight into account. One way to make this information have an effect on path-finding is to represent a heuristic function of a search algorithm as a weighted sum of tactics. In this paper we consider the problem of learning heuristic to optimize path-finding based on given tactical information. What is meant by learning is to produce a good weight vector for a heuristic function. Training examples for learning are given by a game level-designer and will be compared with search results in every search level to update weights. This paper proposes a learning algorithm integrated with search for tactical path-finding. The perceptron-like method for updating weights is described and a simulation tool for implementing these is presented. A level-designer can mark desired paths according to characters' properties in the heuristic learning tool and then it uses them as training examples to learn weights and shows traces of paths changing along with weight learning.

Key words : Computer game, Tactical pathfinding, Heuristic learning

요 약

기존의 경로 찾기는 장애물을 피하는 짧은 경로를 찾는 것에 집중되어 왔다. 그러나 컴퓨터 게임이 점점 복잡해지면서 경로 찾기에 매복지점이나 적으로부터의 가시성과 같은 전술적 정보를 포함하는 것이 요구되고 있다. 이와 같은 정보를 경로 찾기에 반영하는 한 가지 방법은 탐색 알고리즘의 휴리스틱 함수를 전술들의 가중치 합으로 나타내는 것이다. 본 논문에서는 주어진 전술적 정보에 대해 경로 찾기를 최적화하도록 휴리스틱을 학습하는 문제를 다룬다. 여기서 학습이란 휴리스틱 함수를 위한 좋은 가중치 벡터를 찾아내는 것을 의미한다. 학습용 훈련 예제는 게임 레벨 설계자가 제공하며 매 탐색 레벨마다 실제 탐색 결과와 비교되어 가중치를 갱신하는데 사용된다. 본 논문에서는 전술적 경로 찾기를 위해 탐색과 결합된 학습 알고리즘을 제안한다. 가중치를 갱신하는데 사용된 퍼셉트론 유사 방법을 설명하며 이를 구현한 시뮬레이션 도구를 소개한다. 시뮬레이션 도구에서는 레벨 설계자가 캐릭터의 특성에 따라 바람직한 이동경로를 제시할 수 있고, 이를 훈련 예제로 이용하여 가중치를 학습하며 훈련에 따라 변화하는 경로의 자취를 보여주는 기능을 제공한다.

주요어 : 컴퓨터 게임, 전술적 경로찾기, 휴리스틱 학습

1. 서 론

경로 찾기에 대한 대부분의 연구는 탐색 알고리즘을 이용하여 최단 경로를 찾는데 집중되어 왔으나 최근 컴퓨터 게임이 복잡해짐에 따라 경로를 결정짓는데 이동 거리뿐 아니라 다른 많은 주변의 정보들을 고려할 것이 요구되고 있다. 여기서 정보는 위협으로부터의 거리, 우호적 유닛으로부터의 거리, 적으로부터의 가시권 등 전술에 관한 정보를 의미하며 이와 같은 정보를 이용하여 경로를 찾는 것을 전술적 경로 찾기라고 한다(Millington, 2006).

일반적으로 경로 찾기는 휴리스틱 함수를 이용한 탐색으로 구현되는데(Stout, 1996) 바로 이 휴리스틱 함수를 이용하면 게임에서 필요한 전술 정보들을 표현할 수 있다.

2010년 1월 27일 접수, 2010년 3월 31일 채택
[1)] 덕성여자대학교 컴퓨터학과
주 저 자 : 유견아
교신저자 : 유견아
E-mail; kyeonah@duksung.ac.kr

즉, 전술 정보들의 중요성에 따라 가중치를 부여하고 휴리스틱 함수를 전술 정보들의 가중 합(weighted sum)으로 표현하면 탐색할 때 모든 전술 정보가 제각기 가중치에 따라 비중이 적게 혹은 많이 경로 선택에 반영되는 것이다. 여기서 우리는 가중치의 선택이 휴리스틱을 결정하게 되므로 곧 경로의 정확성, 경로의 길이, 탐색 성능 등 전체적인 경로찾기의 성능에 매우 중요한 요인이 됨을 알 수 있다. 게임 설계시에 레벨 설계자가 정보의 중요도에 따라 가중치를 부여하여 닫힌 형식(closed-form)의 휴리스틱을 만들 수 있으나 설계자가 임의로 배정한 가중치는 정확하지 않은 결과와 나쁜 성능을 초래할 수 있다. 그러므로 레벨 설계자가 처음부터 휴리스틱 함수를 정하고 시작하는 대신, 좋은 경로나 나쁜 경로의 예들을 제공만 하면 이들을 이용하여 탐색에 최적인 가중치를 찾아내도록 하는 방법을 고안하도록 한다. 즉, 주어진 예제의 경로 찾기 문제를 현재의 가중치를 이용하여 탐색하면서 설계자에 의해 주어진 샘플 경로와 비교하고 탐색된 경로가 건전하지 않으면 이를 보상하는 방향으로 가중치를 갱신함으로서 휴리스틱을 최적화하도록 하는 것이다.

탐색과 통합하여 휴리스틱을 학습하는 것에 대한 연구는 Daume 등(2005)이 제안하고 Xu 등(2007)이 인공지능 계획 문제에 적용하였던 LaSO(Learning as Search Optimization) 프레임워크가 있으며 본 논문에서는 이를 응용하도록 한다. 인공지능 계획 문제의 특징 중 하나는 비슷한 수준의 좋은 답이 여러 개 존재할 수 있다는 것인데 전술적 경로 찾기 문제는 목표에 이르는 경로가 여러 개일 수 있다는 면에서 이와 비슷하다. 그러나 지나가면 안 되는 루트가 명확히 있다는 차이점이 있다. 그러므로 본 논문에서는 Xu 등(2007)의 방법과 유사하게 너비-우선 빔 탐색으로 휴리스틱을 학습하고자 시도하는데 Xu 등(2007)의 알고리즘처럼 긍정의 훈련 예제(positive training example)만을 이용하는 것이 아니라 부정의 훈련 예제(negative training example)를 사용하여 두 문제 영역의 차이점을 극복하도록 한다. 긍정 및 부정 훈련 예제를 모두 이용하여 좋은 루트를 좇아가도록 하는 동시에 나쁜 루트는 피하도록 가중치를 튜닝하는 학습 알고리즘을 고안하는 것이 본 논문의 주된 목표이다. 본 논문에서는 설계자가 배경 화면과 캐릭터에 따라 예제 경로를 입력할 수 있고, 제안한 알고리즘에 의해 학습을 진행하며 변해가는 경로 트랙을 볼 수 있는 GUI 도구를 구현하여 학습과 경로찾기를 시뮬레이션 해 볼 수 있도록 한다.

본 논문의 구성은 2장에서는 전술적 경로 찾기 분야에서의 연구 동향을 살펴보고 3장에서는 전술적 경로 찾기를 탐색 문제로서 정의한 후, 수정된 알고리즘과 가중치 갱신 방법을 소개한다. 4장에서는 제안된 알고리즘을 구현한 시뮬레이션 도구를 통해 실행한 결과들과 이에 대한 분석 결과를 제시하며 5장에서 향후 과제에 대한 제안과 함께 맺는다.

2. 선행 연구

전술적 경로 찾기는 주변의 전술 정보를 이용하여 경로를 찾는 것을 말한다. 전술적 경로 찾기를 하면 캐릭터들은 이동 목적에 맞도록 다른 루트를 찾을 수 있다. 그러므로 이동 거리나 시간만을 고려하여 경로를 결정하던 기존의 경로 찾기와는 다르게 시작과 목표 지점이 같아도 캐릭터 별로 각기 다른 경로를 찾을 수 있다는 의미이다. 최근 이를 구현하기 위한 전술적 경로 찾기에 대한 몇 가지 결과들이 발표되었다.

Liden(2002)은 웨이포인트에 전술 정보를 생성하는 방법을 제안하고 웨이포인트 그래프를 이용하여 기존의 최단 경로보다 더 안전한 경로를 탐색하도록 하였다. 전술적 정보를 효과적으로 저장하기 위해 연결 및 가시 정보를 비트 스트링 클래스에 저장하고 여러 가지 전술 정보를 미리 계산해 놓아 실행시간에 웨이포인트의 평가가 신속히 이루어질 수 있도록 하였다. 그러나 웨이포인트는 레벨 설계자에 의해 수동으로 생성될 뿐 아니라 미리 계산된 웨이포인트 정보는 모든 캐릭터에 동일하게 적용되어야 하는 한계가 있다.

Millington(2006)과 Sterren(2003)은 전술 정보를 탐색의 비용 함수를 이용하여 표현하였다. Millington(2006)은 비용 함수를 전술 정보들의 일차 결합으로 나타내었고 일차결합을 위한 가중치 할당 부분에서는 각각의 가중치가 가지는 의미를 설명하고 레벨 설계자가 가중치를 할당하는 가이드 라인을 제시하였다. 그러나 정확한 비용 함수를 위해 가중치 할당을 자동화하는 방법에 대해서는 언급하지 않았다. Sterren(2003)은 비용 함수를 거리와 위험의 두 가지 카테고리로 분리하고 표현하고 탐색에 반영하는 방법을 제안하였으나 역시 정보들의 중요도에 따라 통합적으로 위험도에 반영하였으며 가중치에 대한 언급은 하지 않았다.

Straatman(2005)은 컴퓨터 체스 등의 보드 게임에서 보드 구성을 점수화하기 위해 흔히 사용되고 있는 위치 평가 함수(position evaluation functions, Laramee, 2000)를 정의하여 저격수의 위치 선정이나 매복 지점을 선정하

는데 사용하였다. 위치 평가 함수는 게임에서 위치의 좋고 나쁨을 나타내기 위한 모든 특성들로 구성되는데 경로 찾기는 위치 평가 함수를 이용하는 한 가지 예로 소개되었다. A* 알고리즘이 경로 탐색을 위한 기본 알고리즘으로 사용되고 이 때 필요한 비용 함수를 위치 평가 함수를 확장하여 사용하였다. 이와 같은 비용 함수와 함께 A* 탐색을 하면 전술적 경로를 찾을 수 있음을 보이는 한편 대체적으로 탐색을 위해 방문하는 노드의 수가 증가함을 지적하였다. 위치 평가 함수를 비용 함수에 이용할 때 가중치를 부여하고 성격이 다른 캐릭터들마다 가중치를 튜닝하면 캐릭터마다 고유의 이동 경로를 얻을 수 있음을 언급하였으나 구현하지는 않았다.

Kamphuis 등(2005)과 Rook 등(2005)은 전술 정보가 일반 탐색 알고리즘으로 경로를 찾을 때 쉽게 이용될 수 있고 새로운 전술 정보가 생겼을 때 빠르게 정보가 갱신될 수 있도록 전술 정보를 효과적으로 저장하는 방법에 대해 제안하였다. 일단 자유 공간을 복도(corridor)라고 불리는 여러 사다리꼴로 나누고 노드를 각 복도마다 지정하고 이에 전술 정보 값을 할당하면 충분하도록 하였다. 이렇게 노드가 생성되면 일반 탐색 알고리즘으로 전술 정보를 포함한 휴리스틱을 이용하여 탐색하여 경로를 찾게 되는데 이 논문에서는 전술 정보를 표현하기에 충분하도록 복도를 나누는 것과 노드를 정하는 일에 초점이 맞추어 있으며 탐색에 필요한 휴리스틱을 튜닝하는 문제는 다루지 않았다.

3. 전술적 경로 찾기를 위한 가중치 학습

전술적 경로 찾기는 기존의 경로 찾기에 비해서 복잡한 것으로 생각되지만 휴리스틱 함수에 주변 환경의 전술 정보를 포함하고 있기 때문에 일단 휴리스틱이 정해지고 나면 일반 탐색 문제로 전환된다. 그러므로 전술 정보를 포함하기 위한 휴리스틱 함수를 어떻게 정의하는가와 어떤 탐색 알고리즘을 이용하여 학습을 어떻게 통합할 것인지가 전술적 경로 찾기 문제 해결의 방향을 결정짓는 핵심이라고 할 수 있다.

3.1 전술 정보를 표현하는 휴리스틱 함수

전술적 경로 찾기는 게임 캐릭터가 이동할 때 주변 환경의 전술 정보를 고려하게 함으로서 보다 자연스러운 이동을 가능하게 한다. 휴리스틱 함수를 다음 식과 같이 전술들의 일차 결합으로 나타내어 전술 정보를 탐색에 반영한다. 아래 식에서 n은 탐색 노드이며 f_i는 게임에서 지원되는 전술들, w_i는 f_i에 대한 가중치를 나타낸다:

$$H(n) = \sum_i w_i f_i(n) = w \cdot F(n)$$

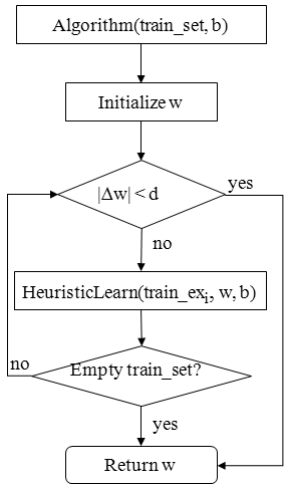
소개된 관련 연구 대부분이 언급하였듯이 전술 정보들을 어떻게 저장하고 휴리스틱 값으로 변환하여 사용하는지가 중요하다. 전술 정보는 두 가지 타입으로 나눌 수 있는데 연결-기반 정보와 위치-기반 정보이다. 연결-기반 정보는 가시성이나 유클리디언 거리와 같이 두 노드 사이에 정의되는 정보로서 미리 계산하여 저장시키고 사용하면 실행 시에 계산 시간을 절약할 수 있으나 노드수가 많아지면 비효율적이 될 수 있을 뿐 아니라 동적인 환경에서는 사용하지 못하는 단점이 있으므로 본 논문에서는 연결이 생성될 때마다 계산하는 방법을 사용한다. 위치-기반 정보는 그 위치의 형태나 기울기 등, 각 노드가 갖고 있는 정보를 의미하며 휴리스틱 값으로 이용되기 위해서는 연결-기반 비용으로 전환되어야 하는데 본 논문에서는 연결을 이루는 두 노드의 정보 값의 평균을 취하여 사용하기로 한다. 두 노드가 멀리 떨어져 있는 경우, 평균을 취하는 방법은 연결 링크의 정보를 표현하기에 문제가 있으나 본 논문에서는 게임 배경을 그리드-기반으로 표현하여 연결이 이웃한 노드 사이에만 발생하므로 평균값을 사용하는 것이 무방하다.

3.2 학습 통합을 위한 탐색 알고리즘

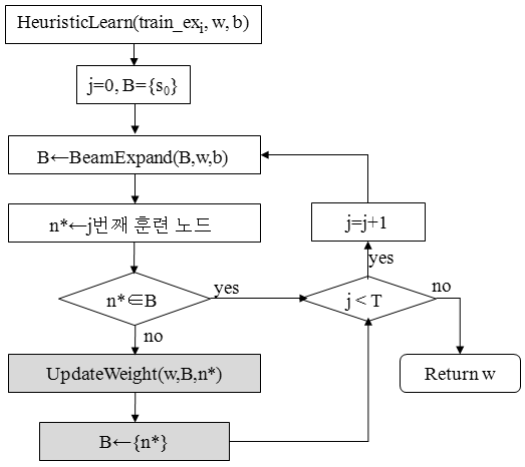
전술적 경로 찾기를 위한 탐색 알고리즘으로는 g(n)+h(n) 형태의 비용함수를 이용하는 대부분의 탐색 알고리즘을 사용할 수 있는데 본 논문에서는 너비-우선 빔 탐색 알고리즘을 사용하고자 한다. 빔 탐색 알고리즘은 오픈 큐에 미리 정해진 수만큼만 후보를 남기는 알고리즘으로 그 중에 너비-우선 빔 탐색은 빔 탐색 알고리즘을 너비-우선 방식으로 전개하는 것이 특징이다. 즉, 각 탐색 레벨에서 현재 레벨에 있는 노드들의 모든 자식 노드를 생성하고, 자식 노드들을 비용함수에 따라서 정렬한 다음, 정렬된 자식 노드 가운데 미리 정해진 수만큼 빔에 남기면서 진행된다. 빔에 목표 노드가 있는지 확인하고 아니면 이 과정을 반복한다. 이와 같은 방식에 의해 오픈 리스트의 구성 요소들은 트리의 깊이 별로 구성되게 되고 주어진 훈련 예제 경로와의 비교가 단계별로 이루어지게 되어 학습 알고리즘을 구현에 용이성을 제공한다.

Xu 등(2007)은 휴리스틱 학습을 탐색과 통합하여 인공지능 계획 분야에 적용하였다. 기존의 탐색 알고리즘과의 차이는 두 가지인데 한 가지는 탐색 오차가 발생하면

가중치를 갱신하는 루틴을 추가하는 것이고 다른 한 가지는 예제 노드가 빔에 없으면 현재 빔의 내용을 버리고 예제 노드로 빔을 구성하여 탐색을 계속하도록 하는 것이다. 그림 1의 흐름도에서 회색으로 채워진 처리 상자가 위에서 언급한 두 가지 수정 부분에 해당한다. 우선 전체 알고리즘(좌측 흐름도)의 입력은 훈련 예제 집합 과 빔 크기 이고 가중치 값의 변화가 일정 기준 이하가 되거나 훈련 예제가 더 이상 없어질 때까지 HeuristicLearn 프로시저를 수행한다. HeuristicLearn 프로시저의 입력은 훈련 예제 train_ex$_i$=(x_i,y_i),현재 가중치 w, 빔 크기 b이다.



(a) 휴리스틱 학습의 메인부분



(b) 빔탐색과 학습의 결합부분

그림 1. 휴리스틱 학습 알고리즘

3.3 전술적 경로찾기를 위해 수정된 알고리즘

전술적 경로 찾기는 (Xu 등.2007)에서 다룬 인공지능 계획 문제와 마찬가지로 목표에 이르는 경로가 여러 개 존재하며 어떤 특정한 경로만이 다른 것들에 비해 우월하다고 볼 수 없는 경우가 많다. 기본 탐색 방법으로 선택한 빔 탐색 알고리즘은 평가 함수에 의해 최고 노드 한 개를 선택하기 보다는 상위 b개를 선택하는 방식으로 인공지능 계획 분야나 경로 찾기 문제에 적합하다. 반면에 전술적 경로 찾기가 인공지능 계획과 다른 점은 전체 상태 공간 대비, 좋은 경로의 비율이 크다는 것이다. 그러므로 좋은 경로를 모두 포함하도록 빔 크기를 크게 정하면 탐색 오차가 발생하지 않아 결과적으로 가중치 갱신이 이루어지지 않는 문제점이 있다. 반대로 빔 크기를 너무 작게 설정하면 좋은 경로중의 일부가 나쁜 경로로 분류되게 되는데 실제로 반드시 피해야 하는 부정적 예가 따로 존재하기 때문에 학습의 모순을 야기시킬 수 있다. 이 문제를 해결하기 위해 기존의 방식대로 좋은 경로를 긍정적 훈련 예제로 사용하는 동시에 반드시 피해야 하는 최악의 경로를 부정적 훈련 예제로 이용하도록 그림 1에서 소개한 휴리스틱 학습 알고리즘을 수정한다.

전체 알고리즘의 입력은 훈련 예제 집합 train_set= {(train_ex$_i$, type)}과 빔 사이즈 b인데 train_set 에 예제가 긍정인지 혹은 부정인지를 나타내는 이진 변수 type이 추가된 것이다. $n^* = (x_i,(s_0,s_1,\cdots,s_L))$를 긍정 혹은 부정의 모든 경우에 i번째 훈련 예제에 대한 j번째 노드까지의 주어진 경로(desired search path)를 나타내는 것이라고 하면 긍정적 훈련 예제의 경우에는 n^*가 빔에 없으면, 부정적 예제의 경우에는 n^*가 빔에 있으면 탐색 오차가 발생하는 것이다. 이와 같이 수정된 부분의 알고리즘을 그림 2에 도시하였다

탐색 오차가 발생하면 가중치를 갱신하는 루틴 $Update\ Weight(w,B,n^*)$를 실행하게 되는데 긍정적 예제와 부정적 예제의 경우로 나누어 표현한 식 (1)을 기반으로 작성되었다. $F(n) =< f_1(n),\cdots,f_m(n) >$은 지원되는 전술 벡터 함수를, w는 가중치 벡터를 나타낸다.

$$w \leftarrow w + \begin{cases} \alpha\left(\dfrac{\sum_{n\in B} F(n)}{|B|} - F(n^*)\right), & positive\ ex \\ \alpha\left(F(n^*) - \dfrac{\sum_{n\in B_n} F(n)}{|B_n|}\right), & negative\ ex \end{cases} \quad (1)$$

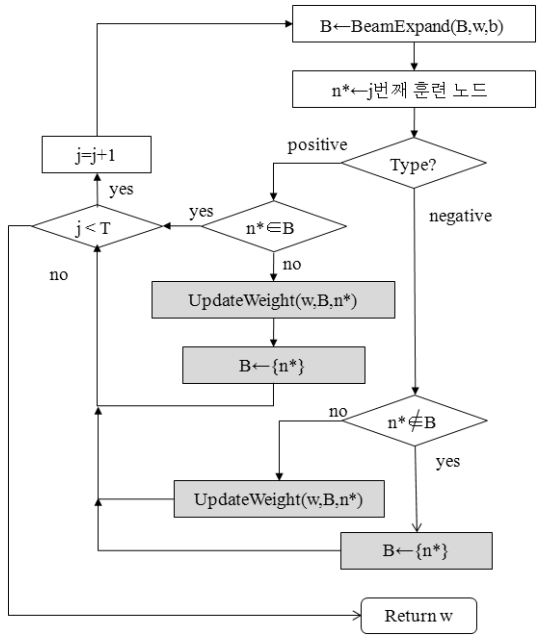


그림 2. 부정 예제를 포함하는 휴리스틱 학습 알고리즘

여기서 $0 < \alpha \leq 1$인 학습 속도를 조절하는 인자이고 $B_n = \{n | n \in B, H(n) \geq H(n^*)\}$이다. 낮은 값이 더 선호되는 휴리스틱의 경우이므로 가중치의 갱신은 바람직한 혹은 좋은 노드의 휴리스틱 값은 더 낮추고, 피해야 하거나 나쁜 노드의 휴리스틱 값은 높이는 방향으로 이루어짐을 알 수 있다.

4. 시뮬레이션 결과

수정된 알고리즘에 의해 휴리스틱 학습이 진행됨에 따라 경로 찾기에 어떤 변화가 있는가를 시뮬레이션을 통해 분석하였다. 경로의 질적인 측면 뿐 아니라 경로의 길이와 경로를 찾기 위해 방문한 노드수 등의 양적 성능을 비교하며 또한 빔 탐색에서 가장 중요한 요소인 빔 사이즈의 변화가 탐색과 학습에 어떻게 영향을 주는가를 분석하였다.

시뮬레이션을 위하여 구현된 도구는 그림 3과 같다. 이 도구에서는 지형을 편집할 수 있고 시작점과 목표점을 지정하고 캐릭터를 선택하면 경로를 탐색하여 결과를 보여주는 기능이 있다. 또한 학습을 위해 레벨 설계자가 선택된 캐릭터에 대해 훈련 경로를 입력하는 기능이 있는데 훈련 경로는 긍정, 부정의 두 가지로 입력할 수 있다. 입력된 예제에 따라 휴리스틱을 학습하는데 학습 결과만 디

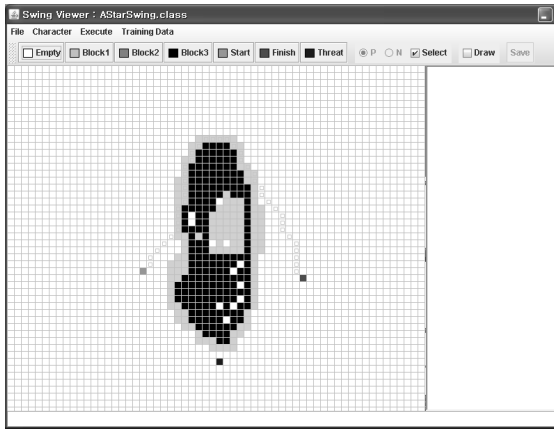


그림 3. 학습 알고리즘의 구현

스플레이할 수도 있고 학습이 진행되면서 변화하는 경로들을 모두 디스플레이할 수도 있어서 이를 참고하여 레벨 설계자는 추가로 훈련 예제를 더 입력하고 학습에 이용할 수 있도록 설계되었다. 그림 3은 사용자에 의해 긍정 예제가 입력된 후의 화면을 나타낸다.

그림 3는 임의로 균등하게 가중치를 할당하고 탐색한 결과로부터 시작하여 학습을 통해 변화하는 가중치로 탐색한 결과를 포함하고 있다. 이 예에서는 레벨 설계자가 입력한 훈련 예제가 가지는 의미를 쉽게 알 수 있도록 이동 거리, 위협요소와의 거리, 보급소로부터의 거리 등, 3가지 전술 요소만을 고려하였으며 학습 속도 α=0.01, 빔 크기 b=5, 초기 가중치는 1/n로 균등 분할하여 시뮬레이션하였다. 흰색과 회색으로 구분한 지형의 차이는 고려하지 않고 검정으로 구분한 장애물만 블록된 것이라고 가정하였다. 지형의 종류를 고려하기 위한 대안은 5장 결론에서 언급한다. 사용자가 입력한 긍정 예제의 예는 그림 3과 같고 초기 균등한 가중치에 의해 탐색 알고리즘을 적용한 결과는 경로 상에 '초기'라고 표시하였다. 그리고 학습이 8회 및 13회 진행됨에 따라 변화하는 과정을 나타내었다. 학습이 13회 반복된 결과가 그림 3에서 입력한 예와 유사함을 알 수 있다.

그림 4에서는 학습이 진행됨에 따라 긍정 예제와의 비교에 의해 탐색 경로가 훈련 예제에 가까워지는 경우를 보여주었으나 그림 5에서는 긍정 예제만으로는 학습되지 않을 수 있음을 보인다. 그림 5 (a)에서 아래쪽 경로와 같이 긍정 예제를 입력하여 학습을 진행하면 초기 가중치에 의한 경로가 거리상으로 근접하여 있으므로 빔 탐색 결과, 탐색 오차로 구분되지 못하고 따라서 학습이 일어나지 않

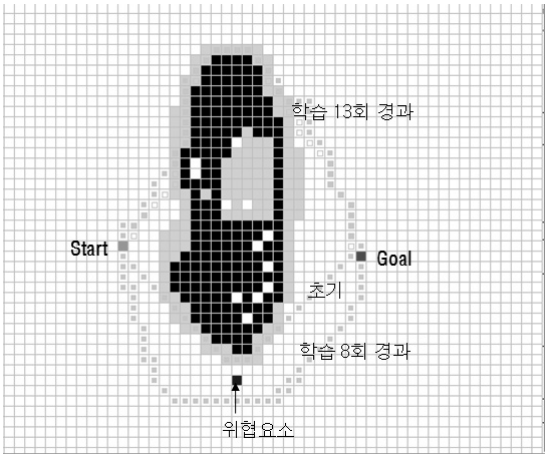
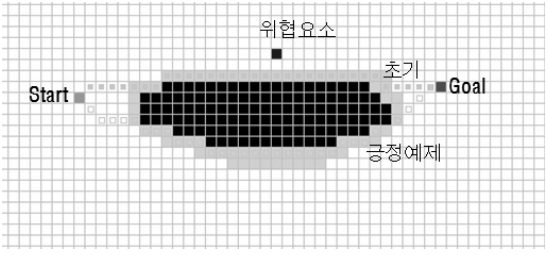
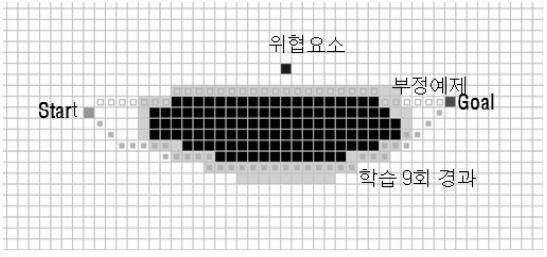


그림 4. 학습 빈도에 따른 경로의 변화



(a) 긍정 예제만으로의 학습이 실패하는 경우



(b) 부정 예제의 제시로 학습이 성공한 경우

그림 5. 부정 예제를 이용한 학습 결과

는다. 즉 지속적인 학습 후에도 초기 경로와 같이 위협요소에 가까운 길을 생성한다. 빔 크기를 작게 조정하여 해결할 수 있으나 너무 작은 빔 크기의 경우, 허용되는 경로에 대해서도 탐색 오류로 분류될 수 있기 때문에 부정 예제를 명시적으로 사용함으로서 이 문제를 해결한다. 그림 5 (b)에서는 거리상으로 가깝지만 위협요소를 피하는 경로를 부정 예제로 제공하였더니 위협요소를 피하는 경로가 학습됨을 보여준다.

빔 탐색 알고리즘에서는 빔 크기가 탐색의 성능을 결정하는 중요한 요인이다. 빔 크기를 크게 할수록 결과는

표 1. 빔 크기에 따른 학습 성능 비교

빔 크기	1	5	10	15
방문노드수	72.2	184.4	286.2	435.0
경로길이	43.2	38.9	35.3	32.5
학습성공률(%)	40	70	100	80

표 2. 긍정 예제만 혹은 부정 예제 모두를 사용한 경우의 학습 성능 비교

	긍정예	긍정 · 부정예
학습성공률(%)	60	100
소요학습수	32	24

최적의 해에 가깝지만 방문하는 노드의 수가 증가하고 따라서 탐색 소요시간이 길어지는 단점이 수반된다. 그와 반대로 빔 크기를 작게 할수록 탐색 소요시간은 줄어들지만 많이 우회하는 경로를 찾게 되어 최적 해와는 많은 차이를 보인다. 본 논문에서는 학습과 통합된 탐색을 하는 경우, 빔 크기에 따라 어떤 영향이 있는지 확인하기 위하여 빔 크기의 변화에 따른 탐색 성능을 비교해 보았다 (표 1).

10가지 경우를 실험한 결과, 빔 탐색 알고리즘을 학습에 이용할 때에는 빔 크기의 변화가 탐색의 경우와 조금 다른 결과를 가져오는 것을 알 수 있었다. 빔 크기가 작아질수록 방문 노드수는 급격하게 감소한다는 장점이 있지만 아예 학습에 실패하는 경우가 많다는 것이 치명적인 단점이다. 경로의 길이는 빔 크기가 커질수록 나아지다가 일정 빔 크기에 다다르면 더 이상 나아지지 않는다. 또한 빔 크기를 너무 크게 하는 경우에도 학습에 실패하는 경우가 있음을 볼 수 있다. 즉 학습을 위해서는 이와 같은 시뮬레이션을 통하여 적절한 빔 크기를 결정하여야 함을 알 수 있다. 표 2는 여러 가지 상황에서 긍정의 예제만으로 학습한 결과와 두 종류 예제를 모두 이용하여 학습한 결과를 비교한다. 이 시뮬레이션에서 학습속도 α는 0.01로 빔 크기는 10으로 세팅하였다.

긍정 예제의 경우, 소요학습수는 성공한 경우들의 평균을 취하였다. 예제의 종류에 따라 학습 성공률은 달라질 수 있으므로 60%와 100%의 숫자가 큰 의미가 있는 것은 아니지만 중요한 것은 긍정 예제만을 이용하였을 때, 학습에 실패하는 것을 부정 예제를 이용함으로서 방지할 수 있다는 사실이며 긍정, 부정 예제를 번갈아 사용함으로서 소요 학습수를 줄일 수 있다는 사실도 주목할 만한 결과이다.

본 논문의 결과는 RTS(real-time strategy) 게임의 제

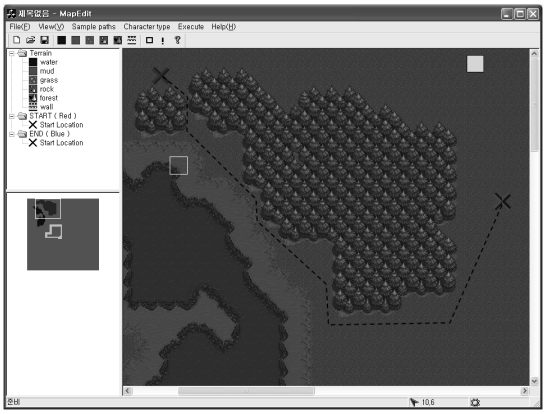


그림 6. 셀-기반 편집기로 제작된 게임에서의 학습

작에 사용되는 셀-기반 편집기로 생성된 게임에 적용 가능하다. 게임 배경에서 셀 하나를 격자 하나로 대응한 격자-기반(grid-based)으로 표현하고 기존의 편집기를 통해 생성된 자료 구조를 최소로 수정하여 전술 정보를 저장하면 빔 탐색 알고리즘으로 학습한 결과를 그림 6에 나타내었다.

5. 결 론

본 논문에서는 전술적 경로 찾기를 휴리스틱 학습 문제로 접근하였다. 경로찾기가 탐색으로 구현되는 점에 착안하여 휴리스틱 학습과 탐색을 통합하여 학습이 이루어지도록 하였다. 경로찾기에서는 해답경로(solution path)로 선택할 만한 경로가 다수 존재한다는 성질 때문에 A* 알고리즘이 아닌 빔 탐색 알고리즘을 이용하여 학습한다. 반드시 피해야 하는 경로도 존재하는 것을 해결하기 위해 학습을 위한 예제로서 긍정 및 부정 예제 모두를 이용할 것을 제안하였다. 부정 예제를 이용한 결과, 긍정 예제만으로 학습하였을 때 발생하였던 실패를 줄일 수 있었으며 학습 속도도 향상되었다. 그러나 4장에서 언급하였듯이 지형의 종류와 같이 경로의 비용이 휴리스틱에만 의존하는 것이 아니라 전체로서 더 중요한 경우, 휴리스틱 대신 전체 비용함수를 학습하는 방식으로 수정하여 이를 해결할 수도 있다. 즉 전체 비용함수를 최적화하도록 알고리즘을 수정한다면 긍정 예제만을 이용해도 유사한 결과를 가져올 수 있을 것이라 판단되며 g+h형식의 평가함수를 이용하는 다른 탐색 알고리즘의 이용도 고려해 볼 수 있다. 또한 본 논문은 탐색과 결합된 학습 알고리즘을 격자-기반으로 표현된 게임에 응용하였는데 기존의 게임들에서는 격자-기반 표현보다 여러 가지 장점을 가진 표현 방식들을 많이 사용하고 있다. 특히 게임에서 많이 사용하고 있는 방식들로는 가시성 그래프, 웨이포인트 그래프, 네비게이션 메쉬 등이 있는데(Tozour, 2004) 이들을 사용하는 경우에는 근접해 있지 않은 두 노드 사이의 전술정보 값들을 처리하는 방법에 대한 연구도 필요하다.

참 고 문 헌

1. A.Kamphuis, M. Rook, and M.H. Overmas, “Tactical Path Finding in Urban Environment,” In Proceedings First International Workshop on Crowd Simulation, pp. 51-60, 2005.
2. B. Stout, “Smart Moves: Intelligent Path finding”. Game Developer Magazine, April, pp. 28-35, 1996.
3. F. Laramee, Chess Programming Ⅵ:Evaluation Functions, http://www.gamedev.net/reference/ articles/article1208.asp, 2000.
4. H. Daume III and D. Marcu, “Learning as Search Optimization: Approximate Large Margin Methods for Structured Prediction”, ICML-05, pp. 169-176, 2005.
5. I. Millington, “Tactical and Strategic AI,” Artificial Intelligence for Games, Morgan Kaufmann, pp. 473-562, 2006.
6. L. Liden, “Strategic and Tactical Reasoning with Waypoints”, AI Game Programming Wisdom, Edited by Steve Rabin, Charles Rive Media, pp. 211-220, 2002.
7. M. Rook and A. Kamphuis, “Path Finding using Tactical Information” In Poster Proc. Eurographics /ACM SIGGRAPH Symposium on Computer Animation, pp. 18-19, 2005.
8. R. Straatman, W. van der Sterren., and A. Beij, “Killzone’s AI: dynamic procedural combat tactics”, In Proceedings of Game Development Conference. http://www.cgfai.com/docs/straatman_ remco_killzone_ai.pdf, 2005.
9. S.Russell and P. Norvig, Artificial Intelligence: A Modern Approach (2nd Edition), Prentice Hall, 2003.
10. Tozour, P., “Search Space Representations” In: Rabin, S. (eds.): AI Game Programming Wisdom 2, Charles Rive Media, pp. 85-102, 2004.
11. W. van der Sterren, “Tactical Path-Finding with A*”, Game Programming Gems 3, Charles River Media, pp. 294-306, 2002.
12. Y.Xu, A.Fern, and S.Yoon, “Discriminative Learning of Beam-Search for Planning” In Proceedings of International Joint Conference on Artificial Intelligence, pp. 2041-2046, 2007.

유 견 아 (kyeonah@duksung.ac.kr)

1986 서울대학교 공과대학 제어계측공학과 학사
1988 서울대학교 공과대학 제어계측공학과 석사
1995 미국 USC Computer Science 박사
1996～현재 덕성여자대학교 컴퓨터학과 교수

관심분야 : 인공지능, 경로계획, 로봇 알고리즘