

# 군중 장면 처리 시스템의 설계 및 구현

안민수<sup>\*○</sup>, 백성민<sup>\*</sup>, 장진욱<sup>\*\*</sup>, 이인권<sup>\*\*</sup>, 이승용<sup>\*</sup>

<sup>\*</sup> 포항공과대학교 컴퓨터공학과

<sup>\*\*</sup> 포항공과대학교 정보통신대학원

{atom, sogmi, tyrann7, iklee, leesy}@postech.ac.kr

Design and Implementation of a Crowd Scene Management System

Minsu Ahn<sup>\*</sup>, Sung Min Baek<sup>\*</sup>, Jin-Wook Chang<sup>\*</sup>, In-Kwon Lee<sup>\*\*</sup>, and Seungyoung Lee<sup>\*</sup>

<sup>\*</sup> Dept of Computer Science & Engineering, POSTECH

<sup>\*\*</sup> POSTECH Information Research Laboratory

## 요 약

본 논문에서는 수많은 애니메이션 인물들이 동시에 출현하는 장면을 만들어내는 군중 장면 처리 시스템을 개발하였다. 이 시스템은 상황에 따라 주어지는 인물의 종류와 동작 특성을 입력받아 이들 인물의 전체적인 움직임을 시뮬레이션한다. 이를 통해 얻어지는 결과는 키프레임을 포함하는 파일로서 필요에 따라서는 애니메이터의 수정이 가능하도록 하였다. 또한 이 시스템의 개발 환경을 마야 플러그인 방식으로 채택함으로써 보다 범용 상품으로서의 가치를 높인다.

## 제 1 절 서론

사람이나 동물들이 무리를 지어 움직이는 것은 실제 생활의 친숙한 부분이다. 애니메이션에서 수많은 인물(character)들이 동시에 등장하는 장면을 군중 장면(crowd scene)이라고 한다. 삼차원 애니메이션에서 군중 장면이 중요한 의미를 가지는 것은 군중 장면은 흔히 볼 수 있는 일상 생활의 한부분이기 때문이다. 군중 장면은 애니메이션 전개상 필요할 뿐만 아니라, 애니메이션의 사실감을 높여 준다.

군중 장면에는 수많은 인물이 등장하기 때문에, 일반 장면을 제작하는 방법으로 제작할 경우, 많은 시간과 노력이 필요하다. 하지만 군중 장면은 전체 장면 중 극히 작은 일부분에 불과하다. 또한 이 장면에 등장하는 인물들은 대부분 주요 인물이 아니다. 따라서 군중 장면은 적은 비용과 노력으로 얻을 수 있어야 하며, 다른 일반 장면을 생성하는 방법과 구분해야 한다.

이와 같은 이유로 사용자의 간단한 조작과 적은 시간으로 자연스러운 군중의 동작을 이끌어 낼 수 있는 군중 장면 처리 시스템이 필요하게 되었다. 그리고 군중 장면 처리 시스템을 널리 사용하는 응용 프로그램의 플러그인(plug-in) 형식으로 개발할 경우, 일회성에 끝나는 도구가 아닌 범용의 상품으로 그 가치가 매우 높다고 할 것이다.

본 논문에서는 마야(Maya)에서 제공하는 개발자 도구를 사용해서 수많은 인물의 움직임을 시뮬레이션하는 시스템을 설계하고 구현한다. 그리고 이 시스템을 설계하는데 해결해야 할 문제들을 살펴본다.

## 제 2 절 관련 연구

삼차원 애니메이션을 제작하는 기업에서는 군중 장면을 생성하기 위해서 각자의 군중 장면 처리 시스템을 제작하였

다 [2]. PDI는 군중 애니메이션 시스템(crowd animation system)을 동작 주기(motion cycle)를 기반으로 군중 시뮬레이터(crowd simulator)와 합성 시스템(blending system)으로 구성하였다 [1]. 군중 시뮬레이터는 정해진 규칙을 고려하여 각 인물의 동작 주기를 선택하고 인물의 행동을 생성한다. 합성 시스템은 인물에 동작을 부여하고, 특정 조건을 만족하도록 동작을 변형시키는데 사용한다. 군중 장면을 위해서 ILM에서는 마야 기반의 시스템을 만들었다. 이 시스템은 군중 장면당 평균 143개의 동작 주기를 사용했으며 이 주기와 관련된 데이터베이스는 70기가 바이트 정도를 사용했다. Pixar 또한 군중 장면을 생성하기 위해서 인물의 행동, 습성, 특성 그리고 상태에 대한 데이터베이스를 만들었다. 이러한 데이터베이스로부터 인물들을 생성하고, 배치 도구를 이용해서 군중의 형태, 위치 그리고 배치 밀도 등을 조절하였다. 마지막으로 시뮬레이션 도구를 사용해서 군중 장면을 완성했다.

## 제 3 절 전체 시스템 개요

군중 장면 처리 시스템은 형태 도구(shape tool), 동작 도구(motion tool), 군중 애니메이터(crowd animator) 등의 세 영역으로 나누어진다. 이 세개의 도구를 거쳐 최종적으로 얻는 결과는 애니메이션이 가능한 키프레임(keyframe)을 가진 파일이다. 이러한 전체적인 흐름도는 [그림 1]와 같다.

형태 도구는 군중 장면에 등장하는 인물들 사이에 차이를 주는 것이다. 형태 도구를 통해 만들어진 인물의 형태에 동작을 부여하는 것이 동작 도구이다. 동작 도구는 인물에 동작을 부여하거나, 주어진 동작을 변화시키는 것이다. 군중 장면을 만들기 위해서는 형태 도구와 동작 도구를 통해 만들어진 기본 인물을 복사, 배치해야 한다. 그리고 복사된

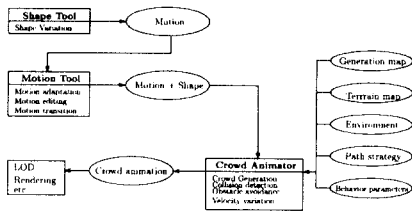


그림 1: System overview

인물에 실제 애니메이션에서 사용할 수 있는 키프레임을 부여해야 한다. 이러한 작업을 담당하는 부분이 군중 애니메이터이다. 즉, 군중 애니메이터는 군중의 생성 및 충돌 발견, 장애물 회피, 그리고 인물 동작 속도 조절 등이 이루어져 원하는 수의 프레임들을 생성하는 것이다.

이 시스템에서는 군중을 그룹으로 나누고, 그룹 단위로 특성을 부여하였다. 각 그룹마다 인물의 이동 전략과 생성 영역에 대한 정보 등을 가지고 있다. 또 각각의 도구 사이에는 필수적인 관계를 맺는 것이 아니라, 필요한 입력 정보만 제공되면 독립적인 도구로 사용할 수 있다.

## 제 4 절 시스템 구성 요소

### 4.1 형태 도구

형태 도구는 군중 장면에서 등장하는 인물들 사이에 차이를 주는 것이다.

### 4.2 동작 도구

인물 골격의 동작에 대한 연결관계를 정의하고, 골격에 맞는 동작을 부여하고나, 수정하는 도구이다. 이런 목적을 가진 동작 도구는 다음과 같은 세부분으로 나눌 수 있다.

**동작 편집 (motion editing)** 주어진 동작을 원하는 형태로 변형시키다. 키프레임으로 정의된 개체 동작의 일부분을 수정했을 때, 전체 동작이 부드럽게 변형되도록 한다.

**동작 적응 (Motion adaptation)** 특정 동작을 다른 골격에 적용할 때, 자연스럽게 보이도록 동작을 수정한다.

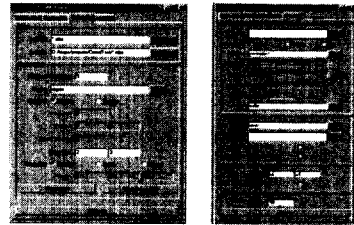
**동작 전이 (Motion transition)** 동작은 주기를 가지는 부분 동작을 연결로 생성된다. 이때 부분 동작들 사이에 생길 수 있는 불연속성을 없애고 적절한 수의 키프레임을 갖는 동작을 만들어 준다.

### 4.3 군중 애니메이터

군중 애니메이터는 크게 두 개의 부분으로 구성된다. 첫번째가 원하는 수의 군중을 생성하고 이를 배치하는 부분이다. 두번째가 생성된 군중 인물들에게 애니메이션을 추가하는 부분이다. 즉, 원하는 수의 키프레임을 얻고, 각각의 그룹마다 특정 동작을 부여하는 부분이다.

군중을 생성하고 애니메이션을 만들기 위해서는 몇가지 정보가 필요하다. 지형, 인물 생성 영역, 군중 내 인물의 수, 그룹의 크기 및 배열 형태 등을 사용자로부터 입력받아야 한다. 입력된 정보를 이용해서 임의의 요소를 적절히 추가

해 자동적으로 군중의 배열을 생성하고 시물레이션한다. 그리고 이러한 정보들은 군중 또는 그룹의 속성으로 키프레임 정보와 같이 저장된다.



(a) (b)

그림 2: 사용자 인터페이스

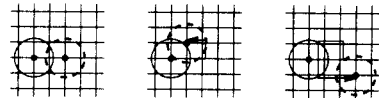
## 제 5 절 애니메이션 알고리즘

군중 장면을 생성하기 위해 군중 애니메이터에서 고려해야 할 사항은 다음과 같다.

### 5.1 인물 생성 및 배치

군중은 사용자가 선택한 영역에만 생성한다. 따라서 인물을 배치할 때 그 위치가 영역 안에 있는지를 검사하는 것이 필요하다. 이를 위해서 임의로 선택된 위치에서  $y$  축 방향으로 광선을 쏘아서 지정된 다면체 영역과 교차하는지를 검사한다. (인물은  $xz$  평면 위에 있다고 가정한다)

그리고 이미 위치한 다른 인물과의 충돌을 검사해야 한다. 인물을 배치한 후 다른 인물과 충돌이 일어날 경우, 그 주위를 격자로 나누어서 각 격자점을 차례로 조사해서 충돌이 일어나지 않는 곳을 찾는다. 이 과정을 [그림 3]과 같다.



(a) (b) (c)

그림 3: 배치시 충돌 회피 알고리즘

인물 배치할 때 반드시 고려해야 할 것은 인물의 어떤 부분도 지형과 교차하지 말아야 한다는 사실이다. 이를 위해 인물이 위치하는 곳의 지형 높이를 알고 있어야 한다. 지형 높이를 계산하기 위해서 인물의 위치  $(x, z)$ 에서  $y$  축 방향으로 광선을 쏘아 지형과 만나는 점을 구한다.

### 5.2 인물의 이동

인물의 이동에는 주어진 경로를 따르는 경우, 임의의 방향인 경우, 그리고 목적지가 있는 경우로 나누어진다. 우선, 경로가 곡선으로 주어진 경우는 각 프레임마다 인물의 진행 방향을 변경한다. 각 프레임마다 경로 곡선 위의 점들 중 인물과 가장 가까운 점을 찾아서 이 점에서의 접선 벡터를 그 인물의 진행 방향으로 한다. 경로를 임의의 방향으로 선택한 경우는 인물들은 임의의 방향으로 나아간다. 이 경우

인물의 진행 방향은 일정 시간마다 임의로 변경한다. 사용자가 목적지를 주면 인물은 목적지를 향하여 이동한다. 각 프레임마다 인물의 진행 방향은 인물의 현재 위치와 목적지의 현재 위치 사이의 차이를 이용하여 계산한다. 목적지가 움직일 경우, 인물의 방향도 바뀐다.

### 5.3 인물들 간의 충돌 탐지 및 회피

각 인물을 포함하는 경계선 원통 (bounding cylinder)를 사용하여 인물 간 충돌 검사를 한다. 이 시스템에서 사용한 충돌 회피 방법에는 인물의 진행 방향을 바꾸어서 충돌이 일어나지 않도록 하는 방법과 인물의 진행을 잠시 멈추는 방법이 있다. 우선 두 개의 인물이 충돌이 일어날 경우, 충돌이 일어나지 않는 각도  $\theta$ 를 구한다. 이 과정은 [그림 4]과 같다.

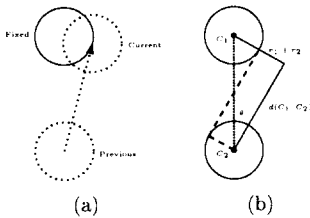


그림 4: 이동시 충돌 회피 알고리즘

$\theta$ 의 값이 한계값을 넘지 않는 경우 진행 방향을  $\theta$ 만큼 진행 방향을 전환하고, 넘을 경우에는 진행을 하지 않는다.

### 5.4 장애물 회피

무한 필드 함수를 통해 장애물이 인물에 미치는 힘을 계산하고 그 힘과 인물의 진행 방향을 고려해서 새로운 진행 방향을 결정한다. 부드러운 방향 전환을 위해 무한 필드 함수는 매 프레임마다 계산한다. 이 경우 부드러운 방향 전환은 물론 장애물들이 모여 있는 부분의 중심을 향하여 움직일 때도 피할 수 있다.

### 5.5 시물레이션 과정

군중 장면을 시물레이션하기 위해서는 등장하는 모든 인물의 정보가 필요하다. 하지만 인물의 형태 정보가 매우 클 경우, 전체 크기가 매우 커져서 시스템에서 사용 가능한 크기 이상이 될 수도 있다. 하지만 시물레이션할 때 인물 형태 정보는 크게 사용하지 않는다. 이 시스템에서는 인물 대신 인물을 포함하는 직육면체 (bounding box)를 사용하여 시물레이션한다. 그리고 직육면체를 사용해서 얻은 정보를 각각의 인물에 재적용하는 방법을 사용한다.

## 제 6 절 시스템 구현 및 사용 예제

군중 장면 처리 시스템은 Alias|wavefront 사에서 제작한 마야라는 삼차원 모델링 도구의 플러그인 형식으로 제작하였다. 시스템의 각 도구들은 마야에서 제공하는 개발 도구를 이용하여 마야에서 실행할 수 있는 명령어들로 만들었다. 그리고 사용자 인터페이스는 마야에서 제공하는 MEL (Maya Embedded Language)이라는 스크립트 언어를 사용

하여 구현했다. 이 시스템은 사용자 인터페이스를 통해 입력된 정보로부터 적절한 명령 구문을 만들어 이를 실행하도록 구현하였다. 이 명령 구문은 각 도구의 기능을 담당하는 명령어들로 구성된다.

군중 장면 처리 시스템을 통해 다양한 군중 장면을 만들 수 있다. 예로 간단한 군대의 행진 장면이나 시장 거리 장면을 있을 수 있다. [그림 5]은 이 시스템을 만들어 제작한 간단한 장면들이다.

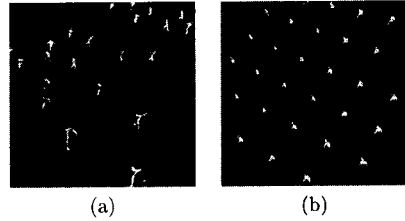


그림 5: 예제 (a) 임의의 분포, 임의의 방향으로 진행 (b) 격자 모양 분포, 같은 방향으로 진행

[그림 5]의 (a)는 인물을 임의의 위치에 배치하고 임의의 방향으로 진행하게 한 것이다. (b)는 인물의 격자 형태로 배치하고 같은 방향으로 진행하도록 한 장면이다.

## 제 7 절 결론 및 향후 연구방향

본 논문에서는 마야에서 제공하는 개발 도구를 사용하여 군중 장면 처리 시스템을 구현하였다. 시스템 구현에 있어서 마야의 플러그인 방식으로 제작하였기 때문에 범용의 상품으로 그 가치가 매우 높다고 하겠다. 또한 그 최종 결과가 애니메이션이 가능한 키프레임을 가진 파일이기 때문에 추가적인 작업이 가능하며, 각 기능이 독립적으로 사용 가능하기 때문에 다른 분야에도 사용할 수 있다.

## 참고 서적

- [1] 1998. <http://www.antz.com/technology.html>.
- [2] CG Crowd: The Emergence of the Digital Extra. SIG-GRAPH 99 Panel discussion, 1999.