

가상공간상의 객체 간 상호작용기법을 이용한 충돌 회피

The Collision Avoidance with Interaction Technique between Objects in Virtual Space

류남훈*, 반경진*, 오경숙*, 송승헌**, 김응곤*
순천대학교 컴퓨터학과*, 광양만권 u-IT 연구소**

NamHoon Ryu*, KyeongJin Ban*, KyeongSug Oh*,
SeungHeon Song**, EungKon Kim*
Dept. of Computer Science, Sunchon National
University*, Ubiquitous Gwangyang & Global IT
Institute**

요약

컴퓨터 그래픽스 기술의 발달은 영화, 게임 등 각종 엔터테인먼트 산업의 발전으로 이어졌으며, 가상의 공간상에 다양한 객체를 생성하고 동작을 제어하는 기술들도 급속도로 발달하고 있다. 특히 객체(캐릭터)는 컴퓨터 애니메이션 분야에 있어서 가상공간에 생기를 불어넣는데 아주 중요한 역할을 하고 있다. 가상의 공간상에서 많은 수의 객체가 애니메이션 될 경우 고정지형이나 객체 간의 충돌 등의 원치 않는 일이 발생하게 된다. 이런 문제의 해결을 위해서는 세부적인 제어기법이 필요하다. 본 논문에서는 가상의 수족관을 배경으로 하여 해저 지형과의 충돌 회피와 또 다른 객체와의 충돌 회피에 대한 제어 기법을 제시하고 구현한다.

Abstract

The development of computer graphics techniques came after development of various entertainment industry such as movies, game and so on, and techniques that make diverse objects in virtual space and control the action of objects are growing rapidly. Especially objects(characters) play an important role in encouraging virtual space at computer animation field. If lots of objects are animated in virtual space, undesired matters of fixed topography or collision between objects etc occur. In order to solve this problem, we need control techniques in detail. This paper presents and implements collision evasion with submarine topography in a virtual aquarium and control technique about collision evasion with other objects.

I. 서론

컴퓨터 그래픽스 기술의 발달로 인해 영화나 게임 등의 각종 콘텐츠에서 3차원의 객체(캐릭터)를 쉽게 접할 수 있다. 3차원 게임이나 시뮬레이션 프로그램에서 객체의 증가는 게임 시나리오 및 여러 가지 요소의 복잡도를 증가시키게 되고, 객체끼리 충돌하는 등의 의도하지 않은 결과가 발생하기도 한다 [1]. 그러므로 시뮬레이션 시스템은 객체들의 행동 애니메이션을 제어할 수 있는 구조로 설계되어야 한다. 본 논문에서는 객체들 간의 상호작용을 위한 에이전트의 메시지 교환방법을 통한 시스템을 제안한다. 에이전트 시스템은 지능적 에이전트 기반 구조로써 환경 데이터 처리, 객체 애니메이션, 객체들 간의 상호작용 제어, 객체들의 다양한 상호 관계를 정의함에 사용되어진다.

본 논문은 II장에서 상호작용 및 에이전트 기법에 대해 알아보고, III장에서는 상호작용 모델을 설계하며, IV장에서는 상호작용기법을 이용한 가상수족관 시스템의 구성 및 구현된 결과를 보고, 마지막 V장에서는 결론을 맺는다.

II. 관련 연구

3차원 게임이나 시뮬레이션 환경 내에는 많은 수의 객체들이 존재한다. 객체로는 움직이지 않고 고정되어 있는 정적인 객체(건물, 책상, 의자 등)와 이동할 수 있는 동적인 객체(사람, 동물 캐릭터 등)로 나눌 수 있다. 객체들은 게임 시나리오의 전개 속에서 논리적인 규칙에 따라 움직이게 된다. 논리적인 규칙에는 게임 객체들 간의 동적인 의사소통 규칙들과 충돌 탐지 및 회피처리, 경로 탐색 등 이동과 관련되는 알고리즘들이 포함되고, 객체의 속성에는 객체 간의 반응에 따른 지능적 처리와 애니메이션 처리를 위한 행동 유형 등 세부적인 사항들이 포함된다. 게임은 정해진 시나리오 및 규칙에 따라 진행되지만 진행도중 객체의 행동유형의 변화 등 여러 가지 변경요소로 인해 게임 디자이너가 원하지 않은 상황들이 발생하게 된다. 게임에서는 스토리 전달의 문제점이 발생할 수 있으므로 게임의 외부와 내부 모듈에서 상호작용을 유동적으로 처리할 수 있어야 한다.

1. 상호작용 기술

가상 캐릭터와 정적 객체간의 상호작용 기술은 처음 OSR (Object-Specific Reasoning) 모듈을 이용한 자연어 명령 기반 시뮬레이션에서 언급되었다[2]. 그러나 주로 작업을 이해하는 측면만 고려하고 복잡한 여러 환경에서의 객체들 간의 조정이나 상호작용 측면은 고려하지 않았다. 지금까지의 주된 관심은 사용자와 주변 환경 객체 간의 직접적 상호작용이었다 [3-4]. 이것은 객체가 적절한 행동을 가지도록 하지는 않는다. 어떤 시스템은 객체의 기능성만을 강조할 뿐 동적 객체와 정적 객체의 상호작용은 고려하지 않는다[5].

가상 객체의 동작 정보를 정의하고 상호작용 능력을 부여한 기법[6]의 경우 스크립트 기반 동작 오토마타와 시각적 상태 머신을 제안하였다. 그러나 순차적인 행동의 결과에 따른 판단 처리만이 설정되어 있고 3차원 게임 환경과 같은 유동적인 환경에서 사용자 객체나 다른 객체의 추론적, 인지적 행동 처리는 부족한 면을 가지고 있다. 객체 상호간의 제어보다는 동작 생성의 제약 조건에 관심을 두는 경우 서로간의 대화 규칙을 통한 제어관계가 결여되어 있다[7]. 특히 3차원 게임의 경우 전체 게임 시나리오의 전개 시 게임 객체 간 많은 상호작용 등이 요구된다.

이러한 상호작용 관계가 올바르게 없을 때 충돌 등의 의도 하지 않은 결과를 접하게 된다. 그러므로 3차원 게임 환경과 같이 애니메이션의 장면이 유동적으로 변하는 상황에서는 게임 캐릭터들뿐만 아니라, 주변의 정적 객체들 사이에 자신의 개별 동작과 상호작용을 제어하기 위한 조정이 필요하다.

2. 에이전트 기법

게임 캐릭터 객체 자신이 정한 목표를 달성하기 위한 내부적인 인지적 능력과 신뢰성 있고 일관성 있게 다른 가상 객체와 반응할 수 있는 요소들이 필요하다. 지능적, 자율성의 속성을 제공하는 에이전트 기법[8-9]은 각 게임 캐릭터의 동작 상태 변동에 따른 자원을 관리하기 위한 구조적인 방법을 제시할 수 있고, 동적인 환경 변화에 대응할 수 있는 방안을 제공할 수 있다. 가상 인체 모델 에이전트 시뮬레이션을 위한 플랫폼을 제안한 ACE는 특징적 모델링 접근법을 이용하여 에이전트와 객체 간의 상호작용을 디자인하고 스크립트된 모듈을 통해 각 에이전트의 추론 및 행동을 제어하였다[10].

III. 상호작용 모델 설계

에이전트 객체는 정해진 시나리오의 전개 속에서 객체 간의 상호작용을 통한 상황인지 능력, 상황에 따른 행동결정 능력을 갖추고 상황 발생에 따른 메시지 전송 기능을 가지고 있다.

1. 행동 유형 제어

에이전트 기술은 상호작용을 위한 지능적인 가상 객체를 형성하는 컴퓨터 그래픽, 가상현실, 3차원 게임과 같은 곳에서 사용된다. 게임 객체는 에이전트 형태로 인지적 능력을 기반으로 한 행동 단위 속성을 가지고 있다[11]. 에이전트의 인지적 능력을 기반으로 한 지능적 능력은 다른 객체와의 상호작용을 위해 사용된다.

1.1 에이전트 객체

에이전트 객체는 실시간으로 각 에이전트의 정보를 분석하여 처리한다.

1.1.1 에이전트 종류

- 동적 에이전트 객체 : 인지적 능력을 가진 객체로 본 시스템에서는 어류가 해당된다.
- 정적 에이전트 객체 : 해저환경을 이루는 암초나 산호초 등이 해당된다.

1.1.2 에이전트 속성

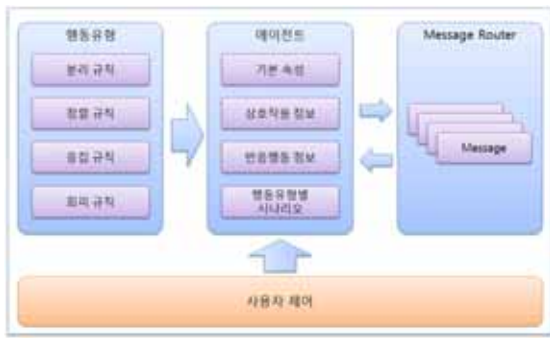
- 기본 객체 정보 : 객체를 구분할 수 있는 이름과 관리를 위한 관리번호, 크기, 속도, 물속에서의 세 축의 좌표 등
- 상호작용 정보 : 다른 객체와의 상호작용을 위한 상태 정보
- 반응 행동 정보 : 상호작용에 따른 에이전트의 반응 행동 정보
- 행동유형별 시나리오 : 행동유형에 따른 이동 경로 및 속도 등

1.2 메시지 처리

에이전트 객체는 의사전달을 위해 메시지를 사용한다[12]. 메시지는 동적 에이전트와의 의사소통을 위한 메시지와 정적 에이전트와의 의사소통을 위한 메시지로 구분된다. 메시지의 내용은 에이전트 간의 의사소통을 위해 객체의 관리번호, 세 축의 좌표, 속도, 행동유형 등의 정보를 포함하고 있다. 메시지는 스크립트를 통한 사용자의 제어나 동적 에이전트의 인지로 인한 충돌 등이 예상될 경우에 발생한다.

2. 상호작용 시스템 구조

에이전트 객체의 상호작용을 통한 경로탐색 및 다른 에이전트와의 상호작용 처리, 충돌 탐지, 상황에 따른 행동유형 결정을 위해 그림 1과 같은 제어 구조가 필요하다.



▶▶ 그림 1. 상호작용 시스템 구조도

IV. 가상 수족관 시스템

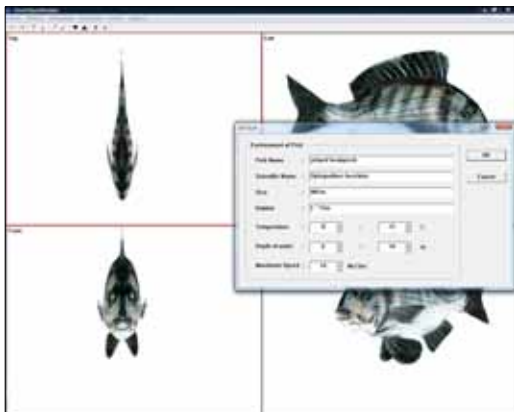
에이전트 객체의 상호작용 기법을 이용한 가상 수족관 시스템은 해저환경 구축 모듈과 어류속성 편집 모듈, 시나리오 생성 모듈, 그리고 행동유형 제어 모듈로 구성된다.

1. 해저환경 구축 모듈

가상 수족관의 해저환경은 정적 객체에 해당하는 암초 및 산호초로 구성되었으며, 이는 각각 2,000 ~ 3,000개 정도의 삼각형으로 모델링되었다.

2. 어류속성 편집 모듈

본 시스템에서는 3 종류의 모델로 어류를 구성하였다. 첫 번째 모델은 동물성 플랑크톤이나 부착조류 등을 섭취하는 초식성 어류이다. 두 번째 모델은 부착조류 및 자신보다 작은 크기의 어류를 섭취하는 잡식성이며, 세 번째 모델은 상어와 같은 육식성 어류이다. 이 모델은 각각 4,000 ~ 5,000 개 정도의 삼각형 형태의 면으로 구성되어 있으며, 꼬리를 흔드는 등의 자연스러운 유영 형태를 표현하기 위해 30개의 프레임으로 모델링하였다. 어류 객체는 객체를 구분하기 위한 기본 정보 및 관리번호, 물속에서의 좌표, 이동 속도 등의 속성을 가지고 있다. 그림 2는 어류 속성 편집 프로그램의 구동 화면이다.



▶▶ 그림 2. 어류 속성 편집 화면

3. 시나리오 생성 모듈

어류는 방추형, 측편형, 원통형, 종편형 등 모양에 따라 각각 다른 특색의 유영 형태를 가지고 있다. 측편형의 경우 꼬리지느러미를 좌우로 흔들며 유영하며, 종편형의 경우 상하로 지느러미를 흔들며 유영한다. 이처럼 어류마다 가지고 있는 유영 특색인 피치, 롤, 요와 같은 세 축으로의 이동 각도 및 속도에 해당하는 거리를 행동유형별로 시나리오로 생성하여 사용한다.

3.1 자유 유영

적으로부터의 위협이 없으며, 정적 객체 및 동적 객체와의 충돌이 예상되지 않는 상태로 가장 일반적인 유영 형태이다.

3.2 응집 유영

어류 객체는 자신이 무리로부터 떨어져 있을 경우 가장 가까운 무리를 향해 모여들게 된다. 이는 자신을 비롯한 많은 객체들이 포식자의 접근을 함께 감시하다가 포식자가 발견되면 순식간에 무리로부터 이탈함으로써 포식자를 혼란스럽게 하여 위협으로부터 탈출하기 위함이다.

3.3 분리 유영

무리를 향해 모여드는 응집 형태로만 유영하게 되면 다른 객체와의 충돌이 발생하게 되므로, 다른 동적 객체와 일정한 간격을 두기 위해 무리의 반대쪽을 향해 유영하는 형태이다.

3.4 정렬 유영

동적인 객체끼리 무리를 지어 다닐 경우 동일한 방향으로 진행하지 않으면 대단히 많은 충돌이 발생하게 된다. 그러므로 무리에 속한 대부분의 객체는 거의 동일한 방향으로 진행을 하게 되며, 선두 그룹의 방향 전환은 곧 무리에 속한 나머지 객체의 방향 전환으로 이어진다.

3.5 리더 따라가기

어느 특정한 객체가 리더가 되어 무리 전체의 흐름을 제어하는 유영 형태이다.

3.6 사냥과 도망

포식자는 배고픔을 느끼게 되면 사냥감을 찾아 나서게 되며, 먹이를 발견하게 되면 기회를 포착하여 최대 속도로 유영하면서 사냥을 하게 된다. 이때 피식자가 포식자를 발견하게 되면 포식자를 피해 도망을 하게 된다. 포식자는 피식자보다 빠른 속도를 이용해 사냥에서 대부분 성공하지만, 피식자가 정적 객체인 산호초 등으로 인해 포식자의 시야에서 벗어나게 되면

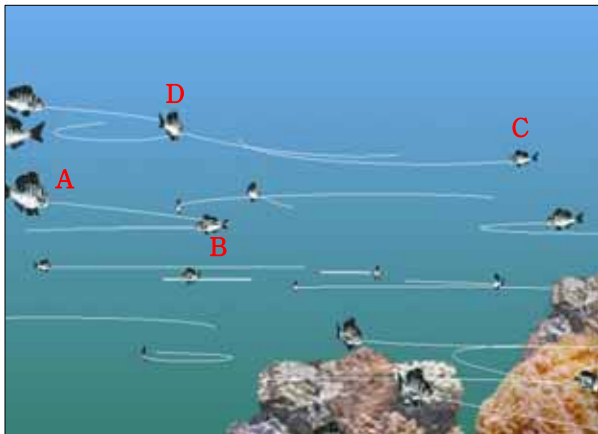
잠시 긴장 상태에서의 유영이 이어지다가 자유 유영 형태나 무리로의 진입을 시도하게 된다.

4. 행동유형 제어 모듈

어류 객체는 각각의 움직임을 제어하기 위해 어류의 기본정보와 유영 형태 등에 대해 초기화의 과정을 거친다. 그런 다음 에이전트를 통해 인지된 상황을 종합하여 정적 객체와의 충돌이나 다른 객체들과의 충돌에 대해 체크하고, 상황에 따라 메시지 라우터로 메시지를 송신한다. 행동 유형 제어 모듈에서는 인지된 상황들 중 가장 위험한 것과 가장 가까이에서 감지되는 것들을 고려하여 결정하게 된다. 객체는 행동 유형이 결정되면 행동 유형에 따른 시나리오대로 행동하게 된다.

5. 구현화면

어류 객체는 에이전트를 통한 상호작용 기법을 이용해 동작을 결정하게 된다. 고정지형인 정적 객체에 대해서는 메시지 라우터에 있는 정적 객체에 대한 메시지를 읽어서 이동 경로를 결정한다.



▶▶ 그림 3. 상호작용 기법을 이용한 충돌 회피

동적 객체에 대해서는 모든 객체가 이동하고 있는 동안 메시지 라우터로 현재의 좌표와 이동 속도, 행동 유형 등을 남기고, 각 객체는 메시지 라우터에서 또 다른 객체의 정보를 가져와서 충돌 등의 위험이 있는지 판단하게 된다. 충돌이 예상될 경우 경로를 수정하게 되며, 충돌의 위험이 적은 경우는 방향을 선회하고, 충돌의 위험이 높은 경우는 속도를 급격히 줄여가면서 방향을 선회하게 된다. 이전에 충돌의 위험으로 인해 방향을 선회하고 있었다면 이전의 회전 방향을 유지하며, 충돌의 위험이 사라지면 점차 속도를 높여서 평상시의 속도까지 높이며, 행동 유형에 따른 시나리오에 따라 이동하게 된다. 그림 3은 에이전트를 통한 상호작용 기법을 이용한 충돌 회피 과정을 보여주고 있다. 화면의 흰 선은 어류 객체에 대해 현 위치부터 100프레임 앞까지의 예상 경로이며, 메시지 정보에 따른 분석에 의해 충돌이 예상될 경우 방향을 선회하게 되며, 충돌의 위험이 없을 경우 예상 경로를 연장하게 된다. 그림에서 객체 A는 메시지 라우터에서 가져온 객체 B의 정보를 분석하여 객체 B와의 충돌을 피하기 위해 경로를 수정한 경우이며, 객체 C도 객체 D와의 충돌을 피하기 위해 경로를 수정한 경우이다.

6. 구현환경

본 시스템은 어류 객체들의 충돌 회피 과정을 비롯한 여러 가지 유영 형태를 보여주는 시스템으로 MFC 환경에서 Visual C++ 6.0과 OpenGL을 이용하여 구현하였으며, 어류 객체는 3D Max를 이용하여 모델링하였다.

V. 결 론

본 논문에서는 객체들 간의 상호작용을 위한 에이전트의 메시지 교환방법을 통해 어류 객체들의 행동 애니메이션을 제어할 수 있는 시스템을 구현하였다. 에이전트는 게임 시나리오 전개 속에서 논리적인 규칙에 따라 움직이게 된다. 객체들은 지능적 에이전트를 통해 정적 객체 및 또 다른 동적 객체와의 사소통을 하며, 충돌 탐지 및 회피에 대해 처리하며 경로를 탐색한다. 에이전트 간의 통신은 메시지 라우터를 통해 스크립트 형태의 언어적 방법을 사용하였고, 사용자의 제어를 통해 행동 유형의 변경이 가능하도록 하였다.

어류 객체를 이용해 에이전트를 통한 상호작용을 구현하기 위해 가상 수족관 시스템을 구현하였으며, 어류의 일반적인 유영 형태인 자유 유영, 응집, 분리, 정렬, 리더 따라가기, 사냥 및 도망 형태의 유영 등을 구현하였다.

향후 연구 과제로는 어류 객체가 접하게 되는 여러 가지 유영 형태를 취하면서 성장 1단계부터 성장 n단계까지의 성장과

정을 보여주는 시스템에 대한 연구가 필요할 것으로 사료된다.

감사의 글

본 연구는 문화관광부 및 한국문화콘텐츠진흥원의 지역문화 산업연구센터(CRC) 지원사업의 연구결과로 수행되었음

■ 참고 문헌 ■

- [1] 안정현, 원광연, "Survey on Crowd Animation", KAIST Tech Memo 2003-4, September 2003.
- [2] L. Levison, "Connecting Planning and Acting via Object-specific Reasoning", PhD thesis, Department of Computer & Information Science, University of Pennsylvania, 1996.
- [3] C. Hand, "A survey of 3D Interaction Techniques", Computer Graphics Forum 16, pp. 269-281.
- [4] 김기현, 김상욱, "3차원 게임에서 객체들의 상호 작용을 디자인 하기 위한 제어 기법", 정보과학회논문지, 제 9권, 제 3호, pp. 322-331, 2003. 6.
- [5] Y. Okada, K. Shinpo, Y. Tanaka and D. Thalmann, "Virtual Input Devices Based on Motion Capture and Collision Detection", Proceedings of Computer Animation 99, Geneva, May 1999.
- [6] M. Kallmann and D. Thalmann, "Modeling Behaviors of Interactive Objects for Real-Time Virtual Environments", Journal of Visual Languages and Computing, pp. 177-195, 2002.
- [7] Wagner da Silva, F., Garcia, L. M. Farias, R. C., Oliveira, A. A. F., "A Control Theory Approach for Real-time Animation of Artificial Agents", Proceedings XIII Brazilian Symposium on Computer Graphics and Image Processing, pp. 211-218, 2000.
- [8] P. Bechieraz, D. Thalmann, "A Behavioral Animation System for Autonomous Actors Personified by Emotions", Proc. Of first Workshop on Embodied Conversational characters(WECC98), Lake Tahoe, California, 1998.
- [9] H. Noser, D. Thalmann, "The Animation of Autonomous Actors Based on Production Rules", "Proceedings of Computer Animation, Geneva, Switzerland, 1996.
- [10] M. Kallmann, J. Monzani, A. Caicedo and D. Thalmann, "ACE: A Platform for the Real Time Simulation of Virtual Human Agents", EGCAS 2000-11th Eurographics Workshop on Animation and Simulation, Interlaken, Switzerland.
- [11] 김기현, 김상욱, "캐릭터 동작 애니메이션 제어를 위한 에이전트 시스템", 정보과학회논문지, 제 7권, 제 5호, pp. 467-474, 2001. 10.
- [12] Chaib-draa, B., Moulin, B., Mandiau, R. and Millot, P., "Chapter 1- Trends in Distributed Artificial Intelligence", Foundations of Distributed Artificial Intelligence, G.M.P.O Hare and N.R. Jennings (eds.), pp. 3-55, John Wiley & Sons Inc., 1996.