Proc. of 2nd KSV Conference
POSTECH, Pohang, Korea, Nov. 14, 2003

CG 제작을 위한 유체 애니메이션

강병권† · 차득현* · 김장희* · 민정기* · 임인성*

Liquid Animation for CG Production

Byungkwon Kang, Dukhyun Cha, Janghee Kim, Jungki Min and Insung Ihm

Abstract

Fluid is an effective element in computer animation. Recently, the techniques from CFD have been actively applied to CG production. In this paper, we describe our fluid animation system which implements a variety of established simulation and rendering methods. We also explain our new techniques such as chemical reaction and hardware-assisted fluid animation that are being developed to enhance the features of our software system.

Key Words : Computer Animation, Computational Fluid Dynamics, Navier-Stokes Equation, Level-Set Method, Ray-Tracing, Photon Mapping, Reaction Kinetics, Programmable Graphics Hardware

1. 서 론

컴퓨터 그래픽스 분야에서 물, 불, 연기, 가스 등과 같은 자연현상의 사실적인 표현은 수준 높은 애니메이션의 제작에 있어 매우 중요한 사항중의 하나이다. 이러한 유체 애니메이션 기법은 최근 영화 '슈렉'이나 '개미'등의 3차원 애니메이션에서 이미 사용된 바 있으며, 사실적인 유체의 표현을 통해 영화의 사실성을 한 차원 높이는 데 큰 역할을 하였다.

본 논문에서는 이러한 유체 애니메이션 기법들을 컴퓨터 그래픽스 분야에 적용할 수 있는 기법들을 이용하여 유체의 흐름을 계산하기 위한 수치 엔진과 사실적인 유체의 렌더링을 위한 렌더링 엔진을 구현하고, 이를 이용하여 유체의 흐름을 제어할 수 있는 기법들에 대하여 제안하고자 한다.

2. 기존 연구

일반적으로 물리 기반 유체 애니메이션 기법들은 주로 정확한 유체 해석에 주안점을 두고 있기 때문에 많은 계산량이 방대할 뿐만 아니라, 애니메이터가 의도하는 대로 유체의 흐름을 제어하기 어렵기 때문에 컴퓨터 그래픽스 분야에 적용하기가 용이하지 않았다. 따라서 컴퓨터의 계산 능력이 현재보다 낮았던 과거에는 물과 같은 유체의 표현을 위해 이러한 물리 기반의 애니메이션 기법을 사용하기보다는, 수작업에 의존을 하거나, 여러 사인 곡선의 합성을 통한 웨이브 곡면을 이용한 기법이나 유체 표면의 높이지도(Height Map)를 이용한 기법들을 통해 유체와 비슷한 모양을 만들어내는 방법을 사용하곤 하였다[1,2,3].

최근에는 여러 연구에서 N. Foster 등은 유체역학에서 사용되는 대표적인 방정식중의 하나인 Navier-Stokes 방정식을 애니메이션 제작에 활용하기 용이한 형태로 풀어내는 기법을 제안하여 보다 사실적인 유체의 움직임을 생성하였으며[3,4], 이러한 유체역학 기반 기법들을 영화 '슈렉' 등에 성공적으로 적용하였다. 또한 J. Stam은 semi-Lagrangian 기법을 이용하여 기존 방법의 불안정성을 해결하고, 계산량을 줄여서 유체의 대화식 제어가 가능하도록 하였다[6]. 이러한 유체 애니메이션 기법들은 지속적으로 발전되고 있으며, 유체 역학에 기반을 둔 유체 흐름의 효과적인 생성뿐만 아니라 컴퓨터 그래픽스 관점에서 중요한 문제들, 즉 유체의 렌더링이나 직관적인 조절 등의 문제에 대하여 활발한 연구가 진행이 되고 있다[5,7,13,14,15].

3. Navier-Stokes 방정식의 풀이

Navier-Stokes 방정식은 전산 유체역학 분야에서 유체의 흐름을 계산하기 위하여 사용되는 대표적인 방정식이다. Navier-Stokes 방정식은 크게 아래와 같이 두 부분으로 나눌 수 있으며, 식(1)은 질량 보존의 법칙을, 식(2)는 운동량보존의 법칙을 통해 유체의 속도

† 책임저자의 소속: 서강대학교 공과대학 컴퓨터학과
E-mail: wormkbk@grmanet.sogang.ac.kr
* 공저자 소속: 서강대학교 공과대학 컴퓨터학과

를 계산하는 식이다.

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

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

위의 식(2)에서 우변의 첫 번째 항은 유체의 점성, 두 번째 항은 대류, 세 번째 항은 압력에 의한 속도 보정을 나타내며 마지막 항은 외부에서 작용하는 힘을 나타낸다. 점성 부분은 일반적인 중앙 차분법(Central Difference)을 이용하고, 대류 현상은 method of characteristic방법을 사용하여 계산한다. Method of characteristic 방법은 유체의 흐름을 나타내는 궤적 위에서의 속도는 모두 동일하다는 것을 이용한 방법으로서, 속도 궤적위를 역추적하여 구하는 속도를 얻을 수 있다. 세 번째 항에 의해 보정된 속도는 식(1)의 질량 보존의 법칙을 만족하는 속도장이 되며, 이러한 압력 보정은 유체의 특성에 따라 두 가지 방법으로 수행 가능하다. 우선, 첫 번째 방법은, 각 셀(cell)의 여섯 면을 통해 흘러들어오고 나가는 유체의 흐름의 차이로부터 생기는 압력차를 라플라스 연산자(Laplacian operator)를 적용하여 생성되는 선형 방정식을 풀어서 속도를 보정하는 방법이다.

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

$$\sum_{n=[ijk]} (p_{n+1} + p_{n-1}) - 6p = \rho \frac{\Delta \tau}{\Delta t} \sum_{n=[ijk]} (u_{n+1} - u_n) \quad (4)$$

식(4)를 모든 셀에 대하여 적용하면 전체 속도장에 대하여 $AP = B$와 같은 형태의 선형방정식을 얻을 수 있다. 본 논문에서는 이러한 선형방정식을 풀기 위하여 PCG(Pre-Conditioned Conjugate Gradients)방법을 사용하였다. 이 선형식을 통해 얻어진 각 셀의 압력을 통해 다음 식(5)와 같이 속도를 보정한다.

$$u_{ijk}^{t+\Delta t} = u_{ijk} - \frac{\Delta t}{\rho \Delta \tau}(p_n - p_{n-1}) \quad (5)$$

두 번째 방법은 Poisson 방정식을 통해 속도를 보정하는 방법이다. 우선 위에서의 방법과 같이 압력 보정이 되기 전까지의 속도장 w를 구한 다음 이러한 속도장을 Helmholtz-Hodge 분해에 의해 divergence free field인 u와 scalar field q로 다음과 같이 분해할 수 있다.

$$w = u + \nabla q \ (6) \longrightarrow \nabla \cdot w = \nabla^2 q \ (7)$$

Poisson 방정식인 식 (7)은 중앙 차분을 통하여 공간적으로 이산화 할 수 있으며, 결국 희소 선형(sparse linear) 계의 형태로 표현된다. 이를 풀어냄으로서 q를 구한다. 이렇게 구해진 scalar field인 q를 이용하여 다음과 같이 보정함으로써, 질량 보존의 법칙을 만족하는 속도장 u를 구할 수 있다.

$$u = Pw = w - \nabla q \quad (8)$$

본 논문에서 구현하고 실험해본 결과, 첫 번째 방법은 물, 진흙과 같은 액체의 성격을 지닌 유체에 적합하고, 두 번째 방법은 연기, 가스와 같은 기체의 성격을 갖는 유체에 적합한 방법임을 알 수 있었다.

마지막으로, 유체의 흐름에 영향을 미치는 외부 힘은 유체에 작용하는 중력, 온도에 의한 부양력, 소용돌이 힘과 같은 것들이 있다.

4. 사실적인 유체의 렌더링 기법

본 논문에서는 사실적인 유체의 렌더링을 위해 파티클 시스템과 광선 추적법을 기본으로, 포톤 매핑과 level set 기법을 적용한 렌더링 엔진을 구현하여 유체를 렌더링하였다.

4.1 파티클 시스템에 의한 유체 렌더링

본 논문에서는 물, 진흙, 그리고 우유와 같은 형태의 유체의 흐름을 표현하기 위해서 파티클 시스템을 이용한 등가면 추출법으로 렌더링을 수행하였다. 유체의 흐름을 나타내기 위한 파티클은 앞서 수치 엔진에서 계산된 속도장에 의해 매 프레임마다 그 위치를 새로 계산하게 되며 이러한 파티클들이 유체의 형태를 나타내는 기준이 된다. 한 지점 x_p에 위치한 파티클에 대한 유체의 표면은 다음과 같이 얻을 수 있다.

$$\Phi_p(x) = \sqrt{(x_i - x_{pi})^2 + (x_j - x_{pj})^2 + (x_k - x_{pk})^2} - r \quad (9)$$

위 식(9)에서 r은 파티클의 반지름을 나타내며, 이 파티클 x_p에서 유체의 표면은 $\Phi_p(x) = 0$에 해당하는 지점이 된다. 따라서 전체적인 유체의 표면은 모든 격자점에서의 값을 그 지점에서 가장 가까운 파티클 x_p에 대한 $\Phi_p(x)$의 함수 값으로 갖는 볼륨 데이터를 구성하고 이 볼륨 데이터에 대하여 Marching Cubes[8]와 같은 등가면 추출법을 수행하여 다면체 모델로 구해낼 수 있다.

4.2 Level Set 기법에 의한 유체 렌더링

위에서 설명한 파티클을 이용한 등가면 추출법은 반지름이 일정한 구의 모양을 이용하여 등가면을 생성하기 때문에, 최종적으로 유체를 생성했을 때, 표면이 부드럽지 못하고 울퉁불퉁한 모양을 만들어 낸다는 문제점이 있다. 이를 해결하기 위해 level set 기법을 사용한다. level set 기법은 유체의 표면으로부터 떨어져있는 거리를 signed distance 함수를 이용해 공간에 정의를 하여 유체를 표현한다[9,10].

Level set 기법으로 정의되는 signed distance ϕ field의 시간에 따른 변화는 다음의 식으로 표현할 수 있다.

$$\phi_t + \vec{u} \cdot \nabla \phi = 0 \quad (10)$$

여기서 $\vec{u}$는 해당 지점에서의 속도를 의미한다. 식에서도 알 수 있듯이 물체의 법선 방향으로의 속도가 시간에 따른 ϕ의 변화량이 된다.

시간에 따라 식 (10)을 계속 진행해 가다보면 signed distance의 성질이 유지되지 않는 시점이 생기게 된다. 따라서 이를 보정해 주어 수치적으로 안정한 상태를 만들어 주어야 하는데 이를 재초기화(re-initialization)이라 한다. 재초기화를 위해서 다음과 같은 형태의 식을 사용한다.

$$\phi_\tau + sgn(\phi)(\nabla \phi - 1) = 0 \quad (11)$$

하지만 이 방법은 유체의 세세한 특징까지 표현하지 못하고, 종종 유체 형태를 과도하게 부드럽게 만들어 버리고 부피가 조금씩 줄어드는 문제점이 있다. 따라

서 이 두 방법의 문제점을 보완하기 위해 기본적으로 level set 기법을 사용하고, 여기에 파티클 시스템을 결합하는 방법을 사용하였다[5]. 이렇게 하면 level set 기법의 단점을 극복할 수 있고 보다 정확히 유체의 흐름을 추적할 수 있게 된다.

4.3 연기, 가스 등과 같은 기체의 렌더링

기존의 광선 추적법이 물체의 표면에서 쉐이딩 계산이 이루어졌던 것과 달리, 연기나 가스의 경우 표면뿐만이 아니라 물질 내부에서도 쉐이딩 계산이 이루어져야 한다. 이와 같은 물질을 렌더링하기 위하여 기존의 광선추적법과는 다른 렌더링 기법이 필요하며, 본 논문에서는 Ray-Marching기법과 포톤 매핑 기법을 이용하여 렌더링을 수행하였다. 본 논문에서 구현한 렌더링 엔진에서는 연기나 가스와 같은 물질의 특성상 누적된 불투명도 값에 따라 효과적인 조기광선종료가 가능한 forward ray-marching 알고리즘을 이용하였다[7].

포톤 매핑(Photon Mapping)이란 효과적인 전역 조명(global illumination)을 계산하기 위하여 개발된 기법이다[11]. 우선 각 광원에서 퍼져 나오는 빛의 입자를 광자(photon)라 하고, 전체 장면에서 이 광자가 부딪히는 부분이 광원으로부터 빛이 도달하는 부분으로 보고, 그 지점의 밝기를 높여주는 기법이다. 이러한 포톤들을 저장하고 있는 정보를 포톤맵이라 하며, 각 쉐이딩 지점에서의 간접 조명은 포톤맵으로부터 N개의 가장 가까운 포톤들을 찾아 그 밀도를 통해 계산할 수 있다(식 12). 빠른 포톤 검색을 위해 포톤맵은 K-D트리로 구성되며, 트리의 깊이가 최적화된 균형K-D트리의 경우 k개의 포톤을 검색하는 데에 $O(k+logN)$의 탐색 시간을 갖는다.

$$L_r(x,\overrightarrow{w}) \approx \frac{1}{\pi r^2}\sum_{p=1}^{N} f_r(x,\overrightarrow{w_p},\overrightarrow{w})\Delta\Phi_p(x,\overrightarrow{w_p}) \quad (12)$$

일반적인 물체의 경우, 포톤의 반응은 물체의 표면에서 일어나지만, 연기와 같이 밀도장으로 이루어진 물질은 물질 표면뿐만 아니라 물질 내부에서도 반응이 일어난다. 이를 통해 연기 내부에서의 빛의 산란을 표현할 수 있다.

밀도장의 경우 각 지점에서 포톤이 물질과 반응하는지 여부는 그 지점까지 진행해온 동안 누적된 밀도값에 대한 확률 함수로 결정되며, 반응이 일어나게 되면 그 지점에서 포톤이 흡수될 것인지 산란된 방향으로 진행할 것인지를 물질의 알베도(albedo)값의 확률 함수로 결정한다. 포톤의 산란이 일어날 때, 이러한 난반사의 방향은 phase function에 의해 구해지는데, 본 논문에서는 Ray-Marching 알고리즘에 적용이 쉽고 구현이 간단한 Henyey-Greenstein Phase Function을 사용하였다(식 (13)).

$$p(\theta) = \frac{1-g^2}{4\pi(1+g^2-2gcos\theta)^{1.5}} \quad (13)$$

이렇게 구성된 포톤맵과 앞서 기술한 Ray-Marching기법을 통해 밀도장의 렌더링을 수행할 수 있다.

5. 화학 반응속도식의 유체 애니메이션 응용

반응 속도론은 여러 물질이 화학적으로 반응하여 다른 새로운 물질로 변환될 때 나타나는 화학 반응의 속도와 메카니즘을 연구하는 것이다. 임의의 화학 반응에 대한 반응식은 다음과 같이 표현된다.(본 연구에서는 균일반응(Homogeneous reaction)만을 고려하였다)

$$aA + bB + \cdots \rightarrow eE + fF + \cdots \quad (14)$$

위 식(14)에서 a,b,...,e,f,...는 화학량적 계수이며, A,B,...,E,F,...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 t에서의 반응 속도 r을 구하기 위한 반응 속도는 다음과 같이 각 반응물의 농도에 대한 함수로 표현된다.

$$r = -\frac{1}{a}\frac{d[A]}{dt} = \cdots = k[A]^{\alpha}[B]^{\beta}\cdots[L]^{\lambda} \quad (15)$$

위 식(15)에서 비례상수 k는 속도상수 혹은 속도 계수라 하며, 온도와 압력에 대한 함수로 표현된다. $\alpha,\beta,\cdots,\lambda$는 각 반응물질이 갖는 차수가 되며, $\alpha+\beta+\cdots+\lambda \equiv n$은 이 반응의 차수라 한다. 이러한 반응 속도식을 적분하여 각 물질의 변화량을 구해낼 수 있다. 본 논문에서는 이러한 적분을 계산하기 위하여 4차 룬지-쿠타 방법을 이용하였다. 또한 이렇게 계산된 반응 속도 r은 유체 애니메이션을 컨트롤하기 위한 여러 가지 요소로 사용될 수 있는데, 속도 r에 비례하도록 유체의 온도를 조절하거나 소용돌이 힘, 압력 등을 조절하여 유체의 흐름을 컨트롤할 수 있다. 비록, 이러한 방법들이 물리학적이나 화학적으로 정확한 방법은 아닐지라도, 실제로 애니메이션을 제작하는 데에 있어 애니메이터들이 유체를 컨트롤할 수 있는 한 가지 성질로서 사용될 수 있다는 점에서 유용한 방법으로 사용될 수 있을 것이다.

6. GPU를 이용한 선형식 계산

최근 보편화되어 사용되고 있는 그래픽스 하드웨어는 이전의 고정된 그래픽 파이프라인과는 달리 프로그래밍이 가능한 고성능의 그래픽 파이프라인을 가지게 되었다. 그로 인하여 여러 가지 사실적인 이미지들을 실시간으로 렌더링할 수 있게 되었을 뿐만 아니라 여러 가지 수치 계산에 대해 CPU보다 빠르게 연산을 수행할 수 있게 되었다. 프로그래밍이 가능한 그래픽스 파이프라인으로 버텍스 쉐이더와 픽셀 쉐이더가 제공되는데, 본 논문에서는 8개의 픽셀 파이프라인을 가지고 있는 픽셀 쉐이더를 이용하여 여러 번 반복 수행되어야 하는 문제를 푸는데 적용하였다.

이 실험에서는 유체 엔진에서 사용되는 Poisson 방정식에 대하여 이러한 방법을 적용할 수 있는 실험을 수행하였다. Poisson 방정식을 풀기 위한 실험에서 가우스 자이델 방법을 픽셀 쉐이더를 사용함으로써 CPU보다 빠르게 계산할 수 있음을 확인하였다. 실험에 사용되었던 PC 사양은 펜티엄4 2.6GHz, 512M DDR PC3200 RAM, GeForceFX5900 Ultra이였다. 또한 [16]에서 제안한 최적화 기법을 적용하면 전체 연산의 수가 30~40%정도 줄일 수 있기 때문에 더 빠른 수행이 가능하였다. 결과로 40x40x40의 해상도를 갖는 유체 애니메이션에 대하여 픽셀 쉐이더를 사용하여 프로그램을 수행하는 것이 CPU에서의 계산보다 4배 이상 빠름을 확인하였다.

7. 구현 및 결론

지금까지 본 연구실에서 개발 중인, 자연스러운 애니메이션 제작을 위한 유체 역학 기반 애니메이션 소프트웨어에 대하여 간략히 설명을 하였다. 물리 기반 기법은 3차원 애니메이션이나 3차원 게임 등에 현실감 있는 장면을 생성하는데 중요한 역할을 하나, 적용하기에 많은 어려움이 있는 것이 사실이다. 실제로 이러한 유체 애니메이션이 효과적으로 사용된 영화 '슈렉'에서는 영화 제작 과정 중에 진흙과 맥주와 같은 유체를 표현하는 작업이 가장 어려웠던 작업 중의 하나였다고 할 만큼 도전적인 작업이었다. 이러한 애니메이션 과정을 용이하게 하기 위해서는 전기한 바와 같이 애니메이터가 원하는 방식으로 유체를 조절할 수 있도록 컴퓨터 애니메이션 기법과 유체 역학의 기법을 효과적으로 결합해주는 방법들이 개발되어야 할 것이다.

본 논문에서 구현한 수치 엔진과 렌더링 엔진을 이용하여 물, 진흙, 우유 등과 같은 액체의 특성을 갖는 유체와 연기, 가스와 같은 기체의 특성을 갖는 유체에 대하여 사실감 있는 애니메이션을 구현할 수 있다. 오프라인 애니메이션에서는 계산 시간이 큰 문제는 아니나, 아직도 실시간 애니메이션에 적용하기에는 계산량이 많다고 할 수 있다. 따라서 본 논문에서 기술한 그래픽스 하드웨어를 이용한 선형식 계산기법을 통하여 최적화하는 것은 향후 수년간 중요한 연구 주제로 대두가 될 것이다.

참 고 문 헌

[1] Fournier, A. and Reeves, W.T., "A Simple Model of Ocean Waves," ACM SIGGRAPH '86, pp. 75-84, 1986.

[2] Kass, M. and Miller, G., "Rapid, Stable Fluid Dynamics for Computer Graphics," ACM SIGGRAPH '90, pp. 49-57, 1990.

[3] Foster, N. and Metaxas, D., "Realistic Animation of Liquids," Graphical Models and Image Processing, Vol. 58, No. 5, pp. 471-483, 1996.

[4] Foster, N. and Fedkiw, R., "Practical Animation of Liquids," ACM SIGGRAPH 2001, pp. 23-30, 2001.

[5] Enright, D., Marschner, S. and Fedkiw, R., "Animation and Rendering of Complex Water Surfaces," ACM SIGGRAPH 2002, pp. 736-744, 2002.

[6] Stam, J., "Stable Fluids," ACM SIGGRAPH '99, pp. 121-128, 1999.

[7] Fedkiw, R., Stam, J. and Jensen, H.W.,"Visual Simulation of Smoke," ACM SIGGRAPH 2001, pp. 23-30, 2001.

[8] Lorensen, W.E. and Cline, H.E., "Marching Cubes: a High Resolution 3D Surface Reconstruction Algorithm," ACM SIGGRAPH '87, pp. 163-169, 1987.

[9] Sethian, J., Level Set Methods and Fast Marching Methods, Cambridge University Press, 1999.

[10] Osher, S., and Fedkiw, R., The Level Set Method and Dynamic Implicit Surfaces, Springer-Verlag, New York, 2002.

[11] Jensen, H.W., Realistic Image Synthesis Using Photon Mapping, AK Peters, 2001.

[12] Sande, L.R., Shrek: The Story Behind the Screen, ACM SIGGRAPH 2001 Course Note #19, 2001.

[13] Rasmussen, N., Nguyen, D., Geiger, W., and Fedkiw, R.F., "Smoke Simulation for Large Scale Phenomena," ACM SIGGRAPH 2003, 2003(To appear).

[14] Treuille, A., McNamara, A., Popovic, Z., and Stam, J., "Keyframe Control of Smoke Simulations," ACM SIGGRAPH 2003, 2003(To appear).

[15] Stam, J., "Flows on Surfaces of Arbitrary Topology," ACM SIGGRAPH 2003, 2003(To appear).

[16] 오진상, "프로그래밍이 가능한 GPU상에서의 선형식을 위한 버텍스 프로그래밍의 최적화 기법", 석사논문, 서강대학교 대학원 컴퓨터학과, 2002년 2월.

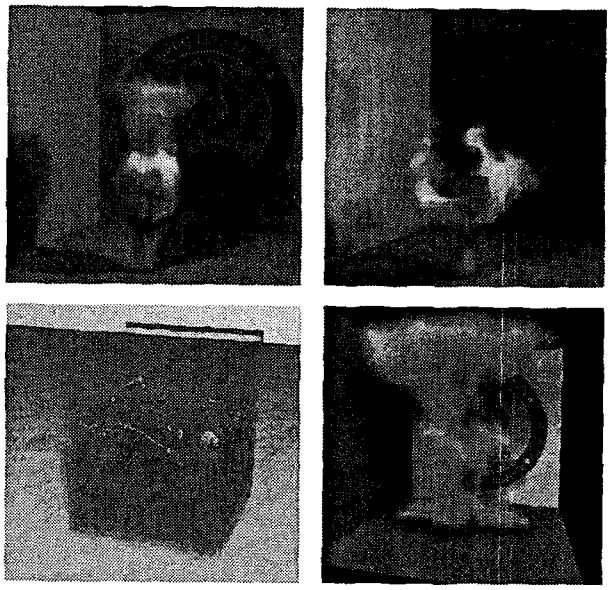

그림 1. 구현 결과