

# 실시간 특수효과를 위한 라이브러리 및 생성기 개발

## Development of Library and Generator for Real-time Special Effects

송승헌, 박경욱\*, 김응곤  
순천대학교, 전남대학교\*

Song Seung-Heon, Park Kyoung-Wook\*, Kim Eung-Kon  
Suncheon National Univ., Chonnam National Univ.\*

### 요약

정보통신의 미래가 인터넷과 디지털 콘텐츠 기술 확보에 달려있다는 전문가들의 예측은 이제 현실이 되어 있으며, 영상 산업 및 게임, 가상현실 등의 디지털 콘텐츠 산업에서는 소비자들에게 시각적으로 사실적이고 화려한 영상을 제공하기 위하여 다양한 특수효과를 필요로 한다. 이중 불꽃, 폭발, 연기, 액체, 눈, 비, 먼지와 같은 실시간 특수효과 재현에 관한 연구는 무엇보다 시급한 상태이며 제작의 편리함을 위해 실시간으로 구현된다. 본 논문에서는 각종 디지털 콘텐츠 상에 자연현상 등의 특수효과를 손쉽게 적용할 수 있는 상위 수준의 그래픽스 라이브러리 형태인 입자시스템 라이브러리를 구현하고 위치, 속도, 색상, 투명도, 크기, 수명, 2차 위치, 2차 속도 등의 속성 조절을 통해 손쉽게 특수효과를 생성할 수 있는 실시간 특수효과 생성기를 개발하고자 한다.

### Abstract

Simulation of dynamic particle systems has been used in computer animation for several years and has more recently been used in real-time simulation and games to enrich the visual appearance of the virtual worlds. A particle system is composed of one or more individual particles. The goal of this paper is to develop particle system graphics library and generator for real time processing.

## I. 서론

정보통신의 미래가 인터넷과 디지털 콘텐츠 기술 확보에 달려있다는 전문가들의 예측은 이제 현실이 되어 있으며, 영상 산업 및 게임, 가상현실 등의 디지털 콘텐츠 산업에서는 소비자들에게 시각적으로 사실적이고 화려한 영상을 제공하기 위하여 다양한 특수효과를 필요로 한다. 이중 불꽃, 폭발, 연기, 액체, 눈, 비, 먼지와 같은 실시간 특수효과 재현에 관한 연구는 무엇보다 시급한 상태이며 제작의 편리함을 위해 실시간으로 구현된다.

디지털 콘텐츠 상에서 특수효과를 표현하기 위해 입자시스템 API를 상위 수준의 그래픽 라이브러리 제공하면 유체속성이나 기상현상과 같은 자연현상 특수효과를 사실적으로 표현할 수 있으며 개발자가 원하는 형태의 특수효과가 구현될 때까지 파라미터 값만을 바꿈으로서 원하는 특수효과를 쉽게 생성할 수 있다는 장점을 가지게 된다.

그리고 손쉽게 다양한 매개변수를 변경할 수 있는 인터페이스를 갖춘 GUI 환경이 제공된다면 기존의 수작업에 의해 많은 시간이 걸리는 합성 작업을 거치지 않고도 콘텐츠에 적용될 특수효과가 제대로 제작되었는지 확인하고 그 결과에 따라 제작

업을 바로 할 수 있어 제작기간을 획기적으로 줄일 수 있으며, 사용자에게 실시간으로 다양한 디지털 콘텐츠를 제공할 수 있다.

본 논문에서는 각종 디지털 콘텐츠 상에 자연현상 등의 특수효과를 손쉽게 적용할 수 있는 상위 수준의 그래픽스 라이브러리 형태인 입자시스템 API를 구현하고 위치, 속도, 색상, 투명도, 크기, 수명, 2차 위치, 2차 속도 등의 속성 조절을 통해 손쉽게 특수효과를 생성할 수 있는 실시간 특수효과 생성기를 개발하고자 한다.

## II. OpenGL과 입자시스템 API

### 1. OpenGL

OpenGL만 이용하여 입자시스템을 구현하면 각각의 오브젝트마다 연산되어지는 과정이 달라지기 때문에 정밀한 효과는 낼 수 있으나 코드의 길이가 길어지고 연산과정에서의 계산량 때문에 시스템에 부하가 생길 수 있다는 단점을 가지고 있다.

반면 OpenGL과 입자시스템 API를 이용하면 광원효과나 윈도우 창 생성 같은 것은 OpenGL이 처리하고 입자들의 운동 방향, 생성, 소멸과 같은 입자들의 전반적인 상태를 입자시스템

API가 처리하여 각각 독립적으로 작동되기 때문에 코드의 길이가 줄어들고 시스템의 부담을 덜게 돼 연산속도를 증대시킬 수 있다.

본 연구에서 OpenGL의 역할은 입자시스템 API에 앞서 가상세계를 초기화하고 입자의 모양을 정의하며 장면이 표현될 윈도우를 생성하는 역할을 담당한다.

## 2. 입자시스템 API

실시간 특수효과 라이브러리는 특수효과 발생을 위해 OpenGL 그래픽스 라이브러리에 기반한 상위 수준의 그래픽스 API를 기반으로 구성된다.

본 입자시스템 API는 각종 디지털 콘텐츠에서 실시간 효율성과 프로그래머가 손쉽게 다양한 효과를 만들어낼 수 있는 유연성을 가지고 있으며, 개개의 파라미터에 대해 독립성을 유지하고 확장성이 있는 상위 수준의 입자 시스템으로서 다양한 특수효과 구현 및 응용을 위해 실시간 입자시스템의 생성을 지원한다.

## 3. 실시간 특수효과 생성 알고리즘

기존의 유체역학 분야에서 사용하는 Navier-Stokes 방정식 같은 경우 빠르게 처리해야 하는 컴퓨터 그래픽스 응용분야에는 계산량이 많아 적합하지 않다.

따라서 본 연구에서는 증강현실 등의 컴퓨터 그래픽스 응용 분야에 적합한 빠르고 효율적인 연기, 불꽃, 폭발 효과를 위하여 기존의 유체역학에서 사용하는 수치 모델을 다음과 같이 변경하여 단순화시킨 시뮬레이션 알고리즘을 개발한다.

연기, 불꽃, 폭발현상의 기체를  $u=(u,v,w)$ 로 표현되는 속도 모델을 다음 식 (1)과 (2)의 Euler 방정식으로 나타낸다.

$$\nabla \cdot u = 0 \quad (1)$$

$$\frac{\partial u}{\partial t} = -(u \cdot \nabla)u - \nabla p + f \quad (2)$$

이 두 방정식에서 속도는 질량과 운동량의 보존성을 유지해야 하는 조건을 나타내며,  $p$ 는 기체의 압력,  $f$ 는 외부에서 작용하는 힘이다. 이들 방정식을 풀기 위해 다음 두 단계를 거친다.

[Step 1] 압력  $p$ 의 항 없이 시간 간격  $\Delta t$ 에 대하여 방정식 (2)를 풀어 식 (3)과 같이 중간 속도  $u^*$ 를 계산한다.

$$\frac{u^* - u}{\Delta t} = -(u \cdot \nabla)u + f \quad (3)$$

[Step 2] 다음의 Poisson 방정식으로부터 압력  $p$ 를 계산한다.

$$\nabla^2 p = \frac{1}{\Delta t} \nabla \cdot u^* \quad (4)$$

식 (5)와 같이 중간 속도와 압력의 그래디언트로부터 속도  $u$ 를 구한다.

$$u = u - \Delta t \nabla p \quad (5)$$

온도  $T$ 와 기체의 밀도를 다음과 같은 식 (6)과 (7)으로 표현한다. 이 두 스칼라 양은 기체의 속도에 따라 단순히 변화한다고 가정한다면 다음과 같다.

$$\frac{\partial T}{\partial t} = -(u \cdot \nabla)T \quad (6)$$

$$\frac{\partial \rho}{\partial t} = -(u \cdot \nabla)\rho \quad (7)$$

## III. 실시간 입자시스템 생성기

본 연구에서의 실시간 입자시스템 생성기는 penGL을 기반으로 하여 입자시스템 API를 적용함으로써 윈도우즈 기반 응용 프로그램으로서 동작하게 된다.

### 1. 입자시스템 생성기의 동작

윈도우즈 화면상에 출력은 GDI 객체인가 OpenGL 객체인가에 따라서 명령의 처리가 디바이스 컨텍스트와 렌더링 컨텍스트로 각각 구분되어 이루어지는데, OpenGL의 명령어에 의해 OpenGL 렌더링 컨텍스트가 먼저 수행되고 난 후 윈도우의 디바이스 컨텍스트와 연결되어 윈도우와 상호작용을 수행함으로써 입자시스템 특수효과를 화면상에 출력하게 된다.

### 2. 입자시스템 생성기의 구성

실시간 입자시스템 생성기의 구성은 입자 속성 모듈, 입자 함수 모듈, 속성과 영역 모듈, GUI 모듈로 크게 4부분으로 나누어진다.

#### 2.1 입자 속성

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

다음 그림 1은 입자속성 함수의 설정의 예이다.

```

PARTICLEDLL_API void pColor
(float red, float green, float blue, float alpha = 1.0f);
PARTICLEDLL_API void pColorD
(float alpha, PDomainEnum dtype,
float a0 = 0.0f, float a1 = 0.0f, float a2 = 0.0f,
float a3 = 0.0f, float a4 = 0.0f, float a5 = 0.0f,
float a6 = 0.0f, float a7 = 0.0f, float a8 = 0.0f);
    
```

▶▶ 그림 1. 입자속성 함수의 설정

2.2 입자 함수

라이브리리는 입자 그룹함수, 속성상태 함수, 액션함수, 액션 리스트 함수 등 4가지 종류의 함수로 구성한다. 개발할 라이브러리 함수 이름은 pFunctionName 형태를 취한다. 프로그래머가 사용하기 쉽고 확장성이 있는 라이브러리 함수를 만들기 위하여 다음과 같이 고려한다.

▶ 입자 그룹 함수

모든 입자는 같은 힘에 의해 작용하는 입자의 집합인 입자 그룹 내에 존재하며, 몇 개의 입자 그룹들이 동시에 존재할 수 있으나 모든 라이브러리 함수들은 현재의 입자 그룹에만 적용되며, 모든 액션들은 현재의 입자 그룹 내의 모든 입자에 적용되도록 한다.

다음 그림 2는 입자그룹 함수의 설정을 보여준다.

```

PARTICLEDLL_API void pCopyGroup(int p_src_group_num,
int index = 0, int copy_count = P_MAXINT);
PARTICLEDLL_API void pCurrentGroup(int p_group_num);
PARTICLEDLL_API void pDeleteParticleGroups(int p_group_num,
int p_group_count = 1);
    
```

▶▶ 그림 2. 입자그룹 함수의 설정

▶ 액션 함수

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

다음 그림 3과 같은 의사코드 형태를 나타내며 그림 4와 같이 정의된다.

```

for each particle group j
    pCurrentGroup(j)
    for each time step per render frame
        pSource(...)
        other actions...
        pMove(...)
    end for
    pDrawGroup(...)
end for
other drawing...
    
```

▶▶ 그림 3. 액션함수의 호출

```

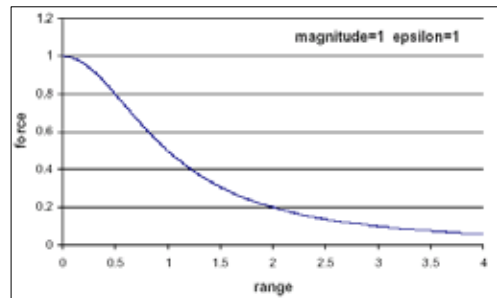
PARTICLEDLL_API void pAvoid
(float magnitude, float epsilon, float look_ahead,
PDomainEnum dtype,
float a0 = 0.0f, float a1 = 0.0f, float a2 = 0.0f,
float a3 = 0.0f, float a4 = 0.0f, float a5 = 0.0f,
float a6 = 0.0f, float a7 = 0.0f, float a8 = 0.0f);
    
```

▶▶ 그림 4. 액션함수의 정의

많은 액션함수들은 입자가 어떤 영향권에서 멀어짐에 따라 효과를 감소시키는 감쇠함수를 필요로 하는데 본 연구에서는 라이브러리의 동질성을 위하여 감쇠함수를 필요로 하는 액션 함수들에 대하여 다음과 같은 감쇠함수를 사용한다.

$$f_{m,r}(r) = \begin{cases} 0 & r \geq r_{max} \\ \frac{m}{r^2 + \epsilon} & r < r_{max} \end{cases}$$

다음 그림 5는 액션함수에서 사용된 감쇠함수에 대한 값의 변화를 그래프로 보여준다.



▶▶ 그림 5. 액션함수의 감쇠함수에 대한 값의 변화

▶ 액션 리스트 함수

액션 리스트를 사용함으로써 응용 프로그램과 입자 동력학을 계산하는 하드웨어 사이의 통신을 줄여주고, 필요한 액션들의 조합을 인식하여 최적화된 함수호출을 하게 되므로 성능을 크게 향상시킬 수 있다.

다음 그림 6은 액션 리스트 함수의 호출이다.

```

PARTICLEDLL_API void pCallActionList(int action_list_num);
PARTICLEDLL_API void pDeleteActionLists
(int action_list_num, int action_list_count = 1);
PARTICLEDLL_API void pEndActionList();
PARTICLEDLL_API int pGenActionLists(int action_list_count = 1);
PARTICLEDLL_API void pNewActionList(int action_list_num);
    
```

▶▶ 그림 6. 액션 리스트 함수의 호출

▶ 속성과 영역

새로운 입자를 생성할 때 입자의 색상, 속도, 크기, 수명 등의 속성을 지정해 주어야 한다.

다음 그림 7과 그림 8은 각각 상태설정 함수와 영역설정 함수의 예를 보여준다.

```

PARTICLEDLL_API void pColor(float red, float green, float
blue, float alpha = 1.0f);
PARTICLEDLL_API void pColorD(float alpha,
PDomainEnum dtype,
float a0 = 0.0f, float a1 = 0.0f, float a2 = 0.0f,
float a3 = 0.0f, float a4 = 0.0f, float a5 = 0.0f,
float a6 = 0.0f, float a7 = 0.0f, float a8 = 0.0f);
PARTICLEDLL_API void pSize(float size_x, float size_y =
1.0f, float size_z = 1.0f);

```

▶▶ 그림 7. 상태설정 함수

```

PARTICLEDLL_API enum PDomainEnum
{
    PDPoint = 0, // Single point
    PDLine = 1, // Line segment
    PDTriangle = 2, // Triangle
    PDPlane = 3, // Arbitrarily-oriented plane
    PDBox = 4, // Axis-aligned box ;
}

```

▶▶ 그림 8. 영역설정 함수

#### IV. 실시간 입자시스템 생성기의 구현

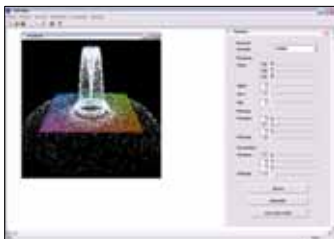
특수효과 생성기는 매개변수의 입력을 용이하게 하기 위해 MFC를 이용하여 GUI 기반의 인터페이스를 구축하였으며, 에디터의 오른쪽에 주요 도메인의 선택과 파라미터 입력을 위한 슬라이더와 입력 창을 생성하고 리셋 버튼과 시물레이션 버튼으로 화면을 갱신, 생성할 수 있게 하였다.

특수효과 생성기의 구현 환경은 다음 표 1과 같다.

[표 1] 구현 환경

소프트웨어	하드웨어
개발환경 : OpenGL, Visual C++	Graphic Card : ATI X800 pro
운영체제 : Windows XP Professional	CPU : Pentium4 3.2Ghz

다음 그림 9는 구현된 특수효과 생성기의 모습을 나타낸다.



▶▶ 그림 9. 특수효과 생성기의 구현

간 특수효과 생성기는 실시간 동적 입자 시스템으로 게임이나 가상현실 등의 다양한 디지털 콘텐츠 상에서 자연현상, 기상현상, 불, 불꽃 등의 특수효과를 적용하는데 활용성과 유연성 등이 뛰어나며 개발 시간과 비용을 크게 줄인다는 장점을 가지고 있으며 다양한 특수효과를 위한 각 파라미터들 간의 연관성있는 세밀한 조정에 많은 시간이 걸리지 않는다는 장점을 가지고 있다.

본 논문에서는 실시간 특수효과 라이브러리 형태인 입자 시스템 API를 구현하였으며 윈도우즈의 GUI 인터페이스인 수치 입력, 슬라이더, 버튼 등을 활용하여 속성을 결정하는 요소인 입자의 위치, 속도, 색상, 투명도, 크기, 수명 등 손쉽게 조정하도록 하는 실시간 특수효과 생성기를 구현하였다.

본 과제(결과물)은 교육인적자원부, 산업자원부, 노동부의 출연금으로 수행된 산학협력중심대학육성사업의 연구결과물입니다.

#### ■ 참고 문헌 ■

- [1] Reeves, W. T. "ParticleSystems - A Technigine for Modeling A Class of Fuzzy Objects". Proc. of SIGGRAPH '83, Detroit, Michigan, July, 1983.
- [2] Reeves, W. T. and R. Blau "Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems" Proc of SIGGRAPH '85, San Francisco, California, July, 1985
- [3] McAllister, D. K. "The Design of an API for Particle Systems" <http://cs.unc.edu/~davemc/Particle>, 1999
- [4] <http://www.opengl.org>

#### V. 결 론

본 연구의 특수효과 라이브러리인 입자 시스템 API와 실시