

무리 이동 개체의 군집 형태 제어 기법*

강 영민^{○*}, 조 창식^{**}

동명대학교 게임공학과*, 한국전자통신연구원**

ymkang@tu.ac.kr* cscho@etri.re.kr**

Method for Controlling the Formation of Flocking Crowds

Young-Min Kang^{○*}, Chang-Sik Cho^{**}

Tongmyong University*, ETRI**

요 약

본 논문은 무리를 지어 움직이는 군집의 전체적인 외형을 제어할 수 있는 기법을 제안한다. 본 논문의 기법은 입자로 구성된 서로 다른 두 군집이 있을 때, 하나의 군집을 다른 군집으로 가장 잘 변환할 수 있는 선형 변환을 찾아 군집의 움직임을 제어하는 데에 사용한다. 본 논문의 기법은 특정한 모양을 유지하며 이동하는 새나 물고기 떼의 움직임을 표현하는 데에 유용하며, 게임 등의 다양한 콘텐츠에서 특별한 효과를 위해 정해진 모양을 가진 군집을 표현하는 데에도 적용할 수 있다. 제안된 기법은 군집의 외형을 유지하기 위해 많은 양의 계산을 수행하지 않아도 되므로, 빠른 반응이 요구되는 게임 등의 분야에 적합하다.

1. 서 론

새와 물고기 등이 무리를 지어 이동하는 모습을 관찰하면 군집의 전체적인 모양이 일정하게 유지되는 현상을 발견할 수 있다. 이러한 모양은 강제처럼 군집을 구성하는 입자들의 위치적 상호 관계가 고정적으로 유지되는 것은 아니지만, 전체의 외형이 원래의 모양을 유추할 수 있을 정도의 관계는 유지한다. 이러한 외형의 유지뿐만 아니라 개별 개체들은 어디에 구속되지 않고 자유롭게 움직이는 양상을 보인다.

군집을 다룰 때에, 일반적인 관심은 개별 개체의 행동 규칙을 모델링하는 데에만 집중하여 군집 전체의 움직임을 제어하는 문제는 많이 다뤄지지 않았다. 본 논문의 군집의 외형이 특정한 형태를 유지할 수 있도록 군집 구성 개체의 움직임을 제어하는 기법을 제안한다.

제안된 기법은 외부 입력을 통해 움직임이 제어

되는 군집의 개별 개체들이 각각 어떠한 점을 목표로 이동해야 하는지를 결정하는 것을 핵심적인 요소로 한다. 이를 위하여 외부 제어를 통해 입력된 군집의 이동 방향과 위치에 따라 정렬된 이상적 목표 군집 형상을 현재의 군집 상태로 가장 비슷하게 옮겨 놓는 선형 변환을 구한다. 이 선형 변환을 적용하여 얻는 군집의 상태는 현재의 군집 개체들이 가장 최소의 에너지를 소비하고 원래의 형상을 만들어 낼 수 있는 상태가 된다. 이와 함께 군집의 전체적인 크기를 원래의 모양과 일치시키는 목표점을 같이 제시하여 군집이 제어의 방향과 위치를 따르면서 전반적인 모양은 원래 지정된 형태를 유지할 수 있도록 한다.

본 논문에서 제안된 기법은 특정한 모양을 유지하며 이동하는 새나 물고기 떼의 움직임을 표현하는 데에 유용하며, 게임 등의 다양한 콘텐츠에서 특별한 효과를 위해 정해진 모양을 가진 군집을 표현하는 데에도 적용할 수 있다.

* 본 연구는 지식경제부의 산업원천기술개발사업[10035184, 실시간 스트리밍 기반의 게임 서비스 기술]의 일환으로 수행되었습니다.

2. 관련 연구

새와 물고기처럼 일정한 무리 모양을 형성하고 이동하는 집단의 움직임을 흉내 내기 위한 연구는 Reynolds의 선구적인 연구를 통해 시작되었다[1]. 이 연구에서는 간단한 규칙을 따르는 개체를 보이드(boid)라고 명명했으며, 이들의 단순한 동작이 복잡한 군집 행동으로 나타날 수 있음을 보였다.

Tu와 Terzopoulos는 물고기 집단의 움직임을 표현하기 위해 간단한 인지 능력을 가진 물고기 개체의 행동 규칙을 모델링하였다[2]. 이후에도 군집 행동을 보이는 개체의 행동을 모델링하려는 다양한 연구들이 있었다[3,4]. 하지만 이러한 연구들의 단점은 개체의 행동에만 관심을 두어 군집 전체의 모양을 제어하는 문제를 고려하지 않았다.

군집의 움직임에 대해 강한 제한조건을 적용하는 기법에 대한 연구는 Anderson 등에 의해 이루어졌다[5]. 하지만 이 기법은 계산 복잡도가 지나치게 높다는 문제를 가지고 있다.

군집의 모양을 제어하는 연구는 Xu 등에 의해서 수행되었다[6]. 이 기법은 군집의 외형을 제어하기 위해 개별 개체의 위치를 군집이 따라야 할 외형을 정의하는 메시(mesh) 위에 투영하는 방식을 사용한다. 이러한 방식은 외형을 따를 수는 있지만, 자연스러운 개체 행동을 제한할 수 있으며, 현재의 군집이 최소의 에너지로 외형을 유지하는 방향으로 움직이는 행동을 표현할 수 없다.

본 논문에서 제안하는 기법은 최소의 에너지를 개체에 작용하여 외형을 유지하려고 한다. 이를 위해서는 군집의 현재 상태와 목표가 되는 외형을 이루는 상태 사이의 기하적 변환을 파악해야 한다. 이러한 연구는 컴퓨터 비전(vision) 등의 분야에서 활발히 연구된 분야인데, 최근 관심을 받고 있는 기법은 Mueller 등이 물체의 변형을 표현하기 위해 사용한 형상 일치 기법이다[7]. 본 논문은 이러한 형상 일치 기법을 군집의 제어에도 활용할 수 있음을 보이고 이를 통해 군집의 외형을 일정한 형태로 유지하는 방법을 제안한다.

3. 군집 행동 제어

본 논문에서는 새와 물고기 등을 표현할 수 있는 군집이 사용자가 지정한 경로를 따라 이동한다는 것을 가정한다. 이러한 경로는 다양한 방식으로 시스템에 입력될 수 있다. 이것은 본 논문의 주제가 아니므로 상세한 설명을 생략한다.

초기 상태의 군집이 가진 외형을 군집이 따라야 할 외형이라고 가정하자. 이 초기 상태에서 개별 개체 i 의 위치를 x_i 라고 하고, 각 개별 개체의 질량이 모두 같다고 가정하고, 초기 군집의 중심, x_{cm}^0 를 다음과 같이 질량 중심으로 정한다.

$$x_{cm}^0 = \frac{1}{n} \sum_{i=1}^n x_i^0 \quad (\text{식 1})$$

군집을 구성하는 개별 개체는 이 중심 기준의 상대 위치로 표현하는 것이 편리하다. 개체 i 이 이러한 상대 위치는 r_i 로 표현하고, 초기 상태의 상대 위치 r_i^0 는 $r_i^0 = x_i^0 - x_{cm}^0$ 로 정의된다.

시간 t 에서 군집의 개별 개체는 x_i^t 의 위치에 있다. 따라서 x_{cm}^t 와 r_i^t 도 비슷한 방식으로 정의된다.

$$x_{cm}^t = \frac{1}{n} \sum_{i=1}^n x_i^t \quad (\text{식 2})$$

$$r_i^t = x_i^t - x_{cm}^t$$

군집의 초기 상태에서 군집에 적용된 회전 변환 R^0 은 항등행렬 I 라고 할 수 있다. 군집이 주어진 경로를 따라 이동하게 되면, 군집의 무게 중심이 경로를 따라 이동할 것이며, 더 이상 초기의 방향을 유지할 수 없고 경로를 따라 방향을 바뀌어야 할 것이다. 이렇게 방향을 바꾸는 회전 변환을 R^t 라고 할 수 있다.

만약 군집이 외형이 전혀 변화하지 않고 강체처럼 움직인다면 군집을 구성하는 특정 개체는 다음과 같은 위치로 이동해 있을 것이다.

$$x_i^t = R^t r_i^0 + x_{cm}^t \quad (\text{식 3})$$

하지만 군집 애니메이션의 특성상 개별 개체들은 자율적으로 운동해야 하므로 모든 개체에 적용되는 이러한 회전 변환이 존재할 수 없다. 개체가

초기 군집 외형을 따르기 위해서는 다음과 같이 원래의 군집을 현재의 군집과 가장 유사한 형태로 변형하는 회전 행렬 \tilde{R}^t 를 구해 원래의 군집에 이 회전변화를 적용한 상태 쪽으로 현재의 개체가 움직이게 하는 것이다. 이 회전을 구하는 방법은 최소자승법을 이용하여 다음 식을 최소화하는 \tilde{R}^t 을 구하는 최적화 문제가 된다.

$$\min \sum_i (\tilde{R}^t r_i^0 - r_i^t) \quad (\text{식 } 4)$$

이 문제를 풀어 \tilde{R}^t 를 구하여 원래의 군집에 적용하면 개별 개체가 따라야 할 목표 지점을 구할 수 있다. 하지만, 이렇게 구한 목표 지점은 현재의 군집이 가지고 있는 전체적인 크기나 휨 등과 관계없이 원래의 외형을 강체 변환하여 얻는 상태를 목표로 설정하기 때문에 개별 개체 입장에서는 매우 무리한 목표치가 될 수도 있다.

이러한 문제를 해결하기 위해 우리는 최적화의 대상을 회전 행렬이 아니라 선형 변환 A로 설정하였다. 따라서 최적화의 대상은 다음이 된다.

$$\min \sum_i (A^t r_i^0 - r_i^t) \quad (\text{식 } 5)$$

Mueller 등의 연구에서 언급된 바와 같이 A의 모든 계수에 대한 도함수가 0이 되도록 설정하면 최적 변환 A를 다음과 같이 구할 수 있다.

$$A = \left(\sum_i r_i^t r_i^{0T} \right) \left(\sum_i r_i^0 r_i^{0T} \right)^{-1} \quad (\text{식 } 6)$$

군집의 개별 개체 i 는 이제 최소의 에너지를 이용하여 원래의 외형을 표현하기 위해서 다음과 같은 위치를 목표 지점 g_i^t 로 설정할 수 있다.

$$g_i^t = A r_i^0 + x_{cm}^t \quad (\text{식 } 7)$$

개별 개체가 초기의 외형을 표현할 수 있는 위치로 이동하도록 제어하기 위해, 본 논문의 기법은 (식 7)의 목표 위치로 빠르게 이동할 수 있도록 다음과 같은 외형 복원력 f_i^t 를 개별 입자에 가한다.

$$f_i^t = \frac{\kappa m_i}{h} (g_i^t - x_i^t) \quad (\text{식 } 8)$$

이때 κ 는 외형을 복원하는 힘의 경직도이며 m_i 는 개체 i 의 질량이다.

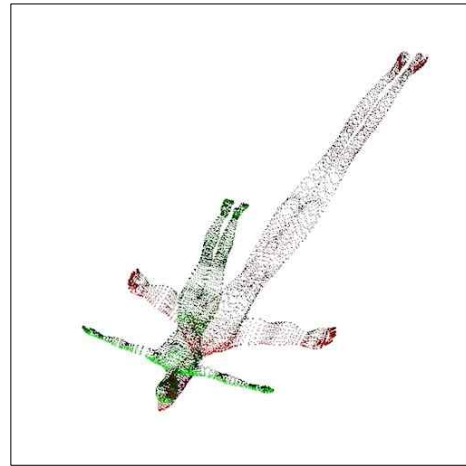
(식 8)의 힘을 적용할 경우 개별 개체에 최소의 힘을 가해 원래의 외형을 유지할 수 있다. 하지만, 이때 적용되는 선형 변환 A는 크기 변형을 포함한 선형 변환이기 때문에 군집의 크기는 초기의 크기로 복원되지 않는다는 단점을 가진다. 따라서 군집의 모양은 원래의 모양과 비슷하지만 지나치게 늘어나고 개별 개체가 거의 움직이지 않는 형태의 애니메이션이 생성되게 된다. 따라서 우리는 이러한 문제를 해결하기 위해 입력된 경로를 통해 얻어지는 회전행렬 R과 앞서 구한 (식 6)의 A를 동시에 고려한 목표치를 사용한다. 목표 위치는 다음과 같이 수정된다.

$$g_i^t = \alpha R r_i^0 + (1 - \alpha) A r_i^0 + x_{cm}^t \quad (\text{식 } 9)$$

적용되는 힘은 여전히 (식 8)과 같다. 군집의 외형 복원 특성은 (식 9)의 α 를 변경하여 제어할 수 있다. 이때 α 는 0에서 1 사이의 값이 된다.

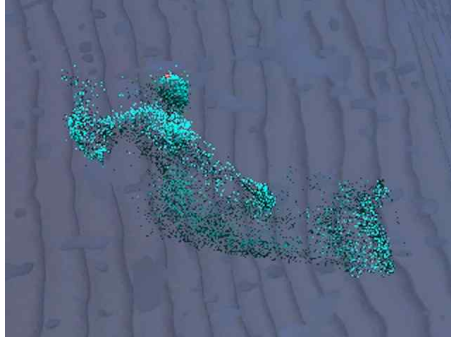
4. 실험결과

본 논문의 기법을 구현하기 위하여 사용한 시스템은 Mac OS X 운영체제를 사용하는 3.2 GHz Intel i3 CPU, 4Gb DDR3 RAM, ATI Radeon HD 5670 GPU의 컴퓨터이다.

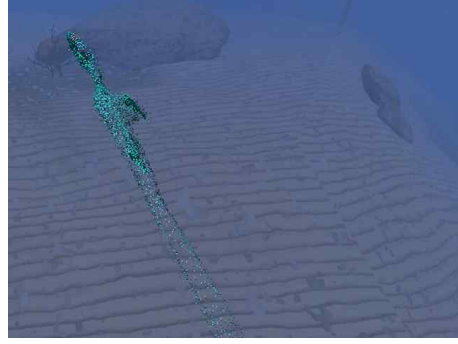


[그림 1] 두 개의 목표 외형

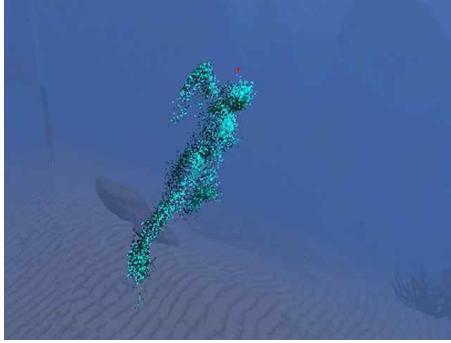
군집의 모양은 메시(mesh) 모델을 읽어 들여 그 정점을 이용하여 설정하였다. 실험에 사용된 메시



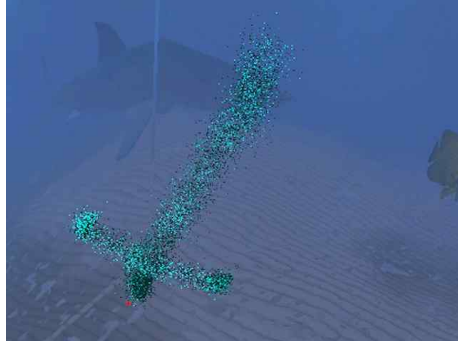
(a) $\alpha = 1$



$\alpha = 0$

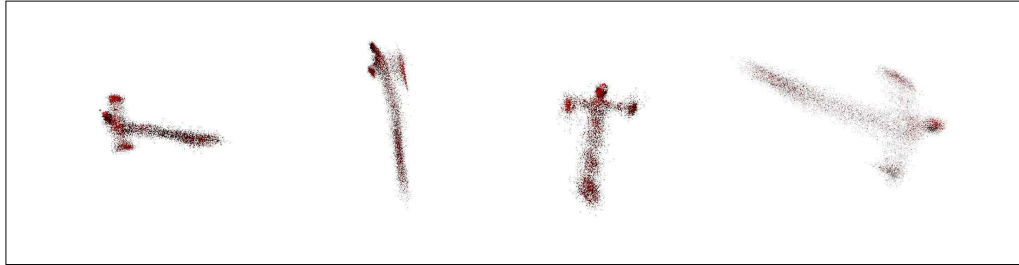


(c) $\alpha = 0.5$



(d) $\alpha = rand(0,1)$

[그림 2] α 값의 조절을 통한 외형 제어의 결과



[그림 3] 외형을 유지하며 이동하는 군집의 애니메이션

는 총 9276개의 정점으로 이루어져 있으며, 군집을 구성하는 개체의 수도 동일하다.

[그림 1]은 군집이 따라야할 목표 위치를 가시화하고 있다. 그림에는 메시의 외형을 따르는 두 가지 목표 위치가 나타나 있다. 두 목표 위치 가운데 원래의 메시와 동일한 크기를 가진 외형이 경로에 의해 결정되는 회전 변환만을 적용한 목표이며, 길게 늘어나 있는 목표 위치는 현재의 군집을 최소로 변형하여 주어진 외형과 닮은 형태를 만들 수

있는 목표 위치이다. 두 목표 위치 가운데 어떠한 위치를 더 중요하게 여길 것인지는 (식 9)의 매개변수 α 로 제어한다.

9276 개의 개체가 무리를 지어 이동하며 본래의 외형을 유지하는 시뮬레이션을 본 논문의 기법으로 구현하여 앞서 언급한 시스템에서 수행했을 때 얻는 성능은 초당 83 프레임 정도였다. 이 결과는 하드웨어 가속을 사용하지 않고 하나의 스레드(thread)만을 사용한 것이며, GPU 가속 등을 사용

할 경우 더욱 높은 성능을 얻을 수 있을 것이다.

[그림 2]는 본 논문의 기법을 적용하여 생성한 군집 애니메이션의 결과를 보이고 있다. [그림 2]의 (a)는 (식 9)의 제어 매개 변수인 α 값을 1로 설정하여 경로를 정확히 따르는 회전 변환을 초기 군집에 적용하여 얻은 목표치를 적용한 것으로 강제처럼 원래의 외형과 동일한 모습을 목표로 한다. 경로가 변경됨에 따라 방향이 급격히 변화할 경우, 그림처럼 외형을 정확히 따르지 못하는 개체가 많아진다. (b)는 이와 달리 α 값을 0으로 설정한 결과로 개체들이 쉽게 원래의 모양을 표현할 수 있는 동작을 보이지만, 현재의 위치에서 최소의 이동만을 보이므로 군집 애니메이션에서 나타나야 하는 자유로운 움직임이 둔화된다. 따라서 이 두 결과를 적절히 합성하는 것이 바람직하다. (c)에 보인 결과는 α 를 0.5로 하였으며, (d)는 α 값을 랜덤 함수를 통해 0에서 1 사이의 값으로 생성한 결과이다.

실험을 통해 우리는 α 값을 임의의 랜덤 값으로 설정했을 때에 가장 자연스러운 군집 행동을 보임을 확인하였다. [그림 3]은 이렇게 임의의 랜덤 값을 α 에 적용하여 얻은 군집 애니메이션에서 포착된 이미지들이다.

5. 결 론

본 논문은 무리를 지어 움직이는 군집의 전체적인 외형을 제어할 수 있는 기법을 제안하였다. 본 논문에서 제안한 기법은 하나의 군집을 다른 군집으로 가장 잘 변환할 수 있는 선형 변환을 찾아 군집의 움직임을 제어하는 데에 사용한다.

제안된 기법은 매우 빠르게 개별 개체의 목표 위치를 계산할 수 있어 실시간 애니메이션에서 특별한 형태를 갖춘 군집 행동을 효율적이며 효과적으로 표현할 수 있다. 따라서 게임과 같은 상호작용적 콘텐츠에서 특별한 효과를 위해 군집의 형태를 제어할 필요가 있을 경우, 유용하게 활용할 수 있는 기술이다. 본 논문에서 제안된 기법은 빠르게 두 군집 사이의 선형 변환을 찾기 때문에 다수의 개체들을 실시간 환경에서 애니메이션 할 수 있다.

참고문헌

- [1] C. Reynolds, "Flocks, herds, and schools: a distributed behavior model", Computer Graphics, 4:25-34, 1987.
- [2] X. Tu, D. Terzopoulos, "Artificial fishes: physics, locomotion, perception, behavior", Proc. of ACM SIGGRAPH '94, pp.43-50, 1994.
- [3] U. Erra, R. D. Chiara, V. Scarano, M. Tatafiore, "Massive simulation using GPU of a distributed behavioral model of a flock with obstacle avoidance", Proc. of the Vision, Modeling, and Visualization Conference 2004, pp. 233-240, 2004.
- [4] O. Bayazit, J. Lien, N. Amato, "Roadmap-based flocking for complex environments", Proc. of 10th Pacific Conference on Computer Graphics and Applications, pp.104-115, 2002.
- [5] M. Anderson, E. McDaniel, S. Chenney, "Constrained Animation of Flocks", Eurographics/SIGGRAPH Symposium on Computer Animation 2003.
- [6] J. Xu, X. Jin, Y. Yu, T. Shen, M. Zhou, "Shape-constrained flock animation", Computer animation and Virtual Worlds, 19:319-330, 2008.
- [7] M. Mueller, B. Heidelberger, M. Teschner, M. Gross, "Meshless Deformation Based on Shape Matching", Proc. of ACM SIGGRAPH 2005, 24(3), 2005.