

# 보이드들의 생태계 행동 모델링을 위한 퍼지 플로킹 기법

권일경<sup>o</sup> 이상용

공주대학교 컴퓨터공학과

{psent95<sup>o</sup>, sylee}@kongju.ac.kr

## Fuzzy flocking method for boid's ecosystem behavior modeling

Ilkyoung Kwon<sup>o</sup> Sangyong Lee

Dept. of Computer Engineering, Kongju National University

### 요 약

게임 세계에 존재하는 수많은 보이드들의 지능적인 집단 행동을 모델링하기 위한 방법으로 플로킹 기법이 많이 사용되고 있다. 특히 생태계에 존재하는 객체들의 행동을 재현하기 위한 연구가 활발하게 진행되고 있는 실정이다. 따라서 본 연구에서는 생태계에서 흔히 볼 수 있는 먹고 먹히는 관계를 퍼지 논리를 통해 게임의 보이드 행동을 모델링하고 구현한다.

## 1. 서론

온라인 게임이 발전하자 많은 사람들은 인공지능은 게임에서 더 이상 중요하지 않은 분야라고 생각하기 시작하였다. 그러나 오히려 게임이 멀티플레이 성격을 갖추수록 인공지능의 필요성은 더욱더 크게 자리잡기 시작하였다. 특히 게임에서 인간과 상호 작용하는 NPC 및 여러 보이드(boid)들의 움직임은 지능적인 측면을 많이 갖추고 있으며, 이러한 게임 요소들은 인공 생명이라는 예측 불가능한 인간의 생각 및 행동을 묘사하거나 모방하기 위한 방법을 통해 게임에 적용되기 시작하였다.

본 연구에서는 예측 불가능한 생명체들의 움직임을 모델링하는 플로킹 기법(flocking)을 이용하여 생태계에 존재하는 무리들의 먹고 먹히는 관계를 좀더 사실적으로 표현하고자 한다. 이를 통해 현실적인 보이드들의 행동 모델링이 가능하고 시간이 지남에 따라 먹이 사슬이 점차 파괴되는 기존의 문제점을 효과적으로 개선하고자 한다.

## 2. 관련 연구

### 2.1 플로킹(flocking)

플로킹은 일종의 무리를 모델링하는 기법으로 1987년 Craig Reynolds가 처음으로 발표하였다. 플로킹의 아버지로 불리는 Reynolds는 그림 1의 세가지의 간단한 기본 규칙들(조타 행동들)을 이용하여 보이드(boid)라고 불리는 존재들이 생물과 비슷한 집단 행동을 취하게 됨을 보여주었다[1].



그림 1 분리, 정렬, 응집 규칙

- 분리(Separation) : 주변 보이드들과 충돌하지 않도록 방향을 돌림
- 정렬(Alignment) : 주변 보이드들과 같은 방향을 가리키도록 함
- 응집(Cohesion) : 주변 보이드들과 평균 위치 쪽으로 방향을 돌림
- 회피(Avoidance) : 주변의 장애물이나 적과 충돌하는 것을 피함

회피(Avoidance) 규칙은 새롭게 추가된 규칙으로 그림 2와 같이 지형 장애물 및 천적 무리로 인해 충돌이 발생할 경우 이를 피하기 위한 방법이다. 충돌을 피하기 위해서는 군집들이 자연스럽게 작은 군집들로 나뉘며 이렇게 떨어져나간 보이드들은 서서히 다른 주변 보이드들과 다른 군집을 형성하게 된다[2].

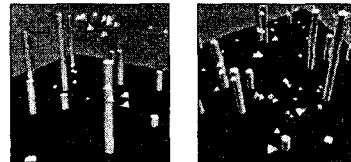


그림 2 장애물 지형의 플로킹

플로킹은 대부분의 게임의 행동 구현 방식과는 달리 이동 갭 사이에서 정보를 유지할 필요가 없는 알고리즘을 사용함으로써 자신에 대한 정보를 계속 지니고 다니게 하는 경우보다 메모리를 덜 소비하게 만들고 좀더 실시간으로 반응하도록 만들 수 있다. 플로킹의 이러한 특징은 인공지능 구현에 많은 자원을 할당하기 어려운 게임 장르인 RTS(Real Time Strategy)의 유닛 대형 문제와 자연 유닛 등의 사실적 행동 구현에 유용하게 쓰인다.

### 2.2 퍼지이론(fuzzy theory)

인간과 비슷하게 생각하고, 행동하는 컴퓨터를 만들고자 하는 인공 지능 연구가 활발하게 진행되고 있다. 컴퓨터가 인공지능을 가지고 인간이 원하는 바를 제대로 수행하기 위해서는 인간이 사용하는 숫자는 물론이고 애매한 표현도 처리할 수 있어야 한다. 이러한 인간의 애매한 표현을 처리할 수 있는 이론적인 바탕을 제공하는 것이 바로 퍼지 이론이다. 퍼지 이론은 애매하게 표현된 자료를 유용한 자료로 만들기 위해서, 퍼지 집합, 퍼지 논리, 퍼지 관계등의 개념을 포함하고 있으며 수학적인 계산 방법도 잘 개발되어 있다[3].

예를 들어, 퍼지는 NPC의 감정을 흥내내는 데 사용할 수 있다. 게임의 NPC가 단순히 "배고픔" 과 "배부름"의 느낌을 가지는 것이 아니라, "매우 배고픔", "조금 배부름" 등의 애매한 감정을 가질 수 있게 한다.

3. 생태계 행동 모델링

3.1 생존 규칙

단순한 형태의 보이드 행동 모델링을 좀더 흥미롭게 하기 위하여 플로킹 기법에 새로운 규칙으로 "생존"을 추가하였다. 생존 규칙은 먹이가 보이면 먹이가 보이는 쪽으로 향하고 천적 무리가 보이면 도망간다는 단순한 원리이다. 특히 회피 규칙과 다르게 천적 관계의 움직임은 장애물을 만날 경우 효과적으로 작동한다.

3.2 다양한 보이드들

생태계는 다양한 종류의 생물체들이 물, 공기, 토양을 근거로 물과 산소 그밖에 영양분을 섭취하고 환경에 적응하며 살아가는 생명 유지 체계를 뜻한다. 생태계를 유지하기 위한 조건 중 가장 중요한 것이 환경과 먹이사슬이며 이것이 깨질 경우 더 이상의 생태계를 유지하는 것은 힘들다.

먹이 사슬은 다양한 종류의 생물체들이 서로 쫓고 먹고 먹히며 그 개체 수를 적당하게 유지하는 것이 가장 중요하다. 본 연구에서는 생태계 구조의 최상위에 속하는 포식자1과 중간의 포식자2 및 최하위에 속하는 먹이 보이드를 설정하여 생태계 먹이 사슬을 표현한다. 이들간의 먹이 섭취 유형이나 이동 속도, 시야는 표 1과 같다. 먹이의 경우 먹이 관계에서 최하위에 속하며 그 개체수는 임의 재생 방법을 통해 조절된다.

표 1 먹이사슬 모델링

	포식자1	포식자2	먹이
이동속도	평균속도이상	평균속도	평균속도이하
시야	평균시야이상	평균시야	평균시야이하
먹이	포식자2	먹이	먹히기만 함

3.3 먹고 먹히기

보이드들 간의 먹고 먹히는 관계를 만들기 위해 "식욕"에 대한 제어 변수가 필요하다. 특히 하나의 보이드가 수십마리의 먹이들을 순식간에 먹어치울 경우 전혀 사실적이지 못하다. 이러한 문제를 해결하기 위해 보이드들의 "식욕" 욕구 수준을 나타내는 멤버 변수를 추가하여 일정 시간이 흐르면 감소하는 형태로 구현한다. 식욕은 0과 100사이의 값을 가지며 0일 때 식욕이 가장 왕성(가장 배가 고프는 상태)해진다. 특히 포식자 보이드들은 식욕과 거리변화, 보이드 속도를 이용하여 모델링한다.

먹이 보이드들은 포식자로부터 잘 도망칠 수 있어야한다. 먹이가 포식자로부터 잘 도망치기 위해서는 포식자의 속도와 움직임에 따른 거리 변화에 민감하게 동작할 수 있어야한다. 그러기 위해 포식자와의 거리와 거리변화, 속도를 이용하여 모델링한다. 그러나 아주 빠르고 굵주린 포식자로부터 완벽하게 회피하는 경우는 매우 드물 것이다.

4. 퍼지 제어

4.1 포식자 행동 시나리오 모델링

생태계 행동 모델링을 위한 포식자 퍼지 논리 모델링을 위하여 다음의 두 가지 사항(언어적 변수)을 고려한다.

- 보이드와 먹이 사이의 "거리 변화(Distance)"
- 포식자 보이드의 "식욕(Appetite)" 정도

앞서 고려된 사항에 의해 다음과 같은 포식자가 가질 수 있는 퍼지 규칙을 얻을 수 있다. "식욕"과 "거리 변화"에 의한 포식

자의 "속도(Speed)"라는 행동 퍼지 집합이 정의된다.

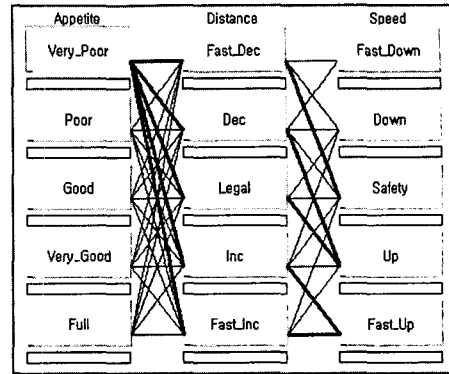


그림 3 "식욕"과 "거리변화", "속도"간의 규칙

표 2, 표 3, 표 4는 그림 3의 규칙들을 설명하는 변수들로부터 개별적인 퍼지 집합을 만들어낸 것이다.

표 2 "식욕" 퍼지 집합 정의

"식욕"	퍼지 집합
Very_Poor	먹이 75~100 정도 섭취 가능
Poor	먹이 15~45 정도 섭취 가능
Good	먹이 35~65 정도 섭취 가능
Very_Good	먹이 55~85 정도 섭취 가능
Full	먹이 0~25 정도 섭취 가능

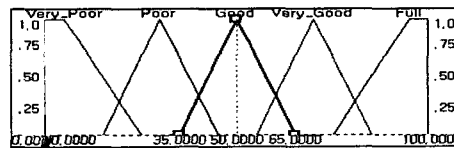


표 3 "거리변화" 퍼지 집합 정의

"거리 변화"	퍼지 집합
Fast_Dec	보이드 속도 반(-) 수준
Dec	0보다 작음
Legal	0 정도
Inc	0보다 큼
Fast_Inc	보이드 속도 반(+) 수준

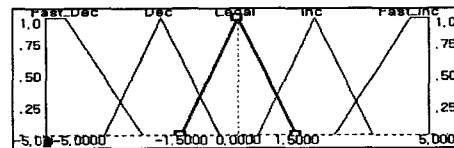
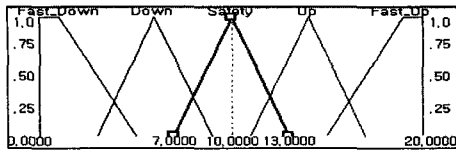


표 4 "속도" 퍼지 집합 정의

"속도"	퍼지 집합
Fast_Down	속도를 반으로
Down	현재속도의 반 정도 감속
Safety	아무것도 행동하지 않음
Up	현재속도의 반 정도 가속
Fast_Up	속도를 두배로



이제 만들어진 하나의 퍼지 제어 시스템은 비퍼지화 방법을 통해 최종적인 결과를 얻을 수 있다. 비퍼지화 방법에는 무게 중심(Center of Gravity)법과 최대 값 평균(Mean of Maximum) 방법이 있으며 본 연구를 통해 두 방법의 결과를 간단하게 비교할 것이다.

#### 4.2 먹이 행동 시나리오 모델링

먹이는 먹히지 않기 위한 퍼지 행동 모델링을 위하여 다음과 같은 두 가지 사항을 고려한다.

- 보이드와 먹이 사이의 "거리"
- 보이드와 먹이 사이의 "거리 변화"

먹이는 포식자가 다가올 경우 더욱 빠르게 회피하여야 할 수 있다. 다음은 포식자로부터 회피하기 위한 규칙(그림 4)이다.

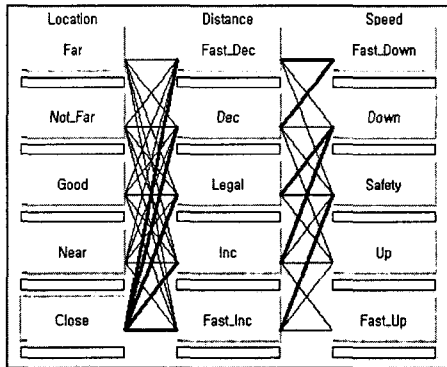


그림 4 "거리"와 "거리변화", "속도"간의 규칙

포식자와 마찬가지로 먹이의 퍼지 제어 모델링이 가능하다. 먹이의 퍼지 제어 모델링이 완성됨으로써 포식자와 먹이 간의 쫓고 쫓기는 관계는 더욱 사실감 있게 표현 가능하다. 특히 먹이에 관심이 없는 포식자로부터 회피에 성공할 가능성을 높일 수 있어 포식자의 탐색을 방지할 수 있다.

#### 5. 실험 방법 및 결과

플로킹 기법의 생명은 속도이다. 이를 위하여 본 연구에서는 이미 비디오 게임과 같은 속도가 중요한 어플리케이션을 위해 최적화된 API인 오픈소스 퍼지 논리 클래스 라이브러리(Free Fuzzy Logic Library : FFLL)를 사용하여 실험하였다[4].

퍼지 조종 언어(Fuzzy Control Language : FCL)을 이용하여 퍼지 논리 모델을 생성하였으며, 그림 5는 포식자의 행동을 하나의 퍼지 모델로 생성하기 위한 FCL을 보여준다. FUZZIFY 블록에 두 개의 입력으로 Appetite와 Distance가 있으며 출력으로 Speed가 있다. RULEBLOCK엔 앞서 생성한 규칙들이 상술되어 있다. 또한 DEFUZZIFY방법을 상술 할 수 있으며 무게 중심법(CoG)과 최대 값 평균법(MoM)을 간단하게 변경할 수 있어 결과값의 비교를 간단하게 할 수 있다.

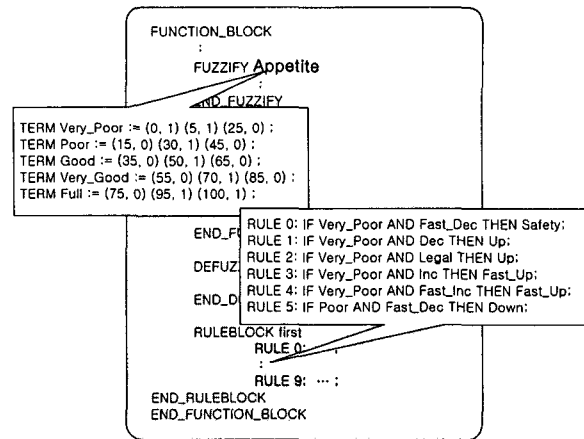


그림 5 포식자의 행동 퍼지 모델 생성을 위한 FCL

API 형태의 FFLL을 사용하여 모델링된 FCL 파일을 임포트하고 모델을 테스트한다. 그림 6은 테스트 프로그램의 간단한 결과 화면이다.

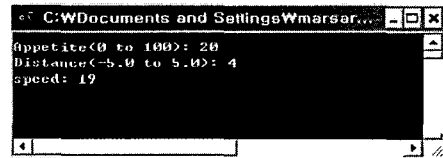


그림 6 모델의 테스트 결과

그림 6의 결과는 포식자의 식욕(20)이 굶주림(표 2 참조)을 의미하며 먹이와의 거리변화가 4(+)로 증가하고 있다. 이때 포식자는 19의 속도(표 4 참조)로 급히 추격하는 것을 의미한다.

#### 6. 결론 및 향후 연구 방향

일반적인 생태계 행동 모델 플로킹 기법의 경우 포식자의 배고픔 정도나 회피를 난수 발생 방법을 통하여 간단히 해결함으로써 생태계의 균형이 깨지는 경우가 빈번히 발생한다. 그러나 퍼지를 이용한 생태계 모델 방법은 먹이가 충분히 유지되도록 스스로 조절 가능한 모델링이 가능하며 적용 가능하다. 게다가 자연스러운 행동 모델링을 위한 보이드의 속도 및 식욕에 대한 균형도 자연스러운 형태로 모델링이 가능함을 알 수 있었다.

현재 대형 유지를 위한 플로킹 기법이 연구되기 시작하였다. 앞으로 효과적인 대형을 유지하며 먹이를 사냥하거나 회피하는 퍼지 모델을 연구할 계획이다.

#### 참고문헌

- [1] Reynolds, C. W., "Flocks, Herds and Schools: A Distributed Behavioral model" Computer Graphics, 21(4), SIGGRAPH '87 Proceedings, pp. 25-34, 1987.
- [2] Woodcock, Steven, "Flocking: A Simple Technique for Simulating Group Behavior" Game Programming Gems, Charles River Media, pp. 305-318, 2000.
- [3] 이상용, "인공지능" 도서출판 창조사, pp. 161-188, 2000.
- [4] Steve Rabin, "AI Game Programming Wisdom" Charles River Media, pp. 90-102.