

3차원 퍼즐 시스템을 위한 구조물 조립 기법

김진모[†], 조형제^{**}

요 약

컴퓨터 하드웨어의 급격한 발전과 다양한 게임 기술의 발전으로 인해 게임 품질이 향상되고 게임 장르도 다변화 되었다. 이에 비해 퍼즐게임은 기존의 단순한 게임방식을 고수함으로써 점점 위축되고 있다. 따라서 본 연구는 현재 퍼즐게임의 한계를 벗어날 돌파구를 3차원 퍼즐게임에서 모색하려는 첫 시도이다. 3차원 퍼즐게임에서의 사실적인 퍼즐 조립을 표현하기 위해 조립에 적합한 구조물 처리 과정과 객체간의 정교한 충돌 방법, 물리구조를 활용한 3단계로 구성된 구조물 조립 기법을 설계하고 이를 통해 조립과정에서 연산 비용을 최소화하면서도 구조물 간의 정교한 퍼즐 조립이 가능함을 간단한 실험으로 보인다.

Structure Assembling Method for 3D Puzzle System

Jinmo Kim[†], Hyungje Cho^{**}

ABSTRACT

With the rapid development of computer hardware and invention of various game-related technologies, the quality of games has been greatly improved and the genres of games have been diversified. Yet, in comparison, puzzle games have held to the past simple way of games, which has only led to a depression of the puzzle game market. This paper is the first attempt to overcome limits of the current puzzle games, by finding a breakthrough in three-dimensional puzzle games. In order to describe a realistic puzzle assembly in a three-dimensional puzzle game, this paper aimed to design a structure assembly technique made up of three steps that used a proper process to handle structure suitable to assembly, an elaborate way of collisions between individual objects and a physical structure. Through this technique, we intend to show in a simple experiment that it is possible to minimize the calculation cost in the assembly process and at the same time to achieve an elaborate puzzle assembly between structures.

Key words: 3D Puzzle-Game(3차원 퍼즐게임), Collision Detection(충돌 검출), Assembling Structure(조립 구조)

1. 서 론

컴퓨터 게임은 1970년대 이후 등장한 새로운 여가 형태의 하나로 '기억능력이 있는 실리콘 칩 컴퓨터 회로에 작동되는 모든 놀이의 총칭으로 정의할 수 있다[레퍼런스: 브리테리커]. 전자오락 게임, 비디오 게임 등 다양한 용어들과 함께 사용되는 컴퓨터 게임

은 전 세계적으로 아동, 청소년뿐만 아니라 성인들도 즐기는 여가 활동의 하나로 인식되고 있으며 그 정도는 시간이 흐를수록 증가되는 추세이다[1].

컴퓨터 하드웨어 기술과 컴퓨터 그래픽스 기술의 급격한 발전으로 인하여 컴퓨터 게임은 2D 스프라이트(sprite)와 텍스트 위주의 단순한 2차원 게임에서 MMORPG (Massively Multi-player Online Role

* 교신저자(Corresponding Author): 조형제, 주소: 서울특별시 중구 필동 3가 26번지(100-715), 전화: (02)2260-3344, E-mail: chohj@dgu.edu
접수일: 2008년 8월 20일, 완료일: 2008년 11월 28일

[†] 준회원, 동국대학교 멀티미디어학과 박사과정
(E-mail: mythsjin@dongguk.edu)

^{**} 정회원, 동국대학교 멀티미디어학과 교수

Playing Game), 전략 시뮬레이션, FPS(First Person Shooting), 캐주얼 등 다양한 장르의 게임으로 발전되고 있다[2]. 이러한 여러 분야로의 발전과 더불어 게임을 즐기는 고객층의 폭이 넓어지면서 단순하면서 쉽게 접할 수 있는 캐주얼 게임이나 웹보드 게임의 관심도 높아지고 있다[3]. 그러나 퍼즐게임의 경우 기존의 게임방식과 구성에서 크게 벗어나지 못하고 다양한 시선에서 접근이 이루어지지 못하면서 점차 관심이 낮아지고 있다. 뿐만 아니라 MMORPG를 비롯한 다른 장르의 게임들은 콘텐츠의 양은 물론 기술적으로도 다양한 시도와 깊이 있는 연구를 통해 많은 발전을 이루고 있는데 반하여 퍼즐게임과 같은 웹보드 게임은 전혀 다양화되지 못하고 있다[4]. 이러한 문제점을 보완하고 차별화된 퍼즐게임을 제작하기 위한 하나의 방안으로 본 연구에서는 3차원의 사실적인 조립식 퍼즐 시스템을 고안하였다.

정교하면서 사실적인 3차원 퍼즐 시스템을 제작하기 위해서는 두 가지 조건이 고려되어야 한다. 첫 번째는 구조물간의 세밀한 충돌 검출을 얼마나 효율적으로 처리하는지 여부이다. 조립과정에서 정교한 충돌 검출로 인한 계산 비용이 커진다면 지연(delay) 문제로 인해 사용자는 정상적인 조립을 진행해나가기 어려울 것이다. 두 번째로 조립을 하고 있다는 사실적인 느낌을 사용자가 느낄 수 있게 하는 것이다. 만약 사용자가 실제와 같은 조립과정처럼 느끼지 못한다면 게임에 대한 흥미가 떨어질 것이다. 이러한 이유로 구조물들 사이의 효율적인 충돌 검출 방법이 정의되어야 한다. 하지만 현재 게임에서 사용되는 충돌 검출 방법만으로는 제안한 퍼즐 시스템의 구현에 한계가 있다. 기존의 단일 객체간의 충돌 검출 방법은 크게 두 가지가 존재하는데 충돌 검출에 소모되는 비용이 적어 효율은 좋지만 객체의 미세한 부분까지 충돌을 고려하지 못하는 경계불림을 이용한 방법과 객체의 정교한 부분까지 충돌 검출을 해나가지만 많은 비용을 소모하는 방법이 있다[5]. 각각의 단점들로 인하여 구조물간의 사실적인 조립과정을 진행해나가기에는 두 가지 방법 모두 문제가 존재한다.

앞서 언급한 충돌 검출 방법의 한계를 극복하고 사실적인 조립 표현을 위해 본 연구에서는 실제 조립 과정에 앞서 조립에 최적화된 구조물 설정을 위한 전처리(pre-processing)과정과 최적화된 구조물의 사실적이며 효율적인 조립을 위한 정렬(alignment),

결합(assembly), 완성(completion)의 3단계로 구성된 조립처리(assembly processing)과정인 구조물 조립 기법을 도입하였다. 이는 다양한 형태로 제공되는 구조물 사이의 조립에 있어 최적화된 조립을 진행하기 위해 설계한 알고리즘이다. 이를 통해 3차원 환경에서 조립에 사용되는 각기 다른 구조물을 효율적으로 관리할 수 있는 기능을 제공할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 현재 퍼즐게임의 동향과 발전방향에 대한 조사와 현재 게임에서 사용되는 다양한 충돌 검출 방법에 대해 분석한다. 3장에서는 조립과정 이전에 조립에 적합한 구조물 설정 작업인 전처리 과정과 사실적이고 효율적인 조립처리 과정을 위해 설계한 구조물 조립 기법을 설명한다. 4장에서는 설계한 조립 기법을 게임에 적용하여 가능성을 실험하고 구조물간 조립과정에서 일반적인 충돌 검출 방법과 설계한 조립 기법간의 성능을 비교 분석한다. 마지막 5장에서는 결론을 맺는다.

2. 연구배경 및 관련연구

2.1 퍼즐게임의 동향 및 발전 방향

국내 게임시장은 초기에 MMORPG 장르를 중심으로 게임에 투자하는 시간이 많은 일부 고객층을 대상으로 발전해왔다. 하지만 기술이 발전하고 게임에 대한 관심이 높아지기 시작하면서 더욱 다양한 장르의 게임을 개발하기 시작하였다. 2005년 '카트라이더', '프리스타일'과 같은 캐주얼 게임이 출시되면서 게임은 더 이상 특정한 고객층이 아닌 누구나 가볍게 즐길 수 있는 형태로 기획, 제작되기 시작하였다. 이러한 분위기 속에서 누구나 친근하게 느낄 수 있는 캐주얼 게임이나 쉽게 즐길 수 있는 웹보드 게임에 대한 관심이 높아졌으며 국내 여러 게임 개발사들은 웹보드 게임을 기반으로 한 게임 포털사이트를 차례로 오픈하면서 본격적으로 캐주얼 게임의 다양화를 꾀하기도 하였다. 또한 2007년 게임을 즐기는 이용자의 선호 장르를 살펴보면 전체이용자의 19.6%가 캐주얼 게임과 웹보드 게임을 선호하고 있음을 그림 1을 통해 알 수 있다.

여러 방면으로 게임 산업이 발전하면서 현재 제작되는 게임 콘텐츠들은 많은 성장을 이루고 있다. 이런 상황 속에서 웹보드 게임 중 특히 퍼즐게임의 경

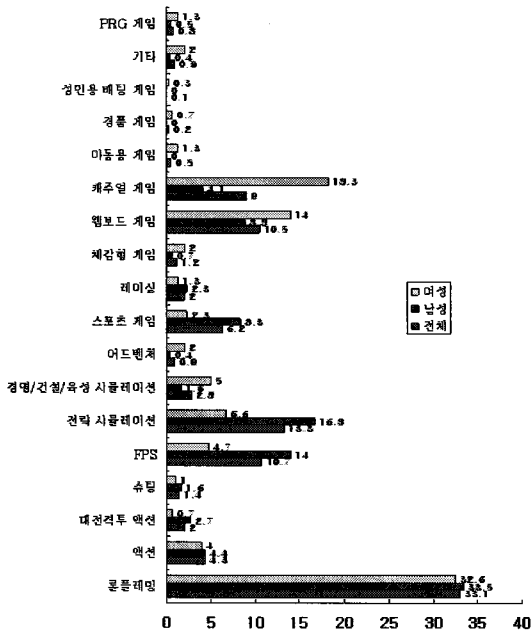


그림 1. 게임 이용자의 선호 장르(3)

우 발전이 거의 이루어지지 않으며 현재는 출시되는 게임이 손가락에 꼽힐 정도로 퇴보하고 있다[4]. 테트리스, 핵사, 조각 맞추기, 틀린 그림 찾기 등 대부분의 퍼즐게임은 과거에 제작되던 퍼즐게임의 방식과 전혀 차별화 되지 못하고 있고 그래픽적인 요소만 신경을 쓰고 있으며 기존의 방식에 아이템만 몇 개 추가하여 게임의 재미요소를 부가시키는 데에 그치고 있기 때문에 게임을 즐기는 이용자들은 쉽게 식상해질 수밖에 없다. 또한 최근에 제작되는 대부분의 장르들이 3차원 환경에서 새로운 접근과 다양한 시도를 하는데 반하여 퍼즐게임 장르는 유독 발전 속도가 미미하고 대부분의 게임이 2차원의 단순한 구조를 벗어나지 못하고 있는 실정이다[6](그림 2참조). 최근에는 Wii와 같은 체감형 게임 플랫폼의 등장과 안경 착용 식 입체영상을 활용한 사실적인 게임의 개발 등 여러 분야와의 접목을 통한 새로운 시도가

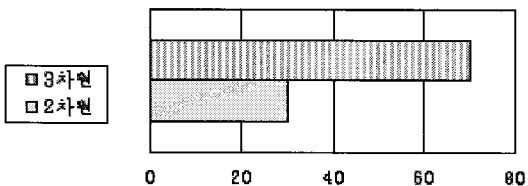


그림 2. 국내 100위 이내의 3차원 게임의 비율(6)

이루어지고 있어 이러한 추세를 반영하여 퍼즐게임 분야 또한 게임을 즐기면서 퍼즐 조각들이 맞추어지는 상황들이 입체적으로 느껴진다면 게임에 대한 몰입도가 높아지고 흥미를 더욱 유발시킬 수 있을 것이다. 따라서 퍼즐게임 장르 역시 발전하고 변화하는 현 게임시장과 더불어 입체적이고 사실적인 퍼즐 시스템으로 발전해나간다면 퍼즐게임 장르의 다양한 발전 방향을 모색할 수 있는 하나의 계기가 될 것으로 생각된다.

2.2 게임에서의 충돌 검출 기법

3차원 공간에서 입체적이며 사실적인 퍼즐게임을 제작하기 위한 중요한 요소는 조립과정이 이루어지는 객체들 사이의 충돌 검출 방법이다. 일반적으로 현재 게임에서의 단일 객체의 충돌 검출 방법은 크게 두 종류로 나눌 수 있다. 계산속도 개선을 위한 방법으로 경계볼륨을 통한 충돌 검출 방법, 정교한 충돌 검출을 위한 점과 점, 점과 선, 삼각형과 삼각형 등의 객체를 이루는 기본단위에서의 충돌 검출 방법 이 두 가지 방법이다[5]. 여기에 이러한 방법을 좀 더 개선하기 위해 객체 또는 경계볼륨을 트리구조나 그룹 등으로 관리하는 방법들도 있다[7,8].

2.2.1 경계볼륨을 통한 충돌 검출

객체를 둘러싸는 구나 상자 형태의 경계볼륨을 생성하여 객체간의 직접적인 충돌 검출이 아닌 경계볼륨간의 충돌 검출을 다루는 방법이다.

(1) 경계 구와 경계 구

경계구의 중심점과 반지름을 이용하여 경계구의 중심 간의 거리가 경계구의 반지름의 합보다 작게 되면 충돌 검출로 판별하는 방식이다[5].

(2) 경계상자와 경계상자-1 (AABB - AABB)

경계상자를 이용하여 충돌을 검출하는 방법은 두 가지로 나누어 생각할 수 있다. 우선 그 첫 번째 방법으로 경계상자의 각 모서리가 3차원 월드 좌표계의 X,Y,Z축에 평행한 모양을 하고 있는 AABB(Axis-aligned Bounding Box)방식이다. AABB의 충돌 검출은 8개의 꼭지점이 다른 상자 내부에 포함되어 있는지 여부로 계산한다. X,Y,Z 세 개의 축으로 투영하여 세 개의 축 모두 겹치는 부분이 발생하면 충돌로

판정하는 방법이다[9].

(3) 경계상자와 경계상자-2 (OBB - OBB)

경계상자를 이용한 두 번째 방법으로 3차원 환경에서 객체의 좌표축을 기준으로 경계상자를 생성하는 OBB(Oriented Bounding Box)가 있다. 이는 게임 속에서 자주 움직이는 객체에 충돌 검출을 하는 경우 쓰이는 방법이다. 객체 각각의 OBB는 3개의 면의 법선벡터, 3개의 선의 방향을 가지고 있으므로 $3+3+3*3 = 15$ 개의 축을 분리축으로 사용한다. 총 15개의 축으로 각 객체를 투영하여 모두 겹치는 부분으로 확인되면 충돌로 판정한다[10].

2.2.2 기본 구성단위의 충돌 검출

(1) 선과 삼각형 평면과의 충돌 검출

두 점 사이의 선분을 매개변수 t 를 도입한 매개방정식을 구하고 이를 주어진 삼각형을 표현하는 평면방정식에 대입하여 구해진 t 값이 0~1사이의 값으로 존재한다면 선과 삼각형 평면이 교차하는 것으로 판정할 수 있다[5].

(2) 삼각형 평면간의 충돌 검출

각각의 객체들은 객체를 이루는 점들이 삼각형을 형성하고 이들 삼각형들의 연결로 그 형상이 드러난다. 따라서 객체들 간의 정교한 충돌 처리를 위해서는 객체를 이루고 있는 삼각형간의 충돌 검출이 필요하다. 이를 위해서는 삼각형의 평면방정식($Ax + By + Cz + D = 0$)을 구하고 평면방정식과 충돌 검출을 할 다른 객체의 삼각형의 선분과의 교점을 구하여 해당 교점이 삼각형 내부에 존재하는 지 여부를 통해 충돌 검출을 한다[5]. 이는 삼각형단위의 정교한 충돌 검출이 가능하지만 객체를 이루는 삼각형의 개수가 많아질수록 충돌 검출과정에서 큰 부하를 가져올 수 있다[5,11].

그 밖에도 충돌 객체 그룹설정, Sweep 알고리즘 그리고 트리구조 설계 등 효율적인 충돌 검출 방법들이 존재한다. 하지만 기존의 방법들은 효율적 연산 시간과 정교한 충돌 검출 두 가지를 모두 만족해야 하는 본 퍼즐 시스템에 적용하기에는 한계가 있다. 이러한 한계를 극복하기 위한 방법으로 본 연구에서는 구조물 조립 기법을 도입한다.

3. 연구 내용

본 연구에서는 기존의 퍼즐 시스템과 차별성을 지닌 새로운 방향의 퍼즐게임을 제작하기 위한 하나의 방법으로 입체적이고 사실적인 조립 형태의 3차원 퍼즐 시스템을 설계함을 목표로 한다. 따라서 본 연구의 핵심은 조립과정을 거치는 구조물들 사이의 정교한 충돌 검출과 조립과정에서 구조물들의 사실적인 움직임을 보여주는 것이므로 이 두 가지 목적을 효과적으로 처리하기 위하여 두 과정을 거친다. 우선 구조물들 사이의 조립과정은 조립을 하는 능동적 돌출부와 조립이 되어지는 수동적 도입부의 결합으로 정의될 수 있다. 하지만 돌출부와 도입부는 사람이 직관적으로 알 수 있는 것과 달리 가상환경에서 컴퓨터는 이들 정보를 확인할 수 있는 방법이 없으므로 조립에 앞서 최적화된 돌출부, 도입부 정보를 설정하는 과정이 첫 번째인 전처리 과정이며 이는 객체 편집기에서 이루어진다. 다음으로 최적화된 구조물들 사이의 효율적이며 사실적인 조립을 위해 3단계의 구조물 조립 기법을 도입한 조립처리 과정이다.

아래에서 두 처리 과정을 통해 사실적인 표현과 정교한 검출이 동시에 적용되는 조립과정에 대해 설명한다(그림 3참조).

3.1 객체 편집기를 통한 전처리(pre-processing)

조립에 가장 적합한 형태의 구조물 정보를 추출하는 과정으로서 이 작업을 효과적으로 처리하기 위한 사전 편집도구가 객체 편집기이다. 이는 구조물을 이루는 돌출부와 도입부의 정점, 인덱스, 삼각형 구성 정보를 조립이 진행되기 전에 미리 설정할 수 있도록 저장, 관리하는 시스템이다.

객체 편집기의 작업구조를 살펴보면 우선 조립에

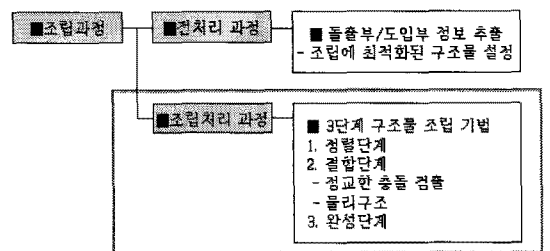


그림 3. 조립과정 내부 구조

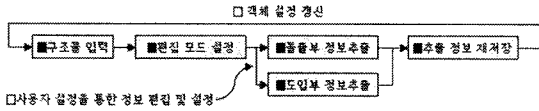


그림 4. 객체 편집기 작업구조

사용되는 구조물들을 차례로 불러온다. 이들을 3차원 가상공간에 배치하고 돌출부와 도입부를 순차적으로 사용자의 키보드와 마우스를 통해 설정한다. 각각의 정보가 단계적으로 설정되고 나면 임시버퍼에 저장된 구조물의 돌출부와 도입부 정보를 정점, 노말, 텍스처 좌표 등으로 구성된 구조물 정보에 추가로 저장한다. 이러한 과정을 반복하면서 조립에 사용될 구조물의 조립 정보를 효과적으로 관리할 수 있는 사전 준비를 완료하게 된다. 그림 4는 이러한 객체 편집기의 작업구조를 나타낸 것이다.

전처리 과정은 조립에 최적화된 구조물을 설정하는 것만을 목표로 하기 때문에 그 과정이 단순하면서도 빠르게 진행되어야 한다. 따라서 객체 편집기 인터페이스 또한 간결하며 정보 추출 과정도 빠르게 진행될 수 있어야 한다.

그림 5의 (a)는 객체 편집기의 화면 인터페이스를 보여주는 것으로 메뉴는 꼭 필요한 두 가지 옵션으로 나누었음을 확인할 수 있다. (b)는 화면의 구조물로부터 객체 편집기를 사용하여 돌출부/도입부의 정보를 4단계로 간결하게 추출함을 확인할 수 있다.

3.2 구조물 조립 기법을 도입한 조립처리(assembly processing)

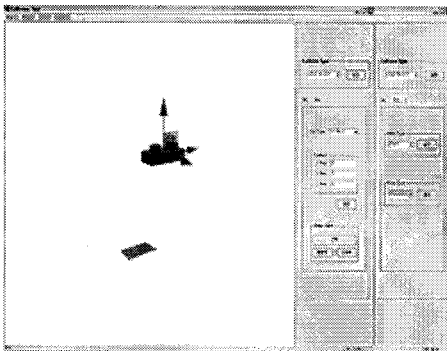
전처리 과정을 통해 조립에 적합한 구조물 설정이

완료되면 가상환경 내에서 이 구조물을 불러들여 조립이 진행될 수 있다. 조립과정은 우선 정교한 결과에 앞서 구조물간의 간격이 결합되기에 충분한 거리 내에 존재하는 가를 AABB 충돌 검출 방식을 통해 판정한다. 이 조건을 만족하면 구조물들 사이의 조립 과정이 진행되는데 이때 본 연구에서 설계한 3단계 구조물 조립 기법이 적용된다.

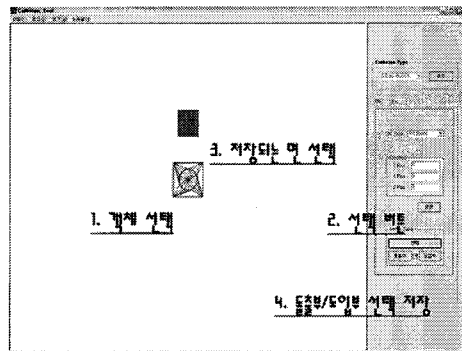
구조물 조립 기법에서는 돌출부와 도입부 사이의 충돌 검출 방법과 구조물 간의 물리구조를 중심으로 정교함과 효율성 그리고 사실적 표현을 목표로 설계 된다.

3.2.1 조립 기법의 3단계 기본 구조

구조물 조립은 세 가지 단계를 거쳐서 실행이 된다(그림 6 참조). 우선 첫 번째 단계는 구조물의 돌출부와 도입부가 조립에 올바르게 배치되었는지를 판단하는 정렬단계이다. 만약 올바르게 정렬이 되지 않았다면 다음 단계로 진행되지 않고 구조물의 외벽 기리 맞닿아 조립은 진행되지 않는다. 반대로 구조물이 올바르게 정렬이 되었다면 두 번째 단계인 구조물을 끼우거나 돌리는 등의 작업이 수행되는 결합 단계로 넘어간다. 이 단계에서 전처리 과정을 통해 설정된 구조물의 돌출부와 도입부와와의 충돌 연산이 실행되고 사실적인 결합을 위한 물리연산이 추가된다. 마지막으로 결합단계를 거쳐 두 구조물이 완전히 결합이 되었는지를 판단하는 완성단계를 통해 3단계의 구조물 조립이 종료된다. 다음은 두 구조물이 근접하여 조립이 되기까지의 과정을 나타낸 코드이다.



(a)객체 편집기 인터페이스



(b) 돌출/도입부 선택 단계

그림 5. 객체 편집기와 돌출/도입부 선택 단계의 예

```

< 3단계 구조물 조립 기법 기본 구조 >
if( Object→BoundingBoxCollision() ) // 구조물이
결합될 근접 범위에 들어왔다면 수행
{
    if( Object→GetAlignment() ) { // 구조물이
    올바르게 정렬이 되었는지 판별
        if( Object→GetCompletion() ) // 구조물이
        결합이 완료되었다면 종료
        else Object→SetAssembly(); // 결합이 완
        료되지 않았다면 결합단계를 수행
    }
    else {
        // 구조물이 정렬이 되지 않았다면 구조물의
        외벽 충돌 검사만 수행
    }
}
    
```

3.2.2 결합시의 돌출부와 도입부 사이의 충돌 판정

구조물 조립 기법이 적용되지 않는 범위에서의 충돌 검출은 정교함을 필요로 하지 않기 때문에 간단히 경계상자를 이용하여 충돌 검출을 실행한다. 하지만 구조물이 결합을 하기위한 단계로 들어섰을 때에는 정교한 결합을 위해 세밀한 충돌 검출이 필요하다. 여기서 사용되는 충돌 검출은 기존의 삼각형단위의 교차 판정 방법을 응용하여 다음과 같이 충돌 검출을 진행한다.

우선 두 개의 삼각형을 각각 삼각형-1, 삼각형-2로 정의하면 우선 삼각형-1의 평면방정식을 구한다. 평면방정식의 일반형은 $Ax + By + Cz + D = 0$ 이고 삼각형의 두 벡터의 외적을 얻으면 그 값은 A, B, C 와 일치한다. $D = -(Ax + By + Cz)$ 으로 x, y, z 에 폴리곤의 정점 중 하나를 선택하여 대입하면 D 의 값을 구할 수 있다. 다음으로 삼각형이 포함되는 평면에 삼각형-2의 세 선분들이 교차되는지를 판정한다. 삼각형-2의 선분을 이루는 두 정점을 선택하여 $X_0(x_0, y_0, z_0)$,

$X_1(x_1, y_1, z_1)$ 라고 하면 두 점을 지나는 직선상의 임의의 정점 $P(x, y, z)$ 는 식(1)로 정의된다.

$$X_0(x_0, y_0, z_0) * t = X_1(x_1, y_1, z_1) * (1-t) \tag{1}$$

식(1)의 매개 변수 방정식을 평면의 방정식에 대입하면 식(2)를 얻을 수 있다.

$$A*(x_0*t + x_1*(1-t)) + B*(y_0*t + y_1*(1-t)) + C*(z_0*t + z_1*(1-t)) + D = 0 \tag{2}$$

이를 t에 대해 정리하면 식(3)이 된다.

$$t = -(A*x_1 + B*y_1 + C*z_1 + D) / (A*(x_0 - x_1) + B*(y_0 - y_1) + C*(z_0 - z_1)) \tag{3}$$

이를 이용해서 t값을 구하면 우리는 선분 X_0X_1 와 삼각형-1을 포함하는 평면이 교차하는 지점 t를 구할 수 있게 된다. 이때 t값은 반드시 0과 1사이의 값이어야 하며 그렇지 않다면 선분과 평면은 교차하지 않는 것이다.

삼각형-1과 앞서 찾은 교점을 XY, YZ, ZX 평면 중 하나에 투영시킨다. 평면에 투영된 삼각형과 교점과의 관계를 계산하여 삼각형-1과 삼각형-2와의 교차 여부를 확인한다. 이후 평면에 투영된 삼각형의 각 정점들을 이용해 삼각형을 이루는 세 직선의 방정식을 구성하고, 삼각형의 중점 $((X_0 + X_1 + X_2)/3, (Y_0 + Y_1 + Y_2)/3)$ 을 구한다. 이 점은 삼각형의 내부가 어떤 방향인지 알려주는 하나의 기준으로 쓰인다. 교점을 대입했을 때 삼각형을 이루는 세 선분 내부에 모두 존재해야만 삼각형 내부에 존재한다고 확신할 수 있다[11-13].

3.2.3 결합시의 구조물 사이의 물리구조

결합단계에서 구조물 조립부의 정교한 충돌 검출 만으로는 정교한 조립은 가능하지만 사실적인 조립을 표현할 수 없다. 결국 현실감 있는 조립과정을 표

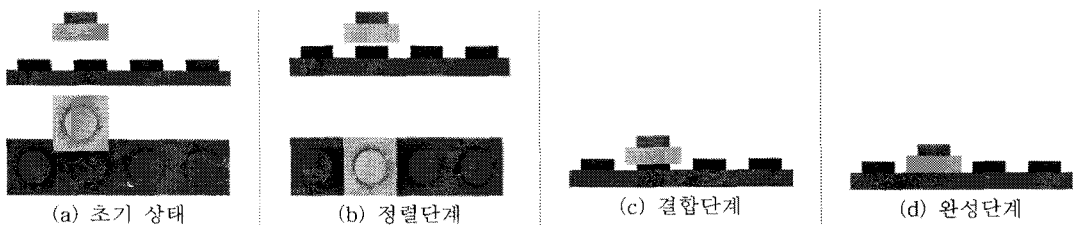


그림 6. 3단계 구조물 조립의 진행과정

현하기 위해서는 조립과정에서 구조물 간의 물리구조가 설계되어 있어야 한다. 따라서 다양한 조립 형태를 갖춘 구조물의 성격에 맞추어 물리구조를 설계하여 사용자가 손쉽게 적용할 수 있는 방법으로 구조물 조립 기법을 구현하여야 한다.

본 연구에서는 끼워 맞추는 블록 구조물과 맞물려 조립하는 나사 구조물 두 가지 형태만을 우선적으로 실험하였다.

첫 번째는 끼워 맞추는 블록 구조물로 돌출부와 도입부가 끼워 맞춰지는 과정에서 실제 블록 면에 마찰의 개념을 적용하여 힘을 주어 꼭 끼우는 느낌을 표현한다. 이 때 쓰이는 운동 마찰력은 식(4)와 같이 정의된다.

$$F_k = -\mu_k N \tag{4}$$

여기서 N 은 물체가 표면에 가하는 힘의 법선 성분이며 μ_k 는 운동 마찰계수로 플라스틱의 재질에 맞게 설정하여 마찰 성분이 도입된 블록의 느낌을 연출할 수 있다.

두 번째로 나사 구조물의 경우 정렬단계가 지나면 회전을 하며 돌출부와 도입부가 맞물려 돌아 들어가게 되는데 이를 표현하기 위하여 간단한 식을 정의하였다.

나사가 360° 회전하였을 때 나사의 이동거리가 이 음세를 한 바퀴 돌아 이음새간의 거리(p)의 2배 만큼 이동하게 되므로 회전 각도를 θ , 이음새 간의 거리를 p 라 할 때 나사의 진행거리 D 는 식(5)와 같이 정의할 수 있다(그림 7 참조).

$$D = \theta / 360^\circ * 2p \tag{5}$$

3.2.4 구조물 조립의 최종 단계

전처리 과정을 통해 설정된 돌출부와 도입부의 정점과 인덱스 정보를 통해 구조물간의 정렬 상태를 확

인한 후 두 번째 단계인 결합단계로 넘어가는데 이 때 앞서 설계한 정교한 충돌 검출이 적용된다. 또한 결합되고 있는 구조물이 어떤 종류의 구조물인지 판별하고 조건에 맞는 물리구조가 적용된다. 최종적으로 결합이 완성되면 조립과정은 종료된다. 다음은 이런 세부적 조립 진행 과정을 코드로 정리한 것이다.

```
bool GetAlignment() // 정렬단계 확인 함수
{
    // 구조물-1의 돌출부와 구조물-2의 도입부가 정렬이 되었다면
    // 또는 구조물-2의 돌출부와 구조물-1의 도입부가 정렬이 되었다면
    // 정렬 상태를 반환한다.
    if(check( Object1->bit, Object2->leadin )
        || check( Object2->bit, Object1->leadin ) )
        return true;
    return false;
}

void SetAssembly() // 결합단계 진행 함수
{
    if(Type == Block) // 블록 구조물간의 결합이라면
    {
        // 주어진 구조물의 정지 마찰력 이상의 움직임이 들어온다면 결합을 진행
        if(Object->move > Object->friction)
            Move(Object->move, direction); // 해당 방향으로 이동
    }
    if(Type == Spiral) // 나사 구조물간의 결합이라면
    {
        angle = Object->move;
        distance = angle / 360 * 2 * Spiral->p;
        RotationAndMove( angle, distance);
    }
}
```

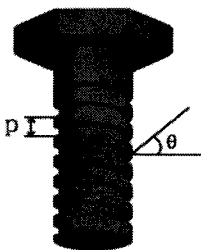


그림 7. 나사의 회전 성분

설계한 구조물 조립 기법을 통해 실제 구조물간의 사실적인 조립과정이 이루어지는 지 여부를 가상의 3차원 공간에 구조물들을 배치하여 확인해 보았다. 아래 그림 8과 같이 블록 구조물의 경우 돌출부와 도입부가 정확히 정렬이 되면 도입부의 홈으로 돌출부가 끼워지며 결합됨을 확인할 수 있다. 그리고 나사 구조물의 경우 정확한 정렬이후 맞물려 돌아가며 결합이 되는 것도 확인할 수 있다. 각 조립부분의 아

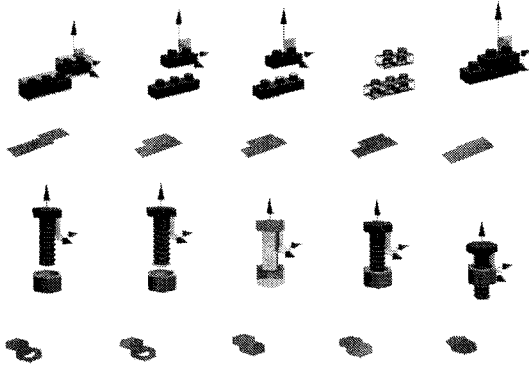


그림 8. 구조물의 조립과정의 예시

래에 나타난 것은 조명을 위쪽에 둔 경우의 그래픽 객체에 대한 그림자이다.

4. 실험 및 성능 평가

본 논문에서 제안하는 기법의 시뮬레이션 시스템 제작을 위해 Microsoft Visual Studio 6.0과 DirectX SDK 9.0b를 이용하여 프로그래밍 하였고 Adobe Photoshop CS, 3DS Max 8.0을 사용하여 그래픽 작업을 진행하였으며 조립 작업을 수행하는 구조물들 사이의 사실적이고 효율적인 진행을 위하여 설계한 3단계 구조물 조립 기법을 적용하였다.

구조물 조립 기법 설계의 1차적 목표는 다양한 퍼즐게임 제작을 위한 하나의 방향을 제시하는 데에 있어서 사실적인 조립과정 연출과 게임에서 사용 가능한 효과적인 연산이 가능한가를 점검하기 위함이다. 이를 위해 설계한 구조물 조립 기법을 실제 게임 인터페이스에 맞추어 간단한 형태의 3차원 퍼즐게임에 적용해 봄으로써 가능성을 실험해 보았다.

또한 정교한 조립과정을 효율적으로 처리하기 위하여 설계한 구조물 조립 기법이 게임에서 정교한 충돌 검출을 위해 사용되던 일반적인 검출 방법과 비교하여 얼마나 성능이 개선되었는지를 초당 프레임 수 비교를 통해 정량적으로 확인하였다.

4.1 구조물 조립 기법을 적용한 퍼즐 시스템 제작 실험

전처리 과정과 조립처리 과정을 통해 진행되는 구조물 조립의 사실적이며 효율적인 처리를 위해 새롭게 설계한 구조물 조립 기법을 단순하게 제안이 아닌 실제 퍼즐 시스템에 사용될 수 있는 형태로 구현하여

그 가능성을 실험하였다.

전체적인 퍼즐 시스템은 게임 제작 구성 요소인 기획, 그래픽, 프로그래밍 단계를 병렬적으로 진행하였다. 먼저 기획단계는 완성된 하나의 조립품을 설정하고 주어진 조립 매뉴얼을 통해 사용자가 단계적으로 실제 조립과정과 유사하게 진행하여 사용자가 현실세계에서 조립하는 것과 동일한 과정으로 가상공간에 표현한다.

게임 진행은 그림 9와 같은 흐름으로 진행된다. 먼저 사용자는 화면의 조립 설명서(manual)를 통해 조립진행과정을 확인한 후 조립구조물들이 담겨있는 저장 공간에서 필요한 구조물을 선택하여 가상의 공간에 불러온다. 이를 3차원 가상공간에 배치하고 마우스를 사용하여 구조물들을 원하는 위치에 조립을 진행한다. 이러한 과정을 반복하면서 최종 완성품이 조립될 때까지 진행한다.

전체적인 퍼즐 시스템 진행 흐름을 기획하고 최종 완성품 모델이 결정되면 완성 모델에 필요한 각각의 구조물 제작 작업을 진행한다. 구조물의 종류는 앞서 설계한 두 가지 형태로 블록 구조물과 나사 구조물을 기반으로 이들을 변형하여 제작하는 것을 기본으로 한다. 그림 10은 제작된 기본형의 구조물이다.

기획과 그래픽 제작이 완료되면 마지막으로 실험 목적에 부합한 프로그래밍 로직(logic)을 설계한다. 프로그래밍 흐름의 첫 번째는 초기화 단계로 구조물 입력, 카메라 설정, 그래픽 장치 설정 등 퍼즐 시스템에 필요한 모든 구성요소들을 설정하는 단계이다. 다음 단계는 전처리 과정을 통해 최적화된 구조물들을 사용자 입력을 통해 구조물 조립 기법을 적용하여 조립과정을 처리하는 것으로 이 부분이 퍼즐 시스템 제작 실험의 가장 핵심이 되는 부분이다. 두 번째 단계를 거쳐 처리된 조립과정을 화면에 보여주는 렌더링 단계가 세 번째이며, 마지막으로 모든 실험이 완

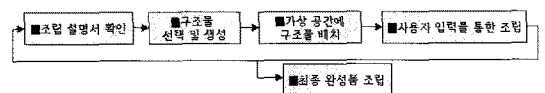


그림 9. 퍼즐 시스템 제작 실험 진행 단계



그림 10. 제작된 구조물의 기본형

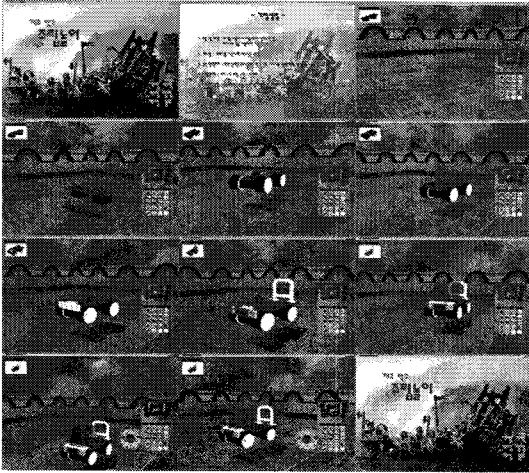


그림 11. 조립과정이 적용된 퍼즐 시스템 제작

료되면 초기화한 객체를 해제시켜주는 해제 단계가 네 번째 단계이다. 총 네 단계를 통해 프로그래밍 흐름이 구성된다. 그림 11은 최종적으로 제작한 퍼즐 시스템을 통해 조립과정이 효율적이며 사실적으로 진행되는 과정을 나타낸 것이다.

4.2 성능 분석

여기서는 구조물들 사이의 조립과정에서 일반적으로 사용되는 삼각형단위의 정교한 충돌 검출 방법과 새롭게 설계한 구조물 조립 기법간의 성능을 비교하였다. 이를 위해 사용된 환경은 CPU 펜티엄 4-630, RAM 1GB, VGA Geforce 6600이다. 구조물이 하나씩 추가되고 조립되는 환경에서 실험이 진행되었고 1초당 프레임 수(FPS, Frames per Second)를 확인하고 값의 변화를 기록하였다(그림 12).

조립 기법을 사용하지 않고 일반적인 충돌 검출

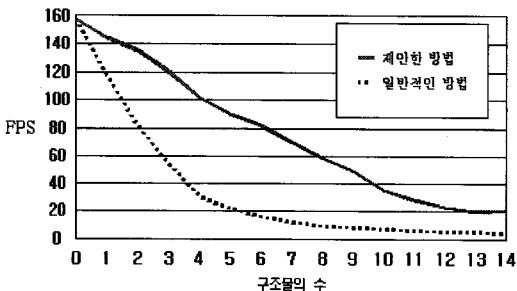


그림 12. 일반적인 충돌 검출 방법과 구조물 조립 기법의 성능 비교

방법만으로 조립을 하게 될 경우 구조물이 추가될 때 마다 초당 프레임수가 큰 차이를 보임을 알 수 있다. 일반적인 충돌 검출 방법에서는 어떤 경우에도 구조물을 이루는 모든 삼각형단위로 충돌 검사를 진행하기 때문에 조립과정에서 불필요한 연산을 수행하게 되어 구조물의 수가 증가할 때 연산의 양이 더 많이 증가되므로 처리할 수 있는 FPS가 상대적으로 더 빨리 감소된다. 일반적인 방법으로 조립을 진행할 경우 구조물의 수가 4개만 넘어가도 많은 연산 양으로 인해 30FPS 미만으로 떨어짐을 확인할 수 있다. 그리고 7개가 넘어가면서 부터는 10FPS 밑으로 떨어져 원활한 조립과정이 불가능해 짐을 그림 12를 통해 알 수 있다. 하지만 본 연구에서 제안한 구조물 조립 기법을 사용할 경우 구조물의 수가 10개로 증가하여도 30FPS를 유지하고 있어 구조물의 수가 증가함에 따라 성능의 차이가 확연히 달라지는 것을 확인할 수 있다.

그 성능의 차이를 구체적으로 예를 들어보면, 기본형 블록 구조물의 경우 블록을 이루는 삼각형의 수가 40개라고 가정하면 두 개의 기본 블록을 일반적인 방법으로 검사하였을 때 $40 \times 40 = 1600$ 번의 연산을 수행한다. 여기서 기본 블록의 돌출부를 이루는 삼각형의 수가 7개, 도입부를 이루는 삼각형의 수가 8개라고 하면 구조물 조립 기법의 경우 1600번의 연산이 아닌 $(7 \times 8) \times 2 = 112$ 번의 연산만을 수행하게 된다. 이와 같이 불필요한 연산을 최소화 하고 조립에 필요한 연산만을 정리하여 처리하게 되는 구조물 조립 기법이 조립과정에서 더욱 효율적임을 실험을 통해서 확인하였다.

5. 결론

본 논문에서는 다양한 발전과 변화를 기반으로 성장하고 있는 게임 산업의 현 추세와 더불어 퍼즐게임의 새로운 시도로 3차원 조립형 퍼즐게임을 제안하였다 제안한 사실적인 3차원 퍼즐 시스템을 위해서는 조립 부속 구조물들 사이의 효율적이면서 사실적인 조립이 중요한 요소이다. 따라서 이러한 문제를 해결하기 위해 본 논문은 구조물 간의 효과적인 처리 기법을 설계하였다.

본 논문에서 설계하는 구조물사이의 처리 과정은 크게 두 가지로 나누어진다. 첫 번째는 조립에 앞서

조립에 사용될 구조물을 결합에 최적화된 상태로 정보를 설정하는 전처리 과정이다. 여기서 결합에 최적화된 상태란 많은 점, 선으로 표현되는 구조물의 복잡한 다각형 정보 중 조립에 사용되는 조립부 즉, 돌출부와 도입부 정보를 추출하여 재설정된 것을 의미한다. 전처리 과정은 본 논문에서 객체 편집기를 통해 구현되었다.

조립에 효과적인 구조물 재설정 작업이 완료된 후에는 이들 구조물을 불러들여 조립이 진행될 수 있는데 이를 조립과정이라 한다. 이는 실제로 가상공간 내에 구조물간의 조립이 효율적이고 사실적으로 처리될 수 있도록 하는 과정으로 본 연구에서는 3단계로 나눌 수 있는 구조물 조립 기법을 설계하였다.

구조물 조립 기법의 3단계는 정렬, 결합, 완성으로 구성된다. 먼저 정렬단계는 전처리 과정에서 재설정된 구조물의 조립부 정보를 통해 조립을 하는 두 구조물이 올바르게 정렬이 되었는지 판별하는 것으로 결과가 확인된 후에 결합단계가 실행되는데 이는 효율성과 사실성에 초점을 둔다. 먼저 정교한 결합을 위해 삼각형단위의 충돌 검출이 이루어지고 여기서 일반적으로 모든 정점 정보를 검사하는 것이 아닌 재설정된 구조물의 조립부 정보만의 검사가 진행된다. 따라서 계산 비용을 대폭 감소시키게 된다. 다음으로 사실적인 표현을 위해 각 구조물의 결합 형태에 따른 물리구조를 함수화함으로써 각각의 결합 형태에 따른 사실적인 연출이 가능해 진다. 이와 같은 사실적이면서 정교한 결합단계가 진행된 후 결합이 끝까지 이루어져 더 이상 진행될 수 없는 상태가 되면 마지막 단계인 완성단계가 호출됨으로써 조립이 완료된다.

설계한 구조물 조립 기법이 실제 게임 안에서 사실적으로 표현되는지 또한 조립형 퍼즐게임 내부에 적용 가능하지 여부를 확인하기 위하여 간단한 실험 게임을 제작, 적용해봄으로써 가능성을 확인하였다. 이 결과 실제로 사실적인 조립이 표현되며 게임으로 사용할 수 있는 정도의 효율적인 연산이 가능함을 실험을 통해 확인할 수 있었다. 또한 정교한 조립을 고려하여 설계한 구조물 조립 기법이 일반적으로 객체간의 정교한 충돌 검출에 사용되는 방법과 비교하여 어느 정도 계산 효율이 향상되었는지를 초당 프레임 수 비교를 통해 확인하였다. 결과적으로 동일하게 정교한 충돌 검출이 진행되면서 성능이 나아졌음을

확인할 수 있었다.

향후 사실적인 표현을 유지하면서 좀 더 편리하게 조립을 진행하기 위하여 돌출부와 도입부가 일정 범위 이내에 도달하면 자동으로 정렬이 되는 자동 정렬 기능이 추가되어 사용자 하여금 정렬 시간을 줄이고 바로 결합단계로 넘어갈 수 있도록 설계한다면 불필요한 충돌 연산과 조립시간을 더욱 감소시킬 수 있는 하나의 방법이 될 것으로 본다. 그리고 본 연구에서 제작한 퍼즐게임은 가장 기본적인 요소만으로 구성된 것이나 게임 요소를 추가한다면 퍼즐게임의 다양한 발전방향을 모색할 수 있는 초석이 될 것으로 보인다.

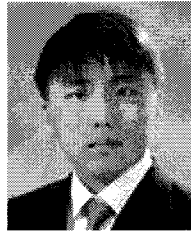
그 밖에 전자부품으로 구성된 전자제품, 작은 부품으로 구성된 기계장치와 가구 등 조립형 구조를 가지는 다양한 제품들을 가상공간에서 미리 조립 시뮬레이션을 해 보는 가상 조립분야로의 활용 또한 가능할 것이다. 이와 같이 구조물 조립 기법은 조립 형태의 모든 객체에 적용 가능한 방법으로 설계되었기에 퍼즐 게임을 포함한 다양한 조립 구조물의 가상 조립분야에 폭넓게 활용될 것으로 기대된다.

참 고 문 헌

- [1] 임희경, "컴퓨터 게임의 과거, 현재, 미래 : 발전 동향에 관한 비교 분석 연구," 한국 일러스트 아트학회, 통권, 제12호, pp. 90-101, 2003.
- [2] 이은아, 박용범, "게임장르 분류를 위한 제안," 신소재 연구 논집, 제12권, 제2호, pp. 65-76, 2004.
- [3] 문화관광부 한국 게임 산업진흥원, 2007 대한민국 게임백서 상, 문화관광부 한국 게임 산업진흥원, 서울, 2007.
- [4] 문화관광부 한국 게임 산업진흥원, 2007 대한민국 게임백서 하, 문화관광부 한국 게임 산업진흥원, 서울, 2007.
- [5] K. Kaiser, 3D Collision Detection, *Game Programming gems*, Charles River Media, Hingham, Massachusetts, 2000.
- [6] Game Chart Homepage : www.gamechart.co.kr
- [7] S. Lee, "An Efficient Collision Detection in The Dynamic Spatial Subdivisions for an MMORPG Engine," *Journal of Korea multi-*

media society, Vol.17, No.12, pp. 1729-1736, 2004.

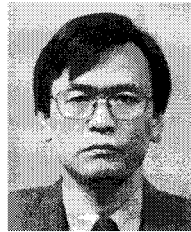
- [8] 남승우, "BSP 트리를 이용한 효율적인 게임객체 충돌 탐지에 관한 연구," 호남대 정보산업대학원, 광주, 2003.
- [9] G. Van and D. Bergen, "Efficient collision detection of complex deformable models using AABB trees," *Journal of Graphics Tools (USA)*, Vol.2, No.4, pp. 1-13, 1997.
- [10] S. Gottschalk, M.C. Lin, and D. Manocha, "OBB Tree: a hierarchical structure for rapid interference detection," *Proceedings of 23rd International Conference on Computer Graphics and Interactive Technique (SIGGRAPH 96)*, pp. 4-9, 1996.
- [11] E. Lengyel, *3D Game Programming & Mathematics for Computer Graphics*, Charles River Media, Hingham, Massachusetts, 2001.
- [12] Game Programming Gems Study Homepage : www.gpgstudy.com
- [13] Game Developer Homepage : www.gamedev.net



김진모

2006년 2월 동국대학교 멀티미디어공학과 졸업(공학사)
 2008년 2월 동국대학교 멀티미디어학과 졸업(예술공학석사)
 2008년 3월~현재 동국대학교 멀티미디어학과 박사과정

관심분야 : 게임 프로그래밍, 게임 알고리즘, 영상처리



조형제

1973년 부산대학교 전자공학과(학사)
 1975년 한국과학기술원 전기·전자공학과(공학석사)
 1986년 한국과학기술원 전기·전자공학과(공학박사)
 1986년~현재 동국대학교 멀티미디어학과 교수

관심분야 : 컴퓨터 그래픽스, 게임공학, 컴퓨터비전, 디지털사운드처리