시스템 성능 향상을 위한 하이브리드 기법을 적용한 플로킹 시스템 설계 및 구현

Design and Implementation of Flocking System for Increasing System Capacity with Hybrid Technique

류남훈*, 반경진*, 오경숙*, 송승헌**, 김응곤*

순천대학교 컴퓨터과학과*, 광양만권 u-IT 연구소**

Nam-Hoon Ryu(kmpi9560@hanmail.net)*, Kyeong-Jin Ban(multwave@sunchon.ac.kr)*, Kyeong-Sug Oh(oksug10@naver.com)*, Seung-Heon Song(fsaa@sunchon.ac.kr)**, Eung-Kon Kim(kek@sunchon.ac.kr)*

요약

컴퓨터 애니메이션 기법이 적용된 영화나 온라인 게임 등의 보급으로 인해 다수의 캐릭터들이 등장하는 장면을 쉽게 접할 수 있다. 대규모 군중 애니메이션의 경우 장면의 사실성을 높이다 보면 시스템의 성능이 저하되고, 시스템의 성능을 높이다 보면 장면의 사실성이 떨어지게 된다. 본 논문에서는 해저 환경을 배경으로 한 대규모 군중 애니메이션을 구현함에 있어서 어류의 행동 유형에 영향을 미치는 요소에 대해 분석하여 적용하였으며, 군중의 개념을 이용하여 집단이나 개별 객체 각각에 대해 행동 유형을 제어할 수 있도록 하였으며, 애니메이션을 위한 계산 방법으로 실시간 계산법과 혼합 계산 방법인 하이브리드 계산법을 비교 분석함으로써 시스템의 성능을 높이면서도 자연스러운 장면의 표현 방법을 찾고자 한다.

■ 중심어 : | 가상 수족관 | 군중 애니메이션 | 플로킹 알고리즘 |

Abstract

Due to spread of movies or online games which are applied with computer animation techniques, we can easily see scenes where numerous characters appear. In the case of large-scale crowd animation, if one were to increase reality of the scene, features of system would be lowered, and if one were to increase functioning of system, reality of the scene would be lowered. In realizing large-scale crowd animation with seafloor environment as background, the paper analyzed and applied elements that affect behavioral types of fishes; and by using concept of crowd, the paper enabled each group or object to control their behavioral type; by comparing and contrasting real-time calculation method as calculation method for animation and hybrid calculation method which is mixed calculation method, the paper seeks to find a method that increases functioning of the system while also expresses natural scenes.

■ keyword : | Virtual Aquarium | Schooling Animation | Flocking Algorithm |

* 본 연구는 문화관광부 및 한국문화콘텐츠진흥원의 지역문화산업연구센터(CRC) 지원사업의 연구결과로 수행되었습니다.

접수번호 : #080414-003

접수일자 : 2008년 04월 14일

심사완료일 : 2008년 07월 10일

교신저자 : 김응곤, e-mail : kek@sunchon.ac.kr

I. 서 론

컴퓨터 그래픽스 기술의 발달로 인해 애니메이션이 점차 증가하고 있으며, 이에 따라 영화, 게임 등의 엔터테인먼트 산업에서는 현실 세계와 흡사한 군중 애니메이션의 요구가 계속적으로 증가하고 있다. 군중 애니메이션은 다수의 캐릭터를 이용해 장면의 사실성을 높이는 기술이다. 캐릭터의 수가 증가함에 따라 애니메이션 성능은 반비례하여 감소하지만, 가상세계의 몰입감은 비례하여 증가하게 된다. 다수의 캐릭터에 대해 각각의 움직임을 보다 사실적으로 표현해야 하며, 시스템의 효율을 높여서 보다 부드럽게, 그리고 사용자와의 상호작용을 원활하게 해야 한다.

대규모 군중 애니메이션을 구현함에 있어서 캐릭터들의 세부적인 움직임을 각각 묘사하는 일은 매우 많은 노력을 필요로 한다. 따라서 대규모의 캐릭터가 등장하는 애니메이션일수록 자동화의 필요성이 대두되고 있다. 하지만, 캐릭터의 움직임을 자동화할 경우 여러 가지 문제가 발생한다. 단위 모션간의 자연스럽지 못한 연결이 발생하게 되며, 군중이 목적하고자 하는 행동과 일치하지 않는 캐릭터의 움직임이 발생하게 되며, 캐릭터간의 충돌이나 고정 장애물과의 충돌 등의 문제가 발생하게 된다[1]. 이에 어류의 일반적인 행동 유형들을 분석하여 시나리오로 작성해 두고, 사용자의 제어 혹은 상황의 변화로 인해 발생하는 이벤트에 의한 행동 유형의 변화에 대해서는 실시간으로 계산하는 방식을 제안한다.

본 논문에서는 Ⅱ장에서 군중 애니메이션의 대표 알고리즘인 플로킹 알고리즘에 대해서 알아보고, Ⅲ장에서는 가상 수족관 시스템을 구성하는 요소들과 속성, 그리고 이 요소들을 편집할 수 있는 어류 속성 편집기에 대해서 알아본다. Ⅳ장에서는 구현된 Fish 군중 애니메이션 시스템에 대해서 알아보고, Ⅴ장에서는 계산 방식에 따른 실험 결과를 분석하고, 마지막 Ⅵ장에서는 결론을 맺는다.

II. 플로킹 알고리즘

Craig W. Reynolds는 1987년에 SIGGRAPH에서 발표한 "Flocks, Herds, and Schools : A Distributed Behavioral Model"에서 그래픽스 분야에서는 처음으로 무리의 의미인 플로킹(Flocking) 기법을 도입하였다[2]. 이 논문에서 Reynolds는 자동화된 에이전트(보통 보이드(boid)라고 부른다)들의 집단이 새떼나 물고기떼 또는 벌떼와 비슷한 집단행동을 보이도록 만들기 위한 세 가지 조타 행동(Steering Behavior)을 제시하였다. Reynolds는 이후의 논문들을 통해 네 번째 규칙이라고 불리우는 회피(Avoidance) 기법을 추가하였다[3].

Steven Woodcock는 Game Programming Gems 2권을 통해 보이드들에 새로운 특징인 '굶주림'이라는 특징을 추가함으로써 집단 행동을 하면서도 다소 개별화하는 형태로 확장하였다. 그리고 보이드들이 서로를 피해 다니도록 하기 위하여 다른 보이드들을 잡아먹는 포식자(Predator) 보이드들을 추가하였다. 그리하여 다섯 번째 규칙으로 생존(Survival) 규칙을 도입하였다[4].

1. 분리(Separation) 규칙

보이드들이 무리를 짓기 위해 모여들기만 할 경우 서로 부딪히게 된다. 그러므로 보이드들이 주변의 다른 보이드들이나 고정 장애물과의 적당한 거리를 유지하도록 하는 것이 분리 규칙이다. 무리의 각 보이드마다 주변의 다른 보이드들 혹은 부근의 고정 장애물과의 거리를 판단하고 적당한 거리를 유지할 수 있도록 방향을 돌리는 방식으로 이 규칙을 구현한다.

2. 정렬(Alignment) 규칙

보이드가 주변의 다른 보이드들과 동일한 방향이나 속도를 유지하도록 하는 것이다. 이 규칙에 의해 보이드들이 모인 무리가 마치 하나의 보이드인 것처럼 움직이게 된다. 보이드의 방향과 속도를 주변 보이드들의 평균적인 속도와 방향으로 조정함으로써 이 규칙을 구현한다.

3. 응집(Cohesion) 규칙

보이드들이 무리와의 거리가 멀어지면 무리를 짓기 위해 무리의 중심으로 모이게 하는 역할을 한다. 보이드들의 방향을 무리를 짓고 있는 보이드들의 평균 위치로 갈 수 있도록 조정함으로써 이 규칙을 구현한다.

4. 회피(Avoidance) 규칙

보이드들이 무리지어 모여 있을 경우 주변의 다른 보이드들이나 고정 장애물과 충돌하지 않도록 하는 역할을 한다. 이는 보이드 자신의 예정 경로 중 일정 시간 동안의 경로를 내다보고 다른 보이드들이나 고정 장애물과의 충돌이 예상된다면 속도와 방향을 조정함으로써 이 규칙을 구현한다.

5. 생존(Survival) 규칙

현실 세계에 먹고 먹히는 관계인 생태계가 존재하듯이 보이드들 중에도 다른 보이드들을 잡아먹는 포식자와 잡혀 먹히는 피식자가 존재한다. 포식자는 피식자를 발견하면 피식자 방향으로, 피식자는 포식자를 발견하면 포식자의 반대 방향으로 방향을 조정하며 속도를 높이는 규칙을 구현한다.

플로킹의 기본 규칙들을 적용한 예제로는 플로킹의 아버지라 불리우는 Craig Reynolds의 홈페이지[5]와 riversoftavg[6] 사이트를 들 수 있다. [그림 1]은 플로킹 알고리즘을 적용한 애니메이션으로 돌고래가 무리지어 있는 물고기떼를 향해 헤엄치는 화면이다. 물고기떼는 무리의 중심을 기준으로 일정 거리 이상 떨어지면 무리의 중심을 향해 방향을 선회한다. [그림 2]는 플로킹 데모 프로그램의 실행화면으로 체크박스를 통해 적용할 행동 유형들을 선택할 수 있으며, 행동 유형별 가중치도 조절이 가능하다. 객체가 화면의 원 및 사각형과 충돌할 위험이 있는 경우는 속도를 줄인 후 반대방향으로 선회한다. [그림 3]은 펭귄들이 무리지어 걷고 있는 플로킹 애니메이션 실행화면이다[7].

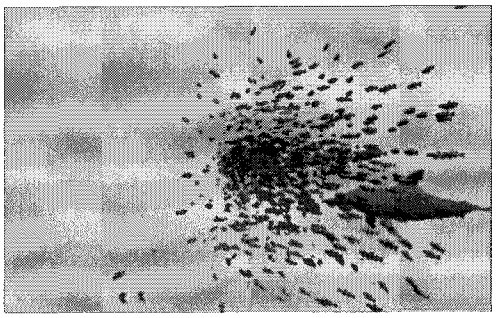


그림 1. 플로킹 애니메이션 실행화면

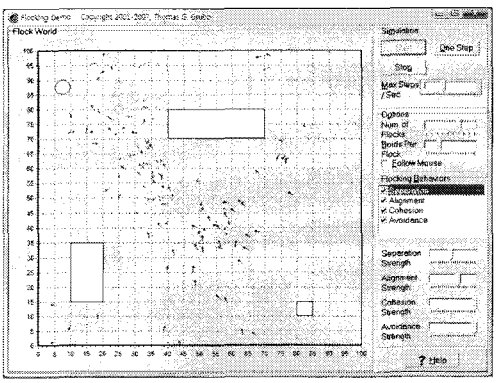


그림 2. 플로킹 데모 화면

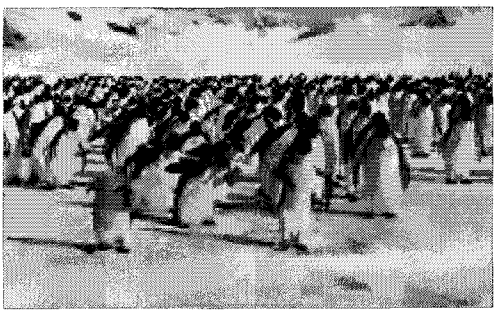


그림 3. 플로킹 애니메이션 실행화면

III. 가상 수족관 시스템

1. 시스템 개요

본 시스템은 해저 환경 구축 모듈, 어류 속성 편집 모듈, 시나리오 생성 모듈, 행동유형 제어 모듈로 구성되

며, 행동유형 제어 모듈은 다시 군중 애니메이션 제어 모듈과 행동유형 제어 모듈, 자율행동 처리 모듈로 나뉜다. [그림 4]는 가상 수족관 시스템의 구성도를 보여준다[8].

2. 해저 환경 구축 모듈

가상 수족관의 해저환경은 깊은 바다의 느낌으로 표현하기 위해서 산호초 등으로 이루어진 암초를 두었으며, 이는 각각 2,000 ~ 3,000개 정도의 삼각형으로 모델링되었다. 깊은 바다 및 먼 바다의 느낌을 표현하기 위해서 안개효과를 두었으며, 산호초 그림을 텍스처링한 후 반사 하이라이트를 적용하는 형태로 조명효과를 표현하였다. 암초는 삼각형 형태의 면으로 구성되었으며, 보이드가 자신의 예상경로 안에 충돌 가능한 해저환경이 존재하는지를 체크하는데 필요한 면들이다.

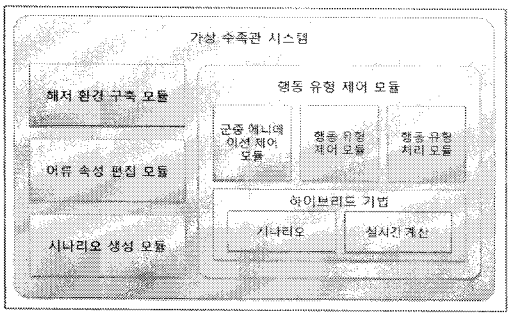


그림 4. 가상 수족관 시스템 구성도

3. 어류 속성 편집 모듈

본 시스템에서는 어류의 행동 유형과 관련된 내용 중 군중 애니메이션과 관련한 내용들만을 기초로 [표 1]과 같이 주요 속성을 구성[9][10]하였으며, [그림 5]는 이 속성을 편집할 수 있도록 개발한 스마트 오브젝트 모델러의 구현화면이다. 본 모델러는 4개의 분할 화면을 통해 객체의 회전 및 확대, 축소가 가능하며, 대화상자를 통해 어류의 기본적인 속성을 편집 후 저장할 수 있다. [표 1]은 어류의 행동 유형에 영향을 미치는 기본적인 속성들이다.

본 시스템에서는 3가지 형태의 모델로 구성하였으며, 첫 번째 모델은 동물성 플랑크톤이나 부착조류 등이 주 먹이인 초식성이다. 두 번째 모델은 부착조류나 자신보다 작은 크기의 어류가 주 먹이인 잡식성이며, 세 번째 모델은 상어와 같은 육식성이다. 이 모델들은 4,000 ~ 5,000개 정도의 삼각형 형태의 면으로 구성되어 있으며, 자연스러운 모션을 표현하기 위해서 30개의 프레임으로 모델링하였다.

표 1. 어류의 기본적인 속성

속성	내 용
기본 정보	어류를 구분하기 위한 기본적인 정보로 이름, 학명, 분류, 체색, 산란시기, 서식장소, 분포지역 등이 해당된다.
모양	모양에 따라 방추형, 측편형, 원통형, 종편형으로 나눌 수 있으며, 이에 따라 유영 형태가 달라진다.
탐지 거리	해저 지형이나 포식자, 먹이를 탐지할 수 있는 거리로 일반적으로 시야 각도가 넓을수록 탐지 거리는 짧아진다.
식성 / 먹이	초식성, 잡식성, 육식성 등으로 나뉘며, 초식성의 먹이는 동물성 플랑크톤이나 부착조류이고, 잡식성의 경우는 부착조류 및 수생곤충, 그리고 자신보다 작은 크기의 어류를 먹는다. 육식성의 경우는 자신보다 작은 크기의 어류를 먹는다.
식사량	어류마다 포만감을 느끼게 되는 식사량이 있으며, 사냥에 성공할 경우 먹이의 종류에 따라 포만감 수치는 올라간다. 포만감 수치가 100 이상이면 사냥감을 만나도 사냥하지 않는다.
최대 수심 / 수온	어류의 유영에 있어서 최대 수심까지만 내려갈 수 있으며, 최대 수심에 도달하면 현재의 수심에 위치하거나 위로 올라가게 된다. 또한 적정 수온이 있어서 최대 수심까지 접근하지 않았더라도 적정 수온보다 수온이 낮으면 위로 올라가게 된다.
최고 속도	포식자의 경우 이동하는 먹이를 발견했을 때, 피식자의 경우 위험 반경 내에 포식자의 출현시 최대속도 이내로 이동하게 되며, 일반적으로 포식자는 피식자보다 빠르므로 포식자를 사냥할 수 있지만, 피식자의 회피과정 중 해저 지형 등으로 인해 포식자의 시야에서 벗어나거나 포식자가 다른 대상을 사냥하게 되면 위험에서 벗어나게 된다.

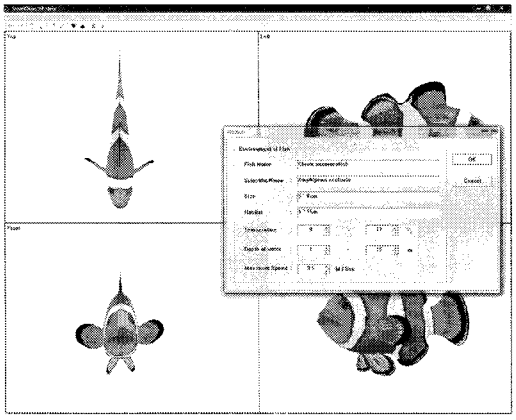


그림 5. 스마트 오브젝트 모델러 구현화면

4. 시나리오 생성 모듈

어류마다 각기 다른 특색의 유영 형태를 가지고 있다. 평상시의 유영 형태와 해저 지형 등의 장애물을 만났을 때의 유영 형태, 그리고 먹이를 발견했을 때와 포식자로부터 회피하는 유영 형태가 모두 다르다. 유영 형태는 시스템의 실행 시 발생하게 되는 이벤트에 따라 반응하는 행동들에 대해 방향성을 가지는 피치, 롤, 요에 해당하는 세 축의 각도와 속도를 표현할 수 있는 이동거리로 표현이 가능하다. [그림 6]은 시나리오 작성 예로, 초당 30 프레임으로 구성하였다.

Frame 30
Frame Time 0.033333333

X_Trans	Y_Trans	Z_Trans	X_Rot	Y_Rot	Z_Rot	X_Scale	Y_Scale	Z_Scale
5.27	16.59	-3.26	6.32	2.91	-3.22	3.62	3.62	3.62
6.12	16.86	-3.37	6.54	3.27	-3.00	3.62	3.62	3.62
6.63	17.08	-3.46	6.12	3.69	-2.71	3.62	3.62	3.62
7.35	17.30	-3.44	6.01	3.91	-2.46	3.62	3.62	3.62
8.20	17.59	-3.86	6.26	4.27	-2.23	3.62	3.62	3.62
8.92	17.96	-3.97	6.49	4.69	-1.98	3.62	3.62	3.62
9.77	18.37	-3.72	6.71	4.91	-1.75	3.62	3.62	3.62
9.52	18.59	-3.49	6.93	4.79	-1.53	3.62	3.62	3.62
9.23	18.47	-3.27	7.20	4.68	-1.26	3.62	3.62	3.62
8.81	18.36	-3.00	7.56	4.39	-0.90	3.62	3.62	3.62
9.32	18.07	-2.64	7.98	4.68	-0.48	3.62	3.62	3.62
10.04	17.65	-2.22	7.56	5.04	-0.26	3.62	3.62	3.62
10.55	17.54	-2.00	7.45	5.46	0.16	3.62	3.62	3.62
10.28	17.79	-2.12	7.70	5.68	0.38	3.62	3.62	3.62
9.99	18.02	-2.23	7.93	5.90	0.26	3.62	3.62	3.62
9.77	18.24	-2.52	7.81	5.78	0.15	3.62	3.62	3.62
10.49	18.51	-2.94	7.70	5.67	-0.14	3.62	3.62	3.62
11.34	18.92	-3.05	7.41	5.38	-0.56	3.62	3.62	3.62
12.06	19.57	-2.88	6.99	5.67	-0.67	3.62	3.62	3.62

그림 6. 시나리오 작성 예

X_Trans, Y_Trans, Z_Trans는 세 축으로의 이동거리이며, X_Rot, Y_Rot, Z_Rot는 세 축으로의 회전 각도이다. 이상 6개의 값을 불러와서 객체의 좌표 및 각도를 초당 30회에 걸쳐서 조정한다. X_Scale, Y_Scale, Z_Scale은 객체 모델링 과정 후 실제 시뮬레이션 시 원하는 크기로 확대하거나 축소하기 위한 값이다. 모델링한 객체와 유사한 객체의 시나리오 작성 시 재사용할 수 있도록 하기 위해 텍스트 형태를 취했으며, 3D 모델과 병합하지 않음으로써 시나리오만 재사용할 수 있도록 하였다.

객체를 이동시키는 방법은 크게 두 가지로, 이벤트가 발생할 때마다 계산하는 방법과 해당 모델에 대한 움직임을 미리 계산하여 시나리오로 작성해 두는 방법[11]이 있다. 기존의 연구에서는 이벤트가 발생할 때마다 실시간으로 계산하는 방법을 사용하였다. 본 시스템에서는 3가지 형태의 모델에 대해 기본적인 유영 형태에 대해서는 미리 계산된 시나리오를 불러와서 사용하는 방식으로 구현하였으며, 이벤트의 발생으로 인한 다른 시나리오로의 변환 과정에 대해서는 실시간으로 계산하는 방식인 하이브리드(Hybrid) 계산법을 사용하여 구현하였다.

5. 행동유형 제어 모듈

5.1 군중 애니메이션 제어 모듈

가상공간에 대규모의 군중 애니메이션을 표현하기 위해서 군중의 계층을 군중(Crowd), 집단(Group), 개인(Individual)의 단계로 구성하였다[12-14]. 개인의 행동 규칙은 개인이 속해 있는 집단으로부터 결정되며, 집단은 속해 있는 군중의 영향을 받는다.

개인은 개개의 움직임을 표현하기 위해서 초기화의 과정을 거친다. 이 때 모델의 기본정보와 유영 형태 등 여러 가지 습관 등을 초기화한다. 그런 다음 각종 상황을 인지할 수 있는 센서를 통해 수온이나 유속 등의 환경정보를 파악하며, 해저 지형이나 다른 보이드들과의 충돌에 대해 체크하고, 포만감 등을 체크하여 행동 유형을 결정하게 된다. 행동 유형이 결정되면 동작 제어기에서 적절한 모션을 시뮬레이션하게 된다. 개인은 사용자로부터 행동 유형 변화 메시지를 접하게 되면 메시지에 따른 행동 유형으로 전환하며, 집단이 메시지를 접하게 되면 그 메시지는 집단 전체에게 행동 유형의 변화를 지시하게 된다. [그림 7]은 군중 애니메이션 제어 모듈의 구성도를 나타낸다.

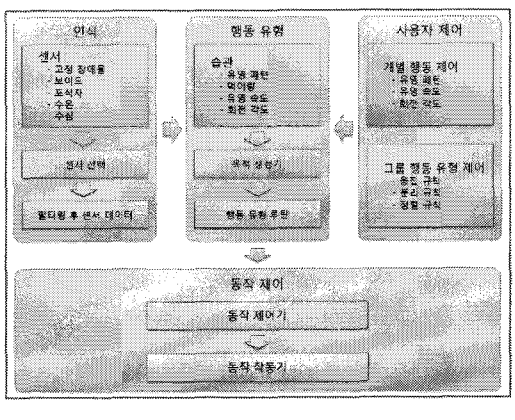


그림 8. 행동 유형 제어 모듈

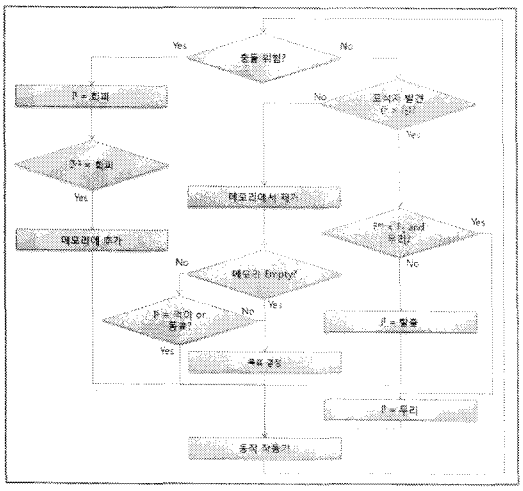


그림 9. 행동 유형 처리 모듈

Ⅳ. Fish 군중 애니메이션 시스템

본 시스템은 해저환경을 배경으로 어류들의 군중 애니메이션을 구현한 시스템으로 MFC 환경에서 Visual C++ 6.0과 OpenGL을 이용하여 구현하였다. 초기화 과정을 거친 어류 객체들은 포식자로부터의 공포감도 없으며, 포만감이 가득한 상태이므로 고정 장애물만 피하는 형태로 무리지어 다니면서 자유 유영을 하게 된다. [그림 10]은 Reynolds가 플로킹 알고리즘에서 정한 3가지의 조타 행동(Steering Behavior) 중 정렬 규칙의 실행화면이다.

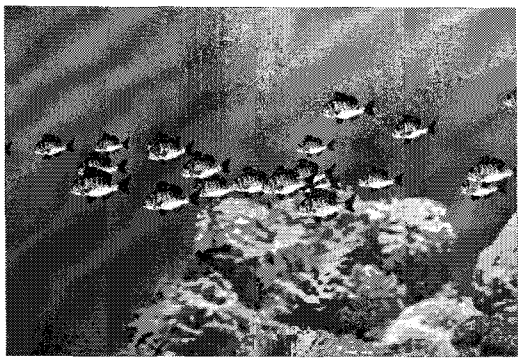


그림 10. 정렬 규칙 실행화면

각 객체는 센서를 통해 얻은 정보를 활용하여 충돌이 예상되는 경우 방향을 선회하게 되며, 충돌 가능성이 높을수록 급선회와 동시에 속도를 줄여서 회피하게 된다. 예상경로 상에 해저 지형 등의 고정 장애물이나 다른 객체와의 충돌 가능성이 없을 경우 다른 센서 정보에 대해 반응하게 된다. 단, 포식자와 사냥 대상이 되는 피식자 간에는 포식자의 입 부근에 피식자가 위치하는 경우, 사냥에 성공한 것으로 처리하였으며, 포식자는 피식자의 크기에 따라 다른 양의 포만감을 얻게 되며, 피식자는 화면에서 사라지게 처리하였다. [그림 11]은 객체의 예상 경로를 표시하여 해저 지형과 다른 객체들을 피하는 과정을 표현한 충돌 회피 실행화면이다.

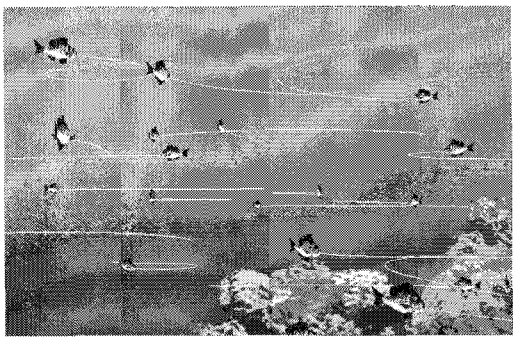


그림 11. 충돌 회피 실행화면

시간의 흐름에 따라 포만감이 줄어들게 되고, 포만감이 줄어든 객체는 먹이를 찾아 나서게 되며, 피식자는 탐지거리 내 포식자의 출현을 센서를 통해 인지하게 된다. 포식자는 피식자를 발견하게 되면 최고속도까지 속도를 높이면서 사냥을 하게 되며, 피식자 또한 최고속도까지 속도를 높여서 피하게 된다. 일반적으로 포식자의 속도가 피식자보다 빠르므로 사냥에 성공하게 되지만 해저 지형 등의 장애물로 인해 시야각도로부터 벗어나게 되면 포식자는 다른 먹이를 찾아 나서게 되며, 피식자는 다음 행동 유형으로 전환하게 된다.

객체들은 공포감과 포만감에 따라 무리지어 다니는 자유 유영 형태와 포식자와 피식자 간의 사냥 및 회피 행동 유형으로 전환하게 된다. 기본 행동 유형들은 세 축의 각도와 이동거리를 통해 속도를 표현한 시나리오

에 의해 동작되며, 행동 유형의 변화 시에는 실시간 계산을 통해 모션간의 연결을 자연스럽게 처리하였다. [그림 12]는 포식자와 피식자 간의 사냥 및 도망 형태를 표현한 실행화면이다.

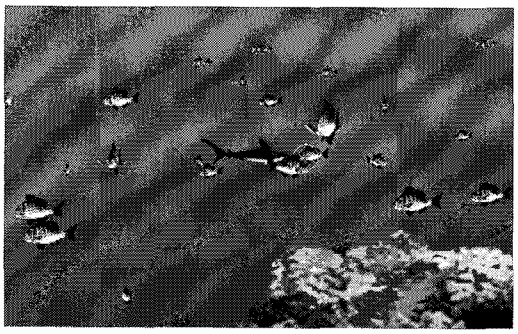


그림 12. 사냥 및 도망 실행화면

V. 실험 및 결과

애니메이션을 통해 행동 유형의 변화를 보여주는 방법은 크게 2가지가 있다. 첫 번째는 일반적인 행동 유형과 이벤트로 인한 행동 유형의 변화 모두를 실시간으로 계산하는 방법이다. 이 방법은 사용자로부터 이벤트가 발생하거나, 충돌 감지, 포식자의 발견 시 뿐만 아니라, 충돌의 위험이나 포식자로부터의 위험이 없는 경우에도 어류의 행동 습관에 근거한 자료를 활용하여 세 축으로의 각도 및 이동거리를 실시간으로 계산하는 방식이다.

두 번째는 본 논문에서 제안하는 방식인 하이브리드 계산법이다. 이 계산법은 일반적인 행동 유형은 사전에 어류의 행동 유형에 대해 면밀히 분석하고 계산하여 시나리오로 저장해 두고, 이벤트에 의한 행동 유형 변화의 경우에 한해 실시간으로 계산하는 방법이다. 사전에 어류의 행동 습관에 근거한 자료를 활용하여 세 축으로의 각도 및 이동거리를 계산하여 시나리오로 저장해 둔 후, 저장된 행동 유형이 선택될 경우 사용하는 것이다.

본 연구에서는 프레임 레이트를 체크하는 방법으로 실험하였다. 실험 결과 개체수가 50인 경우 12%가 증가하였으며, 개체수가 200과 250의 경우 모두 33%의 증가를 보였고, 개체수가 300인 경우 40%의 증가를 보였다. 이는 개체수가 적을 경우에는 큰 차이가 발생하지 않았지만 개체수가 많을수록 실시간 계산을 미리 계산해 둔 시나리오로 대체하는 것이 훨씬 효율적이라는 것을 알 수 있었다. [표 2]와 [그림 13]은 개체수별 프레임 호출 현황을 보여주고 있다.

표 2. 개체수별 프레임 호출 현황

(단위 : 프레임수/초)

구분 \ 개체수	50	100	150	200	250	300
실시간 계산법	60	30	20	15	12	10
하이브리드 계산법	67	36	26	20	16	14
증가율(%)	12	20	30	33	33	40

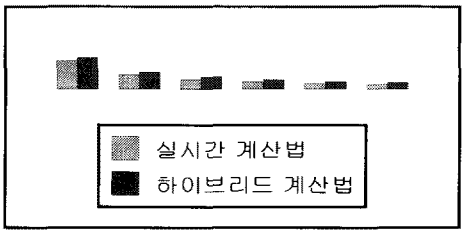


그림 13. 개체수별 프레임 호출 현황 그래프

VI. 결론

대규모 군중 애니메이션에서 장면의 사실성을 높이기 위해서는 행동 유형에 대해 보다 상세하게 적용하여야 하겠지만, 시스템의 성능 향상과는 상반되는 개념으로 동시에 만족시키기는 어렵다. 장면의 사실성을 높이기 위해서는 객체들의 속성, 행동 유형, 그리고 객체와 객체간의 상호작용에 관한 분석이 필요하다.

본 연구에서는 행동 유형을 결정지을 수 있는 주요 속성들에 대해 분석 및 편집이 가능한 Smart Object Modeler를 개발하였으며, 3가지 모델에 대해서 행동 유형에 영향을 미치는 주요 속성인 모양, 탐지 거리, 시야각도, 식성 및 최고 속도 등을 분석 적용하였으며, 연결동작의 자연스러운 처리를 위하여 기본적인 행동 유형은 미리 계산된 자료를 활용하여 시나리오로 작성하여

사용하였으며, 행동 유형의 변화 시에는 실시간 계산을 통해 처리하였다. 하지만 시나리오의 작성 및 편집은 일반적인 문서편집기를 사용함으로 인해 시나리오 변경의 효율이 떨어지는 단점이 있다. 처리 과정은 군중, 집단, 개인으로 구분되는 군중의 개념을 이용하여 사용자가 개인이나 집단에 대해 세부적인 행동 유형을 제어할 수 있도록 하였으며, 행동 유형의 결정에 있어서 공포감, 포만감, 주변 환경 등에 행동 유형 가중치를 적용해 포식자로부터의 출현이나 충돌에 대해 최우선적으로 반응하도록 하였다.

본 시스템은 가상공간을 통해 해저 생태계를 표현하므로 해양박물관이나 해양 생태계를 보여 주고자 하는 장소에 활용되어질 수 있을 것으로 기대된다. 향후 연구과제로는 해저 지형이나 다른 객체와의 충돌회피에 대해 보다 효율적인 알고리즘을 연구하여 해저환경의 사실성을 높이는 것이다.

참 고 문 헌

[1] 안정현, 원광연, *Survey on Crowd Animation*, KAIST Tech Memo, 2003.

[2] C. Reynolds, "Flocks, herds and schools : A distributed behavioral model," ACM SIGGRAPH 87, pp.25-34, 1987(7).

[3] T. Xiaoyuan and T. Demetri, "Artificial Fishes : Physics, Locomotion, Perception, Behavior," ACM SIGGRAPH 94, pp.43-50, 1994(7).

[4] Mark A. Deloura, *Game Programming Gems 2*, Charles River Media, 2002.

[5] http://www.red.com

[6] http://www.riversoftavg.com/flocking.htm

[7] http://www.massivesoftware.com

[8] 김기호, 박재형, 안재우, "가상현실 기법을 이용한 가상수족관 시스템 개발", 대한인간공학회, 학술대회논문집, 제2권, pp.166-170, 1996.

[9] P. Rick, *Computer Animation Algorithms and Techniques*, Morgan Kaufmann, 2002.

[10] 강경헌, 정승문, 이현철, "사이버 아쿠아리움 구축을 위한 어류속성 편집기 개발", 한국콘텐츠학회 추계종합학술대회 논문집, 제4권, 제2호, pp.528-532, 2006.

[11] 남지승, 강미영, 이형옥, "시나리오 기반의 3D 객체 재사용 알고리즘", 한국콘텐츠학회 논문지, 제6권, 제11호, pp.302-309, 2006.

[12] S. Musse, C. Babski, and T. Capin, "Crowd modeling in collaborative virtual environments," ACM VRST 98, pp.115-123, 1998(11).

[13] S. Musse and D. Thalmann, "A model of human crowd behavior : Group inter-relationship and collision detection analysis," Workshop of Computer Animation and Simulation of Eurographics 97, pp.39-51, 1997(9).

[14] 김종찬, 조승일, 김응곤, "가상해저 환경구축을 위한 Fish 군중행동 모델러", 한국정보과학회 학술발표논문집, 제33권, 제2호, pp.158-161, 2006.

저 자 소 개

류 남 훈(Nam-Hoon Ryu) 준회원

- 2007년 2월 : 한국방송통신대학교 컴퓨터과학과(이학사)
- 2007년 3월 ~ 현재 : 순천대학교 컴퓨터과학과 석사과정 재학 중

<관심분야> : 영상처리, 컴퓨터 그래픽스, 알고리즘

반 경 진(Kyeong-Jin Ban) 정회원

- 2003년 2월 : 순천대학교 컴퓨터과학과(이학사)
- 2005년 2월 : 순천대학교 컴퓨터과학과(이학석사)
- 2007년 8월 : 순천대학교 컴퓨터과학과 박사 수료

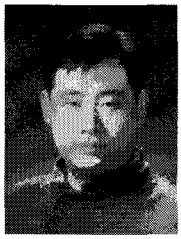
<관심분야> : 컴퓨터 그래픽스, RFID, USN

오 경 숙(Kyeong-Sug Oh) 준회원

- 2007년 8월 : 한국방송통신대학교 컴퓨터과학과(이학사)
- 2007년 9월 ~ 현재 : 순천대학교 컴퓨터과학과 석사과정 재학 중

<관심분야> : 영상처리, 컴퓨터 그래픽스, HCI

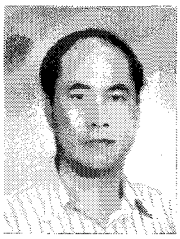
송 승 헌(Seung-Heon Song) 정회원



- 2000년 2월 : 순천대학교 컴퓨터과학과(이학석사)
- 2001년 9월 : 순천청암대학 겸임교수
- 2006년 2월 : 순천대학교 컴퓨터과학과(이학박사)
- 2007년 5월 ~ 현재 : 광양만권 u-IT 연구소 팀장

<관심분야> : 영상처리, 컴퓨터 그래픽스, 멀티미디어, HCI, USN

김 응 곤(Eung-Kon Kim) 종신회원



- 1980년 2월 : 조선대학교 전자공학과(공학사)
- 1986년 2월 : 한양대학교 컴퓨터공학과(공학석사)
- 1994년 2월 : 조선대학교 컴퓨터공학과(공학박사)
- 1993년 3월 ~ 현재 : 순천대학교 컴퓨터과학과 교수

<관심분야> : 영상처리, 컴퓨터 그래픽스, 멀티미디어, HCI