

결정트리 학습 알고리즘을 활용한 축구 게임 수비 NPC 제어 방법*

조달호*, 이용호*, 김진형**, 박소영**, 이대웅**
 상명대학교 대학원 컴퓨터과학과*, 소프트웨어대학 디지털미디어학부**
 chodalho@msn.com, yhlee512@gmail.com, ghostliving@naver.com,
 {ssoya, rhee219}@smu.ac.kr

NPC Control Model for Defense in Soccer Game
 Applying the Decision Tree Learning Algorithm

Dal-ho Cho*, Yong ho Lee*, Jin Hyung Kim**, So-Young Park**, Dae-Woong Rhee**
 SangMyung University Dept of Computer Science, Graduate School
 Division of Digital Media, College of Computer Software & Digital Media**

요 약

본 논문에서는 결정트리 학습 알고리즘을 활용한 축구 게임 수비 NPC 제어 방법을 제안한다. 제안하는 방법은 실제 게임 사용자들의 이동 방향 패턴과 행동 패턴을 추출하여 결정트리 학습 알고리즘에 적용한다. 그리고 학습된 결정트리를 바탕으로 NPC의 이동방향과 행동을 결정한다. 실험결과 제안하는 방법은 결정트리 학습에 시간이 다소 걸리지만, 학습된 결정트리를 바탕으로 이동방향이나 행동을 결정하는 시간은 약 0.001-0.003 ms(밀리초)가 소요되어 실시간으로 NPC를 제어할 수 있었다. 또한, 제안하는 방법은 현재 상태 정보 뿐만 아니라 이를 분석한 관계정보, 이전 상태 정보도 함께 활용하므로, 기존방법인 (Letia98)에 비해 이동방향 결정 시 높은 정확도를 나타냈다.

ABSTRACT

In this paper, we propose a defense NPC control model in the soccer game by applying the Decision Tree learning algorithm. The proposed model extracts the direction patterns and the action patterns generated by many soccer game users, and applies these patterns to the Decision Tree learning algorithm. Then, the proposed model decides the direction and the action according to the learned Decision Tree. Experimental results show that the proposed model takes some time to learn the Decision Tree while the proposed model takes 0.001-0.003 milliseconds to decide the direction and the action based on the learned Decision Tree. Therefore, the proposed model can control NPC in the soccer game system in real time. Also, the proposed model achieves higher accuracy than a previous model (Letia98); because the proposed model can utilize current state information, its analyzed information, and previous state information.

Keywords : Decision Tree, Soccer Game, Machine Learning, Game AI

접수일자 : 2011년 8월 18일 일차수정 : 2011년 12월 01일 심사완료 : 2011년 12월 13일

교신저자(Corresponding Author) : 이대웅 (rhee219@smu.ac.kr)

* 본 연구는 서울시 산학연 협력과제(JP090973) 지원을 받아 수행된 연구 결과입니다.

1. 서 론

현재 시장에 출시되어 있는 축구 게임은 시뮬레이션 게임과 매니지먼트 게임의 2가지 유형으로 구분된다. 시뮬레이션 게임은 사용자가 시합에 참가하는 게임이고 매니지먼트 게임은 감독이나 구단주가 되어 축구팀을 운영하는 게임을 말한다. 우리나라의 경우는 EA Sports의 피파 시리즈와 Konami의 위닝 일레븐 시리즈로 대표되는 시뮬레이션 게임이 인기를 끌고 있다.

온라인 게임으로 한정해서 살펴보면 피파 온라인이 2006년 6월 서비스 이후 최고 동시접속자 15만 명을 기록하였다. 이후 현재까지도 국내 게임 순위에서 상위권을 차지하며 스포츠 게임 분야에서 독보적인 위치를 차지하고 있다[1]. 피파 온라인의 인기에 힘입어 국내 게임 업체에서도 다양한 온라인 축구 게임을 출시하였다. 하지만 프리스타일 풋볼을 제외하고는 시장에 연착륙한 게임이 없는 실정이다. 이미 완성도가 높은 해외 게임이 소개되어 국내게임이 비교우위를 확보하기 어려웠기 때문이다.

이러한 점은 특히 AI 부분에서 두드러지게 나타난다. AI 기술의 부재로 인해 현재 온라인 게임의 경우 프리스타일 풋볼같이 골키퍼만을 AI로 구현하여 서비스 하는 중이다. 이러한 접근방법은 사용자들이 수비수보다는 공격수를 선호하며, 종종 게임도중 무단이탈하여 NPC(Non Player Character)로 대체되는 경우에 적절히 대응할 수 없다는 한계가 있다.

따라서 본 논문에서는 축구 게임 NPC를 자동으로 제어하는 방법을 제안한다. NPC가 적극적으로 공격할 경우 사용자들의 흥미를 떨어뜨릴 수 있으므로, NPC는 수비역할에 충실하도록 설계하였다. 앞으로 2장에서는 NPC 제어와 관련된 기존 연구에 대해 살펴본다. 3장에서는 제안하는 축구 게임 수비 NPC 제어방법에 대해 자세히 설명하고 4장에서는 제안하는 방법이 실제 사용자와 얼마나 유사하게 게임을 진행하는지 실험을 통해 평가한

다. 마지막으로 5장에서 결론을 맺는다.

2. 관련연구

게임에서 NPC를 제어하기 위해서는 다음과 같은 특징을 고려해야 한다. 무엇보다 실시간으로 NPC를 제어할 수 있도록 의사 결정이 고속으로 처리되어야 한다. 또한, 게임 진행패턴이 단조롭지 않아서 사용자가 실제 다른 사용자와 경기를 하고 있다는 느낌이 들어야 한다. 이러한 점을 고려하여 그동안 FSM(Finite State Machine), 퍼지논리, 유전자 알고리즘, 신경망, 결정트리 등을 이용한 접근방법이 제안되었다[2].

FSM은 NPC가 행동해야 하는 유한개의 상태를 정의하고, 조건에 따라 상태를 전이함으로써 행동을 제어한다. FSM은 구현이 쉽고 구조가 안정적인 장점이 있지만, 단순하여 행동의 결과가 쉽게 예측될 수 있다[3,4]. 반면 FSM을 복잡하고 정교하게 구성하면 설계자의 주관적인 직관이나 경험에 의존하는 경우가 많아 개발과 수정 보완, 확장성 측면에서 어려움이 많다[3].

퍼지 논리는 애매한 표현을 처리하기 위해 사용되는 AI 기법이다. 참 또는 거짓으로 표현되는 전통적인 논리가 아닌 “어느 정도”나 “얼마나”같은 말로써 수식되는 것들에 대한 논리이다[5,6]. 하지만 퍼지 논리를 활용하다 보면, 서로 연관되는 집합이 너무 많이 늘어나는 조합 폭발이라는 상황이 발생할 수 있다. 이러한 경우 퍼지 논리 체계의 연산 속도가 느려지고 유지, 보수하기가 힘들어지는 문제가 발생한다[6].

유전자 알고리즘은 자연의 진화과정과 유전법칙에 영향을 받은 메타 휴리스틱 알고리즘이다. 유전자 알고리즘은 일반적으로 개체의 모집단 정의 적합도에 따른 선택 단계, 후손을 생성하기 위한 교배 단계, 그리고 새로운 후손발생을 유도하기 위한 돌연변이 단계를 거치며 최적의 해를 탐색한다. 유전자 알고리즘은 복잡한 해공간의 탐색능력이 우

수하여 변수와 제약이 많은 대형 수리문제를 푸는데 적합한 기법이다. 또한 범용성이 좋아 다양한 문제에 적용이 용이하다는 장점을 갖는다[7,8]. 유전자 알고리즘에서 출현할 수 있는 염색체들의 개수는 모형 안의 유전자의 개수에 따라 지수 승으로 증가한다. 따라서 연산을 하는 데 상당한 시간이 필요하기 때문에 게임 AI에 실시간으로 적용하기에는 어려움이 따른다[10].

신경망은 단순성을 유지하면서도 기계 학습의 장점들을 제공하는 기법이다. 사람의 두뇌가 어떻게 작동하는지를 연구함으로써 인식과 지능에 대한 모델을 만든 것이다. 신경망의 큰 특징은 학습 능력으로, 규칙을 정해주지 않아도 학습을 통해 지속적으로 더 나은 행동을 찾아 낼 수 있는 것이다[9]. 신경망의 성능은 입력 층, 출력 층간 연결가중치의 전체적인 상태에 따라 결정된다. 따라서 성능을 높이려면 연결이 많아져야 한다. 따라서 게임에 실시간으로 적용하기에는 시간이 너무 오래 걸릴 수 있다[3,10].

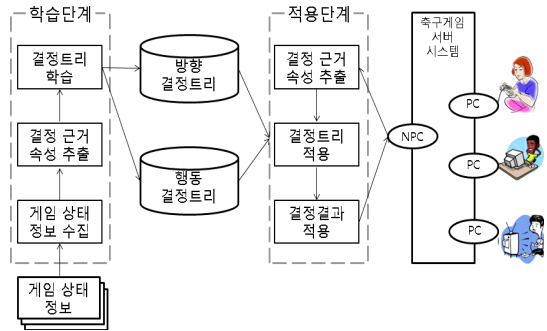
결정트리는 몇 가지 입력 속성에 기초하여 예측 결과를 반환하는 기법이다. 결정트리는 분류나 예측의 근거가 명확하며 어떠한 속성들이 결과에 결정적인 영향을 주는 가를 쉽게 파악할 수 있다. 영향력이 큰 속성을 찾는 학습시간은 다소 걸리지만, 일단 트리가 구축 되면 결정 결과를 찾는 속도가 빠르다. 그리고 학습을 통해 결정트리를 구축하므로 유지, 보수와 확장이 용이하다[11,12,13]. 그러나 주어진 결정 근거 속성이 단순하면 정확도가 낮을 수 있으므로, 결정 근거 속성을 정교하게 정의해야 한다.

이러한 점을 고려하여, 본 논문에서는 NPC가 실제 사용자와 보다 유사하게 게임을 진행할 수 있도록, 게임의 현재 상태 정보뿐만 아니라 NPC와 선수들의 관계 정보 및 과거 상태 정보도 함께 활용한다. 그리고 FSM은 성능개선 및 확장이 어렵다는 문제점을 고려하여 제안하는 방법은 인공지능 학습기법을 사용한다. 그리고 게임에 실시간으로 적용하기에는 퍼지논리, 유전자, 신경망 알고

리즘의 적용속도가 다소 느릴 수 있으므로, 제안하는 방법은 결정트리 학습 알고리즘을 이용한다. 다음 장에서는 제안하는 축구 게임 서버 NPC 제어를 위한 학습 및 적용방법에 대해 살펴본다.

3. 결정트리를 이용한 축구 게임

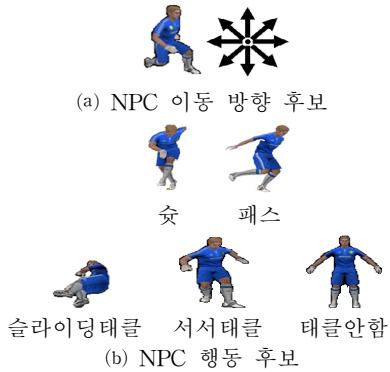
본 논문에서 제안하는 방법은 [그림 1]과 같이 학습 단계와 적용 단계로 나눌 수 있다. 학습 단계에서는 먼저 숙련된 사용자에게 실제 게임을 수행하도록 하여 게임 상태 정보를 저장한다. 저장된 게임 상태 정보에서 결정 근거 속성과 결정 결과를 추출하여 자동으로 결정트리를 학습한다[12]. 한편, 축구 게임 서버 시스템에서 현재 게임 상태를 제안하는 방법의 적용 단계에 전달하면, 게임의 현재 상태에서 결정 근거 속성을 추출하고 결정트리의 결정 결과를 게임에 적용한다. 이해를 돕기 위해 제안하는 방법의 적용 단계를 먼저 설명하고, 이를 어떻게 학습했는지를 설명한다.



[그림 1] 시스템 구성도

3.1 적용단계

제안하는 방법을 적용하는 축구 게임 서버 시스템에서는 사용자에게 캐릭터의 이동방향 및 행동 정보를 키보드와 마우스로 입력받는다. 축구 게임 서버 시스템에서 캐릭터의 이동방향은 [그림 2]의 (a)와 같이 동, 북동, 북, 북서, 서, 남서, 남, 남동,



[그림 2] 게임화면과 이동방향 및 행동 예

이동방향 없음의 9가지 경우가 가능하다. 캐릭터의 행동정보는 [그림 2]의 (b)와 같이 슛, 패스, 서서태클, 슬라이딩태클, 태클안함의 5가지 경우가 가능하다.

따라서 사용자는 [그림 2]의 (c)처럼 공을 가진 상대 공격수가 NPC 가까이에서 있고, NPC가 태클에 실패해도 도와줄 수 있는 같은 팀 선수가 주위에 있다면, [그림 2]의 (d)처럼 공을 빼앗기 위해 슬라이딩 태클을 시도할 수 있다.

동일한 게임 상태에 대해, 제안하는 방법은 다음과 같은 순서로 NPC를 제어한다. 먼저, 축구 게임 서버 시스템에서는 [그림 2]의 (c)와 같은 현재 게임 상태를 [그림 3]의 (a)와 같이 각 캐릭터와 공의 위치를 좌표 값으로 표현하고, 이를 제안하는 방법의 적용단계에 전달한다. 이동 방향이나 행동에 더 적합한 속성을 찾기 위해서, 제안하는 방법은 전달받은 위치정보를 분석하고, [그림 3]의 (b)와 같은 결정 근거 속성을 추출한다.

이때, NPC와 공의 x, y좌표는 그 자체가 유용할 수 있으므로 결정 근거 속성으로 그대로 사용하고, NPC와 공과의 거리도 중요하므로 분석하여 결정 근거 속성에 추가한다. 이전 이동방향이나 이전 행동 값도 중요하므로 포함한다. 한편, 같은 팀

NPC 위치	공의 위치	같은팀 선수A 위치	상대팀 선수A 위치	같은팀 선수B 위치	상대팀 선수B 위치	...
(-14,20)	(-10,20)	(-19,-25)	(-10,20)	(124,32)	(-10,20)	...

(a) 현재 게임 상태

NPC 위치	공의 위치	NPC와 공의 거리	NPC 이전 이동방향	NPC 이전 행동	NPC 주변 같은팀 선수 수	NPC 주변 상대팀 선수 수	공 주변 같은팀 선수 수	공 주변 상대팀 선수 수
(-14,20)	(-10,20)	4	동쪽	태클 안함	2	2	2	2

(b) 결정 근거 속성

NPC와 공의 거리 < 10 :	
NPC의 이전 이동 방향이 북쪽 :	↑
NPC의 이전 이동 방향이 동쪽 :	→
NPC의 이전 이동 방향이 서쪽 :	←
NPC의 이전 이동 방향이 남쪽 :	↓
NPC의 이전 이동 방향이 북동쪽 :	↗
NPC의 이전 이동 방향이 북서쪽 :	↖
NPC의 이전 이동 방향이 남동쪽 :	↘
NPC의 이전 이동 방향이 남서쪽 :	↙
...	

(c) 이동 방향 결정트리 적용 예

NPC와 공의 거리 < 10 :	
NPC주변 상대팀 선수 수 <= 2 :	
NPC주변 같은 팀 선수 수 >= 2 :	슬라이딩태클
NPC주변 같은 팀 선수 수 < 2 :	서서태클
NPC주변 상대팀 선수 수 > 2 :	태클안함
...	

(d) 행동 결정트리 적용 예

[그림 3] 결정트리 적용 예

선수들과 상대팀 선수들의 위치정보 자체 보다는 NPC나 공 주변에 있는 각 팀의 선수 수가 이동방향이나 행동에 더 큰 영향을 끼칠 수 있으므로 이를 분석하여 결정 근거 속성에 추가한다. 이렇게 추출한 결정 근거 속성 값을 [그림 3]의 (c)와 (d)와 같은 결정트리에 적용하여 결정 결과를 추론한다. 마지막으로, 추론한 결정 결과를 축구 게임 서버 시스템에 전달하고, 실제로 이를 바탕으로 NPC를 제어한다.

예를 들어, [그림 2]의 (c)와 [그림 3]의 (a)의 현재 게임 상태는 NPC 가까이 공이 있고 NPC 주변에 같은팀 선수 2명과 상대팀 선수 2명이 있다는 것을 나타낸다. NPC가 태클을 하지 않으면서 동쪽으로 이동 중 이었다고 가정하면, [그림 3]의 (b)와 같은 결정 근거 속성 값을 추출할 수 있다. 이를 [그림 3]의 (c) 이동방향 결정트리에 적용하면, 제안하는 방법은 NPC와 공의 거리가 10 미만으로 매우 가깝고 이전에 동쪽으로 이동 중이었으므로 계속 동쪽으로 이동하라고 결정한다. 그리고 [그림 3]의 (d) 행동결정트리에 적용하면, 제안하는 방법은 NPC와 공의 거리가 10 미만으로 매우 가깝고 NPC 주변 상대팀 선수가 2명이므로 NPC 주변 같은 팀 선수가 2명 이므로 슬라이딩 태클을 시도한다. 이러한 결정 결과를 바탕으로 축

구 게임 서버 시스템에서는 [그림 2]의 (d)와 같이 동쪽방향으로 슬라이딩 태클을 시도한다.

3.2 학습단계

결정트리의 학습 단계는 [그림 1]에서 보듯이 게임의 현재 상태를 포함하고 있는 게임 상태 정보를 수집하는 단계, 현재 게임 상태 정보로부터 결정 근거 속성 값을 추출하는 단계, 그리고 추출된 결정 근거 속성 값으로부터 결정트리를 학습하는 단계로 진행된다.

첫째, 게임 상태 정보를 수집하는 단계에서는 실제 축구에 대한 경험이 많은 사용자들을 대상으로 10:10 축구 게임을 실제로 진행하여 캐릭터의 행동을 게임 상태 정보에 저장한다. [그림 4]는 게임 상태 정보를 통해 얻은 학습 집합의 일부를 보여준다. 본 논문에서는 결정트리를 이용하여 수비 NPC의 움직임을 제어하는 것이 목적이다. 따라서 한 번의 단위 시간 당 입력된 사용자들의 움직임 중 수비 팀 선수들의 움직임만을 분석하여 사용한다.

둘째, 결정 근거 속성 추출하는 단계에서는 수집된 게임 상태 정보에서 NPC 제어에 영향력이 있는 의미 있는 속성을 찾아낸다. 즉, 공이나 선수

NPC 위치	공의 위치	같은팀 선수A 위치	상대팀 선수A 위치	같은팀 선수B 위치	...	NPC 위치	공의 위치	NPC와 공의 거리	NPC 이전 이동방향	NPC 이전 행동	NPC 주변 같은팀 선수 수	NPC 주변 상대팀 선수 수	공 주변 같은팀 선수 수	공 주변 상대팀 선수 수	이동 방향	행동
(-14,20)	(-10,20)	(-19,-25)	(-10,20)	(124,32)		(-14,20)	(-10,20)	4	동쪽	태클 안함	2	2	2	2	→	슬라이딩 태클
(10,90)	(0,-20)	(-24,32)	(-20,40)	(10,-90)		(10,90)	(0,-20)	110	북쪽	태클 없음	1	3	1	1	↑	태클 없음
(4,-20)	(10,100)	(70,-93)	(14,56)	(25,10)		(4,-20)	(10,100)	120	남쪽	태클 없음	0	1	2	3	↘	태클 없음
(200,-22)	(200,-20)	(-20,74)	(200,-24)	(183,25)		(0,-22)	(-100,-20)	100	남쪽	서서태클	3	4	4	6	↓	서서태클
(-20,-10)	(10,-20)	(-13,48)	(500,-90)	(0,30)		(-20,-10)	(10,-20)	31	서남쪽	태클 없음	0	2	6	3	↙	태클 없음
...						...										

(a) 현재 게임 상태

(b) 결정 근거 속성 추출 결과

(c) 결정결과

[그림 4] 학습 집합의 예

들의 단순한 위치정보에서 NPC 주변의 상대팀 선수 수나 공 주변의 같은 팀 선수 수를 분석하여 추출한다.

셋째, 결정트리를 학습하는 단계에서는 이러한 다양한 결정 근거 속성들 중에서 NPC의 이동 방향과 NPC의 행동에 영향을 많이 주는 속성을 바탕으로 결정트리를 생성한다. 이동방향과 행동에 영향을 미치는 속성이 서로 다르기 때문에 [그림 3]의 (c)와 (d)처럼 각각 결정트리를 학습한다.

일반적으로 결정트리 학습 방법은 먼저 각 결정 근거 속성에 대해 정보 이득량 (Information Gain)을 계산한다. 계산된 값에 따라 학습 집합을 가장 잘 분류하는 최적속성을 선택하고, 이를 바탕으로 학습 집합을 분류한다. 학습 집합에 대해 일관성 있는 가설을 찾을 때까지 최적속성의 선택과 정과 학습 집합의 분류과정을 반복하여 결정트리를 확장해나간다.

4. 실험 및 평가

숙련된 사용자들이 골키퍼를 제외한 10:10 축구 게임을 수행하고 28,950개의 게임 상태 정보를 수집하였다. 게임 상태 정보는 각 선수 및 공의 위치 정보를 바탕으로 사용자가 어떻게 캐릭터를 제어했는지를 포함한다. 수집된 게임 상태 정보의 90%인 26,055개는 결정트리 학습을 위해 사용하고, 10%인 2,895개는 결정트리의 정확도를 분석하기 위해 사용한다. 정확도는 결정트리에 적용한 결정 결과와 실제로 사용자가 캐릭터를 제어한 결과가 일치하는 비율로 평가한다. 또한 축구 게임서버 시스템에 실시간 적용이 가능한지 확인하기 위해 하나의 패턴을 입력하였을 때 결과가 나오는 시간을 intel(R) core(TM) i7 920 cpu, 6기가 램, Windows XP 32bit 환경에서 측정하여 평균값으로 계산하였다.

정확도를 평가하는 실험에서는 [표 1]과 같이 11개의 결정 근거 속성을 사용한다. 이들은 NPC

[표 1] 중요 속성들의 상세설명

		결정 근거 속성 설명
위치 정보	nx	npc의 x좌표
	ny	npc의 y좌표
	bx	공(ball)의 x좌표
	by	공(ball)의 y좌표
	nb	npc와 공(ball)과의 거리
선수들 연관 정보	bo	공(ball) 주변 상대팀(opponent team) 선수 수
	bm	공(ball) 주변 같은팀(my team) 선수 수
	no	npc 주변 상대팀(opponent team) 선수 수
	nm	npc 주변 같은팀(my team) 선수 수
과거 상태 정보	pd	npc의 이전 방향(previous direction)
	pa	npc의 이전 행동(previous action)

와 공의 위치정보, 이전, 선수들 연관정보로 분류할 수 있다. 전체 속성 중 정확도를 가장 높일 수 있는 속성을 하나씩 추가하면서, [표 2]와 같이 정확도의 변화추이를 살펴보았다. 결정 근거 속성 전혀 없이 이동방향을 가장 잘 추론할 수 있는 방법은 항상 “이동방향 없음”을 선택하는 것이다. 9가지 이동방향 중 “이동방향 없음”은 [표 2]의 ‘없음’처럼 전체 실험집합에서 33.71%로 가장 많이 나타났다.

실험집합에서 정확도가 가장 높은 속성은 “NPC의 이전방향(pd)”으로 64.28%의 정확도를 보였다. 이는 연속적으로 움직임이 이루어지는 축구 게임의 특성상 캐릭터가 동쪽방향으로 이동 중 이라면 한동안 계속 동쪽방향으로 움직이기 때문이다. “NPC의 이전방향(pd)”에 “공 주변 상대팀 선수 (bo)”를 함께 고려할 때 정확도가 0.11%로 가장 많이 증가하여 64.39%가 되었다. 이는 주변에 상대팀이 없으면 방향을 바꾸지 않고 계속 이동하지만, 상대팀이 나타나면 이동방향을 바꾸기 때문이다. 그리고 이 두 속성에 “공의 x좌표(bx)”를 추가하면 정확도가 0.24%가 향상되어 64.63%가 되었다. 이는 캐릭터는 공이 있는 위치로 이동하려는

경향이 있기 때문이다.

[표 2] NPC 이동 방향의 정확도

	학습		실험	
	정확도	증감	정확도	증감
없음	33.46	-	33.71	-
pd	67.65	-	64.28	-
pd, bo	67.70	△0.05	64.39	△0.11
pd, bo, bx	68.55	△0.85	64.63	△0.24
pd, bo, bx, bm	70.40	△1.85	61.90	▽2.73
pd, bo, bx, bm, by	72.47	△2.07	59.62	▽2.28
pd, bo, bx, bm, by, nm	75.34	△2.87	56.89	▽2.73
pd, bo, bx, bm, by, no, nm	77.03	△1.69	54.89	▽2
pd, bo, bx, bm, by, no, nm, ny	78.55	△1.52	54.30	▽0.59
pd, bo, bx, bm, by, no, nm, ny, pa	78.71	△0.16	53.33	▽0.97
pd, bo, bx, bm, by, no, nm, ny, pa, nb	81.27	△2.56	52.92	▽0.41
pd,bo,bx, bm,by, no, nm,ny,pa, nb, nx	81.87	△0.6	49.22	▽3.7

네 가지 이상의 속성을 사용할 경우 학습 집합의 정확도가 증가하지만 실험 집단에서의 정확도는 오히려 감소되었다. 이는 결정 근거 속성을 충분히 활용할수록 학습 집합의 정확도는 개선되지만, 실험집합은 자료부족 문제로 인해 정확도가 떨어지기 때문이다.

한편, [표 3]은 행동결정 정확도를 가장 높일 수 있는 속성을 하나씩 추가할 때 정확도가 어떻게 변화하는지 나타낸다. 결정근거 전혀 없이 행동을 가장 잘 추론할 수 있는 방법은 항상 ‘태클 없음’을 선택하는 것이다. 슬라이딩 태클, 서서태클, 태클 없음의 세 가지 수비 행동 중 태클 없음은 전체 실험 집합에서 98.17% 거의 대부분을 차지하였다. 결정근거 속성을 추가하여도 대부분의 경우 98.17%의 정확도를 넘지 못하였다. 단, NPC와 공

[표 3] NPC 행동의 정확도

	학습		실험	
	정확도	증감	정확도	증감
없음	98.13	-	98.17	-
nb	98.13	-	98.17	-
nb, no	98.13	△0	98.17	-
nb, no, pd	98.15	△0.02	98.20	△0.03
nb, pd, no, bm	98.14	▽0.01	98.17	▽0.03
nb, pd, no, bm, pa	98.14	△0.0	98.17	-
nb, pd, no, bm, pa, ny	98.27	△0.13	97.96	▽0.21
nb, pd, no, bm, pa, ny, bo	98.32	△0.05	97.96	-
nb,pd,no,bm,pa,ny,bo,bx	98.38	△0.06	97.89	▽0.07
nb,pd,no,bm,pa,ny,bo,bx,nx	98.44	△0.06	97.89	-
nb,pd,no,bm,pa,ny,bo,bx,nx,by	98.48	△0.04	97.69	▽0.2
nb,pd,no,bm,pa,ny,bo,bx,nx,by,nm	98.50	△0.02	97.75	△0.06

과의 거리(nb), NPC 주변 상대팀 선수 수(no), 이전 이동방향(pd)을 고려할 경우 정확도가 0.03%증가하여 98.20%가 되었다. 이는 공을 가지고 있는 상대팀이 주변에 있을 때 태클을 시도하려는 경향이 있기 때문이다.

마지막으로 기존방법과 제안하는 방법의 성능을 비교분석하기 위해서, [표 4]와 같이 동일한 실험 환경에 대해 (Letia98)[13]을 적용하였다. (Letia98)의 방법은 선수들 간의 거리와 각도만을 이용하여 결정트리를 생성하였다. [표 4]에서 제안방법은 [표 2]와 [표 3]에서 가장 높은 정확도를 보인 3가지 속성을 이용한 경우를 나타낸다.

[표 4] 기존방법과의 성능 비교

		(Letia98)	제안방법	
이동 방향	정확도	학습	88.53	68.55
		실험	41.52	64.63
	속도	학습	0.098	0.001
		실험	0.077	0.002
행동	정확도	학습	98.48	98.15
		실험	98.55	98.20
	속도	학습	0.020	0.001
		실험	0.020	0.003

(Letia98)의 경우는 이동방향의 학습 집단에서 매우 높은 정확도를 보인 반면 실험집합에서는 매우 낮게 나타났다. 이는 결정 근거 속성이 너무 많아 자료부족문제가 심각하게 나타났기 때문이다. 반면, (Letia98)은 행동의 실험집합은 매우 높은 정확도를 보였다. 태클을 시도할 때 NPC와 공과의 거리, NPC와 상대팀 선수와의 거리정보가 중요한데, (Letia98)은 이들을 모두 활용할 수 있지만 제안하는 방법은 NPC와 공과의 거리만 고려할 수 있기 때문이다. 또한 (Letia98)에 비해 제안방법의 처리속도가 몇 십 배 이상 빠르다. 이는 제안방법에서 사용하는 결정 근거 속성의 수가 (Letia98)에 비해 작고, 주변 선수 수의 속성 값은 10가지 이하로 한정적이지만, (Letia98)에서 사용하는 거리정보는 실수 값으로 매우 다양하게 나타날 수 있다.

5. 결 론

본 논문에서는 결정트리 학습방법을 이용하여 축구 게임에서 수비 NPC를 제어하는 방법을 제안하였다. NPC가 적극적으로 공격할 경우 사용자들의 흥미를 떨어뜨릴 수 있으므로, NPC는 수비역할에 충실하도록 설계하였다. 제안하는 방법의 특징은 다음과 같다.

첫째, 축구 게임 시스템에서 실시간으로 NPC를

제어할 수 있다. 결정트리를 학습하는 과정은 다소 시간이 걸리지만, 학습된 결정트리를 게임에 적용하여 NPC의 행동을 결정하는 시간은 0.001 ~ 0.003ms 이었다. 제안하는 방법을 축구 게임 서버 시스템에 적용한 결과 실시간으로 움직이는 것을 확인하였다.

둘째, 제안하는 방법은 실제 10:10 축구 게임에서 단순한 위치정보뿐만 아니라 선수들 간의 관계 정보, 이전상태 정보도 학습에 활용하였다. 일반적으로 사용자는 한 방향으로 계속 이동하려는 경향이 있으므로, 이동방향을 결정할 때에는 이전 이동방향 속성이 큰 영향을 끼친다. 제안하는 방법은 각 선수들 간의 거리나 각도정보를 이용한다 (Letia98)보다 이동방향을 결정할 때 높은 정확도를 보여주었다.

셋째, 제안하는 방법은 결정트리 학습 방법을 이용하므로, 유지보수 및 확장이 용이하다. 일단 결정트리의 속성 값들이 정해지게 되면, 숙련된 사용자들의 게임 상태 정보를 추가하여 좀 더 정확한 학습결과를 생성할 수 있다.

향후 추가 연구를 통하여 수비뿐만 아니라 공격도 할 수 있도록 결정트리를 학습 하고자 한다. 즉 슈트와 패스를 포함한 축구 게임에서 일어날 수 있는 모든 행동을 학습하여 축구 게임 NPC를 위한 결정트리를 구축 할 수 있다. 또한 팀 전술을 위한 인공지능을 추가하여 단순히 NPC 개개인의 움직임만이 아닌 진형을 유지하는 인공지능을 구현하고자 한다.

참고문헌

- [1] 게임트릭스, <http://www.gametrics.com/>
- [2] 임차섭, 김태용, “게임 NPC지능 개발을 위한 부하분산과 그룹 행동을 지원하는 유연한 플랫폼 구조”, 전자공학회 논문지, 43권, CI편 제 2호, 2006년 3월
- [3] Mat Buckland, “Programming Game AI by Example”, Wordware Publications, 2005.
- [4] 양정모, 조경은, 엄기현, “적응형 NPC 생성을

- 위한 FSM의 동적 활용 방안”, 한국멀티미디어 학회지 제11권 제9호, pp.1258-1266
- [5] 문성원, 조형제, “퍼지 확장 기법을 이용한 온라인 게임에 적합한 지능적 AI기법”, 게임학회 논문집, 8권, 3호, pp.77-85, 2008
- [6] Mark Deloura의 공저, “Game Program Gems (Mason McCuskey-비디오 게임을 위한 퍼지 논리)”, pp.416-428, 정보문화사, 2001
- [7] 조성형, 강신진, “유전자 알고리즘을 사용한 타워 디펜스 공격대의 자동 구성 기법”, 게임학회 논문집, 11권, 2호, pp.19-28, 2011
- [8] 강신진, 신승호, 조성현, “유전자 알고리즘을 사용한 게임 레벨 디자인 기법”, 컴퓨터그래픽스학회논문지 제15권 제4호, pp.13-21
- [9] 권오광, 박종구, “유전 알고리즘과 신경망을 이용한 RPG 게임 캐릭터의 제어”, 한국게임학회지 제6권 제2호 13-22
- [10] Steve Rabin의 47인, “AI Game Programming Wisdom(Alex J. Champandard - 잘 알려지지 않은 신경망 기법)”, pp.855-868, 정보문화사, 2003
- [11] 박소영, 곽용재, 정후중, 황영숙, 임해장, “한국어 구문분석의 효율성을 개선하기 위한 구문제약규칙의 학습”, 정보과학회논문지, 29권, 9호, pp.755-765, 2002
- [12] J. Ross Quinlan, “C4.5: Programs for Machine Learning”, Morgan Kaufmann Publishers, 1993.
- [13] Ioan Alfred Letia, Marius Joldos, Calin Cenan, Diana Zaiu and Alina Andreica, “Decision trees and rule induction in simulated soccer agents”, Lecture Notes in Computer Science, 1998, Volume 1456, Collective Robotics, pp. 110-122



조 달 호 (Cho, Dal-ho)

2003.2 상명대학교 소프트웨어학과 학사
2005.2 상명대학교 대학원 컴퓨터학과 석사
2005.3-현재 상명대학교 컴퓨터과학과 박사 재학 중

관심분야 : 게임 프로그래밍, 인공지능, 영상처리



이 대 응 (Rhee, DaeWoong)

1986.2 서울대학교 계산통계학과 이학사
1988.2 서울대학교 대학원 계산통계학과 이학석사
1996.8 서울대학교 대학원 계산통계학과 이학박사
1990.4-현재 상명대학교 소프트웨어대학
디지털미디어학부 교수

2001.9-2008.2 상명대학교 디지털미디어 대학원 원장 역임
2003.1-2004.1 상명대학교 소프트웨어 대학 학장 역임
2006.1-2008.2 상명대학교 소프트웨어 대학 학장 역임
2008.3-2011.10 상명대학교 일반대학원 원장 역임

관심분야 : 게임 기획, CT, 게임 프로그래밍



이 용 호 (Lee, Yongho)

2011.2 상명대학교 소프트웨어대학 컴퓨터과학과 학사
2011.3-현재 상명대학교 컴퓨터과학과 석사 재학 중

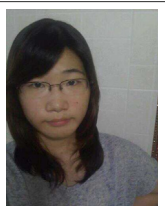
관심분야 : 게임 프로그래밍, 네트워크 트래픽 관리



박 소 영 (Park, So-Young)

1997.2 상명대학교 전자계산학과 (이학사)
1999.8 고려대학교 컴퓨터학과 (이학석사)
2005.2 고려대학교 컴퓨터학과 (이학박사)
2007.3-현재 상명대학교 디지털미디어학부 조교수

관심분야 : 자연어처리, 기계학습, 텍스트마이닝



김 진 형 (Kim, JinHyung)

2009.3-현재 상명대학교 디지털미디어학부 학사 재학 중

관심분야 : 게임 프로그래밍, 인공지능, 기계학습