

아카데미상 영화에서 유체 시뮬레이션 기술

김명규 오승택 최병태

한국전자통신연구원 콘텐츠연구본부
{mgkim, stoh, btchoi}@etri.re.kr

Fluid Simulations in Academy Awarded Movies

Myunggyu Kim Seungtaik Oh ByoungTae Choi

ETRI Content Research Division

요약

유체 시뮬레이션은 영화나 게임에 등장하는 물, 연기, 불, 폭발, 모래 등과 관련된 애니메이션을 생성하기 위해 유체 운동 방정식을 풀어 형상화하는 컴퓨터 그래픽스 기술이다. 본 리뷰 논문은 유체 시뮬레이션에 대한 최근의 연구 동향을 영화를 중심으로 분석한다. 이를 위해 먼저 컴퓨터 그래픽스 분야에서 유체 시뮬레이션을 하는 방법과 이와 관련된 효율적 속도, 사실적 형상화 등의 기술적 이슈들을 격자 방법과 입자 방법으로 분류하여 설명한다. 다음으로 2008년 Sci-Tech 오스카상을 수상한 유체 시뮬레이션 연구자 및 개발자들의 업적을 중심으로 그들이 제작한 영화 속에 사용된 기술의 특징을 설명한다. 이 과정을 통해 향후 영상 콘텐츠 분야의 유체 시뮬레이션 연구 개발의 발전 방향을 짚어 보면 유체와 변형체나 고체 사이의 상호작용, 상변이에 의한 위상 변화 표현, 유체 기반 크리처 생성과 크리처와 유체 상호작용 표현 기술이 주목을 받을 것으로 예상된다.

Abstract

Fluid simulation for computer graphics is a field of generating the realistic movements of water, smoke, fire, explosion, sand and related phenomena to be used in motion pictures and video games. In this paper we review the fluid simulation technologies and present a trend analysis for the simulation methods used in the recent movies. First of all, for this purpose, the two methods that are most widely used for fluid simulation are explained as well as their technical issues. These two methods are classified into Eulerian grid-based and Lagrangian particle-based approaches. Next, focusing on the achievements of the scientists and engineers that the 2008 Sci-Tech Oscar Awards are given to, the features of their fluid simulation technologies are analyzed. Finally, we anticipate that there are and will be the needs for visualizing fluid interaction with rigid and soft bodies and topological change among solid, fluid and gas, creating digital creatures based on fluid simulation and presenting interaction between creature and fluid.

키워드: 유체, 크리처, 나비에-스톡스, 모래
Keywords: Fluid, Creature, Navier-Stokes, Sand

1. 서론

아카데미상 즉 오스카상은 전 세계인이 주목하는 영화 제전으로 매년 2월 말 즈음에 열린다. 메인 오스카상 수 주 전에 영화 제작을 위한 과학적 공학적 기술적 분야의 인물들에게 수여하는 Sci-Tech Oscar 상 시상식이 별도의 행사로 개최된다. 2008년의 경우 Oscar 시상식은 2월 24일 일요일에 Sci-Tech Oscar 시상식은 2월 9일 토요일에 개최되었다. Sci-Tech Oscar 수상의 성격은 전년도만이 아니라 수 년 간의 영화 제작 기술에 공헌한 공학자, 수학자, 과학자들에 대한 명예의 전당과 유사하다. 올해의 경우 흥미롭게도 Sci-Tech Oscar 10개 부문 중 6개 부분이 유체 시뮬레이션 소프트웨어를 연구/개발하는 팀들에게 주어졌다.

CG에서의 유체 애니메이션은 영화 속에서 물, 연기, 불, 바람, 안개, 모래 등의 움직임 특수 효과를 표현하는 기술이다.

유체 애니메이션은 모델링, 시뮬레이션, 형상화와 렌더링 순으로 진행된다 [1]. 이 중 모델링은 시뮬레이션 대상이 되는 유체와 이와 상호작용하는 물체를 격자 합수나 입자들 위치로 표현하는 방식을 의미한다. 시뮬레이션을 진행하기 위한 초기 조건과 경계 조건의 설정도 이 과정에 포함된다. 시뮬레이션은 유체의 움직임을 표현하는 동역학 방정식을 시간 진행에 따라 해를 구해가는 과정이다. 매 시간마다 시뮬레이션을 통해 얻어진 유체 움직임 표현 자료 구조로부터 형상화와 렌더링 작업을 수행하여야 한다. 이 중 형상화에서는 격자 합수 값이나 입자 위치값으로부터 물의 표면이나 가스의 외형을 추출해 나가는 과정이고, 렌더링은 유체에 투여 되어서 관측자에게 보여지는 빛의 종류와 양으로부터 최종 영상을 생성하는 과정이다. 본 논문에서는 렌더링은 다루지 않고 시뮬레이션을 중심으로 모델링과 형상화도 같이 다루는데 이들 세 가지 개념을 함께 시뮬레이션이라고도 칭한다.

2와 3장에서는 각각 초창기의 유체 애니메이션과 이의 문제점으로부터 요구된 유체 시뮬레이션의 필요성에 대하여 논한다. 4장에서는 유체 시뮬레이션의 연구 역사와 방법을 격자화 입자 방법으로 나누어 살펴 본다. 5장에서는 2008년 Sci-Tech Oscar 수상 SW들의 특징과 4장의 연구 결과가 어떻게 반영되었는지 영화를 중심으로 정리한다. 6장에서는 5장에서 살펴본 문제점들을 해결하기 위한 최근의 유체 시뮬레이션 기술 개발 동향을 유체 중심 상호작용, 유체 위상 변화와 유체 기반 크리처 시뮬레이션 관점에서 분석한다. 마지막 장에서 결론을 맺는다.

2. 초창기의 유체 애니메이션

CG 유체 특수 효과는 학계에서 1980년대부터 연구되었다 [2, 3]. 실제 영화 속에서 유체 표현은 1983년 “스타 트렉 2”와 1989년 “어비스(The Abyss)”를 들 수 있다. “스타 트렉 2”에서는 화염에 싸이는 행성을 표현하기 위해 입자 시스템을 사용하였다.



그림 1: 영화 “어비스 (1989)” (20th Century Fox/Courtesy Everett Collection)

1980년대 유체 표현은 수작업에 의해 프레임별로 수행되었다. 예를 들어 (그림 1)의 영화 “어비스”에서의 외계 생명체의 물 촉수는 전통적인 셀 애니메이션처럼 프레임 단위로 고품질의 몸체 위에 물 피부를 씌우는 단순 작업의 결과물이다.

3. 유체 시뮬레이션의 필요성

유체 표현을 자동화하기 위해서는 유체 지배 방정식에 기반한 시뮬레이션이 필요하다. 유체 지배 방정식은 고전 뉴턴 역학으로 유도되는 Navier-Stokes (NS) 방정식이나 이를 특수한 경우에 단순화한 Shallow Water 방정식이나 파동 방정식을 의미한다. 유체 시뮬레이션이란 이들 방정식을 풀어 시간별로 유체의 공간 형상을 얻어내는 과정을 의미한다. Shallow Water 방정식은 유체의 운동이 얇은 물로 국한되면서 상대적으로 운동의 강도가 약할 때 NS 방정식으로 부터 유도되며, 파동 방정식은 비선형인 Shallow Water 방정식을 선형 근사하여 구할 수 있다 [4]. 본 논문에서는 이와 같이 상대적으로 정적인 Shallow Water 방정식이나 파동 방정식 기반 유체 시뮬레이션은 다루지 않고 상대적으로 동적인 NS 기반 유체 시뮬레이션 및 영상 특수 효과에 대해서만 분석한다.

뉴턴 동역학으로부터 유체의 밀도장과 속도장은 다음과 같

은 편미분 방정식을 만족함을 유도할 수 있다.

$$\rho_t + \rho \nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\rho(\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}) = \rho \mathbf{g} - \nabla p + \mu \nabla^2 \mathbf{u} + (\zeta + \mu/3) \nabla(\nabla \cdot \mathbf{u}) \quad (2)$$

여기서 첫째는 질량 보존 방정식이고, 둘째는 운동량 보존을 만족하기 위한 NS 방정식이다. 위 식에서 ρ 와 \mathbf{u} 는 각각 유체의 밀도와 속도장이다. ρ_t 와 \mathbf{u}_t 는 각각 밀도와 속도의 시간에 대한 편미분이다. \mathbf{g} 는 중력, 부력, 바람 등과 같은 외부에서 주어지는 힘이다. p 는 압력 스칼라장이다. μ 와 ζ 는 점성 계수들이다. 위의 두 식은 ρ , μ 와 ζ 가 유체 내부에서 균일하다고 근사화하여 얻어진 방정식이다 [4].

유체 시뮬레이션은 이미 오래 전 부터 과학이나 공학 분야에서 정밀한 물리 현상을 설명하거나 예측하기 위해 사용되고 있는데, 미국 Los Alamos 국가 연구소에서는 핵 폭발후의 공기 에너지의 흐름을 예측하기 위해 유체 시뮬레이션을 수행하고 있다. 이와 같은 연구 분야를 계산 유체 역학 (Computational Fluid Dynamics: CFD)라고 부른다 [5]. 수학 분야의 경우 NS 편미분 방정식은 클레이 수학 연구소 (Clay Mathematics Institute: CMI)가 제시한 21세기에 해결하여야 할 7개의 수학 문제들 중의 하나이다. 임의의 물리적으로 합리적인 경계 조건과 초기 조건에 대하여 NS 방정식의 시간 진행 해가 발산하지 않는다는 것을 엄밀하게 증명하거나 발산하는 예를 발견하면 CMI에서 수여하는 백만 달러의 상금을 받을 수 있다 [6].

CG에서 최초로 유체 시뮬레이션을 시도한 사람은 Kajiya와 Von Herzen인데, 그 당시는 컴퓨터 파워가 열악하여 3차원 시뮬레이션에서 의미 있는 결과를 생성하지 못하였다 [7]. Foster와 Metaxas는 3차원 NS 방정식 유체 시뮬레이션을 최초로 시도하여, coarse 격자 기반으로 3차원 회오리치는 액체 운동을 표현하였다 [8]. 하지만 explicit 적분법을 사용하여 시간 스텝이 작은 경우에만 안정적 시뮬레이션이 가능하여 큰 영역에서 빠른 유속을 보여주기 위해서는 시뮬레이션 속도가 너무 느렸다.

시뮬레이션 유체를 사용하는 최초의 영화는 (그림 2)와 같은 1998년 Dreamworks작 “개미”다. 하지만 이 영화에서 사용된 유체 시뮬레이션 방법은 매우 복잡하고 오랜 시간이 걸려 이후로 거의 사용되지 않았다.

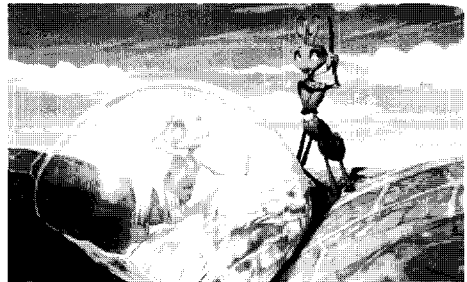


그림 2: 영화 “개미 (1998)” (Dreamworks/Courtesy Everett Collection)

격렬한 홍수, 대규모 폭발, 덮쳐오는 파도 장면과 같은 복잡한 유체 특수 효과를 표현하기 위해서는 수작업 애니메이션으로는 시간이 너무 많이 걸려 경제성이 없다. 공학에서 사용하는 물리적으로 정밀한 유체 시뮬레이션은 계산량이 방대하여 CG에서 효율성이 낮다. 따라서 CG에서는 CPU 타임, 연산 메모리량, 데이터 저장 용량과 같은 계산 자원을 줄여 물리적 정

밀성을 어느 정도 희생한다. 또한 시간 진행에 따른 방정식 풀이 과정이 안정적으로 수행되어 되도록 많은 프레임의 영상 획득이 가능해야 한다. 이 경우도 시각적으로 그럴듯한 관측의 느낌을 최대한 유지하여야 한다.

4. 유체 시뮬레이션 방법

유체 시뮬레이션은 크게 Eulerian 격자 방법과 Lagrangian 입자 방법으로 나눌 수 있다. Eulerian 격자 방법은 공간을 격자로 나누고 각각의 셀의 속도와 압력의 시간 진행을 풀어내는 방법이다. 전체 셀은 유체 내외부를 모두 포함한다. Lagrangian 입자 방법은 공간을 격자로 나누지 않고 유체가 입자들로 구성된다고 모델링하고 각각의 입자의 위치와 속도의 시간 진행을 풀어내는 방법이다. 이때 입자는 유체 내부만 기술한다.

Eulerian 격자나 Lagrangian 입자 시뮬레이션 모두 시간을 적절한 간격으로 이산화 하여 각 시각 별로 즉 때 프레임별로 유체의 형상을 계산해내야 한다.

4.1 Eulerian 격자 방법

Eulerian 격자 방법에서는 속도장과 압력장의 공간 함수로써 유체를 기술한다. 유체가 비압축 즉 밀도가 상수라고 가정하면 식 (1)과 (2)에서 주어진 유체 방정식은 다음과 같이 단순화된다.

$$\nabla \cdot \mathbf{u} = 0 \quad (3)$$

$$\mathbf{u}_t = \mathbf{g} - \mathbf{u} \cdot \nabla \mathbf{u} + \nu \nabla^2 \mathbf{u} - \frac{1}{\rho} \nabla p. \quad (4)$$

이 식에서 $\nu = \mu/\rho$ 이다. 이전 시간에서 얻어진 속도장을 이용하여 현재 시간의 각각의 항을 순차적으로 적분하여 전체 NS 방정식의 해를 구할 수 있다는 사실은 이미 증명되었으며, 이런 항 별 풀이 방법을 operator splitting이라고 부른다 [9].

처음 푸는 항은 식 (5)에서 주어진 외력 효과이다.

$$\mathbf{u}_t = \mathbf{g} \quad (5)$$

$$\mathbf{u}_0 \xrightarrow{\mathbf{g}} \mathbf{u}_1 \quad (6)$$

$$\mathbf{u}_1 = \mathbf{u}_0 + \Delta t \mathbf{g}. \quad (7)$$

외부힘에 의한 현재 시간의 속도장 값은 식 (7)에서 주어진 1차 explicit 적분만으로도 대부분 큰 에러 없이 안정적으로 구할 수 있다. 하지만 외부힘이 위치 벡터나 속도 벡터에 의존하면 stiff 하다면 implicit 알고리즘을 이용해야 안정성을 확보할 수도 있다 [10].

다음 푸는 항은 식 (8)의 대류항(advection)이다.

$$\mathbf{u}_t = -(\mathbf{u} \cdot \nabla) \mathbf{u} \quad (8)$$

$$\mathbf{u}_1 \xrightarrow{\text{advection}} \mathbf{u}_2. \quad (9)$$

대류항은 유체 지배 방정식을 비선형으로 만든다. 이런 비선형성에 의한 시뮬레이션의 불안정성이 1999년 이전 (그림 2)의 “개미”같은 영화 제작의 유체 시뮬레이션이 지나치게 많은 시간을 요구하는 주된 이유들 중의 하나였다.

식 (8)은 다음 식 (10)처럼 스칼라 s 대신에 벡터 \mathbf{u} 의 각각의 성분을 대입한 것으로 일반화할 수 있다.

$$s_t = -(\mathbf{u} \cdot \nabla) s. \quad (10)$$

\mathbf{u} 가 주어졌을 때에 식 (10)의 s는 특징 곡선 (characteristic curve)을 이용한 semi-Lagrangian 기법을 이용하여 해를 구할 수 있다는 것이 수치 해석 분야에서 알려져 있었다 [11]. 임의의 하나의 특징 곡선 상에서는 밀도, 온도 등의 물리량 s가 변하지 않으므로, 현재 격자의 물리량 $s(\mathbf{x})$ 는 그 점에 도착하는 특징 곡선 상의 이전 점에서의 물리량 $s(\mathbf{x}')$ 과 같다. 따라서 semi-Lagrangian 기법에서는 우선 Forward Euler 적분으로 특징 곡선의 이전점 \mathbf{x}' 을 구한 후, \mathbf{x}' 을 둘러싼 격자점들에서 물리량 값을 보간법에 의해 이전점에서의 물리량 $s(\mathbf{x}')$ 을 구한 후 이 값을 현재 격자점의 물리량으로 정한다 [12].

1999년 Stam은 식 (10)에서 s에 대하여 밀도, 온도 뿐 만이 아니라 속도장 \mathbf{u} 자체를 특징 곡선 방법으로 다루어 식 (8)을 풀었다. 이 접근법은 수학적으로는 엄밀한 풀이가 아니나 CG에서 사용하기에는 시각적으로 그럴듯하며 어떤 크기의 시간 간격에도 매우 안정적인 시뮬레이션이 가능하였다 [13]. 이전 점 \mathbf{x}' 을 구하는데 있어서 Forward Euler 방법이 대부분의 경우 충분하나, 정확도를 높이기 위해 2차나 4차 Runge-Kutta 방법이 사용되기도 한다 [14]. 또한 이전점에서의 물리량을 주위의 격자점들의 값으로 보간법 하는데 있어서도 [15]에서는 기상학에서의 기법을 사용하여 선형 보간법 대신에 고차의 보간법을 사용하여 정확도를 높였는데 이 방법은 밀도나 온도를 시간 진행하는데 특히 효율적이다.

다음으로 적분하여야 할 부분은 식 (11)에서 주어진 점성항(viscosity)으로 이 항은 기체의 경우는 보통 무시하고 점성 액체의 경우만 고려한다.

$$\mathbf{u}_t = \nu \nabla^2 \mathbf{u} \quad (11)$$

$$\mathbf{u}_2 \xrightarrow{\text{viscosity}} \mathbf{u}_3. \quad (12)$$

점성항은 확산항(diffusion)이라고도 불린다. 점성항은 implicit 방법으로 선형 방정식을 풀어 해를 구할 수 있다.

대류항과 점성항은 식 (13)과 같이 순서가 바뀌어 계산되기도 한다.

$$\mathbf{u}_1 \xrightarrow{\text{viscosity}} \mathbf{u}_3 \xrightarrow{\text{advection}} \mathbf{u}_2. \quad (13)$$

마지막으로 풀어야 할 부분은 식 (14)에서 주어진 압력장에 의한 유속 변화이다.

$$\mathbf{u}_t = -\frac{1}{\rho} \nabla p \quad (14)$$

$$\mathbf{u}_3 \xrightarrow{\text{pressure}} \mathbf{u}_4. \quad (15)$$

압력장에 의한 유속 변화는 속도장을 비압축 해들의 공간으로 투영하여 푼다 [16]. 구체적으로 설명하면 우선 식 (14)를 Forward Euler 적분하여 다음 식 (16)을 얻는다.

$$\mathbf{u}_4 = \mathbf{u}_3 - \frac{1}{\rho} \Delta t \nabla p. \quad (16)$$

식 (16)의 \mathbf{u}_4 를 식 (3)으로 표현된 질량 보존 방정식에 대입하면 다음 Poisson 방정식을 얻는다.

$$\frac{1}{\rho} \Delta t \nabla^2 p = \nabla \cdot \mathbf{u}_3. \quad (17)$$

식 (17)을 풀어 얻어진 압력장 p를 식 (16)에 대입하여 최종 속도장 \mathbf{u}_4 를 구한다. 이렇게 얻어진 \mathbf{u}_4 는 비압축($\nabla \cdot \mathbf{u}_4 = 0$)으로 밀도가 균질한 유체에 대하여 질량 보존을 성립하게 한다. 큰 폭발과 같이 압축이 가능한 유체에 대한 질량 보존을 하려면 밀도가 시간에 따라 바뀌어야 한다 [17].

식 (17)의 Poisson 방정식은 대칭이고, 적어도 한 개의 Dirichlet 경계 조건이 사용되면 양정치(positive-definite)이다. 따라서 [15]에서처럼 conjugate gradient 방법으로 풀 수 있다. 이때 물과 같은 액체의 경우 공기의 압력을 0으로 보고 물 표면(free surface)의 압력을 설정하는 것이 보통이고 표면 장력 효과를 고려한다면 여기에 표면 곡률량이 추가된다 [18, 19, 20]. 점성까지 고려된 일반적인 경우는 점성 스트레스 텐서까지 추가된 물 표면의 Dirichlet 경계조건을 생각할 수 있다 [21, 4]. [22, 19]에서는 적층적으로 조밀화된 격자인 octree를 사용하여 유체 시뮬레이션을 향상시킨다.

식 (11)과 (17)에서 각각 주어진 점성 부분과 부영 부분은 대형 sparse linear system과 관계가 있다. Sparse linear system을 푸는 방법으로는 [23]의 multi-grid 알고리즘, [15]에서 사용된 conjugate gradient 방법, [24]에서 사용된 Gauss-Seidel relaxation 방법 등이 있다. [13]에서는 [25]에서 설명된 FISHPAK 라이브러리의 해법을 사용하였다.

4.1.1 1999년 Stam의 업적

식 (8)의 대류항에 대한 semi-Lagrangian 적분법과 식 (11)의 점성항과 식 (17)에 대한 implicit 적분법이 핵심인 [13]의 방법은 무조건 안정적으로 빠른 유체 시뮬레이션을 가능하게 하였다. Stam이 1999년에 제안한 이 방법은 CG 유체 시뮬레이션 분야에서 계속 발전되면서 많은 유체 시뮬레이터에 채택되었고 2008년 Sci-Tech Oscar 상을 수상하였다. 하지만 Stam의 방법은 1차 적분 방식이라 너무 많은 수치 감쇄(numerical dissipation)를 발생시키는 문제가 있다.

Stam의 방법은 이후로 많은 연구자들에 의해 더욱 개선되었다. 예를 들어 [26]에서는 Constrained Interpolation Profile (CIP) 기반 대류 방법을 이용하여 수치 감쇄를 줄인다. 이 방법에서는 잠재적으로 감쇄인 셀을 물방울이나 거품으로 변경하여 더 낮은 감쇄를 줄인다.



그림 3: 영화 “반지의 제왕 1 (2001)” (New Line Cinema)

Stam의 방법은 다양한 애니메이션 SW에 채택되어 많은 영화의 유체 애니메이션 효과를 생성한다. Stam 방법이 바로 적용된 영화는 2001년 “반지의 제왕 1-반지 원정대”와 2003년 “캐리비안의 해적 1”이다. “반지의 제왕 1”에 사용된 유체 SW는 Digital Domain 사의 FSIM 유체 시뮬레이션 시스템이다. 이 영화에서는 (그림 3)과 같이 계곡에서 장대하게 흘러내리는 물 애니메이션에 FSIM이 사용되었다.

4.1.2 연기와 같은 기체

수치 감쇄가 물리적 점성이나 분자적 확산보다 훨씬 큰 coarse 격자 기체에 대해서는 NS 방정식의 점성항을 무시할 수 있다. 이 경우 비점성 비압축 일정 밀도 유체가 되어 연기와 같은 기체를 모델링할 수 있으며, 비압축 NS 방정식은 비압축 Euler 방정식으로 단순화된다. 연기와 공기가 같이 있는 경우 연기

의 밀도와 온도는 연기의 속도장에 영향을 준다. 이와 같은 영향을 [15]에서는 다음 부력을 식 (5)에 주어진 외력 \mathbf{g} 에 더하여 모델링 하였다.

$$\mathbf{g}_{buoy} = -\{\alpha\rho + \beta(T - T_{amb})\}\mathbf{z}. \quad (18)$$

이 식에서 ρ 와 T 는 기의 밀도와 온도이고 T_{amb} 는 공기 상온이며, α, β 는 양의 상수이고, $\mathbf{z} = (0, 0, 1)$ 은 수직 위 방향이다. 연기와 공기의 혼합물에서는 많은 회전 및 난류 구조가 생긴다. 하지만 4.1.1절에서 설명한 것처럼 지나친 수치 감쇄가 발생되어 회전 및 난류 등의 소규모 구조체를 새로운 효과로 생성할 수 없다. 이와 같은 문제를 해결하기 위한 하나의 방법은 흐름 속에 소규모 랜덤 섭동을 발생시키는 것이다. 예를 들어 Kolmogorov spectrum을 이용하여 다이버전스가 없는 속도장을 생성하고 이것을 흐름장에 더하여 원하는 효과를 생성한다 [27]. 하지만 이 경우 소규모 구조체가 물리적으로 정확한 곳에 생성되지 않고 지나치게 역동적으로 보이는 문제가 있다.

Fedkiw와 Stam의 2001년 논문 [15]에서는 이와 같은 문제점들을 해결하는 모델을 제안한다. 이 모델은 안정적이고 빠르며, 지나친 수치 감쇄가 생기지 않으며, 비교적 정확한 위치에 소규모 구조체를 생성한다. 이 모델은 CFD에서 이미 알려진 vorticity confinement 기술을 사용한다 [28]. CFD에서는 헬리콥터 주위의 복잡한 난류를 계산하기 위해 이 방법이 고안되었다. 이 방법은 수치 감쇄에 의해 사라진 에너지를 보충하기 위해 가상의 힘을 추가한다. 추가된 힘이 어떤 임계치 이내이 기간 하면 [15]의 모델도 4.1절 Stam의 방법만큼 빠르고 안정적이다. 이 모델에서 사용되는 추가된 가상의 힘은 다음 식과 같다.

$$\mathbf{g}_{conf} = ch(\mathbf{N} \times \boldsymbol{\omega}) \quad (19)$$

$$\mathbf{N} = \frac{\nabla |\boldsymbol{\omega}|}{|\nabla |\boldsymbol{\omega}||}, \boldsymbol{\omega} = \nabla \times \mathbf{u}. \quad (20)$$

c 은 가상힘을 제어하는 양의 상수이고, h 는 격자 간격이다. h 가 0으로 접근하면 가상힘도 사라진다. [15]에서는 또한 격자 영역 내부에 고체 경계를 다룰 수 있어서 물체 주위를 타고 도는 연기를 시뮬레이션 할 수 있다.

[29]에서는 연기 밀도, 유체 속도나 이미지 색상 같은 물리량들을 대류시키는데에 Back and Forth Error Compensation and Correction (BFEC) 방법을 사용하여 여러 규모의 난류 구조체를 유지한다. 이 방법은 semi-Lagrangian 기술의 안정성 장점을 그대로 유지하면서 수치 감쇄 단점을 감소시킨다.

[30]에서는 detonation shock dynamics를 NS와 결합하여 생동감 있는 불을 시뮬레이션 한다. 이 방법에서는 다음 절에서 설명할 등위 집합(level set) 방법을 사용하였고, 화염의 주름과 셀 패턴을 사실적으로 표현한다.

4.1.3 액체 표면 형상화

3차원 유체의 표면을 찾아내는 작업은 단순하지 않다. 유체 표면은 implicit 표면을 이용하여 표현된다. Implicit 표면은 비관성 입자 방법, 등위 집합 방법이나 이 둘을 결합한 방법으로 구할 수 있다.

비관성 입자란 질량이 없는 입자를 의미하며 비관성 입자 방법에서는 매 프레임 입자의 속도는 그 입자를 둘러싼 격자 속도장의 tri-linear 보간법에 의해 구하며, 이 속도를 적분하여 입자의 위치를 구한다. 이렇게 구한 모든 입자의 위치로부터 모든 입자들을 삼각형들로 결합한 후에 유체 전체의 연결

구조와 표면을 찾아낸다. 이 방법으로 유체 표면을 알아내는 일은 매우 어려운 작업이며, 또한 계산한 표면이 일반적으로 smooth 하지 않다는 단점을 가지고 있다.

유체 표면을 구하는 다른 방법으로 등위 집합 방법이 있다. 등위 집합은 각각의 격자 점으로부터 유체의 표면까지의 거리를 나타내는 추가 스칼라장 $\phi(x)$ 이다. 유체의 표면은 implicit 함수 $\phi(x)$ 의 zero들로부터 구할 수 있다. 최초 시각의 유체 표면에 해당하는 $\phi(x)$ 를 계산하고 이후 매 시각 $\phi(x)$ 를 식 (10)에서 주어진 대류 방정식으로 부터 구하는데, 이러한 방법을 CFD에서 등위 집합 방법이라고 부른다 [31]. 그런데 식 (10)의 방정식은 유체의 mass evolution을 나타내기 때문에 시간 진행 방법이 세밀하지 않다. 따라서 몇몇 연구자들이 대류 항 적분에 고차 upwind differencing 과정을 사용하였다 [32]. 그런데 이 방법은 CG에 사용되는 coarse 격자에서는 심각한 부피 손실을 발생시킨다. 예를 들어 물의 일부분이 splash 되어 분리되고 다시 결합하는 경우 떨어진 부분이 너무 작아 등위 집합 방법의 분해능으로는 사라진 것처럼 보인다.

입자 방법은 Lagrangian 접근법으로 흐름 복잡도가 높을 때 디테일한 표면 표현이 가능하지만 입자 수가 적을 때에는 시각적 artifact 단점을 갖는다. 등위 집합 방법은 Eulerian 접근법으로 항상 smooth 하지만 흐름 복잡도가 높을 때에 부피 손실이 발생한다. [33]에서는 이 두 방법의 장점을 취하는 hybrid 방법을 제시하였다. 이 방법에서는 우선 입자 위치와 iso-contour를 모두 시간 진행 시키는데 매 순간 표면에서 멀리 떨어진 입자들은 제거하고 표면을 구성하는 입자들은 추가하여 나간다. 다음으로 매 순간 표면의 샘플 점들마다 곡률을 구하여, 곡률이 작은 곳에서는 iso-contour를 사용한다. 곡률이 큰 곳에서는 입자들을 이용하여 $\phi(x)$ 의 값을 변경한다. [33]에서는 이 방법으로 $\phi(x)$ 에 대한 smooth iso-contour를 얻어 비자연스러운 겹침이나 각점을 감소시킬 수 있었다.

등위 집합은 드문드문한 격자 점으로부터 구해지므로 표면의 디테일을 표현하기 부족한 단점을 갖는다. [34]에서는 입자 등위 집합 방법을 이용하여 이 단점을 감소한다. 등위 집합으로 구한 표면 렌더링을 위해서는 $\phi(x)$ 방정식을 풀어 장선 추적하거나, marching cube 알고리즘을 사용하여 폴리곤 메쉬를 얻으면 된다.

[35]에서는 등위 집합을 explicit 적분을 이용하는 대신 semi-Lagrangian 방법과 octree 공간분할을 이용하여 업데이트 하여 매 시각 폴리곤 메쉬를 추가로 생성해 나가는 방법을 제안한다. 이 방법의 큰 장점은 색깔 또는 텍스처 좌표와 같은 유체 표면 성질의 추적이 가능하다는 것이고 regular 격자 대신 정교한 octree 분할을 사용함으로써 [34]에서 설명한 입자 등위 집합 방법과 같은 정도의 부피보존을 보인다.

[33, 34]에서 제시된 유체 표면 형상화 방법은 모두 스탠포드 대학교의 Fedkiw와 그의 동료나 학생들이 이룬 연구 결과이다. 이와 같은 연구 결과를 활용하여 Fedkiw는 ILM사의 유체 시뮬레이터인 Zeno 개발에 관여하였고, 이 업적으로 2008년 Sci-Tech Oscar 상을 수상하였다.

4.2 Lagrangian 입자 방법

Lagrangian 입자 방법에서는 유체를 질량, 위치와 속도를 갖는 입자로 샘플링한다. 분수나 폭발과 같은 입자들이 서로 떨어지는 간단한 경우에는 상호작용을 무시하는 자유 입자 시스템으로 유체를 표현할 수 있다. 하지만 이런 특별한 경우를 제외하고는 대부분 입자 사이의 상호작용을 고려하여야 한다. CG 유체 표현을 위한 상호작용하는 입자 시스템 시뮬레이션으로 Smoothed Particle Hydrodynamics (SPH)가 최근 들어 활발

히 연구되어 오고 있다.

4.2.1 Smoothed Particle Hydrodynamics

SPH는 원래 천체 물리 분야에서 고안된 방법인데 Monaghan에 의해서 경계와 상호작용하는 비압축 유체 시뮬레이션으로 확장되었다 [36, 37]. SPH에서는 유체가 불연속적인 입자 유체 요소들로 구성되어 있다고 모델링한다. 각각의 입자들은 점입자가 아니고 h 로 표시되는 smoothing length 라고 불리는 support 반경을 갖고 있다. 임의의 공간점에서의 물리량은 입자들 위치한 점들의 물리량과 반경 h 와 커널 함수 $W(\mathbf{x}-\mathbf{x}_j, h)$ 로 부터 얻어 진다. 각각의 입자는 뉴턴 동역학에 따라 시간 적분되는데 여기서 사용되는 입자에 가해지는 힘은 상태 방정식으로부터 유도된다. SPH에서는 연속체 PDE를 푸는데 등장하는 대류항이 필요 없어 계산이 간단하므로 게임과 같은 실시간 응용에 관심을 받아 왔지만 최근에는 실시간이 아닌 영화 특수 효과에도 많이 사용되고 있다.

SPH에서는 \mathbf{x} 에서의 함수값 $A(\mathbf{x})$ 는 입자의 위치에서의 샘플값 A_j 에 의해 다음과 같이 얻어진다.

$$A(\mathbf{x}) = \sum_j A_j \frac{m_j}{\rho_j} W(\mathbf{x} - \mathbf{x}_j, h). \quad (21)$$

이 식에서 m_j 와 ρ_j 는 각각 \mathbf{x}_j 에 위치한 입자 j 의 질량과 밀도이다. 커널 함수 $W(\mathbf{x}, h)$ 는 $\int_V W(\mathbf{x}, h) d^3\mathbf{x} = 1$ 을 만족한다. 커널로는 가우스 함수나 compact support를 갖는 다항식이 사용되는데 후자로는 cubic spline이 있다. 밀도 ρ 를 식 (21)의 A 에 대입하면 다음과 같은 식을 얻는다.

$$\rho(\mathbf{x}) = \sum_j m_j W(\mathbf{x} - \mathbf{x}_j, h). \quad (22)$$

함수 W 가 미분 가능하다면 식 (21)의 $A(\mathbf{x})$ 에 대한 미분은 부분적분을 이용하여 다음과 같이 얻는다.

$$\nabla A(\mathbf{x}) = \sum_j A_j \frac{m_j}{\rho_j} \nabla W(\mathbf{x} - \mathbf{x}_j, h). \quad (23)$$

식 (1)과 (2)로 부터 SPH에서의 입자의 운동 방정식을 유도하자. 우선 입자의 숫자는 일정하고 각각의 입자는 일정한 질량을 갖고 있으므로 질량은 보존되어 식 (1)은 자동적으로 만족된다. SPH에서는 식 (2)의 좌변의 두 항이 합쳐져서 대류항이 필요 없고, 또한 비압축 유체인 경우 우변에 $\nabla \cdot \mathbf{u} = 0$ 이 적용되어, 다음과 같이 표현된다.

$$\begin{aligned} \rho_i \dot{\mathbf{u}}_i &= \mathbf{f}_i^{\text{external}} + \mathbf{f}_i^{\text{viscosity}} + \mathbf{f}_i^{\text{pressure}} \\ &= \rho_i \mathbf{g}(\mathbf{x}_i) + \mu \nabla^2 \mathbf{u}_i - \nabla p_i. \end{aligned} \quad (24)$$

제일 먼저 중력이나 외부 충돌에 의해 주어지는 $\mathbf{f}_i^{\text{external}}$ 은 입자에 직접 적용된다.

다음으로 식 (24)의 점성항에 대하여 식 (23)을 사용하면 식 (25)와 같은 힘이 생긴다. 이 힘은 2개의 입자 i, j 의 상호작용이 비대칭이다.

$$\mathbf{f}_i^{\text{viscosity}} = \mu \nabla^2 \mathbf{u}_i = \mu \sum_j \mathbf{u}_j \frac{m_j}{\rho_j} \nabla^2 W(\mathbf{x}_i - \mathbf{x}_j, h). \quad (25)$$

점성력은 입자들의 속도 차이에만 의존하기 때문에 비대칭 점성력을 다음과 같이 대칭화할 수 있다.

$$\mathbf{f}_i^{\text{viscosity}} = \mu \sum_j m_j \frac{\mathbf{u}_j - \mathbf{u}_i}{\rho_j} \nabla^2 W(\mathbf{x}_i - \mathbf{x}_j, h). \quad (26)$$

마지막으로 식 (24)의 압력항에 대하여 식 (23)을 사용하면 다음과 같은 비대칭 힘이 생긴다.

$$\mathbf{f}_i^{pressure} = -\nabla p_i = -\sum_j p_j \frac{m_j}{\rho_j} \nabla W(\mathbf{x}_i - \mathbf{x}_j, h). \quad (27)$$

커널의 그래디언트는 입자의 중심에서 사라지기 때문에 입자 i 에 주어지는 압력은 자신 이외의 다른 입자에 의해서만 발생한다. 비대칭인 압력을 대칭화하는 여러 방법들이 제안되었는데, 안정적이고 빨리 계산이 가능하며 간단한 방법으로 다음 식이 있다.

$$\mathbf{f}_i^{pressure} = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(\mathbf{x}_i - \mathbf{x}_j, h). \quad (28)$$

식 (26)과 (28)이 모두 요구하는 ρ_j 는 식 (22)로부터 계산된다. 식 (28)이 요구하는 p 는 이상 기체 상태 방정식으로부터 다음과 같이 얻는다.

$$p = k(\rho - \rho_0). \quad (29)$$

이 식에서 k 는 유체 상수이고, ρ_0 는 기준 밀도이다. $\mathbf{f}_i^{pressure} = -\nabla p$ 이므로 수학적으로는 오셋 ρ_0 는 압력항에 영향을 미치지 않는다. 하지만 식 (28)에서 주어진 대칭화된 $\mathbf{f}_i^{pressure}$ 에서는 ρ_0 가 영향을 미치며 다행히도 시뮬레이션을 수치적으로 더욱 안정화시킨다.

Eulerian 격자 방법에서는 식 (17)에서 주어진 Poisson 방정식을 풀어 압력을 계산하여 비압축성을 엄밀히 유지한다. 하지만 SPH에서는 식 (28)과 (29)로부터 주어진 압력항으로부터는 비압축성을 유지할 수 없어 튀어 보이는 것 같은 부자연스러운 시각 표현이 생긴다. [38]에서는 Moving Particle Semi-implicit (MPS) 방법을 이용하여 비압축 유체를 표현하는 기술을 제시한다. 또한 [38]은 다상(multi-phase) 유체를 기술하며 표면 생성을 위해 물리적 입자와 등위 집합을 결합하여 인상적인 시각 효과를 보인다. [39]에서는 Eulerian 격자 방법에서와 같은 Poisson 방정식을 풀어 압력을 결정하여 SPH의 시뮬레이션 시간 간격을 늘려 효율성을 높이는 방법을 설명한다.

SPH는 전통적 격자 기술보다 다음과 같은 장점을 갖고 있다. 첫째 입자들이 각각 보존되는 질량을 가지고 있으므로 추가 계산 없이도 질량 보존이 자동 성립한다. 둘째 SPH에서는 Poisson 방정식과 같은 선형 시스템을 푸는 수 없이 주변 입자들로부터 압력을 바로 계산한다. 따라서 속도 향상의 장점이 있지만 비압축성이 유지되지 않는 단점이 있다. 셋째 2가지 종류의 유체가 상호작용하는 경우에 격자 방법에서는 경계면을 추적하여야 하지만 SPH에서는 경계면을 바로 생성한다.

SPH 시뮬레이션에서 격자 방법 수준의 해상도를 제공하려면 많은 수의 입자로 샘플링하여야 한다. 이 경우 대부분의 입자들이 렌더링되지 않는 부피를 채우는데 사용된다. 유체 표면 형상화의 정확도를 높이기 위해서는 4.1.3절의 입자 등위 집합과 같은 격자 기반 기술을 같이 사용하여야 한다 [34].

시뮬레이션의 품질을 높이는 다른 방법으로는 적응적 시뮬레이션이 있다. 즉 좀 더 많은 샘플이 필요한 곳에서는 입자 하나가 여러 입자로 나누어져 세밀한 표현을 가능하게 하고, 밀도가 거의 균일한 곳에서는 여러 입자가 하나로 합쳐져 연산 속도를 높인다. 질량 m_i , 부피 m_i/ρ 인 하나의 입자가 N 개의 입자로 나누어지는 경우에는 결과로 나타난 입자의 질량은 m_i/N , 부피는 $m_i/\rho/N$ 가 되어 전체 부피 및 밀도가 유지된다. 여러 입자가 하나의 입자로 합쳐지는 경우에는 새 입자의 위치는 이전 입자들의 질량 중심이며, 새 입자의 질량 및 속도는 원래의 질량 및 선형 운동량이 보존되도록 결정 된다 [40].

[41]는 SPH를 CG에 최초로 도입한 연구로 불과 가스 같은 현상을 기술하였는데, “warped blobs”으로 구성된 밀도를 위해 대류-확산 방정식을 푼다. [42]에 의하면 변형체가 SPH로 시뮬레이션 될 때 표면은 질량 밀도의 iso-surface로 정의할 수 있다. [42]에서는 고 변형체의 애니메이션을 위한 상태 방정식을 풀어, 강성으로 부터 짐성까지 다양한 영역의 물질의 애니메이션을 얻는다. [43]은 implicit 표면을 속도장에 의하여 시간 변화하는 동적 표면으로 향상시켜서, 표면 장력이나 부피 보존과 같은 특징을 표면 모델에 부가할 수 있다. [44]은 짐성을 온도장과 열전도와 결합하여 용암 흐름을 애니메이션한다. [45]은 머리카락을 유체 같은 연속체로 시뮬레이션 하는데 변형된 SPH를 사용하였다. [46]은 새로운 커널을 가진 SPH 기반 방법으로 자유 표면을 가진 유체를 5000개의 입자까지 상호작용이 가능하도록 시뮬레이션 한다. 또한 이 논문에서는 표면 장력 효과를 모델링한다. [47]은 Lagrangian 유체와 메쉬 기반 변형체의 상호작용을 모델링한다. [48]은 입자 기반으로 유체와 유체의 상호작용을 모델링하여 끓는 물, 물 안에 갇힌 공기 및 라바 램프 (lava lamp)와 같은 현상을 시뮬레이션 한다.



그림 4: 맥주에서 거품의 생성 (ETRI)

[49]에서는 (그림 4)에서 보여 지듯이 맥주나 샴페인 같은 액체가 접에 부어질 때 생성되는 거품의 움직임과 액체의 흐름을 생동감 있게 보여준다. 이 논문에서는 녹은 기계에서 거품의 생성, 거품과 액체의 상호작용과 거품들 사이의 상호작용에 대한 적절한 물리 모델에 기반하여 SPH 시뮬레이션을 수행한다.

Next Limit사 유체 시뮬레이터 RealFlow가 SPH 기술을 채택한다. RealFlow는 (그림 5)에서 보여 지는 2006년 영화 “포세이돈”에 사용되었고 2008년 Sci-Tech Oscar 상을 수상하였다.

5. 2008년 아카데미상 유체 SW

올해의 경우 흥미롭게도 Sci-Tech Oscar 10개 부문 중 6개 부문이 유체 시뮬레이션 소프트웨어를 연구/개발하는 팀들에게 주어졌는데, Technical Achievement Awards 7개 중 4개가 각각 RealFlow SW 개발, FELT 유체 동역학 시스템 개발, Maya 유체 효과 시스템의 설계/구현과 FlowLine 유체 특수 효과 시스템 개발의 공로로, Scientific and Engineering Awards 2개 모두 각각 Zenos와 FSIM 유체 시뮬레이션 SW 연구 개발의 공로로 수여되었다. 각 수상 SW의 제작사, 수상자와 대표 영화는 표 1과 같다 [50, 51]. 다음 절에서는 각 SW의 기술적 특징 및 장단점을 분석하고, 이 장 마지막에 표 2에서 정리한다.

5.1 RealFlow (Next Limit)

Next Limit은 스페인 마드리드에 위치한 수상자들이 창립한 회사이다 [52]. Next Limit에서 개발한 RealFlow는 최초로 널리 상용화된 사용하기 쉬운 영화 시각 특수 효과용 액체 시뮬레이션 시스템이다. RealFlow는 디지털 아티스트가 복잡한 훈련 없이도 사용하기 쉬운 직관적이고 단순한 User Interface (UI)를 제공하여 영화 산업계에서 가장 널리 사용되는 프로그램들 중의 하나가 되었다.

SW	제작사	수상자	대표 영화
RealFlow	Next Limit	Victor Gonzalez, Ignacio Vargas, Angel Tena	포세이돈 (2006)
FELT	Rhythm & Hues	Jonathan Cohen, Jerry Tessendorf, Jeroen Molemaker, Michael Kowalski	해피 피트 (2006)
Maya Fluids	Autodesk	Duncan Brinsmead, Jos Stam, Julia Pakalns, Martin Werner	고스트 라이더 (2007)
Flowline	Scanline	Stephan Trojansky, Thomas Ganshorn, Oliver Pilarski	300 (2007)
FSIM	Digital Domain	Doug Roble, Nafees Bin Zafar, Ryo Sakaguchi	투모로우 (2004)
Zeno	ILM	Nick Rasmussen, Ron Fedkiw, Frank Losasso, Petterson	캐리비언의 해적 3 (2007)

표 1: 2008년 Sci-Tech Oscar 수상한 유체 SW



그림 5: 영화 “포세이돈 (2006)” (Warner Bros./MPC)

RealFlow는 4.2.1절에서 설명한 입자 기반으로 유체의 운동을 쉽게 조정할 수 있다. RealFlow는 중력과 같은 외력을 국부나 전역에 추가하거나 제거하는 기능을 제공한다. 이와 같은 내부 유체 행동 제어와 외력 추가 기능을 통해 아티스트가 다양한 특수 효과를 생성할 수 있다.

RealFlow는 multi-thread 시뮬레이션을 지원하며 메쉬 생성은 네트워크를 이용한 분산 작업이 가능하다. 기존에 독립적으로 존재한 RealWave를 통합하여 파동 방정식 기반의 open water 시뮬레이션과 입자 시뮬레이션의 유기적 결합이 가능하다. 유체 뿐만 아니라 강체와 변형체에 대한 고성능 시뮬레이션이 가능하며 이들 모두 유체와 쌍방향 상호작용한다. 현재 물과 같은 액체 시뮬레이션만 가능하다.

RealFlow가 사용된 대표적인 영화로는 (그림 5)에서 보여 지는 2006년 “포세이돈”이 있다. RealFlow는 이 영화에서 복도를 통하여 거대한 파도의 물이 격렬하게 물러오는 장면에서 사용되었다. 이 장면에서 물러오는 파도 물 표현은 실감적이지만, 물에 휩쓸리는 부스러기와 의자의 움직임이 부자연스러운 단점이 있다. 이 문제점을 개선하려면 유체와 고체 사이의 상호작용 표현 기술의 향상이 요구된다.

5.2 FELT (Rhythm & Hues)

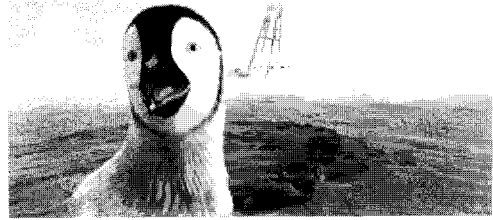


그림 6: 영화 “해피 피트 (2006)” (Warner Bros./Courtesy Everett Collection)

Rhythm & Hues는 Los Angeles에 기반을 둔 유체 동역학 플랫폼 개발 스튜디오다 [53]. 여기서 개발한 유체 시뮬레이션 프로그램 FELT는 다른 스케일의 유체들을 각각 따로 시뮬레이션한 다음에 이들을 하나의 영상으로 결합하는 작업이 가능하게 한다. 예를 들어 (그림 6)에서 보이는 2006년 영화 “해피 피트”에서 주인공 펭귄이 높은 얼음 절벽으로부터 다이빙 한 후에 장거리 수영하는 장면에서는 파도, 거품, 물방울을 따로 따로 다른 기술로 시뮬레이션 한 후에 하나의 영상으로 결합하였다.

FELT는 정확도와 속도 증대를 위한 새로운 시뮬레이션 기술과 부피 데이터를 다루기 위한 고유한 스크립팅 언어를 사용하여 아티스트가 액체와 기체의 실감나는 애니메이션을 생성하게 한다. FELT는 균일 격자를 넘어서는 구조를 가지는데, 그 구조는 튜브의 휘어짐과 같은 것들을 시뮬레이션하기 위한 3차원 구조이다. 이와 같은 향상된 구조를 사용하여 혼돈적으로 복잡한 물 흐름 시뮬레이션이 가능하다.

FELT 개발자들은 시각 효과란 실재를 시뮬레이션 하는 것이 아니라는 점을 강조한다. 실재는 따분하며 따라서 시뮬레이션은 현실 이상의 극적인 효과를 생성할 수 있어야 한다.

5.3 Maya Fluid Effects System (Autodesk)

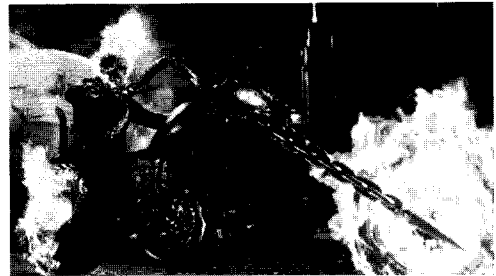


그림 7: 영화 “고스트 라이더 (2007)” (Columbia Pictures/Courtesy Everett Collection)

Maya는 폭넓게 사용되는 컴퓨터 그래픽스 툴로서 거대한 폭풍우, 승털 같은 구름, 폭발 화염, 천체 등을 표현할 수 있다 [54]. Maya Fluids는 Maya와 호환되는 기체 및 액체 시뮬레이터로 4.1절에서 설명한 1999년 논문의 저자인 Stam이 참여하여 개발하였다. 따라서 이 시스템은 무조건적으로 안정적인 semi-Lagrangian 풀이법을 제공한다.

유체란 액체뿐만 아니라 연기, 불 및 폭발을 포함하는데 Maya Fluids는 특히 연기와 불 특수효과를 잘 표현한다. 예를 들어 (그림 7)의 2007년 영화 “고스트 라이더”에서 주인공 몸에 계속 존재하는 화염을 표현하기 위해 이 시스템이 사용되었다. Maya Fluids는 굽이치는 연기를 표현하는데 강점을 가지며 애니메이터가 극적인 효과를 위해 유체의 일부를 과장하는데 사용할 수 있어 감독의 제어에 의한 효과를 부여하기 용이한 특징을 갖는다.

Maya Fluids의 최대 장점은 많은 스튜디오의 범용 3D 작업 환경인 Maya의 주요기능과 유기적으로 연동이 가능하다는 것이다. 실제로 Maya Fluids는 Maya의 강력한 입자, 강체, 변형체 시물레이션과 결합하여 사용자가 원하는 다양한 효과의 연출이 가능하다.

5.4 Flowline (Scanline)



그림 8: 영화 “300 (2007)” (Warner Bros./Courtesy Everett Collection)

Scanline은 진일보한 hydro-fluid 효과로부터 3차원 feature 및 캐릭터 애니메이션을 다루는 독일에 위치한 선도적 VFX 회사이다 [55]. Scanline에서 개발한 Flowline은 디테일하고 실감나는 유체 효과를 지원하기 위해 고도의 병렬 연산을 사용하는 유연한 유체 시물레이터이다. Flowline은 다른 격자 기반 시물레이션에서 표현하기 힘든 물의 격렬한 움직임을 다스려 과장되게 시물레이션하고 있으나 시각적으로 결점을 찾기는 매우 힘들다.

(그림 8)의 2007년 영화 300의 폭풍우 장면은 Flowline 기반으로 물보라나 먼지와 같은 미립자 애니메이션을 유체 시물레이션과 자연스럽게 합성하였다.

5.5 FSIM (Digital Domain)



그림 9: 영화 “투모로우 (2004)” (20th Century Fox/Courtesy Everett Collection)

Digital Domain은 캘리포니아 Venice시에 위치한 CG 특수효과 스튜디오로 1999년 Stam의 알고리즘을 처음 적용한 유체

시물레이터 FSIM을 개발하였다 [56]. 이 FSIM은 2001년 작 “반지의 제왕 1”에서 (그림 3)에서 보이는 계곡을 통과하게 흘러내리는 물 애니메이션 제작에 사용되었다.

이후로 2004년 작 “투모로우”에서 (그림 9)와 같이 뉴욕을 범람하는 홍수를 표현하기 위해 Digital Domain은 새로운 버전의 FSIM을 개발하였다. 이 새로운 FSIM은 기존의 방법을 개선한 알고리즘과 adaptive refinement를 통해 대규모 물 효과를 생성한다. 이 새로운 FSIM은 거품과 안개를 결합할 수 있고, 대규모 시물레이션을 수행할 수 있으며, 아티스트가 유체 애니메이션에 대한 백그라운드 없이도 쉽게 작업할 수 있는 특징을 갖는다.

FSIM 시스템은 대규모의 물 효과를 생성하기 위한 복수의 소프트웨어들의 협업으로 구성된다. 대규모 물 특수 효과를 생성하기 위해서는 몇 단계의 과정이 필요하다. 첫 번째로 물이 흐르는 환경에 대한 3차원 모델을 구축하고 어디서 물이 나올지 초기 조건을 결정한다. 다음으로 중력의 크기나 회오리의 정도와 같은 파라미터의 값들을 입력한 후에 마지막으로 시간 진행 시물레이션을 수행한다. 이 경우 환경에 하수 구멍이 있으면 눈에 보이기 전에 모든 물이 그 구멍으로 새어나갈 수도 있다.

새로운 FSIM은 2007년 영화 “캐리비안의 해적 3-세상의 끝”에서 20개 정도의 유체 시물레이션 장면 제작에 참여하기도 했다.

5.6 Zeno (Industrial Light & Magic)

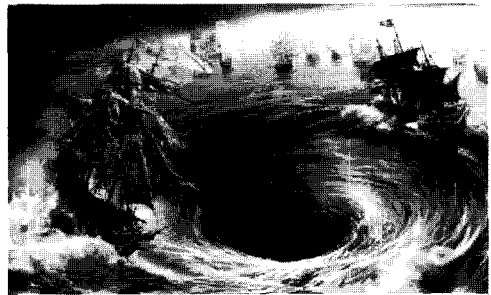


그림 10: 영화 “캐리비안의 해적 3 (2007)” (Buena Vista Pictures/Courtesy Everett Collection)

ILM은 1989년 영화 “어비스”에서 CG 특수 효과에도 참여한 샌프란시스코에 기반을 둔 업체이다 [57]. 이후로 ILM에서 개발한 Zeno 유체 시물레이션 시스템은 2007년 영화 “캐리비안의 해적 3”에서 (그림 10)과 같은 대규모 유체 시물레이션을 수행하였다 [58]. 이 장면에서는 대규모 바다 월풀(whirlpool)에 두 대의 배가 교착 상태에 빠져 있다. 이 그림과 같은 월풀은 ILM의 소프트웨어 디자이너들과 아티스트들 사이의 긴밀한 협력에 의해 가능하였다.

Zeno 유체 시물레이션 시스템은 대규모 물 효과를 생성하기 위해 입자 등위 집합 방법, 병렬 연산과 감독 요구 수용 툴 기술들이 적용되었다. 입자 등위 집합 방법은 Fedkiw이 박사 과정 학생 Doug Enright와 공동으로 고안한 4.1.3절에서 설명한 방법으로 액체의 표면을 매우 자연스럽게 보이게 한다 [34]. 병렬 처리와 결합된 입자 등위 집합 방법은 매우 구체적인 시각 효과를 생성하게 만든다. 감독 요구 수용 툴은 물리 법칙의 제

약 없이 예술적이거나 극적인 효과를 감독이 지시하는 것이 가능하도록 도와준다.

이 장에서 살펴본 유체 SW에 대하여 4장에서 설명한 기술과 함께 특징 및 장단점을 표 2에서 정리하였다.

SW	특징	단점
RealFlow	입자, 4.2.1 절의 SPH, Multi-thread 지원, 편리한 UI, 용이한 외력 추가	날카로운 유체 표면 생성 어려움, CG 유체와 실사 고체 상호작용 표현의 실감도 부족
FELT	Adapted 격자, 스케일별 모듈화 구조 및 합성	다른 스케일의 객체간 비사실적 상호작용 (예 파도-거품 상호작용)
Maya Fluids	격자, 4.1.2 절의 연기 표현, 범용 Maya 기능들과 연동, 굵이치는 연기/불과 바다(호수) 표현에 강점	격렬한 물 표현의 실감도 부족
Flowline	격자, 병렬 처리, 물보라/입자 모듈 구조 및 합성, 다른 격자 SW보다 격렬한 물 표현 가능	다소 과장되어 비물리적
FSIM	격자, 4.1 절의 Semi-Lagrangian 기법 최초 적용, Adaptive refinement 지원하여 LOD별 표현	CG 유체와 실사 고체 상호작용 표현의 실감도 부족
Zeno	격자, 4.1.3 절의 입자 등위 집합 적용하여 액체 형상화에 강점, 병렬 처리, 용이한 외력 추가	대규모 액체 표현은 강점이나 상호작용 부족

표 2: 유체 SW 기술적 특징 및 단점

6. 유체 시뮬레이션 최근 연구/개발 동향

표 2에서 알 수 있듯이 기존의 유체 시뮬레이션 영화는 유체 내부의 물, 파도나 거품과 같은 스케일별 구조들 사이의 상호작용이나 유체 외부의 강체나 변형체와 유체의 상호작용 표현의 실감도가 아직 부족하며 따라서 이를 향상시키려는 연구가 필요하다. 또한 유체 상호작용은 자연스럽게 유체로부터 변형체나 강체로의 상변이(phase transition)나, 유체나 고체의 분할이나 결합과 같은 위상(topology) 변화를 수반하게 된다.

고체와 상호작용하는 유체의 특별한 예로는 운동하는 변형체나 관절체의 표면에서 계속 생성되며 흘러내리는 물, 모래 등 있으며 이와 같은 표현을 최근 영화에서 초자연적인 크리처 표현에 사용하기 시작하고 있다.

이 장에서는 이와 같은 유체 상호작용, 유체 변형과 유체 크리처의 관점에서 최근의 유체 시뮬레이션 연구/개발의 동향을 살펴 보고 향후 이 분야의 연구 개발의 방향을 짚어 보고자 한다.

6.1 유체 상호작용과 유체 위상 변화

유체와 고체의 상호작용과 관련되서는 다음과 같은 연구가 있다. 액체와 고체의 상호작용에서는 고체와 경계를 이루는 액체 표면의 속도상의 경우, 액체의 고체에 대한 상대 속도를 고려한다. 상대 속도의 수직 성분은 보통 0으로 고정하여 액체가 고체로 흐르는 것을 방지하고, 수평 성분에는 아무런 제약 조건을 두지 않아 자유롭게 움직이도록 한다. [33]에서는 고체가 액체에서 움직일 때, 고체에 의해 액체가 밀리는 현상과 고체 주위를 액체가 흐르는 현상을 실감나게 보이게 하는 경계 조건을 제시하였다. [59]에서는 강체를 Eulerian 격자로 투영하여 유체와 강체의 상호작용을 모델링한다. 이 방법에서는 강체와 유체 방정식을 번갈아가며 사용한다. 각 프레임에서 유체와 접촉하고 있는 강체는 유체 격자 내의 강체셀로 투영되고, 부력과 충돌력이 포함된 변형된 NS 방정식으로 그들의 운동을 푼다. 유체 방정식을 푼 후에 강체셀의 속도는 강체 운동으로 투영되고 이 속도를 강체 방정식으로 푼다. 이와 같이 2개의 방정식을 교대로 푸는 과정이 반복된다. [60]에서는 연기 모델에 격자 기반 기술을 사용하는데, 연기와 복잡한 물체의 경계 근처에서 정육면체 대신에 사면체 비구조 메쉬를 사용한다. 이 hybrid mesh 방법은 구부러지거나 복잡한 물체와 연기의 경계를 정확하게 모델링하는 것이 가능하게 한다. 이 논문에서 연기는 질량이 없는 입자로 렌더링한다. [61]에서는 유체와 고체의 상호작용을 빠르고 안정적으로 시뮬레이션 하는 새로운 방법을 제안한다. 이 방법은 coarse 직교 격자를 사용하여 이전의 방법들 보다 10배 이상 더 빨라 실시간 응용에 사용이 가능하다. 이 방법은 식 (16)에서 사용할 압력 p 를 계산하는데 있어서 고전적인 압력 투영법을 운동 에너지 최소화법으로 재해석하는 식을 통하여 유체와 임의의 고체 사이의 안정적인 상호작용 시뮬레이션이 가능하게 한다.

유체와 변형체 사이의 상호작용과 관련되서는 [62]에서는 격자 기반에서 유체의 압력을 이용하여 옷감과 같은 얇은 변형체와 유체와의 쌍방향 상호작용을 처리하는 방법을 제시한다.

최근 유체와 공기의 상호작용과 같은 다상 유체에 대한 활발한 연구가 이루어지고 있다. 예를 들어 [20]에서는 액체와 공기의 경계면에서 표면장력에 의한 압력 비연속성과 점성에 의한 속도 기울기의 비연속성을 다룬다. 이 분야에서는 유체와 공기의 상호작용시 발생하는 공기방울의 움직임과 변화의 표현이 흥미로운 주제이다. [26]에서는 NS방정식과 4.1.3절에서 설명한 등위 집합을 결합하여 다상 유체를 표현하는 공식을 제안하여 물방울과 공기방울을 표현한다. 등위 집합을 [29]의 BFEC에 의하여 대류시키면 물 아래에서 위로 떠오르는 공기방울의 부피가 줄어드는 문제가 있다. [63]에서는 이와 같은 부피 변화에 대한 수학적 모델을 제시하고 부피 손실을 보상하여 부피를 보전하는 방법을 제안한다. [49]에서는 맥주나 샴페인 같은 액체가 컵에 부어질 때 생성되는 거품의 움직임과 액체의 흐름을 SPH기반으로 시뮬레이션 한다. [64]에서는 작은 공기방울이 올라오는 물을 표현하기 위해 격자와 SPH를 결합하는 하이브리드 방법을 제시한다. 이 방법에서는 생동감 있는 작은 공기방울을 표현하기 위해 SPH 입자를 사용한다. 이처럼 유체-공기와 같은 다상 유체에서는 격자 기반 방법과 입자 기반 방법을 혼합 사용하는 접근법이 최근에 많이 시도되고 있다 [65, 58, 64].

유체의 위상 변화나 상변이와 관련되서는 다음과 같은 연구가 있다. [66]에서는 유체 방정식에 온도장과 변화는 점도장을 더하여 액체로 녹을 수 있는 고체를 모델링한다. 이 모델에서는 고체가 매우 큰 점도를 갖는 유체이고 온도가 올라감에 따라 점도가 감소하여 고체가 액체 상태로 상변이한다. 이 방법

으로 모래성을 짓는 모습을 애니메이션하는데, 높은 점도만의 진흙을 모델링하는데 사용된다. [67]에서는 NS 방정식에 탄성항을 더하여 액체와 고체의 성질을 모두 보이는 점탄성 물질을 애니메이션한다. 탄성항은 von Mises의 항복 조건에 의해 조절되고 준선형 소성 모델이 사용된다. 이 모델에서는 유체 격자를 통해 대류하는 변형을 텐서장을 추적하면서 대각 성분은 셀센터에 비대각 성분은 에지 센터에 저장된다. [68]에서는 치약과 같이 점탄성이 있는 물질을 얇은 판 형태로의 변형이나 일부 텅어리로의 분리와 같은 표현을 시뮬레이션한다. 이 논문에서는 고해상도 표면 메쉬를 저해상도 유한 요소 시뮬레이터와 결합한 방법을 이용하여 다양한 점성, 탄성, 소성을 갖는 물질을 매우 빠르면서도 세밀하게 표현한다.

물, 연기나 불이외에도 모래를 유체로 기술하는 다음과 같은 연구가 있다. [69]에서는 입자와 격자를 모두 사용하여 모래를 유체로 기술한다. 우선 Lagrangian 기술로 입자들을 움직이고, 근처 입자들의 속도로부터 격자 점 상에서의 유체의 속도를 얻는다. 이 격자 속도가 식 (17)의 우변의 소스 항에서 사용된다. 이 방법에서는 각각의 격자 셀에서의 변형을 속도를 얻고 이를 이용하여 응력을 계산한다.

6.2 유체 기반 디지털 크리처 생성

유체 상호작용이나 상변이 시뮬레이션을 고려하면 이를 응용한 초자연적인 크리처 표현에 적용이 가능하다. 예를 들어 움직이는 캐릭터를 모두 감싸고 흘러내리는 모래, 물이나 불을 통해 기괴한 크리처의 움직임을 표현할 수 있으며, 이런 초자연적인 크리처가 걷거나 눕거나 증발하는 상변이도 가능하다. 흥미롭게도 최초의 유체 특수 효과 중의 하나인 (그림 1)에서 보여 지는 1989년 영화 “어비스”는 외계 생명체로 등장하는 디지털 크리처 표현에 물 유체를 사용하였다. 1991년 영화 “터미네이터 2”에서 어린 존 코너를 제거하기 위해 미래에서 온 “터미네이터 T-1000”도 액체 금속으로 유체 표현이다. 또한 (그림 7)에서 보여 지는 2007년 영화 “고스트 라이더”도 화염 유체 기반 크리처로 간주할 수 있다.



그림 11: 영화 “스파이더맨 3 (2007)”에서 샌드맨(Sandman)의 탄생 장면 (Columbia Pictures)

2007년 작 “스파이더맨 3”에서 Sony Pictures Imageworks사는 930개의 특수 효과를 생성하였다 [70]. 이들 중 (그림 11)에서 보여 지는 것처럼 샌드맨이 탄생하는 장면은 유체 시뮬레이션이 크리처 생성에 사용된 다른 예이다. 이 장면 생성을 위해 유체 시뮬레이터가 구(sphere) 시뮬레이터와 같은 자료 구조를 공유하는 시스템을 구성한 후 모래 알갱이가 쌓여져 샌드맨이 생성되는 시뮬레이션이 가능하였다.

2008년 영화 “나니아 연대기 - 카스피언 왕자”에서도 클라이막스에 다리를 들어 올려 적군을 강물로 삼켜버리는 강물 거인이 유체 크리처로 등장한다.

이처럼 유체 표현은 디지털 크리처 제작과 연관 지어 왔

영화	유체 크리처	특징
어비스 (1989)	물 축수의 외계인	고형의 축수 몸체에 수작업으로 물 표현
터미네이터 2 (1991)	액체 금속인 악당 터미네이터	액체와 고체 사이의 상변이를 수작업과 Morphing으로 표현
고스트 라이더 (2007)	고스트	고체 몸체에 상존하는 화염 유체 시뮬레이션
스파이더맨 3 (2007)	샌드맨	고체 몸체에 모래 유체가 흘러내리는 구와 유체 시뮬레이션, 고체와 유체의 상호작용
나니아 연대기 2 (2008)	강물 거인	변형체 몸체에 물이 흐르고 뒤편 강물과 상호작용하는 물 유체 시뮬레이션

표 3: 초자연적 유체 크리처들

며, 최근에는 크리처 애니메이션에 유체 시뮬레이션을 적용하는 시도가 표 3에서 보이는 것처럼 주목받고 있다.

2007년 “스파이더맨 3”에는 물탱크에서 쏟아진 물에 휩쓸려 샌드맨이 녹아버리는 장면이 있다. 여기서의 모래 유체 위상 변화는 무언가 부자연스러운 느낌을 준다. 또한 (그림 3, 5, 6)이 보이는 각각 “반지의 제왕 1”, “포세이돈”과 “해피 피트”에서 물에 반응하는 캐릭터가 부자연스러운 자세를 취하거나 바람에 필력은 피부의 변형이 제대로 표현되지 않는 것과 같은 실감성이 떨어지는 문제가 있다. 또한 물로 뛰어들거나 물 속을 헤엄치거나 물에 휩쓸리는 동물 표현이나 바람에 흔들리는 나무잎이나 동물 피부 변형 애니메이션과 같이 유체와 크리처 상호작용 표현의 실감성을 증대하기 위해서는 크리처와 유체 상호작용에 대한 연구도 필요하다.

7. 결론

본 논문에서는 유체 시뮬레이션 방법을 정리하고, 2008년 Sci-Tech Oscar 상을 수상한 유체 시뮬레이션 SW를 중심으로 이들의 기술적 특징과 적용된 영화를 살펴 보았다.

또한 최근의 유체 시뮬레이션 연구/개발 동향을 분석하였는데 상호작용하는 유체와 변형체나 강체를 공통된 프레임워크로 표현, 녹음(melting) 굳음(freezing) 깨짐(fracture) 기화(gasification) 액화(liquefaction)에 수반되는 상변이와 위상 변화 표현, 유체 기반 디지털 크리처 표현이 주요한 연구 개발 방향이 될 것으로 예상된다.

특히 모래와 같은 가루 물질이나 액체, 기체 및 불에 기반한 초자연적인 디지털 크리처 생성과 동식물 크리처와 유체의 상호작용 기술은 초기 연구 단계로 향후 다양한 영상 특수 효과 생성을 위해 지속적인 관심을 기울여야 할 분야로 보인다.

감사의 글

본 연구는 지식경제부/정보통신연구진흥원과 문화부/한국콘텐츠진흥원이 공동으로 추진하는 IT성장동력기술개발사업 [2007-S-051-01, 디지털 크리처 제작 S/W 개발]의 일환으로 수행하였음.

참고 문헌

- [1] R. Bridson, *Fluid Simulation for Computer Graphics*. A K Peters, 2008.
- [2] W. T. Reeves, "Particle systems - a technique for modeling a class of fuzzy objects," *Computer Graphics*, vol. 17, no. 3, pp. 359–376, July 1983.
- [3] A. Fournier and W. T. Reeves, "A simple model of ocean waves," in *SIGGRAPH Proc.*, 1986, pp. 75–84.
- [4] L. D. Landau and E. M. Lifshitz, *Fluid Mechanics*, 2nd ed. Butterworth-Heinemann, 2002.
- [5] "Computational Fluid Dynamics Group," <http://t3.lanl.gov/>.
- [6] "Clay Mathematics Institute," <http://www.claymath.org/>.
- [7] J. T. Kajiya and B. P. V. Herzen, "Ray tracing volume densities," *SIGGRAPH Comput. Graph.*, vol. 18, no. 3, pp. 165–174, 1984.
- [8] N. Foster and D. Metaxas, "Realistic animation of liquids," *Graph. Models Image Process.*, vol. 58, no. 5, pp. 471–483, 1996.
- [9] R. Temam, "Sur l'approximation de la solution des equations de navier-stokes par la methodes des pas fractionnaires," *Arch. Rat. Mech. Anal.*, vol. 33, pp. 377–385, 1969.
- [10] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge University Press, 1992.
- [11] R. Courant, E. Isaacson, and M. Rees, "On the solution of nonlinear hyperbolic differential equations by finite differences," *Communication on Pure and Applied Mathematics*, vol. 5, pp. 243–255, 1952.
- [12] J. Stam and E. Fiume, "Depicting fire and other gaseous phenomena using diffusion processes," in *SIGGRAPH Proc.*, 1995, pp. 129–136.
- [13] J. Stam, "Stable fluids," in *SIGGRAPH Proc.*, 1999, pp. 121–128.
- [14] 강병권, 차득현, 김장희, 임인성, "유체 역학에 기반을 둔 애니메이션 효과," *정보과학회지*, vol. 21, no. 7, pp. 30–38, 2003.
- [15] R. Fedkiw, J. Stam, and H. W. Jensen, "Visual simulation of smoke," in *SIGGRAPH Proc.*, 2001, pp. 15–22.
- [16] A. J. Chorin, "A numerical method for solving incompressible viscous flow problems," *J. Comput. Phys.*, vol. 135, no. 2, pp. 118–125, 1997.
- [17] G. D. Yngve, J. F. O'Brien, and J. K. Hodgins, "Animating explosions," in *SIGGRAPH Proc.*, 2000, pp. 29–36.
- [18] M. Griebel, T. Dornseifer, and T. Neunhoffer, *Numerical Simulation in Fluid Dynamics: Practical Introduction*, ser. SIAM Monographs on Mathematical Modeling and Computation 3. SIAM, 1997.
- [19] F. Losasso, F. Gibou, and R. Fedkiw, "Simulating water and smoke with an octree data structure," in *SIGGRAPH Proc.*, 2004, pp. 457–462.
- [20] J.-M. Hong and C.-H. Kim, "Discontinuous fluids," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 915–920, 2005.
- [21] M. Kang, R. P. Fedkiw, and X.-D. Liu, "A boundary condition capturing method for multiphase incompressible flow," *J. Sci. Comput.*, vol. 15, no. 3, pp. 323–360, 2000.
- [22] L. Shi and Y. Yu, "Visual smoke simulation with adaptive octree refinement," University of Illinois at Urbana-Champaign, Tech. Rep. UIUCDCS-R-2002-2311, 2002.
- [23] W. Hackbusch, *Multi-grid Methods and Applications*. Berlin: Springer, 1985.
- [24] J. Stam, "Real-time fluid dynamics for games," in *Proc. of the Game Developer Conference*, March 2003.
- [25] P. N. Swarztrauber and R. A. Sweet, "Algorithm 541: Efficient fortran subprograms for the solution of separable elliptic partial differential equations [d3]," *ACM Trans. Math. Softw.*, vol. 5, no. 3, pp. 352–364, 1979.
- [26] O.-Y. Song, H. Shin, and H.-S. Ko, "Stable but nondissipative water," *ACM Trans. Graph.*, vol. 24, no. 1, pp. 81–97, 2005.
- [27] J. Stam and E. Fiume, "Turbulent wind fields for gaseous phenomena," in *SIGGRAPH Proc.*, 1993, pp. 369–376.
- [28] J. Steinhoff and D. Underhill, "Modification of the Euler equations for "vorticity confinement": Application to the computation of interacting vortex rings," *Physics of Fluids*, vol. 6, no. 8, pp. 2738–2744, 1994.
- [29] B. Kim, Y. Liu, I. Llamas, and J. Rossignac, "FlowFixer: Using BFEC for fluid simulation," in *Eurographics Workshop on Natural Phenomena*, 2005, pp. 51–56.
- [30] J.-M. Hong, T. Shinar, and R. Fedkiw, "Wrinkled flames and cellular patterns," in *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, 2007, p. 47.
- [31] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," *J. Comput. Phys.*, vol. 79, no. 1, pp. 12–49, 1988.
- [32] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, "A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method)," *J. Comput. Phys.*, vol. 152, no. 2, pp. 457–492, 1999.
- [33] N. Foster and R. Fedkiw, "Practical animation of liquids," in *SIGGRAPH Proc.*, 2001, pp. 23–30.
- [34] D. Enright, S. Marschner, and R. Fedkiw, "Animation and rendering of complex water surfaces," in *SIGGRAPH Proc.*, 2002, pp. 736–744.
- [35] A. W. Bargteil, T. G. Goktekin, J. F. O'Brien, and J. A. Strain, "A semi-Lagrangian contouring method for fluid simulation," *ACM Trans. Graph.*, vol. 25, no. 1, pp. 19–38, 2005.

- [36] J. J. Monaghan, "Smoothed Particle Hydrodynamics," *Ann. Rev. Astron. Astrophys.*, vol. 30, pp. 543–574, 1992.
- [37] J. Monaghan, "Simulating free surface flows with SPH," *J. Comput. Phys.*, vol. 110, no. 2, pp. 399–406, 1994.
- [38] S. Premoze, T. Tasdizen, J. Bigler, A. Lefohn, and R. Whitaker, "Particle-based simulation of fluids," in *Eurographics Proc.*, vol. 22, no. 3, 2003, pp. 401–410.
- [39] S. J. Cummins and M. Rudman, "An SPH projection method," *J. Comput. Phys.*, vol. 152, no. 2, pp. 584–607, 1999.
- [40] B. Adams, M. Pauly, R. Keiser, and L. J. Guibas, "Adaptively sampled particle fluids," *ACM Trans. Graph.*, vol. 26, no. 3, p. 48, 2007.
- [41] J. Stam and E. Fiume, "Depicting fire and other gaseous phenomena using diffusion processes," in *SIGGRAPH Proc.*, 1995, pp. 129–136.
- [42] M. Desbrun and M.-P. Gascuel, "Smoothed particles: a new paradigm for animating highly deformable bodies," in *Eurographics Proc.*, 1996, pp. 61–76.
- [43] M. Desbrun and M.-P. Cani, "Active implicit surface for animation," in *Graphics Interface*, Jun 1998, pp. 143–150.
- [44] D. Stora, P.-O. Agliati, M.-P. Cani, F. Neyret, and J.-D. Gascuel, "Animating lava flows," in *Graphics interface Proc.*, 1999, pp. 203–210.
- [45] S. Hadap and N. Magnenat-Thalmann, "Modeling dynamic hair as a continuum," in *Computer Graphics Forum*, vol. 20, no. 3, 2001, pp. 329–338.
- [46] M. Müller, D. Charypar, and M. Gross, "Particle-based fluid simulation for interactive applications," in *Proc. SIGGRAPH/Eurographics symposium*, 2003, pp. 154–159.
- [47] M. Müller, S. Schirm, M. Teschner, B. Heidelberger, and M. Gross, "Interaction of fluids with deformable solids: Research articles," *Comput. Animat. Virtual Worlds*, vol. 15, no. 3-4, pp. 159–171, 2004.
- [48] M. Müller, B. Solenthaler, R. Keiser, and M. Gross, "Particle-based fluid-fluid interaction," in *Proc. SIGGRAPH/Eurographics symposium*, 2005, pp. 237–244.
- [49] P. W. Cleary, S. H. Pyo, M. Prakash, and B. K. Koo, "Bubbling and frothing liquids," *ACM Trans. Graph.*, vol. 26, no. 3, p. 97, 2007.
- [50] "Scientists and Engineers Get the Oscar for Improving Film Production and Preservation," <http://www.aip.org/isns/reports/2008/005.html>.
- [51] "And the Oscar Goes To: Fluid Simulation Algorithms!" <http://www.popsci.com/entertainment-gaming/article/2008-02/and-oscar-goes-fluid-simulation-algorithms>.
- [52] "RealFlow (Next Limit)," <http://www.nextlimit.com/>.
- [53] "FELT (Rhythm & Hues)," <http://www.rhythm.com/>.
- [54] "Maya Fluid Effects System," <http://www.autodesk.com/>.
- [55] "Flowline (Scanline)," <http://www.flowlines.info/>.
- [56] "FSIM(Digital Domain)," <http://www.d2.com/>.
- [57] "Zeno (Industrial Light & Magic)," <http://www.ilm.com/>.
- [58] F. Losasso, J. Talton, N. Kwatra, and R. Fedkiw, "Two-Way Coupled SPH and Particle Level Set Fluid Simulation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 4, pp. 797–804, 2008.
- [59] M. Carlson, P. J. Mucha, and G. Turk, "Rigid fluid: animating the interplay between rigid bodies and fluid," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 377–384, 2004.
- [60] B. E. Feldman, J. F. O'Brien, and B. M. Klingner, "Animating gases with hybrid meshes," in *SIGGRAPH Proc.*, 2005, pp. 904–909.
- [61] C. Batty, F. Bertails, and R. Bridson, "A fast variational framework for accurate solid-fluid coupling," in *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, 2007, p. 100.
- [62] E. Guendelman, A. Selle, F. Losasso, and R. Fedkiw, "Coupling water and smoke to thin deformable and rigid shells," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, 2005, pp. 973–981.
- [63] B. Kim, Y. Liu, I. Llamas, X. Jiao, and J. Rossignac, "Simulation of bubbles in foam with the volume control method," in *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, 2007, p. 98.
- [64] J.-M. Hong, H.-Y. Lee, J.-C. Yoon, and C.-H. Kim, "Bubbles alive," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–4, 2008.
- [65] J. Kim, D. Cha, B. Chang, B. Koo, and I. Ihm, "Practical animation of turbulent splashing water," in *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2006, pp. 335–344.
- [66] M. Carlson, P. J. Mucha, I. R. Brooks Van Horn, and G. Turk, "Melting and flowing," in *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2002, pp. 167–174.
- [67] T. G. Goktekin, A. W. Bargteil, and J. F. O'Brien, "A method for animating viscoelastic fluids," in *SIGGRAPH Proc.*, 2004, pp. 463–468.
- [68] C. Wojtan and G. Turk, "Fast viscoelastic behavior with thin features," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–8, 2008.
- [69] Y. Zhu and R. Bridson, "Animating sand as a fluid," in *SIGGRAPH Proc.*, 2005, pp. 965–972.
- [70] "Sony Pictures Imageworks," <http://www.imageworks.com/>.