

조종 행동을 이용한 실시간 대규모 군중 제어 기법

Real-Time Crowds Control using Steering Behavior

박교현, Kyohyeon Park*, 김동문, Dongmoon Kim*,
윤태복, Taebok Yoon*, 이지형, Ji-hyong Lee*
성균관대학교 전자전기컴퓨터공학과



요약 동물이나 곤충, 혹은 사람 같은 군중이 등장하는 장면을 영화나 게임 등에서 종종 볼 수 있다. 이러한 대규모의 군중을 제어하기 위해서는 일반적인 에이전트별 제어 방식이 아닌 군중적인 움직임을 보일 수 있는 특별한 군중 제어가 필요하다. 자연스러운 군중의 제어를 위해선 지능적인 행동을 보여야하며 움직임은 자연스러워야한다. 또한 게임 등의 실시간 환경을 위해선 연산의 속도도 고려해야 한다. 이에 본 논문에서는 조종 행동을 이용한 실시간 대규모 군중 제어 기법을 제안한다.

핵심어: Crowds, Steering Behavior, Flocking, AI, Game



1. 서론

영화, 게임, 시뮬레이션 등에서 사람이나 동물, 곤충 등 많은 수의 군중을 표현하는 경우를 흔히 볼 수 있다. 이 때, 군중의 자연스러운 움직임을 위해서는 단일 에이전트를 제어하는 방식과는 다른 특별한 군중 제어가 필요하게 된다. 단일 에이전트 제어 알고리즘에서는 군중적인 움직임의 특성을 고려하고 있지 않기 때문이다. 또한 게임 등의 실시간으로 동작하는 환경에서는 제어 알고리즘의 연산 속도 또한 중요한 요소이다. 제어하는 군중의 숫자와 그 지능의 수준에 따라 필요한 CPU 자원이 달라진다. 따라서 보다 빠르고 효율적이며 지능적인 군중 제어 방식을 필요로 한다.



그림 1. 군중 에이전트가 등장하는 게임

그동안 다양한 군중 제어 기법들이 연구되어왔다. 그러나 몇 가지 문제점들이 존재한다. 군중 제어 기법으로 가장 잘 알려진 것들 중에 첫 번째는 실시간으로 가장 자연스러운 움직임을 표현할 수 있는 것은 조종 행동(Steering Behavior)이다. 그러나 이 방식은 다루는 개체의 수가 증가할수록 연산이 기하급수적으로 증가한다는 단점을 가지고 있다. 두 번째로 전위장(Potential Field)을 이용하는 방법이 있다. 환경 전체의 이동에 영향을 미치는 요소를 고려한 전위장을 생성하고 그에 따른 진행 방향을 계산하는 방식이다. 그 때문에 개체 수에 비해 적은 연산으로도 제어가 가능하나 이동시 고려해야할 요소가 늘어나게 되면 적절한 전위장의 생성이 어려워진다. 세 번째로 탐색(Search)을 이용하는 방법이 있다. 각 개체의 목표점까지의 경로를 미리 전부 탐색하는 방식이다. 따라서 정확한 움직임을 보일 수 있지만 상당한 양의 연산을 필요로 한다. 또한 예측할 수 없는(예를 들면 유저가 조종하는) 동적인 개체에 의해 예측한 경로가 어긋나게 된다면 재탐색을 필요로 하게 된다. 따라서 실시간 환경에는 적합하지 못하다.

본 논문에서는 조종 행동을 이용하여 보다 자연스럽게 지능적인 움직임을 보이면서도 연산 속도를 개선하여 많은 수의 개체를 다룰 수 있는 군중 제어 기법을 제안한다. 퍼지를 사용하여 상황을 인지하면서 Level Of Detail(LOD) 기법을 사용하여 연산 시간을 단축시켰다. 이 방법은 대규모 개체가

등장하는 게임 혹은 유사한 실시간 환경 어플리케이션에 적용하여 군중을 효과적으로 제어할 수 있다.

2. 관련 연구

Craig Reynolds는 SIGGRAPH 1987에서 새떼나 물고기 떼와 같은 집단행동을 흉내 내기 위한 플로킹(Flocking)을 발표하였다.[1] 이는 다음과 같은 분리, 정렬, 응집의 행동들로 이루어져 있다.

분리 : 주변 에이전트와 충돌하지 않도록 방향을 돌린다.

정렬 : 주변 에이전트와 같은 방향을 가리키도록 한다.

응집 : 주변 에이전트의 평균 방향으로 돌린다.

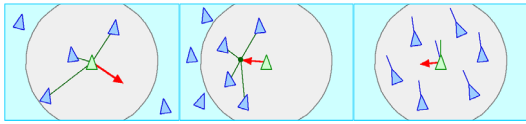


그림 1. Flocking의 3요소. 분리, 응집, 정렬

이후 Game Developer Conference '99에서 플로킹에 추적, 회피, 벽 피하기, 리더 따라가기 등의 행동을 추가하여 이와 같은 행동들을 조종 행동(Steering Behavior)이라고 칭하였다.[2]

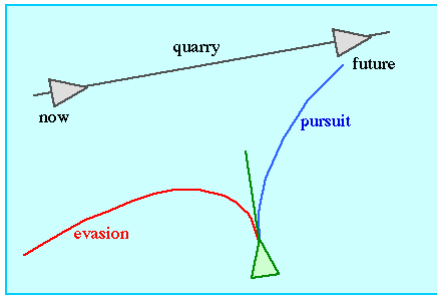


그림 2. 추적과 회피 조타 행동

플로킹의 무게중심 등의 연산 대신 군중 행동에 대한 인간의 지식을 퍼지 규칙을 사용하여 나타내고 이를 이용하여 플로킹과 유사한 군중 행동을 보이도록 접근한 사례도 있다.[3]

플로킹과 비슷하게 별레 무리와 유사한 움직임을 보이는 Swarm 알고리즘이 있다.[4] 이는 플로킹보다 훨씬 단순한 연산으로 이루어져있으나 에이전트간의 충돌은 전혀 고려하지 않는다. 따라서 쥐나 바퀴벌레와 같이 아주 많은 수의 작고 빠르게 움직이는 에이전트의 처리에 사용된다.

플로킹과 같이 각 개체마다 연산을 하는 단점을 가진 제어 알고리즘을 개선하기 위해 계층적인 개념을 도입하여 개체를 개인, 그룹, 군중으로 계층적 분류를 하여 제어하는 ViCrowd도 있다.[5]

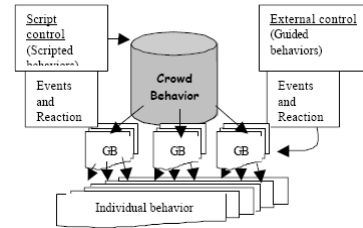


그림 3. ViCrowd의 계층적인 구조

중력장(Potential Field)을 이용한 Continuum Crowds도 있다.[6] 환경 전체를 한번에 계산한다는 장점이 있으나, 여러 요소를 고려하여 처리하지 못한다는 단점이 존재한다.

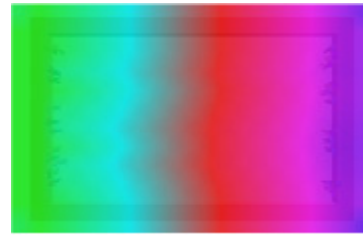


그림 4. Continuum Crowds에서 생성하는 전위장

탐색을 이용하여 이동을 제어하는 방식도 연구되고 있다.[7] 이는 가장 원하는 정답에 가까운 결과를 보여주나 실시간으로 작동하기엔 너무 많은 연산을 필요로 하기에 대규모 군중을 실시간으로 처리하기에는 무리가 있다. 또한 목표점까지의 탐색을 이미 마친 상태에서 동적으로 환경이 변하면 재탐색을 해야 하는 단점도 가지고 있다.

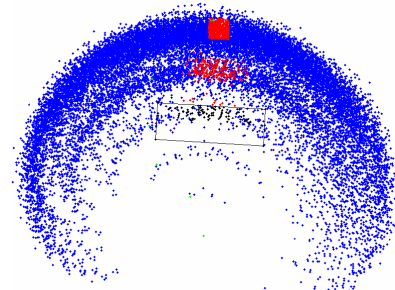


그림 5. 탐색을 통한 제어 방식

3. 제안하는 방법

균중의 지능적이며 빠른 제어를 위해 본 논문이 제안하는 방법은 다음의 세 단계를 거쳐 균중을 제어한다. 첫 번째로는 퍼지 추론을 이용하여 개체마다 자신이 처한 환경을 인지한다. 두 번째로는 인지한 정보를 통해 상태 전이되는 유한 상태 기계를 이용하여 개체 자신이 취할 행동을 결정한다. 지능적인 움직임을 보일 수 있다. 세 번째로는 조종 행동을 이용하여 결정한 행동에 따라 이동한다. 자연스러운 균중의 움직임을 보여줄 수 있는 조종 행동을 이용하여 자연스러우면서도 연산을 줄여서 빠르게 동작한다.

본 논문에서는 다음의 요소로 구성된 가상 환경에서의 균중 제어를 통해 제안하는 대규모 균중의 제어 기법을 설명하도록 하겠다.

- 포식자: 유저에 의해 조종된다. 피포식자를 잡아먹는다.
- 피포식자: AI에 의해 조종된다. 포식자를 두려워하며 가까이 접근하면 도망간다. 주변이 두려움을 느끼면 자신도 두려움을 느낀다.
- 장애물: 개체는 이것을 지나쳐 이동할 수 없다.
- 카메라: 유저의 화면에 보이는 시야범위를 나타낸다.

3.1 퍼지 추론을 이용한 상황 인지

본 논문에서는 여러 가지 환경의 정보를 조합하여 상황을 인지하는 기법으로 퍼지 추론을 사용한다. 퍼지 추론이란 몇 개의 퍼지 규칙으로부터 하나의 다른 근사적인 퍼지 명제를 추론하는 것이다. 예를 들어 직전의 위험도가 높았고 주변의 위험도가 보통 상태이며 포식자와의 거리가 멀다면 현재 피포식자가 느끼는 위험도는 “약간 위험하다” 라는 정도로 표현할 수 있을 것이다. 이를 퍼지적으로 표현함으로써 모호한 경계를 자연스럽게 처리하였다. 퍼지 추론을 사용함으로써 얻을 수 있는 이점은 어떠한 문제에 대한 인간의 지식을 이용해 특별한 공식을 만들지 않아도 입력 값에 따른 결과 값을 추론해 낼 수 있다는 장점이 있다. 따라서 여러 가지 환경 요소로부터 다른 환경 요소 혹은 정보를 얻어낼 수 있다.

본 논문에서는 현재 시간 직전의 위험도, 주변의 위험도, 포식자와의 거리, 그리고 이 세 가지 요소를 이용하여 피포식자가 느끼는 현재 위험도를 퍼지 추론을 이용하여 추론한다. 그림 6은 이러한 요소들을 퍼지 집합으로 나타낸 것이다.

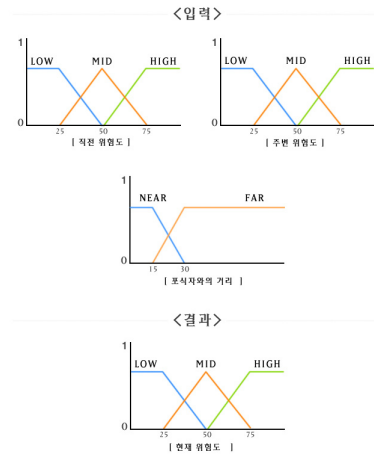


그림 6. 상황 인지를 위해 사용되는 퍼지 집합

피포식자가 느끼는 현재 위험도를 표 1과 같은 퍼지 규칙을 적용하여 현재 상태를 인지하도록 설정하였다.

표 1. 피포식자에 적용된 퍼지 규칙

이전 위험도	주변 위험도	포식자와의 거리	현재 위험도
*	*	Near	High
Low	Low	Far	Low
Low	Mid	Far	Low
Low	High	Far	Mid
Mid	Low	Far	Low
Mid	Mid	Far	Mid
Mid	High	Far	High
High	Low	Far	Mid
High	Mid	Far	Mid
High	High	Far	High

피포식자의 지능적인 행동을 위해 사용된 퍼지 추론은 결과 값을 얻기 위해 많은 양의 연산을 필요로 한다. 이를 줄이기 위하여 입력 값에 따른 출력 값의 매칭 테이블을 사용하여 추론 과정에 사용되는 연산 시간을 감소시켰다.

3.2 유한 상태 기계를 이용한 행동 결정

유한 상태 기계(Finite State Machine, FSM)이란 개체의 상태를 정의하는 것으로 현재 상황의 행동 양식을 제시해준다. 그리고 FSM에서 가장 중요한 부분 중의 하나는 바로 전이이다. 본 논문에서는 퍼지 추론에 의하여 피포식자의 현재 위험도를 계산하고, 그 위험도가 일정 수치 이상이거나 이하

일 때 피포식자의 상태는 다른 상태로 전이된다.

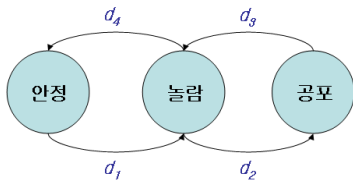


그림 7. 피포식자에 사용된 FSM

피포식자의 상태는 그림 7와 같이 세 가지로 이루어져있다. 상태는 안정과 놀람 그리고 공포로 되어있는데, 안정 상태는 낮은 속도로 무리지어 이동하는 피포식자들의 평화로운 상태를 나타내며, 공포 상태는 포식자의 출현으로 피포식자들이 위협을 느끼고 최대의 속도로 회피하는 상태이다. 그리고 놀람 상태는 공포 상태에서 안정 상태로 전이, 혹은 그 반대의 급격한 행동 변화를 보이는 것을 방지하기 위해 존재한다. 이로 인해 좀 더 자연스러운 모습을 나타낼 수 있도록 하였다.

3.3 조종 행동을 이용한 이동 제어

피포식자에는 무리지어 이동하는 플로킹과 일정 시간 이후에 목표의 위치로부터 회피(evasion) 등 여러 가지의 조종 행동이 적용되었다. FSM의 각 상태마다 이 조종 행동들이 적용될 가중치를 가지고 있다. 결정된 상태에 따라 서로 다른 가중치를 적용하여 합성함에 따라 개체는 어떠한 의도를 가지고 움직이는 듯한 움직임을 보여주게 된다.

3.4 Level Of Detail 를 이용한 연산 감소

가장 연산의 비중을 많이 차지하는 플로킹의 연산 감소를 위해서 Level of Detail(LOD)의 개념을 도입하였다. LOD란 그래픽 분야에서 자주 사용되는 방법으로 표현의 상세한 정도를 달리 하여 잘 보이지 않는 곳의 연산을 줄여서 성능을 향상시키는 방식이다. 이와 유사하게 본 논문에서는 움직임이 세밀하지 않아도 큰 영향이 없을만한 영역은 연산의 횟수를 감소시켜서 연산의 부담을 줄였다. 움직임의 정밀도는 다소 희생되지만 이를 통하여 얻는 연산 감소의 이득은 상당하다. 격자 모양의 공간 분할을 통해 개체를 관리하고 영향 정도에 따른 차등 처리를 하여 연산을 감소한다. 각 단계별 상세 사항은 다음과 같다.

3.4.1 격자 공간 분할

Craig W. Reynolds의 LQ bin lattice 방식을 개선하기

위해 격자 방식의 공간 분할을 사용하였다.[1] LQ bin lattice 방식은 세계에 존재하는 모든 개체의 위치를 비교하거나 공간 분할을 이용해 탐색 공간을 상당량 줄일 수 있으며 군중의 숫자가 증가할수록 보다 효과적이다. 가상 세계를 일정한 크기의 격자로 나눈 후, 각각의 개체는 현재 위치하는 격자ID를 가진다. 그리고 분할된 각 공간은 그림 8과 같이 인근의 분할된 공간으로의 연결과 해당 공간에 존재하는 개체의 링크드 리스트를 가진다.

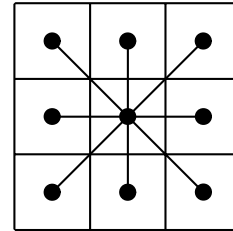


그림 8. 격자 공간 분할

매 루프마다 개체의 위치를 새로 갱신하고 갱신된 자신의 현재 위치를 판단하여 인접 공간으로 이동되었는지 검사하고, 넘어갔다면 인접 공간으로 소속을 옮긴다. 인근 공간으로의 연결이 단순하게 인덱스로 접근하는 것보다 연산을 줄일 수 있다. 그러나 공간을 너무 작게 분할할 경우 오히려 소속 공간을 바꾸는 시간의 증가로 비효율적이 되므로 적당한 크기로 분할하는 것이 중요하다.

3.4.2 영향 정도에 따른 차등 처리

피포식자 무리는 몇 가지 요소에 의해 움직임에 영향을 크게 받는다. 이외의 경우에는 직접적으로 시야에 들어오는 객체가 아닌 경우 세밀한 움직임을 하지 않아도 부자연스럽다고 느껴지지 않는다. 따라서 이러한 객체들의 연산을 감소시켜서 속도의 향상을 이끌어 낼 수 있다. 이 때, 고려할 수 있는 요소들은 다음과 같다.

- 유저의 시야 범위내의 개체
- 포식자와 인접한 개체
- 장애물과 인접한 개체

LOD의 영역을 그림으로 나타낸 것이 그림 9이다.

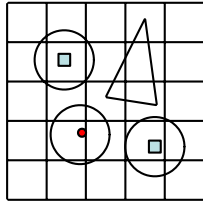


그림 9. 영향력에 인한 LOD 영역

유저의 시야 내에 있는 개체는 가장 자연스러운 움직임을 보여주기 위해 고수준의 LOD 값을 가진다. 포식자와 근접한 개체는 생명 활동을 위한 중대한 상황에 접하였으므로 또한 고수준이 LOD 값을 부여한다. 그리고 장애물과 가까운 개체는 장애물에 충돌하지 않도록 회피 운동을 하기 위해 고수준의 LOD 값을 가진다. 그 외의 개체들은 저수준의 LOD 값을 가지며 이들은 3.4.3의 방법을 통해 연산을 감소시킨다.

3.4.3 조종 행동의 연산 감소

LOD의 정도에 따라 플로킹을 위한 주변 탐색 시에 자신과의 거리 차이 계산을 할 개체의 수를 감소시키고서 연산 속도의 향상을 취한다.

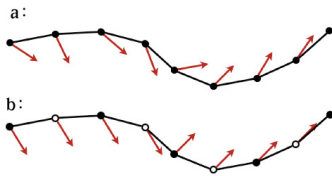


그림 8. 조종 행동 연산의 감소

이 방식을 사용하여 개체에 대한 연산 주기를 두 분류로 나누어서, LOD 값이 높은 개체는 그림 9의 a와 같이 지속적으로 조종 행동의 벡터를 구하였다. 그리고 저수준 LOD인 개체에 대해서는 조종 행동의 연산 횟수를 그림 9의 b와 같이 1/2로 낮추고 이전 시간에 사용된 조종 행동 벡터를 그대로 사용한다. 즉 개체에 대한 연산의 업데이트 주기를 1/2로 줄임으로써 그만큼 속도를 향상시킬 수 있는 것이다. 연산의 주기를 더욱 낮출수록 연산은 줄어드는 반면 움직임의 정밀도가 떨어지므로 적당한 주기로 설정하는 것이 매우 중요하다.

4. 실험 및 결과

본 논문에서 제안한 기법을 실험하기 위해 ATI Sample Framework를 기반으로 그림 10과 같은 포식자와 피포식

자가 등장하는 가상 환경을 구현하였다.



그림 9. 구현된 가상 환경

등장하는 피포식자 무리는 플로킹에 의한 집단적인 이동을 하며 포식자로부터 도망가는 모습을 보여준다. 이 때, 그림 11과 같이 피포식자 간에 이벤트가 전파되는 것과 유사한 움직임을 볼 수 있다. 이는 포식자 주변의 피포식자들이 느낀 위험도가 높아지면서 포식자로부터 도망가게 되고, 주변의 위험을 느낀 다른 개체들도 같이 도망가게 되는 것이다. 인간 지식과 이를 표현한 간단한 퍼지 규칙만으로 특별한 공식을 구하지 않아도 이와 같은 모습을 연출할 수 있는 것이다.

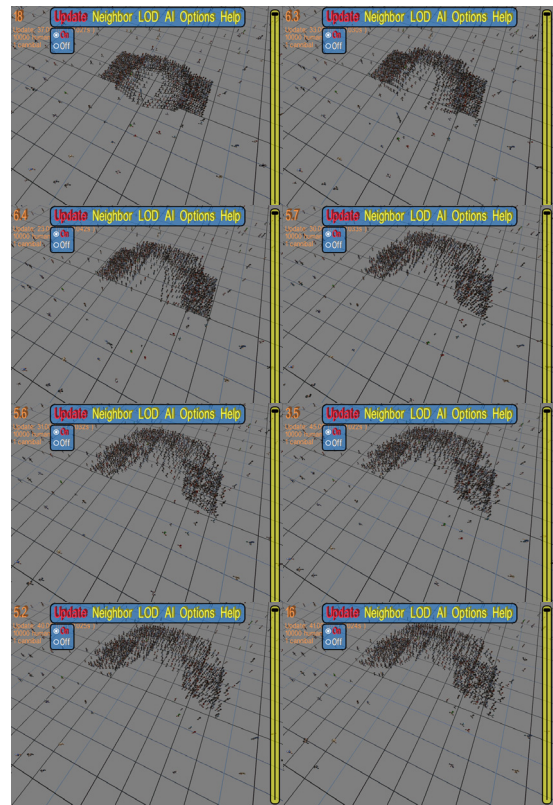


그림 10. 놀란 상태가 전파되는 모습

포식자가 피포식자 무리로 접근하면 무리지어 회피하는 모습을 보인다. 이는 퍼지 추론을 이용한 인지에서 주변의 위험도를 고려하여 현재 위험도를 인지하도록 되어 있기 때문에 위험도가 주변에서 전파되는 것과 같은 자연스러운 인지 모습을 보이는 것이다.

가상 환경을 구현한 프로그램에서 다음과 같은 성능 향상을 확인하였다.

표 1. 설정별 측정된 FPS

LOD	10000	15000	20000
ON	60	48	30
OFF	40	25	15

기록된 수치는 1초를 모든 개체의 인지부터 좌표 이동까지의 모든 업데이트에 걸린 시간으로 나눈 초당 프레임 수 (Frame Per Second)를 나타낸다. LOD에 의해 약 2배의 속도 향상을 볼 수 있다.

5. 결론

본 논문에서는 앞서 설명한 3가지의 방법으로 개선된 Steering Behavior 기법을 통해 보다 지능적이고 빠르게 군중의 제어가 가능하였다. 그러나 대규모의 군중을 다루기 위해서는 AI뿐만이 아니라 많은 수의 개체를 그리기 위한 효율적인 렌더링 방법도 필요로 한다. 실제 어플리케이션에 적용하기 위해선 이에 대한 연구 또한 필요할 것이다.

감사의 글

본 연구는 21세기 프론티어 연구개발 사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅 및 네트워크 원천기반기술 개발사업의 지원을 받았습니다.

참고문헌

- [1] Craig W. Reynolds, "Flocks, herds, anschools: A distributed behavioral model", Proceedings of SIGGRAPH '87, volume 21, pages 25~34, 1987.
- [2] Craig W. Reynolds, "Steering behaviors For Autonomous Characters", Game Developer Conference 1999.
- [3] Iztok Lebar Bajec, "Fuzzy Model for a Computer Simulation of Bird Flocking, DISSERTATION, Univ. of Ljubljana (in Slovenia), 2005.
- [4] Tom Scutt, "Simple Swarms as an Alternative to Flocking", AI Game Programming Wisdom, Charles River Media, 2002.
- [5] S. Raupp Musse, "Hierarchical Model for Real Time Simulation of Virtual Human Crowds", IEEE Transactions on Visualization and Computer Graphics Vol.7, No2, pp.152-164, 2001.
- [6] Adrien Treuille, Seth Cooper, Zoran Popovi ´ c, "Continuum Crowds", ACM Transactions on Graphics (TOG), 2006.
- [7] Manfred Lau, James J. Kuffner, "Precomputed Search Trees: Planning for Interactive Goal-Driven Animation", Eurographics/ ACM SIGGRAPH Symposium on Computer Animation, 2006.