

# 고품질 특수효과를 위한 입자 시스템 라이브러리 개발

Development of Particle System Library for High Quality Special Effects

백기엽, 송승헌, 김응곤  
순천대학교

Baek Gi-Yeob, Song Seung-Heon, Kim Eung-Kon  
Suncheon National University

## 요약

하드웨어의 성능 및 처리속도의 급속한 증가와 그래픽환경의 발전은 화려한 특수영상을 제공하고 있다. 특히, 입자시스템을 이용한 불꽃과 연기, 액체, 눈, 비, 먼지 등의 특수효과는 영화, 게임, 가상현실 등의 없어서는 안 될 중요한 구현효과이다. 그러나 이러한 특수효과 생성은 많은 비용과 제작시간이 소요된다. 본 논문에서는 저비용의 고품질로 각 콘텐츠에 다양한 형태의 입자 특수효과를 구현할 수 있는 입자시스템 라이브러리를 개발해 콘텐츠 산업발전에 기여하고자 한다.

## Abstract

This research results can be used in games or animation application programs that need special effects and can be used as education software in undergraduate or graduate courses. If we use this results effectively in application programs, we can expect no small ripple effect on growing entertainment industry because special effects enhance audience's interest in games, films, and amusements.

## I. 서론

2000년대 이후 게임과 영화는 엔터테인먼트 산업의 중심으로 가장 각광을 받고 있는 분야이다. 특히, 입자 시스템을 이용한 불꽃과 폭발, 연기, 액체, 눈, 비, 먼지 등의 특수효과는 여러 가지 콘텐츠들의 상황에 따라 적용됨에 따라 그 현실감을 더해주고 있다.

이를 위해 사용되는 입자 시스템은 여러 개의 개별적인 입자들의 집합을 의미하며, 컴퓨터 그래픽스에서는 입자들을 사용한 모든 기법들을 지칭한다. 3D Studio MAX와 Maya등의 기존 패키지 제품은 외산으로 고가이면서 오프라인 상에서만 사용가능하고 Open GL과 DirectX등을 이용한 특수효과 응용 프로그램을 개발할 경우는 저수준의 그래픽 라이브러리 사용으로 특수효과 생성에 많은 시간이 소요되

고 개별 알고리즘의 적용으로 코드의 재사용에 어려움이 있다는 단점을 가지고 있다.

따라서, 본 논문에서는 VR(가상현실)응용 프로그램 개발자와 콘텐츠 제작자가 다양한 특수효과를 저비용 고품질의 디지털 콘텐츠에 적용할 수 있도록 물, 불, 불꽃, 폭발현상, 비, 눈, 먼지, 안개 등의 다양한 특수효과를 손쉽게 적용할 수 있도록 입자 시스템 라이브러리를 개발하고자 한다.

1장에 서론에 이어 2장 관련연구에서는 상용패키지나 다른 그래픽 라이브러리에서 사용되는 입자 시스템에 대해 기술한 후, 3장에서는 구현하고자 하는 입자 시스템에 대하여 정의 및 설계하고, 4장에서는 다양한 특수효과를 제공하는 입자 시스템 특수효과 라이브러리를 구현한다. 5장에서 결론을 내리며 향후 연구과제를 제시한다.

## II. 관련연구

Maya나 3D Studio Max와 같은 상용 패키지에서 특수효과를 만들기 위한 입자 시스템을 지원하고 있으나, 온라인상에서는 바로 사용이 불가능하며 대화식으로만 제작할 수 있다. 또한 오프라인 상에서 제작한 특수효과를 사용하거나 자체적으로 특수효과를 내기 위한 기능을 삽입해야 하는데, 이를 위해서는 개발기간이나 비용이 많이 소요된다는 단점을 가지고 있다.

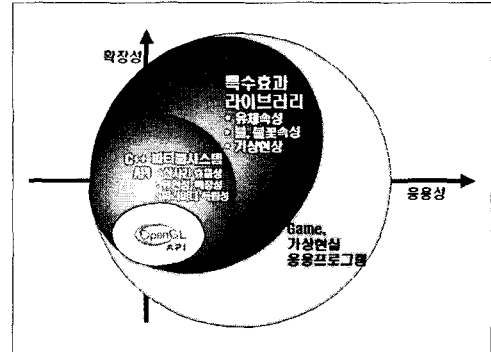
기존의 연구들에서도 유체 속성을 갖는 물질들인 불, 불꽃, 폭발 속성을 갖는 물질의 특수효과나 기상현상의 재현을 위한 특수효과 생성 기법을 제시하고 있으나, 모두 개별적인 알고리즘의 생성과 적용으로 특별한 경우의 효과를 표현하기 위한 것이기 때문에 프로그래머가 게임이나 가상현실 등에서 필요로 하는 특수효과를 쉽게 프로그래밍 할 수 있는 라이브러리는 개발이 미흡한 실정이다. 게다가 게임을 비롯한 많은 그래픽스 응용 프로그램에서는 OpenGL이나 Direct3D를 이용하여 특수효과를 만들어 내고 있으나 실시간 렌더링을 위한 저수준 연산을 수행하는 API만을 지원하고 있는 실정이기 때문에 프로그램 개발자가 특수효과를 내기 위해서는 많은 시간과 노력이 소요되며, 필요한 효과의 구현시 각 기능들을 개별적으로 중복 개발해야 하는 불편함을 감수해야 한다는 단점을 가지고 있다.

## III. 입자 시스템 API

입자 시스템은 OpenGL 그래픽스 라이브러리에 기반한 특수효과 발생을 위한 상위 수준의 그래픽스 API로 구성되기 때문에 유체속성, 불, 불꽃 속성, 기상현상 등의 여러 가지 특수효과에 대해 설정값의 변경만으로 특수효과 라이브러리를 구축할 수 있게 하며, 게임이나 가상현실 등의 응용 프로그램에서 이를 응용하여 사용할 수 있다.

다음 [그림 1]은 Game과 가상현실 프로그램 등에서의 응용성과 특수효과 라이브러리 구축을 위한 입자 시스템 및 OpenGL의 확장성을 보여준다.

입자시스템 API의 세부내용은 다음 [그림 1]과 같다.



▶▶ 그림 1. 입자 시스템과 특수효과 라이브러리의 확장 및 응용

### 1. 입자 속성

입자 시스템 API 내의 입자는 위치(position), 속도(velocity), 색상(color), 알파(alpha), 크기(size), 수명(age), 2차 위치(secondary position), 2차 속도(secondary velocity) 등의 속성으로 구성하였다.

### 2. 입자 함수

API는 입자 그룹함수, 속성상태 함수, 액션함수, 액션리스트 함수 등 4가지 종류의 함수로 구성한다.

#### 2.1 입자 그룹 함수

모든 입자들은 같은 힘에 의해 작용하는 입자들의 집합인 입자 그룹 내에 존재하며, 몇 개의 입자 그룹들이 동시에 존재할 수 있으나 모든 API 함수들은 현재의 입자 그룹에만 적용되며, 모든 액션들은 현재의 입자 그룹 내의 모든 입자들에 적용되도록 한다.

#### 2.2 속성과 영역

새로운 입자를 생성할 때 입자의 색상, 속도, 크기

수명 등의 속성을 지정해 주어야 한다. 대부분의 입자 효과는 입자의 초기 속성 값을 변화시키게 되며, 속성들의 변화를 처리하기 위하여 API에 3차원 공간의 범위를 지정해주는 영역을 설정한다.

### 2.3 액션 함수

액션들은 현재의 입자 그룹 내의 입자들의 속성을 변경시키는 함수들로 물리적인 힘을 시뮬레이션하거나 각 입자들이 어떤 특정위치로 이동하게 하는 파라미터 2차 방정식에 의한 경로를 계산하는 함수들로 구성하였다.

### 2.4 액션 리스트 함수

액션 리스트 생성함수를 호출하게 되면 어떤 특수 효과 생성에 필요한 모든 액션과 상태 변화 호출함수들이 액션 리스트 내에 저장된다.

## IV. 입자 시스템 라이브러리의 구현

### 1. 입자 시스템과 OpenGL

입자 시스템은 OpenGL에 기반하여 만들어진 API의 성격을 가지고 있지만 특수효과를 위한 입자의 생성과 설정, 함수값의 설정은 OpenGL과 독립된 모습을 보여주고 있다. OpenGL은 입자 시스템의 설정 외에 생성되는 가상세계를 초기화 시켜주며, 장면이 표현되기 위한 윈도우의 생성 및 입자를 제외한 여러 오브젝트의 생성을 하는 역할을 하고, 입자 시스템은 입자들의 생성 및 설정을 담당하게 된다.

### 2. OpenGL API에서의 입자 모양 설정

입자 시스템은 OpenGL에 기반하기 때문에 GL.h 헤더 파일에서 정의된 오브젝트 형태 중의 하나를 입자의 모양으로 설정하여야 한다. 이때, 적용할 수 있는 오브젝트의 종류는 임의로 선택할 수 있지만, 본 연구에서는 제공되는 여러 가지 오브젝트 중에서 계

산이 가장 적게 이루어지는 GL\_POINTS를 사용하여 입자를 표현하도록 하였다.

### 3. 입자 시스템 API에서의 도메인 설정

도메인은 어떤 액션이나 상태함수에 파라미터로 사용되기 때문에 입자의 생성 시에 영역 내에서 임의로 선택된 위치에서 입자를 생성하는 역할을 한다.

### 4. 입자의 속성

새로운 입자의 생성 시에는 색상, 속도, 크기, 수명 등의 속성을 지정해 주어야 하는데, 도메인의 설정이 이루어진 다음에는 상태 설정을 위한 함수들 및 Action List, Particle Group, Actions 함수들을 정의하여 준다.

### 5. 특수효과의 생성

게임이나 가상현실 등에서 특수효과의 구현은 C++ 언어로 구현된 상위 수준의 API를 호출함으로써 간단히 구현할 수 있으며, 이때 프로그래머는 각 API의 자세한 구현 내용은 모르더라도 액션 함수의 호출과 각 함수에 따른 파라미터 변경만으로 특수효과를 손쉽게 구현할 수 있게 되었다.

본 연구의 구현을 위한 소프트웨어와 하드웨어 구현환경은 다음 [표 1]과 같다.

[표 1] 구현환경

소프트웨어	하드웨어
운영체제: WindowsXP Professional	Graphic Card : RADEON 9600
개발환경 : OpenGL, Visual C++	CPU : Pentium4 2.9Ghz

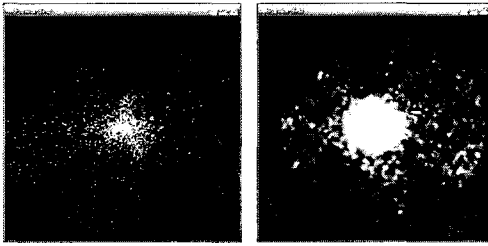
본 논문에서는 기상현상, 불·불꽃, 유체속성에 대하여 다음과 같이 GL\_POINTS와 textured quad를 입자로 사용하여 총 9가지 항목으로 각각 라이브러리화하여 구현하였다.

다음 [표 2]는 측정결과를 도식화한 것이다.

[표 2] 각 특수효과 라이브러리의 적용결과

Special Effect	Frame/Second			
	20,000 Particles		30,000 Particles	
	GL_POIN TS : 640*480	GL_POIN TS : 800*600	textured quad : 640*480	textured quad : 800*600
Water Fall	60	20	53	47
Moving Light	64	56	55	46
Atom	63	55	54	46
Fountain	66	55	49	45
Chaos	59	48	47	40
Rain	57	50	48	42
Explosion	59	47	46	39

다음 [그림 3]은 폭발하는 특수효과를 나타낸다.



▶▶ 그림 3. GL\_POINTS와 textured quad로 구현된 폭발효과

다음 [그림 4]는 폭발소스코드를 나타낸다.

```

void Explosion(bool do_list = true)
{
    if(do_list && action_handle<0)
        action_handle = pGenActionLists(1);
    pVelocityD(PDSphere, 0,0,0,0.01,0.01);
    pColorD(1.0, PDSphere, 0.5, 0.7, 0.5, .3);
    pSize(1.0);
    if(do_list)
        pNewActionList(action_handle);
    pCopyVertexB(false, true);
    pDamping(0.999, 0.999, 0.999);
    static float i=0;
    if(do_list)
        i = 0;
    pOrbitPoint(0, 0, 0, .02, 0.1);
    pExplosion(0, 0, 0, 1, 2, 3, 0.1,
        i+= (1.0f / float(numSteps)));
    pSink(false, PDSphere, 0, 0, 0, 30);
    pMove();
    if(do_list)
        pEndActionList();
}

```

▶▶ 그림 4. 폭발 특수효과의 소스코드

## V. 결 론

본 연구에서는 입자 시스템을 사용하여 객체단위로 프로그래밍이 가능하고 프로그래머가 구현 알고리즘을 모르더라도 그래픽스의 기본 개념만 이해하면 쉽게 특수효과를 구현할 수 있게 하였다.

기존의 다른 연구에서도 유체 속성을 갖는 물질, 불, 불꽃, 폭발 속성을 갖는 물질의 특수효과나 기상 현상의 재현을 위한 특수효과 생성 기법을 제시하고 있으나, 모두 특별한 경우의 특수 효과를 내기 위한 개별적인 알고리즘에 관한 것이고, 프로그래머가 게임 등에서 필요로 하는 특수효과를 쉽게 프로그래밍할 수 있는 라이브러리는 개발되어 있지 않다는 단점이 있었다.

본 연구에서 구현된 실시간 동적 입자 시스템은 게임이나 특수효과를 필요로 하는 애니메이션 응용 프로그램에 직접적인 라이브러리로 활용할 수 있기 때문에 본 연구 결과의 입자 시스템을 직접 호출하거나 C++로 구현되는 물, 불, 불꽃, 폭발현상, 비, 눈, 먼지, 안개 등의 특수효과를 직접 삽입하여 사용함으로써 개발 시간과 비용을 크게 줄일 수 있다.

## ■ 참고 문헌 ■

- [1] Reeves, W. T. "ParticleSystems - A Technique for Modeling A Class of Fuzzy Objects". Proc. of SIGGRAPH '83, Detroit, Michigan, July, 1983.
- [2] McAllister, D. K. "The Design of an API for Particle Systems" <http://cs.unc.edu/~davemc/Particle>, 1999.
- [3] Leech, J. P. and R. M. Taylor. "Intractive Modeling Using Particle Systems". Proc of 2nd Conference on Discrete Element Methods, MIT, 1993.
- [4] Allen, M. B. Flow - a particle animation API. <http://www.dnai.com/~mba/software/flow/>, 1999.
- [5] Neider, J. T. Davis, et al. OpenGL Programming Guide. Addison Wesley, 1993.