

물리 기반의 물과 반응적인 가스의 애니메이션 소프트웨어 제작 사례*

강병권¹ 김장희¹ 임인성¹

¹서강대학교 컴퓨터학과

¹한국전자통신연구원 디지털콘텐츠연구소

¹wormkbbk@grmanet.sogang.ac.kr, ¹rolldice@etri.re.kr, ¹ihm@sogang.ac.kr

Case Study: Software Development for Physically-Based Water and Reactive Gas Animation

Byungkwon Kang¹ Janghee Kim¹ Insung Ihm¹

¹Dept. of Computer Science, Sogang University

¹Electronics and Telecommunications Research Institute

요약

컴퓨터 그래픽스분야에서 유체 애니메이션은 애니메이션 제작에 있어 매우 중요한 표현 기법으로 그 적용 범위가 상당히 넓다고 할 수 있다. 그 중, 물과 가스는 대표적인 유체 표현 기법이며 그 시물레이션에 있어 각각 중요한 특징들을 가지고 있기 때문에 다양한 방법으로 적합한 시물레이션 방법들이 개발되어 왔다. 본 논문에서는 그 동안 본 연구실에서 개발해온 물리 기반의 물 시물레이션 기법에 대해서 소개하고 이를 통해 구현된 물 시물레이션 소프트웨어 제작 사례를 보이고자 한다. 또한, 최근 다양한 특수 효과를 위해 사용되는 폭발이나 화염 등의 유체 시물레이션에 유용하게 사용될 수 있는 반응적인 가스의 시물레이션 기법을 새롭게 제안하고 이 기법의 소프트웨어 제작 사례도 소개하고자 한다.

1. 서론

최근 컴퓨터 그래픽스 분야에서 계산 유체 역학 기법의 적용을 통하여 다양한 유체 애니메이션 효과를 생성하려는 시도가 활발히 진행되어 왔으며, 그 결과 상당히 사실적인 유체 효과를 애니메이션 제작에 적용할 수 있는 수준에 이르고 있다. 지난 수년간의 이러한 물리 기반 유체 애니메이션 기술의 발전에도 불구하고, 유체의

조절이나 고해상도의 유체 시물레이션을 통한 고해상도 영상 생성 등 아직도 컴퓨터 그래픽스 관점에서 해결해야 할 문제들이 산적해 있다.

여러 형태의 유체 중 물 또는 그와 유사한 형태의 유체는 애니메이션 제작에 있어 빈번히 사용되는 형태의 유체로서 그 적용 범위가 상당히 넓다고 할 수 있다. 본 연구실의 개발 경험에 따르면, 물리 기반 기법을 적용하여 유체 효과를 생성하는데 있어서 물은 연기나 가스와 같은 형태의 유체 보다 시물레이션을 하는데 있어 좀 더 까다로운 형태의 유체라 할 수 있으며, 이에 대하여 직관적으로 사용하기 쉬운 물 시물레이션 소프트웨어의 제작은 상당히 중요한 일이라 할 수 있다.

물과 같은 유체 이외의 물리 기반 애니메이션 분야도 다양한 연구가 진행되고 있다. 특히, 연기나 가스와 같이

* 본 연구는 한국과학재단 목적기초연구 (R01-2002-000-00512-0) 및 서강대학교 산업기술연구소의 지원을 받았음.

기체의 성격을 갖는 유체에 대한 애니메이션 기법은 오래 전부터 연구되어 왔으며, 최근에는 폭발이나 화염과 같은 특수 효과를 실제 애니메이션 제작에 적용하려는 연구가 진행되고 있다.

본 논문에서는 물이나 우유와 같은 유체의 사실적인 애니메이션 기법에 대하여 소개하고, 이러한 애니메이션을 제작하기 위한 시스템의 구현 사례에 대하여 소개하도록 하겠다. 그리고, 반응적인 가스를 이용하여 연기나 폭발과 같이 애니메이션 제작에 있어서 다양한 특수 효과에 적용될 수 있는 새로운 물리 기반 유체 애니메이션 기법에 대한 구현 사례도 살펴본다.

2. 기존 연구

90년대 후반 계산 유체 역학 기법들이 컴퓨터 그래픽스 분야에 소개가 되면서, 다양한 부류의 유체에 대하여 컴퓨터 애니메이션 제작에 적합한 시뮬레이션 기법들이 속속 제안되고 있다. 특히 물과 같은 액체의 시뮬레이션 기법들은 다양하게 개발되어 왔는데, [4]에서는 기존의 계산 유체역학에서 사용되던 기법을 3차원 컴퓨터 애니메이션에 적용할 수 있는 기법을 제안하였다. 여기서는 유체 역학 분야에서 널리 사용되고 있는 Navier-Stokes 방정식을 컴퓨터 그래픽스 분야에 적용할 수 있도록 하여 이를 물의 시뮬레이션에 적용하였다.

물과 같은 유체의 시뮬레이션 과정에 있어 중요한 것은 유체의 표면을 동적으로 추적해주어야 한다는 것이다. [4]에서는 기본적으로 표지 파티클(marker particle)을 사용하여 이 문제를 해결하려 하였다. [5]에서는 기존의 Navier-Stokes 방정식 풀이 기법이 갖는 불안정성을 해결하여 좀 더 안정적인 시뮬레이션이 가능하도록 하였으며, 물 표면의 추적에 있어 Lagrange 방식인 파티클 기법과 Euler 방식인 등위 집합(level set) 기법을 혼합하여 좀 더 부드러운 형태로 물 표면을 생성할 수 있었다. [6]에서는 [7]에서 제시한 파티클과 등위 집합의 혼합적 사용을 통해 물 표면을 보다 정확하게 추적할 수 있는 기법을 제안하여 컴퓨터 애니메이션 제작에 적용할 수 있도록 하였다.

물과 같은 유체의 시뮬레이션 기법과 더불어 연기, 가스과 같이 기체의 형태를 지닌 유체의 시뮬레이션 기법도 다양하게 연구되어왔다. [11]에서는 기존의 시뮬레이션 방법들의 계산 과정에서 야기되는 불안정성을 해결하기 위하여 Semi-Lagrangian 기법을 이용한 시뮬레이션 기법을 제안하였다. [12]에서는 이러한 시뮬레이션 기법을 연기의 시뮬레이션에 적용하였고, [13]에서는 이 기법을 기반으로 연기의 흐름을 제어할 수 있는 기법을 제안하였다. 이후 많은 연구에서 물리 기반 유체 애니메이션 기법을 적용하여 불꽃과 폭발, 화염과 같은 다양한

자연 현상을 사실적으로 표현할 수 있는 방법을 제안하였다[14,15,16,18,19].

3. 물, 우유와 같은 유체의 시뮬레이션 기법의 구현

3.1 Navier-Stokes 방정식을 이용한 수치해법

Navier-Stokes 방정식은 계산 유체역학 분야에서 유체의 흐름을 계산하기 위하여 사용되는 대표적인 공식이다. Navier-Stokes 방정식은 크게 아래 식(1)과 (2)와 같이 두 부분으로 나눌 수 있다. 식 (1)은 질량 보존의 법칙에 관한 식으로, 유체가 어떤 식으로 흐르던 그 질량은 보존된다는 것에 대한 식이다. 식(2)는 운동량 보존의 법칙을 통해 유체의 속도를 계산하는 부분이다.

$$\nabla \cdot \vec{U} = 0 \quad (1)$$

$$\vec{U}_t = \nu \nabla^2 (\nabla \vec{U}) - (\vec{U} \cdot \nabla) \vec{U} - \frac{1}{\rho} \nabla P + g \quad (2)$$

여기서 \vec{U} 는 속도를 나타내는 벡터이고 ∇ 는 그래디언트 연산자이다. 식(2)에서의 첫 번째 항은 유체의 점성을 나타내기 위한 부분으로 일반적인 중앙 차분법을 반복 적용하여 간단하게 계산이 가능하다. 두 번째 항은 유체의 대류현상을 계산하는 부분이며 <그림 1>과 같은 method of characteristic 방법을 사용하여 계산할 수 있다.

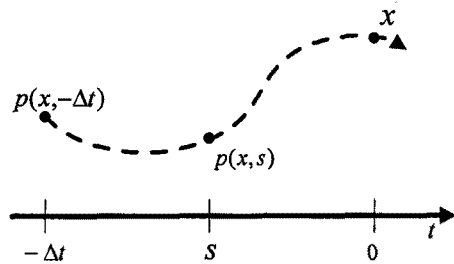


그림 1. Method of characteristic 기법. 과거의 시간으로부터 현재까지 흘러온 유체의 흐름 위에서 속도는 일정하다.

식(2)의 세 번째 항은 8개의 격자점으로 정의되는 셀(cell)의 여섯 면을 통해 흘러들어오고 나가는 유체의 흐름의 차이로부터 생기는 압력차를 라플라스 연산자(Laplacian operator)를 적용하여 생성되는 선형 방정식을 풀어서 보정하는 부분이다.

점성과 대류 부분까지 계산된 속도장에 대하여 질량 보존의 법칙인 식(1)을 적용하여 계산하면 다음과 같다.

$$\nabla^2 p = \frac{\rho \nabla \cdot \vec{U}^*}{\Delta t} \quad (3)$$

이 식은 선형식의 형태로 그 해를 구하게 되며, 이 선형식을 통해 얻어진 각 셀의 압력을 통해 다음 식(4)와 같이 속도를 보정한다. 이렇게 보정된 속도장은 위의 식(1)에서와 같이 질량 보존의 법칙을 만족하게 된다.

$$u_{ijk}^{t+\Delta t} = u_{ijk} - \frac{\Delta t}{\rho \Delta r} (p_n - p_{n-1}) \quad (4)$$

3.2 유체 표면의 추적

물과 같은 액체의 애니메이션에서는 유체의 표면을 추적하는 문제가 특히 중요한 문제이다. 액체의 경우 그 표면을 기준으로 안쪽과 바깥쪽이 분명하게 구별되고 결과적으로 우리가 물과 같은 액체를 보는 것은 그것이 가지고 있는 표면이기 때문이다. 표면을 부적절한 방법으로 추적하게 되면 그것은 결과에 확연히 드러나게 되고 사실적인 느낌을 줄 수가 없게 된다. 따라서 액체의 표면을 정확히 추출할 수 있는 기법의 사용이 무엇보다도 중요하다.

본 논문에서는 파티클 시스템에 의한 유체 표면의 추적 방법과 level set 기법을 사용한 추적 방법을 소개한다. 이 두 방법은 각각 우수한 장점을 가지고 있지만, 단점도 존재하기 때문에, 이 단점을 극복하기 위해 [6]에서 이 두 기법을 혼합한 방법을 제시하였다.

3.2.1 파티클 시스템에 의한 유체 표면의 추적

이 방법은 물, 진흙, 그리고 우유와 같은 형태의 유체의 흐름을 표현하기 위해서 파티클을 이용하여 등가면 추출법으로 유체의 표면을 추적하는 방법이다. 유체의 흐름을 나타내기 위한 파티클은 유체가 있는 부분을 대표한다. 이런 파티클은 일정한 반지름을 지니고 이 반지름에 해당하는 크기를 갖는 구로써 유체의 영역을 나타내게 된다. 그리고 이것은 앞서 수치 엔진에서 계산된 속도장에 의해 매 프레임마다 그 위치를 새로 계산함으로써 유체의 흐름을 추적한다. 이렇게 각 파티클들의 위치를 계산하고 나면 식(5)와 같이 각 파티클에 대한 유체의 표면을 계산할 수 있다.

$$\phi_p(x) = \sqrt{(x_i - x_{pi})^2 + (x_j - x_{pj})^2 + (x_k - x_{pk})^2} - r \quad (5)$$

위 식(5)에서 r 은 파티클의 반지름을 나타내며, 이 파티클 x_p 에서 유체의 표면은 $\phi_p(x)=0$ 에 해당하는 지점이 된다. 이 계산은 모든 파티클에 대해 해당되며 각 파티클들에 대해서 이 값이 가장 작은 값을 얻어 그것을 그 지점에서의 ϕ_p 로 정한다. 즉, 특정 지점에서 가장 가

까운 파티클 x_p 에 대한 $\phi_p(x)$ 을 함수 값으로 전체 격자 점에 대한 볼륨 데이터를 구성하게 된다. 이렇게 구한 볼륨 데이터에 대하여 마칭 큐브(Marching Cubes)[17]와 같은 등가면 추출법을 수행하면 우리가 원하는 유체의 표면에 대한 다면체 모델을 구해낼 수 있다. <그림 2>는 이러한 방법을 통해 얻어진 유체의 표면 다면체 데이터를 나타내고 있다.

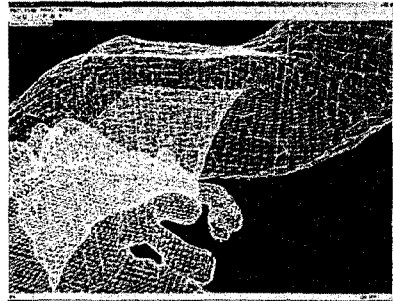


그림 2. 등가면 추출법을 통해 얻어진 유체 표면 다면체 데이터.

3.2.2 등위 집합(Level Set) 기법에 의한 유체 표면의 추적

파티클 시스템을 이용하여 유체 표면을 구하는 방법은 비교적 구현이 간단하다는 장점이 있지만 결과로 얻어지는 유체 표면이 사실적이지 못하다는 단점이 있다. 본 논문에서는 이를 해결하기 위해 [6]에서와 같이 등위 집합 기법을 사용한다. 등위 집합 기법은 유체의 표면으로부터 떨어져있는 거리를 부호 거리(signed distance) 함수 ϕ 를 이용해 공간에 정의를 하여 표현하고자 하는 물체(여기서는 유체)를 표현한다[8,9]. 부호 거리는 크기로는 표면과 떨어져 있는 거리를, 부호로는 정의하는 물체의 안과 밖을 표현한다. 물체가 시간에 따라 움직이고 동시에 표면도 움직이기 때문에 이렇게 표현된 부호 거리 함수는 공간의 함수임과 동시에 시간의 함수이다. 따라서 이를 함수의 형태로 표현하면 공간상의 지점 \vec{x} 와 시간 t 에 대한 함수 $\phi(\vec{x}, t)$ 가 되고 $\phi(\vec{x}, t)=0$ 이 되는 지점이 물체의 표면이 된다. 이렇게 표현된 부호 거리 정보를 이용하여 등가면 추출법을 이용하여 유체 표면을 얻을 수 있다.

등위 집합 기법으로 정의되는 부호 거리 ϕ 장의 시간에 따른 변화는 다음의 식으로 표현할 수 있다.

$$\phi_t + \vec{u} \cdot \nabla \phi = 0 \quad (6)$$

여기서 \vec{u} 는 해당 지점에서의 속도를 의미한다. 식에서

도 알 수 있듯이 물체의 법선 방향으로의 속도가 시간에 따른 ϕ 의 변화량이 된다. 하지만, 시간에 따라 식(6)을 계속 진행해 가다보면 유지되어야 하는 부호 거리의 성질이 깨지게 되는 문제점이 있다. 이 성질이 깨진 상태에서 식(6)을 사용하면 각 지점에서의 법선 벡터를 정상적으로 얻을 수 없기 때문에 오차가 계속 누적되어 원하는 결과를 얻을 수 없게 된다. 따라서 이를 보정해 주어 수치적으로 안정한 상태를 만들어 주어야 하는데 이를 재초기화(re-initialization)이라 한다. 재초기화를 위해서 다음과 같은 형태의 식을 사용한다.

$$\phi_r + \text{sgn}(\phi)(|\nabla\phi| - 1) = 0 \quad (7)$$

여기서 ϕ_r 는 임의의 시간 간격에 대한 ϕ 의 증분을 뜻하고 $\text{sgn}(\phi)$ 는 다음과 같다.

$$\text{sgn}(\phi) = \frac{\phi}{\sqrt{\phi^2 + |\nabla\phi|^2 \Delta x}} \quad (8)$$

여기서 Δx 는 격자의 크기이다. 부호 거리 함수의 성질이 $|\nabla\phi|=1$ 을 만족하는 성질을 가지고 있기 때문에 이를 다시 맞출 때까지 ϕ 를 scale factor $\text{sgn}(\phi)$ 을 사용해 조금씩 움직이며 조정하게 된다. 이 과정은 ϕ 를 직접적으로 수정하기 때문에 현재의 ϕ 가 표현하는 물체의 표면이 움직일 수 있다. 이것은 시뮬레이션의 목적으로 볼 때 적절한 결과라고 볼 수 없기 때문에 이러한 문제점을 최소화하기 위해 fast marching method를 사용한다[8].

3.2.3 등위 집합 기법과 파티클 시스템의 혼합

등위 집합 기법은 부드러운 유체의 표면을 얻기 위한 매우 유용한 방법이다. 하지만 이 방법은 유체의 세세한 특징까지 표현하지 못하고, 종종 유체 형태를 과도하게 부드럽게 만들어버리는 경향이 있다. 또 시간에 따른 유체의 움직임을 이 방법으로 추적할 경우, 유체의 부피를 조금씩 잃어버려 계산상의 정확성이 떨어진다는 문제도 존재한다. 따라서 이 두 방법의 문제점을 보완하기 위해 기본적으로 등위 집합 기법을 사용하고, 여기에 파티클 시스템을 결합하는 방법을 사용하였다[6].

이 방법은 일단 시간에 따른 유체 움직임을 추적해 두 방법을 이용해 독립적으로 수행하고, 이후에 파티클이 표현하는 유체를 이용해 등위 집합 기법이 잃어버린 유체의 세세한 특징이나 잃어버린 부피를 찾아내는 방법이다. 이는 공간상에 정의된 부호 거리를 직접 수정하면서 수행하는데, 이렇게 하면 등위 집합 기법의 단점을 극복할 수 있고 보다 정확히 유체의 흐름을 추적할 수 있게 된다.

3.3 시점의존 적응적 유체 시뮬레이션 기법

지금까지 설명한 유체 시뮬레이션 기법을 이용하여 상당히 사실적인 애니메이션을 얻을 수 있지만, 그 계산량

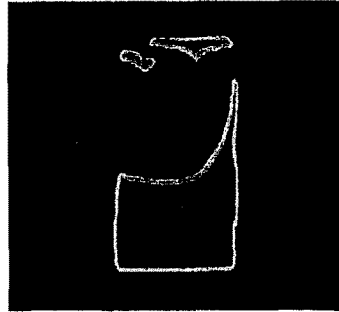


그림 5. 등위 집합 기법에 적용된 파티클들.

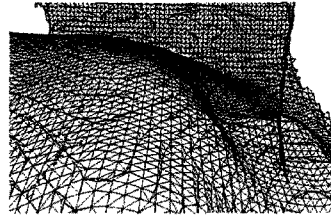


그림 6. 등위 집합 기법과 파티클 시스템의 혼합으로 얻은 부드러운 유체의 표면.

이 상당히 많고 방대한 메모리를 요구하기 때문에 아직까지도 고해상도의 시뮬레이션에는 많은 어려움이 있다. 본 논문에서는 계산 시간과 메모리 사용량을 보다 절약하기 위해 개발한 새로운 시뮬레이션 기법인 시점 의존적 적응적 유체 시뮬레이션 기법을 소개한다. 이 방법에서는 카메라에서 보이는 부분에 대하여 높은 해상도를 적용하는 시점 의존적 기법을 통해 적은 계산량과 메모리를 사용하여 높은 해상도의 시뮬레이션 효과를 구현할 수 있도록 하였다.

적절한 격자 구조를 설계하는 것은 그것이 계산상의 효율을 결정하는 중요한 요소이기 때문에 이 기법과 같은 적응적 시뮬레이션에서 매우 중요한 일이다. 여기서는 시점 의존적 적응적 격자 적용의 구현이 효율적으로 이루어도록 하기 위해 다음과 같은 매우 단순한 적응적 격자 구조를 사용하였다.

우선, 가장 큰 격자를 기준으로 기반 격자를 구성하여 이를 포인터 배열로 구성하고, 이렇게 구성된 격자에 필요한 레벨에 맞추어 상세 격자를 할당하여 포인터로 연결한다. 상세 격자의 개수는 한 좌표축당 2의 멱수로 그 크기를 정하고 인접한 격자간의 레벨차는 최대 1이 되도록 한다. 이렇게 적응적 격자를 구성하면 격자의 구조 유지에 필요한 메모리가 기반 격자의 크기만큼의 포인터 배열밖에 없기 때문에 상당히 작고, 두 번의 메모리 참조만으로 원하는 지역의 데이터에 접근이 가능하다는 장점이 있다.

이렇게 구성된 적응적 격자를 이용하여 카메라에 보이는 부분에만 적응적 격자를 적용하기 위해서 시점 의존적 방법을 사용한다. 만약, 뷰볼륨이 전체 공간 중 일부만 차지하고 있다면 뷰볼륨이 차지하는 공간에 보다 많은 자원을 집중하여 시물레이션의 질은 유지하면서 높은 계산 효율을 얻을 수 있다. 시점 의존 적응적 메쉬 분할을 사용할 경우 일반적으로 고려해야 할 사항들은 다음과 같이 정리할 수 있다.

우선, 보이는 지역의 해상도는 그렇지 않은 지역의 해상도에 비해 높아야 한다. 그리고, 물이 있는 지역의 해상도는 기본적으로 높아야 하고 보이는 지역에서의 물의 표면이 있는 지역은 가장 높은 해상도를 사용한다. 물의 표면이 있는 지역은 비록 뷰볼륨 바깥에 있다 하더라도 어느 정도의 해상도를 보장해 주어야 한다. 여기서의 해상도가 지나치게 낮으면 압력을 계산할 때 오차가 누적되어 전체적인 유체의 흐름이 잘못 계산되게 된다. 마지막으로, 인접한 기반 격자 간의 레벨 차이는 많아야 1만 나도록 한다. 만일, 뷰볼륨의 길이가 상당히 길다면 멀리 있는 지역의 해상도를 가까이 보이는 지역의 해상도보다 낮출 수 있다.

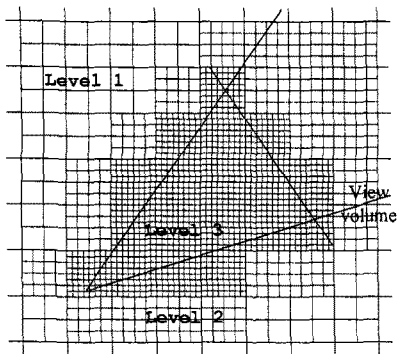


그림 5. 표면에 대한 시점 의존 메쉬 분할의 예.

<그림 5>는 이 기법을 이용하여 구성된 시점 의존 적응적 메쉬 분할의 예를 나타낸다. 이러한 방법을 이용하여 적은 수의 격자를 이용하여 고해상도의 시물레이션에 가까운 애니메이션을 효과적으로 얻을 수 있다.

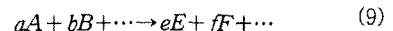
4. 화학 반응 속도식을 이용한 유체 애니메이션

지금까지 설명한 물 애니메이션과 더불어 가스과 같은 형태의 유체 애니메이션도 컴퓨터 그래픽스 분야에서 매우 유용한 분야이다. 이러한 가스의 애니메이션 기법에 대하여 다양한 연구들이 진행되어 왔으며, 최근에는 여러 종류의 가스들이 화학 반응을 일으키는 현상에 대해서도 많은 연구들이 진행되고 있다[24,25]. 실제 자연에서도 여러 물질들이 섞여있는 상황에서 이러한 물질들이 상호 작용하여 화학 반응을 일으키고 이로 인해 발생하는 다양한 자연 현상들이 존재한다. 특히, 폭발이나 연소와 같은 현상은 서로 반응하는 유체의 화학 반응을 이용하는 가장 대표적인 효과이다. 이러한 효과들은 유체 애니메이션 분야에서 사실적인 특수 효과 생성을 위해 유체 애니메이션에 많이 사용되는 주제이며, 많은 연구들에서 다양한 모델들이 제시되어 왔다[14,15,16,18,19,20].

본 논문에서는 실제 애니메이션을 제작하는 과정에서 애니메이션이 화학 반응을 일으키는 유체에 의한 특수 효과를 손쉽게 생성할 수 있는 기법을 제안하고 이를 통하여 다양한 특수 효과를 생성하기 위한 애니메이션 기법을 개발하였다[21]. 본 논문에서 구현한 애니메이션 기법은 [11,12]에서 연기와 가스 등의 유체의 표현을 위해 제안된 수치적 기법에 기반하고 있으며, 서로 다른 물질들 사이의 화학 반응을 시물레이션 과정에 추가하기 위하여 화학 반응 속도식의 개념을 적용하였다. 이 기법은 그 구현이 간단하고, 애니메이션이 화학 반응을 수반한 다양한 특수 효과를 생성하는 데 손쉽게 적용할 수 있도록 하는데 그 장점이 있다.

4.1. 화학 반응 속도론의 개념

화학 반응 속도론은 여러 물질이 화학적으로 반응하여 다른 새로운 물질로 변환될 때 나타나는 화학 반응의 속도와 메커니즘을 연구하는 것이다[22]. 임의의 화학 반응에 대한 반응식은 다음과 같이 표현된다(본 논문에서는 균일반응(homogeneous reaction)만을 고려하였다).



위 식(9)에서 a, b, \dots, e, f, \dots 는 화학량적 계수이며, A, B, \dots, E, F, \dots 는 반응을 이루는 물질들이다. 이러한 반응에서 어떠한 물질이 반응에서 소모되는 속도는 그 계수에 비례하며, 한 반응에서 모든 물질의 반응 비율은

일정하다. 이러한 균일 반응에서 임의의 시간 t 에서의 반응 속도 r 를 구하기 위한 반응 속도는 다음과 같이 각 반응물의 농도에 대한 함수로 표현된다.

$$r = -\frac{1}{a} \frac{d[A]}{dt} = \dots = k[A]^{\alpha}[B]^{\beta} \dots [L]^{\lambda} \quad (10)$$

위 식(10)에서 비례상수 k 는 속도상수 혹은 속도 계수라 하며, 온도와 압력에 대한 함수로 표현된다. 이런 속도 상수는 온도와 압력에 대하여 정의되지만, 대부분의 반응에서 압력에 대한 영향은 극히 적게 일어나기 때문에 일반적으로 압력에 대한 영향은 무시하고 온도에 대한 함수로만 정의된 Arrhenius 방정식을 사용한다. $\alpha, \beta, \dots, \lambda$ 는 각 반응물질이 갖는 차수가 되며, $\alpha + \beta + \dots + \lambda = n$ 은 이 반응의 차수라 한다. 이러한 반응 속도식을 적분하여 각 물질의 변화량을 구해낼 수 있다.

4.2 새로운 기법의 제안

4.2.1 유체 속도장의 계산

기본적으로 유체의 속도장을 계산하기 위하여 Navier-Stokes 방정식을 푸는 방법은 앞의 식(1)과 식(2)를 푸는 방법과 유사한 방법을 사용한다. 단, 연기와 같은 유체의 경우 아주 낮은 점성을 갖는다고 가정하여 식(2)에서 유체의 점성을 제어하는 상수인 ν 를 0으로 하여 단순화한 식을 사용한다. 또한, 격렬한 화학 반응을 통하여 발생하는 폭발과 같은 효과를 내기 위하여 [15]에서 제안한 것과 같이 새로 생성되는 질량을 원래의 유체의 질량에 더하는 다음과 같이 식(1)을 수정한 압력 보정 방법을 사용하였다.

$$\nabla \cdot u = \phi \quad (11)$$

$$\nabla^2 p = -\frac{\rho}{\Delta t} (\nabla \cdot u - \phi) \quad (12)$$

위 식에서 ϕ 는 모든 셀에 대하여 정의된 값으로 애니메이터가 원하는 효과를 생성하기 위하여 정의된 제어 함수로 계산된다.

4.2.2 속도장을 통한 밀도장과 온도장의 계산

각 셀에서의 유체의 속도장이 계산되고 나면, 다음과 같이 확산 상수 x_d 와 x_T 에 대하여 대류-확산 방정식을 이용하여 가스의 밀도장과 온도장을 계산한다.

$$\frac{\partial d}{\partial t} = -(u \cdot \nabla)d + x_d \nabla^2 d \quad (13)$$

$$\frac{\partial T}{\partial t} = -(u \cdot \nabla)T + x_T \nabla^2 T + \sigma_T \quad (14)$$

여기에서 가스의 밀도 d 는 그 물질의 질량 m 과 부피

v 에 대하여 $\frac{m}{v}$ 로 정의된다. 각 물질의 질량은 물질의 분자당 질량과 그 셀 안에서의 물질의 분자량의 곱으로 정의되며, 분자당 질량을 조절하여 무거운 연기와 가벼운 연기의 효과를 만들어낼 수 있다.

4.2.3 화학 반응식의 계산

유체의 흐름에 대한 계산이 끝나면, 화학 반응 $aA + bB + \dots \rightarrow eE + fF + \dots$ 에 대한 계산을 수행한다. 이 단계에서는 유체의 속도장이 정의되어 있는 이산적인 격자점들에 대하여 각 셀의 중심에 각 물질들의 분자량 $c = ([A], [B], \dots, [E], [F], \dots)$ 이 정의되어 있고, 하나의 셀 안에서는 그 밀도 값이 균일하다고 가정한다. 여기에서 애니메이터가 원하는 형태의 애니메이션을 얻어내기 위해서 $r = f_s([A], [B], \dots, [E], [F], \dots)$ 와 같은 반응 속도함수의 적절한 구성은 매우 중요한 부분이다. 애니메이터에 의해 반응 속도함수가 정의되면 반응이 일어난 후의 새로운 분자량 c 는 다음과 같은 미분 방정식을 시간 간격 dt 에 대하여 적분하여 계산한다.

$$\frac{d[A]}{dt} = -a \cdot r, \quad \frac{d[B]}{dt} = -b \cdot r, \quad \frac{d[E]}{dt} = e \cdot r, \dots \quad (15)$$

만약, 반응 속도 r 이 간단한 미분 방정식의 형태로 정의되면, 간단하게 해석적인 방법으로 구현이 가능하지만, 반응 속도식이 복잡하게 정의된 경우에는 복잡한 수치해석 기법이 적용되어야 한다. 매우 격한(steep) 형태의 시스템인 경우, Rosenbrock 방법과 같은 고차의 방법을 적용하여 미분 방정식을 풀어야 정확한 해를 구할 수 있지만 [23], 본 실험에 의하면 2차 수정 오일러 방법으로 충분한 정확도를 갖는 결과를 만들어낼 수 있었다.

4.2.4 외부 힘과 ϕ 값의 계산

본 논문에서 구현한 기법에서는 기존의 방법에 화학 반응식을 계산하는 과정에서 생기는 정보를 이용하여 원하는 형태로 속도장을 제어하는 기능을 추가하였다. 기존의 방법과 같이 밀도장과 온도장은 유체의 속도장에 의해 그 흐름이 계산되며, 이렇게 계산된 밀도장과 온도장은 외부힘 f 와 압력에 관한 함수인 ϕ 값을 통해 다시 속도장에 영향을 미치게 된다. 이러한 외부 힘은 유체 애니메이션에서 애니메이터가 유체의 흐름을 제어하기 위하여 사용될 수 있는 매우 유용한 부분으로 [12]에서 언급된 유체의 부양력과 소용돌이 힘, 사용자에 의해 정의된 힘을 들 수 있다(식 16,17).

$$f_{buoy} = -\alpha dz + \beta(T - T_{amb}) \quad (16)$$

$$f_{conf} = \epsilon h(N \times \omega A) \quad (17)$$

위 식에서 \underline{z} 는 중력의 반대방향을 나타내는 벡터이고, T_{amb} 는 열이 가해지지 않은 상태에서의 상온을 의미한다. 특히 본 논문에서는 화학 반응 과정에서 생성되는 정보를 물질의 온도를 제어하는 데 직접적으로 사용하였다. 우선, 화학 반응 속도에 따른 온도의 변화를 반응 속도 r 을 포함한 다양한 변수들에 대한 함수를 정의하고, 화학 반응이 진행됨에 따라 이 함수를 통해 유체의 온도장의 변화량을 계산하였다. 이렇게 계산된 온도의 변화량을 식(16)에 적용하여 유체의 온도에 따른 부양력을 계산한다.

이러한 외부 힘과 더불어, $f_\phi(r, \dots)$ 혹은 $\frac{d\phi}{dt} = f_\phi(r, \dots)$ 과 같이 반응 속도와 다른 변수들로 정의된 제어 함수인 ϕ 값을 식(12)에 적용하여 유체의 흐름을 제어하는 기법으로 구현하였다. 이 함수를 적절히 정의하여 애니메이션이 터가 화학 반응을 통한 유체의 팽창이나 수축을 자유롭게 제어할 수 있다.

4.3 포톤 매핑기법을 이용한 뜨거운 가스의 렌더링 기법

본 논문에서는 앞서 소개한 반응적인 가스의 시뮬레이션 기법을 통해 얻어진 폭발, 연소, 화염등의 뜨거운 가스를 렌더링하기 위하여 본 연구실에서 개발한 새로운 렌더링 기법을 소개하고자 한다. 이 기법은 높은 온도를 가진 뜨거운 가스가 광원의 역할을 할 때, 기존의 렌더링 기법으로는 표현하기 어려웠던 다양한 광원 효과들을 구현할 수 있도록 하였다. 이 기법에서는 방사 포톤맵(emission photon-map)과 섬광 포톤맵(flash photon-map)이라는 두 개의 새로운 포톤맵을 제안하고 이를 이용하여 다양한 광원 효과들을 구현할 수 있도록 하였다[27].

이 기법에서는 식 (18)과 같은 플랑크의 흑체 복사법칙을 이용하여 광원의 온도에 따른 총 복사에너지를 계산하고, 이를 이용하여 각 지점에서 방출될 포톤의 개수와 에너지를 결정할 수 있도록 하였다.

$$E(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{hc/\lambda kT} - 1} \quad (18)$$

각 지점에서 계산된 총 복사 에너지와 사용자 제어 함수를 통해 계산된 포톤의 개수만큼 방출된 포톤은 기존의 포톤 매핑기법과 같은 방법으로 밀도장 내부를 진행하게 된다. 이렇게 진행하면서 산란되는 방사 포톤들은 방사 포톤맵을 구성하게 되고, 이 포톤맵은 유체 광원을 렌더링하는 데에 사용되게 된다. 밀도장을 빠져나와 공기중으로 진행하게 되는 포톤들이 주변의 물체와 부딪히게 되면 그 곳에 전역 포톤맵으로 저장되고, 이는 기존의 포톤 매핑기법과 동일하게 렌더링되어 유체 광원이 주변을 비추는 광원 효과를 표현하게 된다. 그리고, 식

(19)와 같이 복사 에너지를 계산하는 파장 영역을 조절하여 사용자가 방출되는 빛의 파장을 조절할 수 있도록 하였다.

$$\Phi_{\text{tot}}(T, \lambda_{\min}, \lambda_{\max}) = \pi \int_{\lambda_{\min}}^{\lambda_{\max}} E(\lambda, T) d\lambda \quad (19)$$

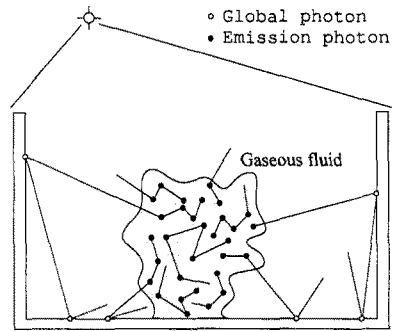


그림 8. 방사 포톤맵의 구성.

또한, 이 기법에서는 폭발 현상과 같이 높은 온도의 광원에 의해 발생하는 섬광 효과를 구현하기 위해서 섬광 포톤맵이라는 새로운 포톤맵을 제안하였다. 방사 포톤맵에서의 경우와 유사하게 유체의 각 격자점이 가진 온도로부터 총 복사 에너지를 계산하여 이 값을 이용하여 방출될 섬광 포톤의 양을 조절하여 섬광 효과를 나타낼 수 있도록 하였다. 각 격자점에서 방출되는 섬광 포톤들은 각각 격자점에서의 복사 에너지에 따라 일정량의 에너지를 갖고 방출되게 된다. 이 포톤들은 밀도장 내부를 산란을 일으키며 진행하게 되고, 밀도장을 빠져나와 공기중으로 진행하는 포톤들은 자신의 에너지를 잃을 때까지 뻗어나가게 된다. 밀도장 내부에서 산란을 일으키는 섬광 포톤들은 방사 포톤들과 유사하게 진행하지만, 매번 산란을 일으킬 때 마다 일정량의 에너지를 잃으면서 진행한다.

밀도장을 빠져나와 공기 중으로 진행하는 섬광 포톤들은 섬광 포톤맵에 포톤들을 저장하면서 진행하게 되는 데, 이렇게 저장된 포톤맵을 이용하여 섬광 효과를 표현할 수 있다. 섬광 포톤맵이 진행하는 거리는 각 포톤이 방출될 때, 유체의 복사에너지의 크기와 밀도장 내부에서 산란되어 감소된 에너지의 양을 통해 결정할 수 있으며, 진행 경로상에 저장되는 섬광 포톤의 밀도는 사용자에 의해 정의된 밀도 함수 δ_p 에 의해 결정된다.

이 기법에서 구현한 섬광 포톤맵을 사용하기 위해서는 기존의 Monte-Carlo 광선추적법이 적절히 수정되어야 한다. 하나의 광선을 따라 진행하다가 섬광 포톤맵이 존

재하는 영역과 만나게 되면 이 지역에서는 유체의 밀도장과 섬광 포톤맵을 모두 고려한 수정된 ray-marching 기법을 수행한다. 각 샘플링 지점 x 에서는 [28]에서의 같은 방법으로 유체의 밀도장과 방사 포톤맵에 의한 셰이딩을 계산하여 $C_{ve}(x)$ 를 계산하고, 밀도장의 불투명도를 $\alpha_{ve}(x) = 1 - e^{-\alpha(x)\Delta x}$ 와 같이 계산한다.

섬광 포톤맵의 밀도는 각 지점에서 유체의 밀도와 함께 적절히 반영되어야 하는데, 사용자 제어 함수에 의해 각 지점에서의 섬광 포톤맵의 색상과 밀도를 (C_p, α_p) 와 같이 계산하면, 샘플링 지점 x 에서의 색상과 불투명도 $(C(x), \alpha(x))$ 는 다음의 식 (20)과 같이 계산된다.

$$\begin{aligned} \alpha(x) &= \alpha_{ve}(x) + \alpha_f(x) \\ \alpha(x)C(x) &= \alpha_{ve}(x)C_{ve}(x) + \alpha_f(x)\Phi_p C_p \end{aligned} \quad (20)$$

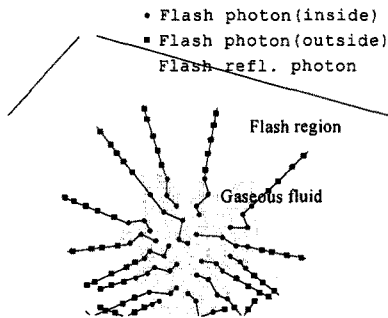


그림 9. 섬광 포톤맵의 구성.

5. 구현 결과

본 논문에서는 3장에서 설명한 유체 시뮬레이션 기법과 등위 집합 기법에 파티클 시스템을 혼합하여 유체의 표면을 생성하는 기법을 사용하여 유체 애니메이션 기법을 구현하였다. 그리고, 움직이는 물체가 유체에 미치는 영향을 계산하여 수치 엔진에 적용하였다. 그리고, 다음과 같이 다양한 예제 애니메이션을 통하여 실제 애니메이션 제작에 적용할 수 있는 결과를 보였다.

<그림 8>은 수조 안에 차 있는 물이 움직이는 공에 의해 영향을 받아 움직임을 보이는 장면이다. 이 장면을 만들기 위해 $70 \times 90 \times 120$ 의 해상도를 사용하였다. 이렇게 계산하면 결과로 부호 거리장이 생성되고 이를 다시 등가면 추출법을 이용하여 다면체 모델을 얻는 작업을

하게 된다.

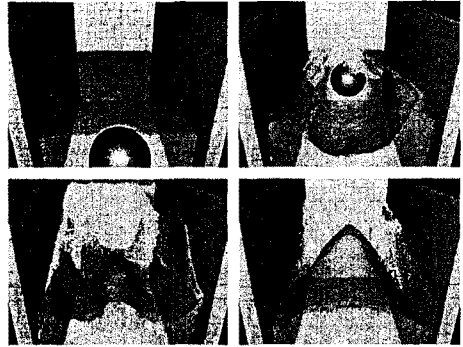


그림 10. 움직이는 공에 의해 움직이는 물의 장면.

<그림 9>는 세워져 있는 물기둥이 중력에 의해 쓰러지면서 구조물이 있는 수조 안에서 움직이는 장면이다. 이 애니메이션도 역시 $70 \times 90 \times 120$ 의 해상도를 사용하였다. 그림에서 알 수 있듯이, 등위 집합 기법에 의해 보다 부드러운 물의 표면을 얻을 수 있었고, 파티클 시스템에 의해 부피의 손실을 막으면서 안정적으로 장면을 얻을 수 있었다. 또 움직이는 물체가 유체에 주는 영향도 안정적으로 얻는 모습을 볼 수 있었다.

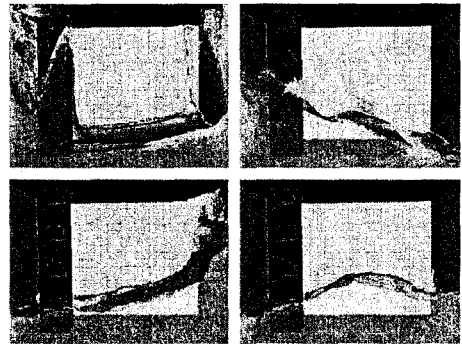


그림 11. 물기둥이 쓰러지는 장면.

<그림 10>에서는 3.3장에서 설명한 시점의존 적응적 격자를 이용한 물의 시뮬레이션 결과를 나타내고 있다. 왼쪽의 이미지는 기존의 균일 격자를 이용한 결과이며 가운데 이미지는 적응적 격자만을 이용한 결과이다. 각 이미지의 밑에 있는 그래프는 각 시뮬레이션의 계산에 사용되는 선형 방정식의 크기를 나타내고 있다. 시점 의존 적응적 격자를 이용한 방법은 균일 격자만을 이용한

방법보다 월등히 적은 계산량을 요구함을 알 수 있다.

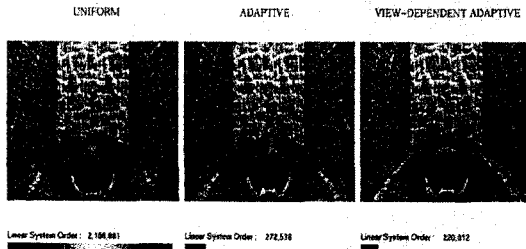


그림 10. 시점의존 적응적 격자를 이용한 시뮬레이션 결과.

또한, 본 논문에서는 기존의 유체 애니메이션 기법에 화학 반응 속도론을 적용하여 실제로 애니메이션 제작에 있어서 화학 반응을 포함한 유체의 특수 효과 제작에 적용할 수 있는 다양한 실험을 수행하였다. 실제로 애니메이션 제작에 효율성을 높이기 위하여 애니메이터가 쉽게 원하는 결과를 나타낼 수 있는 간단한 제어 방법을 구현하는 데 중점을 두고 있다.

<그림 11~13>에서는 본 논문에서 구현한 화학 반응을 이용한 애니메이션 기법을 이용하여 여러 가지 다양한 특수 효과를 제작할 수 있는 예제를 보여주고 있다.

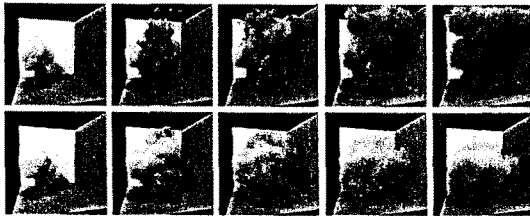


그림 13. 두 가지 연기의 간단한 화학 반응 효과.

<그림 11>에서는 간단히 두 물질의 연기가 만나서 화학 반응의 결과물로 새로운 연기를 만들어 내는 장면을 구현하였다. 그림의 첫 번째 줄은 단순히 두 연기가 만나 화학 반응을 일으켜 새로운 빨간색 연기를 생성하는 과정이다. 두 번째 줄은 같은 장면에 대하여 새로 생성되는 물질의 질량을 무겁게 하여 화학 반응이 진행될수록 연기의 질량이 무거워져서 그림과 같이 밑으로 가라앉는 효과를 구현한 것이다.



그림 14. 간단한 화학 반응식을 이용한 폭발 효과.

<그림 12>에서는 간단한 화학 반응식을 이용하여 폭발과 같은 효과를 표현하였다. 화학 반응식에 사용되는 각 물질의 화학량적 계수를 조절하여 좀 더 많은 양의 연기가 생성되게 할 수 있고, 이를 통해 풍부한 연기의 생성과 더불어 폭발과 같은 효과를 나타낼 수 있다. 이 과정에서 애니메이터에 의해 정의된 제어 함수로 계산되는 ϕ 값과 식(13)을 통해 연기가 팽창하거나 수축하는 현상이 표현 가능하다. 또한, 그림의 네 번째 줄은 온도에 대한 함수인 속도상수 k 를 조절하여 화학 반응의 속도를 조절할 수 있는 결과를 나타낸다.

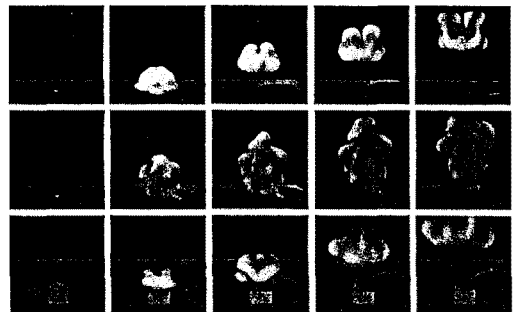


그림 13. 연쇄 반응을 이용한 폭발 효과의 실험.

<그림 13>에서는 복잡한 연립 미분방정식의 형태로 표현되는 연쇄 반응을 계산하여 폭발 현상을 시뮬레이션한 결과이다. 그림의 첫 번째와 두 번째 줄에 나타난 결과에서는 맨 처음 설정되는 각 물질의 초기 밀도장이 균일한 경우와 일정 크기의 노이즈를 적용하여 보다 사실적인 폭발효과를 구현하였다. 마지막 줄에서는 상자 안에서 폭발이 일어나서 터져 나오는 효과를 구현한 결과이다.

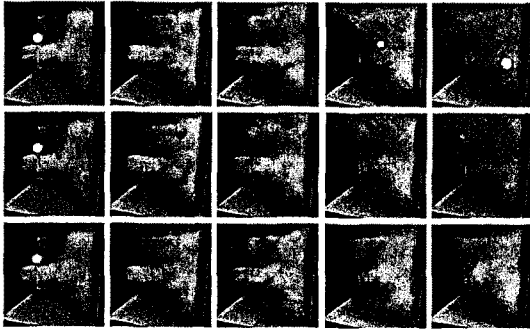


그림 14. 떠다니는 공에 의한 촉매 반응 효과 실험.

<그림 14>에서는 움직이는 물체에 의한 촉매 반응을 나타내었다. 그림에서 떠다니는 공이 연기와 만나면 촉매 반응을 일으켜 빨간색의 연기로 변화하는 과정을 나타내었다. 그림의 각각의 애니메이션에서는 연기가 떠다니는 공과 만나서 촉매 반응을 일으킬 때, 화학 반응 속도에 따른 온도 조절 함수를 조절하여 열을 발생시켜 물체를 떠오르게 하거나 반대로 열을 빼앗아서 물체와 연기가 가라앉는 효과를 구현한 것을 나타내었다.

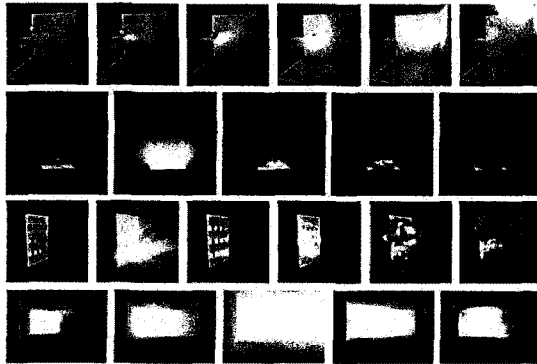


그림 17. 새로운 렌더링 기법의 구현 결과.

마지막으로, <그림 15>에서는 본 논문에서 소개한 새로운 렌더링 기법을 통해 반응적인 가스를 렌더링한 결과이다. 위로부터 첫 번째 줄은 <그림 14>에서 보인 촉매 반응의 결과를 새로운 렌더링 기법을 통해 렌더링한 결과이다. 두 번째 줄은 <그림 12>에서와 같은 간단한 반응식을 이용한 결과를 섬광 포톤맵과 함께 렌더링한 결과이다. 세 번째와 네 번째 줄은 연쇄 반응을 이용한 폭발 현상의 시뮬레이션을 새로운 렌더링 기법을 통해 구현한 결과이며 상당히 사실적인 광원 효과를 구현할 수 있었다.

6. 결론 및 향후 연구과제

본 논문에서는 물, 우유와 같은 액체의 시뮬레이션 기법에 대하여 소개하고 이를 이용하여 구현한 애니메이션 소프트웨어의 제작 사례를 보였다. 본 논문에서 구현한 애니메이션 소프트웨어를 통해 비교적 안정적이고 사실적인 물 애니메이션의 제작이 가능하였으며, 등위 집합 기법과 파티클 시스템의 혼합을 통해 보다 부드러운 유체의 표면을 부피의 손실 없이 찾을 수 있었다. 또한, Navier-Stokes 방정식의 풀이를 통해 유체의 흐름을 비교적 정확히 얻을 수 있으며, 움직이는 물체를 적용하여 유체의 흐름을 우리가 원하는 방향으로의 조정이 가능해졌다. 그리고, 본 논문에서 구현한 소프트웨어를 이용하여 다양한 애니메이션을 제작하여 그 우수성을 보였다.

하지만, 이러한 물 애니메이션 소프트웨어는 아직까지 그 계산량이 상당히 크고 사용되는 메모리의 양이 방대하다는 문제점을 가지고 있다. 본 논문에서는 이러한 문제점을 해결하기 위하여 시점에 의존적인 적응적 격자를 이용한 효과적인 시뮬레이션 기법을 소개하였다.

또한 본 논문에서는 화학 반응 속도식을 유체 시뮬레이션 과정에 접목하여 여러 물질이 반응하는 유체 애니메이션 기법을 제안하고 이를 이용한 애니메이션 소프트웨어를 구현하였다. 이 기법을 통해 간단한 반응적인 가스의 애니메이션부터 폭발, 촉매 반응에 이르기까지 다양한 애니메이션 효과를 구현할 수 있었다. 또한, 애니메이션이터가 유체의 움직임을 원하는 형태로 제어할 수 있는 다양한 요소를 개발하였으며, 이러한 제어 변수들의 직관적인 조절 함수를 정의하여 보다 쉽고 효율적인 애니메이션 제작이 가능하였다. 이러한 기법은 앞으로 애니메이션 제작에 있어서 보다 다양한 특수 효과를 생성해 낼 수 있을 뿐 아니라, 애니메이션이터가 직관적으로 유체를 제어할 수 있게 함으로써, 애니메이션 제작의 효율성을 높일 수 있을 것이라 기대한다.

7. 참고 문헌

- [1] A. Fournier, and W.T. Reeves, A simple model of ocean waves. In *Proc. of ACM SIGGRAPH 1986*, pp. 75-84, 1986.
- [2] D. Peachy, Modeling waves and surf. In *Proc. of ACM SIGGRAPH 1986*, pp. 65-74, 1986.
- [3] M. Kass, and G. Miller, Rapid, stable fluid dynamics for computer graphics. *ACM Computer Graphics (SIGGRAPH '90)*, Vol. 24, No. 4, pp. 49-57, 1990.
- [4] N. Foster, and D. Metaxas, Realistic animation of liquids. *Graphical Models and Image Processing*, Vol. 58, No. 5, pp. 471-483, 1996.
- [5] N. Foster, and R. Fedkiw, Practical animation of liquids. In *Proc. of ACM SIGGRAPH 2001*, pp. 23-30, 2001.
- [6] D. Enright, S. Marschner, and R. Fedkiw, Animation and rendering of complex water surfaces. In *Proc. of ACM SIGGRAPH 2002*, pp. 736-744, 2002.
- [7] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, Vol. 183, pp. 83-116, 2002.
- [8] S. Sethian, *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
- [9] S. Osher, and R. Fedkiw, *The Level Set Method and Dynamic Implicit Surfaces*. Springer-Verlag, New York, 2002.
- [10] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, A PDE-based fast local level set method. *Journal of Computational Physics*, vol. 155, pp. 410-438, 1999.
- [11] J. Stam, Stable fluids. In *Proc. of ACM SIGGRAPH 1999*, pp. 121-128, 1999.
- [12] R. Fedkiw, J. Stam, and H. Jensen, Visual simulation of smoke. In *Proc. of ACM SIGGRAPH 2001*, pp. 23-30, 2001.
- [13] A. Truille, A. McNamara, Z. Popovic, and J. Stam, Key frame control of smoke simulations. *ACM Transactions on Graphics (ACM SIGGRAPH 2003)*, Vol. 22, No. 3, pp. 716-723, 2003.
- [14] G. Yngve, J. O'brien, and J. Hodgins, Animating explosions. In *Proc. of ACM SIGGRAPH 2000*, pp. 29-36, 2000.
- [15] B. Feldman, J. O'brien, and O. Arikan, Animating suspended particle explosions. *ACM Transactions on Graphics (ACM SIGGRAPH 2003)*, 22(3), pp. 708-715, 2003.
- [16] N. Rasmussen, D. Nguyen, W. Geiger, and R. Fedkiw, Smoke simulation for large scale phenomena. *ACM Transactions on Graphics (ACM SIGGRAPH 2003)*, Vol. 22, No. 3, pp. 703-707, 2003.
- [17] W.E. Lorensen, and H.E. Cline, Marching cubes: A high resolution 3D surface reconstruction algorithm. In *Proc. of ACM SIGGRAPH 1987*, p. 163-169, 1987.
- [18] D. Nguyen, R. Fedkiw, and H. Jensen, Physically based modeling and animation of fire. *ACM Transactions on Graphics (ACM SIGGRAPH 2002)*, Vol. 21, No. 3, pp. 721-728, 2002.
- [19] M. Neff, and E. Fiume, A visual model for blast waves and fracture. In *Proc. of Graphics Interface 1999*, pp. 193-202, 1999.
- [20] O. Mazarak, C. Martins, and J. Amanatides, Animating exploding objects. In *Proc. of Graphics Interface 1999*, pp. 211-218, 1999.
- [21] I. Ihm, B. Kang, and D. Cha, Animation of reactive gaseous fluids through chemical kinetics. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 203-212, 2004.
- [22] I. Levine, *Physical Chemistry*, 5th ed. McGraw-Hill, Inc., 2002.
- [23] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C, The Art of Scientific Computing*, 2nd ed. Cambridge University Press, 1992.
- [24] R. Fedkiw, *A Survey of Chemically Reacting, Compressible Flows*, PhD thesis, Dept. of Mathematics, Univ. of California, Los Angeles, 1997.
- [25] W. Gates, *Animation of Reactive Fluids*, PhD thesis, Dept. of Computer Science, The Univ. of British Columbia, 2002.
- [26] G. Turk, Generating textures on arbitrary surfaces using reaction-diffusion. In *Proc. of ACM SIGGRAPH 1991*, pp. 289-298, 1991.
- [27] B. Kang, I. Ihm, and C. Bajaj, Extending the photon mapping method for realistic rendering of hot gaseous fluids. *Computer Animation and Virtual Worlds*, October 2005. (to appear)
- [28] H. W. Jensen, *Realistic Image Synthesis Using Photon Mapping*, A. K. Peters, Ltd, 2001.