

# CG 유체 표현 기술 동향

The Trends of Fluids Simulation in Computer Graphics

유비쿼터스 시대를 주도할  
디지털콘텐츠 기술 특집

표순형 (S.H. Pyo)

CG기반기술연구팀 선임연구원

구본기 (B.K. Koo)

CG기반기술연구팀 팀장

## 목 차

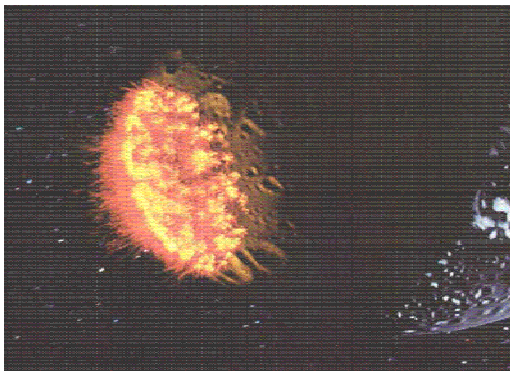
- .....
- I. 서론
  - II. 정적인 대규모 액체 표현
  - III. 동적 액체 및 연기, 불 표현
  - IV. 최근 기술 동향
  - V. 관련 연구 과제
  - VI. 결론

본 논문에서는 영화 및 애니메이션에서 특수효과로 많이 사용되는 유체 표현 기술의 흐름 및 발전 방향을 살펴 본다. 유체의 사실적 움직임을 재현하기 위한 기술은 이미 오랫동안 Computational Fluid Dynamics 분야를 통해 연구되어 왔지만, 그 결과가 컴퓨터 그래픽스 분야에서 적용된 것은 그리 오래되지 않았고 응용 분야의 특성상 아직 해결해야 할 문제들이 많이 남아 있다. 본 논문은 유체 표현을 위해 사용되는 기술의 상이성에 따라 현상을 구분 짓고 각 현상을 표현하기 위한 기술들을 세부적으로 설명한다. 또한 최근에 발표된 기술들이 다루고 있는 이슈들을 짚어 추후 연구 개발의 방향을 제시한다.

## I. 서론

최근 영화에서 차지하는 CG의 비율이 커짐에 따라 CG를 사용한 대표적 특수효과인 물, 불 등의 유체 표현도 영화 및 애니메이션 제작에서 중요한 비중을 차지하게 되었다. 최근에 발표된 한 논문에서는 실제 영화 Shrek[1]의 제작에 참여했던 사람의 글을 빌어 Shrek 제작에 있어 가장 표현하기 어려웠던 장면 중 하나는 컵에 우유를 따르는 장면이었다고 전하고 있다[2]. 시각적으로 보기에 매우 단순하고 쉽게 지나칠 장면이었지만, 그 안에는 많은 기술적 문제들을 내포하고 있다. 본 논문에서는 이러한 CG 유체 표현 기술의 흐름을 짚어 보고 앞으로 진행될 연구의 방향을 제시하고자 한다.

1983년 미국의 Reeves[3]가 Star Trek II: The Wrath of Khan의 한 장면 중 (그림 1)과 같이 화염에 싸이는 행성을 표현하기 위해 particle system을 사용한 이래 근 20년간 컴퓨터 그래픽스 분야에서는 연기, 불, 화염, 물 등의 유체를 사실적으로 표현하기 위한 다양한 연구가 이루어져 왔다. 이들 연구는 오랫동안 물리적 사실성 보다는 시각적인 사실성에 더 비중을 두고 진행되었고 비교적 최근에 들어서야 CFD 분야의 시뮬레이션 기술을 적용하여 물리적으로도 만족할 만한 결과들을 보이고 있다. 이미 오래 전부터 CFD 분야에서는 다양한 상황에서 다양한 유체의 성질을 사실적으로 시뮬레이션 하기 위한 연구가 있어 왔지만, 이 방법 및 결과



(그림 1) Star Trek II: The Wrath of Khan의 한 장면

들이 그래픽스 분야에 적용되기까지 오랜 시간이 걸린 이유는 그 계산량의 방대함에 원인이 있다. 애니메이션 및 다양한 특수효과에 유체역학의 기술들이 적용되고 있는 현 시점에서 정밀한 계산에는 여전히 많은 시간이 소요되어 그래픽스 분야에서는 생산성과 사실성 사이의 균형을 위한 다양한 기술들을 개발하고 있다. 즉, 물리적 사실성에 목적을 둔 CFD와는 달리, 약간의 물리적, 수학적 사실성을 희생하더라도 시각적인 사실성을 유지하는 범위에서 생산성을 높이는 것이 CG 유체 표현 기술의 목표이다.

CG 유체 표현 기술은 크게 모델링(scene representation), 시뮬레이션(simulation), 형상화(reconstruction), 렌더링(rendering)으로 구성된다. 모델링은 시뮬레이션의 대상이 되는 유체 외에 유체와 상호작용이 이루어지는 외부 물체를 시뮬레이션 방법에 적합한 형태로 재구성하는 과정을 말한다. 일례로, 균일한 격자 위치에서 시뮬레이션을 수행하는 방법에 대해서는 상호작용을 하는 물체도 격자 형태로 표현되어야 한다. 시뮬레이션 전에 이루어지는 초기 조건이나 경계 조건의 설정도 이 단계에 속하는 것으로 볼 수 있다. 시뮬레이션은 Navier-Stokes 방정식[4]과 같이 유체의 움직임을 설명하는 식을 사용하거나 또는 간단한 랜덤 요소를 사용하는 등 유체의 움직임을 계산하는 작업을 의미한다. 형상화는 물과 같이 외형을 갖고 이를 유지하는 유체에 대해 시뮬레이션 결과로부터 외형을 계산해내는 과정을 말하며 임의의 점들로부터 외형을 계산하거나 초기에 계산된 외형을 업데이트 해나가는 방법 등이 있다. 끝으로 렌더링은 점, 면 또는 임의의 함수 형태로 구성된 유체와 빛 사이의 관계로부터 최종 영상을 생성하는 과정을 의미한다.

CG 표현을 위해 연구되는 유체를 표현 대상에 따라 분류를 하면, 크게 액체와 기체로 구분될 수 있고 액체는 바다와 같은 정적인 대규모 액체와 컵에 쏟아지는 물, 폭포 등과 같이 동작의 주체가 되는 액체로 구분될 수 있다. 기체의 경우는 그 현상에 따라 연기, 구름, 불, 폭발 등으로 나눌 수 있다. 하지만 적용되는 시뮬레이션 기술 면에서 구분을 지

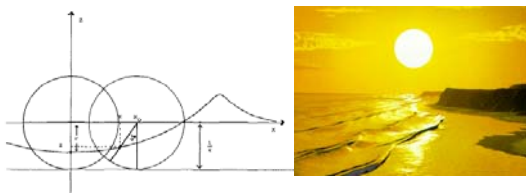
으면, 동작의 주체가 되는 물의 표현은 연기, 불 등과 다르지 않다. 따라서 본 논문에서는 유체 표현을 크게 정적인 대규모의 액체, 즉 바다의 표현과 그 외에 동적인 물, 연기, 불의 표현으로 나누어 설명한다. 동적인 물, 불, 연기는 동일한 시뮬레이션 방법이 적용되나 각각 형상화 및 렌더링 과정에서 다른 형태로 표현이 된다.

II장에서는 정적인 대규모 유체 표현 방법의 동향을 설명하고 III장에서는 동적인 액체 및 불, 연기와 관련된 연구에 대해 논한다. IV장에서는 최근에 발표된 유체 기술을 통해 기술 동향을 파악하고 V장에서 이와 관련된 한국전자통신연구원(ETRI)의 연구 현황을 설명한 후 끝으로 VI장에서 결론을 맺는다.

## II. 정적인 대규모 액체 표현

정적인 대규모 액체의 대표적인 예는 바다로서, 수면에서의 움직임은 있지만, 전체적인 이동이나 변형이 없으므로 정적인 액체라는 표현을 사용하였다. 이에 대한 연구는 물 내부의 시뮬레이션 보다는 주로 파동이론을 이용해 외부에서 본 해수면을 모델링하거나 육지에 가까워짐에 따른 파도의 생성 등에 치중해 있다.

처음 해수면 표현에 대한 연구가 시작된 것은 1980년대 초반부터 이다. 초기의 모델들은 해석학적 함수들로 파도를 모델링 하였고, 주로 height field를 사용해 표현하였다. 1986년 Fournier와 Reeves는 (그림 2)와 같이 해수면의 움직임이 일련의 원 또는 타원 운동을 수행하는 입자들에 의한 것이라는 Gerstner 모델을 도입하여 해수면을 표현하



(그림 2) Gerstner 모델과 Fournier와 Reeves의 연구 결과

였다[5]. 입자의 원운동 궤적을 조절하여 다양한 파도의 생성이 가능했고 또한 수면과 지면 사이의 거리를 이용해 파도의 굴절이나 해수면의 이름에 따른 부서짐도 표현하는 등 해수의 움직임에서 발생할 수 있는 다양한 현상들을 표현하였다. 이들은 해수면을 매개 변수화된 표면으로 구성하고 기존의 렌더링 알고리즘과 환경맵을 사용함으로써 좋은 결과를 생성하였다.

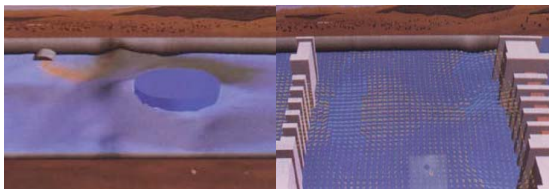
같은 해 Peachey는 파도를 여러 개 겹쳐 해수면을 표현할 수 있도록  $x, y, t$  세 개의 매개 변수를 갖는 함수로 표현하였다[6]. 또한 파도 자체를 파형 프로파일과 위상으로 나누어 표현함으로써 파형을 자유자재로 변형 가능하고 깊이에 따른 변화를 반영할 수 있도록 고안하였다. 해수면 가까이에서는 물마루에서의 원운동 속도가 파도 자체의 속도보다 클 경우 스프레이를 이용해 파도의 부서짐을 표현하였다.

1987년도에는 Ts'O와 Barsky가 렌더링에 사용되는 광선 추적 알고리즘을 적용하여 파도의 굴절을 추적하였고 그 결과를 사인(sine)파를 이용해 해수면 모델링에 적용하였다[7]. 해수면의 렌더링에는 굴절과 반사용 두 개의 텍스처 맵을 사용하였고 수면의 법선 벡터를 고려한 반사 및 굴절 비율에 따라 혼용하여 색을 결정하였다.

1990년대에 들어서는 수면의 움직임에 유체역학적 기술의 도입이 시도되었다. 물의 움직임은 Navier-Stokes 식에 의해 풀이된다는 사실은 이미 알려져 있었으나 이 식을 바로 적용하기에는 당시 컴퓨팅 파워에 문제가 있어 1990년 Kass와 Miller는 표현하는 대상은 얇은 물로 제한하여 식을 단순화시킨 후 적용을 하였다[8]. 또한 기본 방법들이 개개의 파도를 표현하는 데 그친 반면 이들은 반사 파, 흐름, 복잡한 경계, 물의 스퍼들 등의 복합적 현상들을 재현하였다. 하지만 많은 제한 조건과 단순화 과정을 거쳐 시뮬레이션 함으로써 정확성에 한계가 있었고 height map을 이용해 모델링 함으로써 파도의 이중 겹침, 부서짐 등을 표현할 수 없었다.

1995년에 Chen과 Lobo는 처음으로 완전한 Navier-Stokes 식을 2차원 공간에서 수면의 시뮬

레이션에 적용을 하였다[9]. 이들은 (그림 3)과 같이 2차원의 Navier-Stokes 식을 풀어 얻은 각 지점에서의 압력 값을 수면의 높이로 계산해 3차원 물을 표현하는 방법으로 빠른 시간에 물을 시뮬레이션 할 수 있었다. 또한 특정 지점으로 유입 또는 유출되는 속도와 높이의 고저를 나타내는 압력으로부터 삼차원 속도를 계산해 외부 물체를 이동하는 데 사용하였다. 이들의 방법은 사실성과 효율성을 적절히 절충하였으나 근본적으로 2차원적인 접근을 함으로써 표현에 제약을 내포하고 있었다.



(그림 3) Chen과 Lobo의 연구 결과

2000년대에 들어서면서부터는 해수에 대한 연구가 다양성을 띠며 수면의 모델링뿐만 아니라 사실적 렌더링에 대한 연구들이 진척을 보였다. 2000년 Gonzato와 Saec은 파도의 표현에 이용되던 Biesel의 타원 운동 모델이 해안선 가까이에서 지면을 뚫고 들어가는 문제점을 지적하고 Gestner의 원형운동 모델에 사용되는 기준을 적용해 해안선에도 적용이 가능한 파도 모델을 제안하였다[10]. 또한 해안선 근처에서 물마루가 구부러지는 현상 표현을 위해 (그림 4)와 같이 해변 쪽으로 늘어나는 함수와 중력을 반영하기 위한 변위 함수를 추가하여 사실적으로 부서지는 파도를 모델링 하였다. 파도의 굴절과 회절 등의 현상 표현을 위해서는 파동 광선(wave ray)을 이용해 파도의 진행 방향에서 일어나는 충돌을 감지하여 새로운 파도를 생성하거나 적절한 반응을 취하도록 하였다. 기존에 많이 사용되던 height



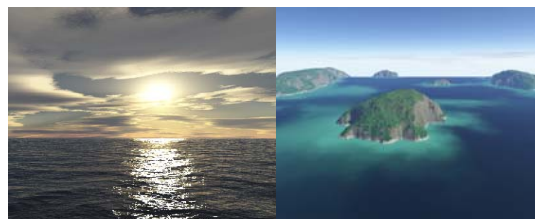
(그림 4) Gonzato와 Saec의 파도 모델

map은 해안선 근처의 파도를 표현하기 어려우므로 이들은 각각의 파도를 변위 맵으로 표현하고 평평한 수면에 반복적으로 적용하여 전체 수면을 모델링 하였다. 렌더링 시에는 에너지의 감쇄와 굴절된 광선이 물체를 만나는 경우 등을 고려하였고 굴절된 광선의 색을 결정하는 데 spectral model을 사용하여 태양과 수평선 사이의 관계와 바다의 색 등을 적용하였다. (그림 5)는 이들의 렌더링 결과를 보인다.



(그림 5) Gonzato와 Saec의 연구 결과

2000년 Utah 대학의 Premoze와 Ashikhmin은 기존에 해양학에서 많이 사용되던 주파수 도메인에서 노이즈를 중첩하는 방식으로 수면을 모델링하고 빛의 전달 과정을 분석해 수중에서 빛이 전달되고 감쇄되는 현상을 표현하였다[11]. 이들은 빛의 전달 과정을 수면에서 발생하는 이벤트와 수중의 전달 과정으로 구분 짓고, 수면에서는 빛의 반사와 굴절을 설명하고 굴절된 빛의 깊이에 따른 래디언스 변화는 물의 광학적 변수와 래디언스 자체의 변화로 표현하였다. 렌더링에는 Monte Carlo path tracer를 사용하였고 외부 환경을 간단한 형태로 모델링해 표현하였다. 이들의 연구는 (그림 6)과 같이 빛과 물 사이의 관계를 잘 표현해 좋은 결과를 생성했으나 수면에서 발생하는 기타 현상에 대한 고려는 배제하였다.

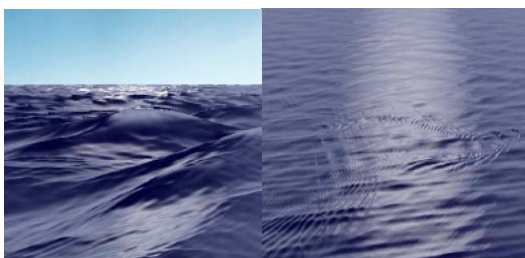


(그림 6) Premoze 등의 연구 결과

Thon 등은 해수면을 사인파로 구성하는 Airy 모델과 날카로운 파도를 표현할 수 있는 Gerstner 모델을 혼합하여 파도를 표현하는 식을 제안하고 이 모델의 계수로 사용되는 진폭, 빈도, 위상, 방향 등을 ocean spectrum으로부터 구하였다[12]. 그러나 샘플링 과정에서 일부 세밀한 정보들이 사라지게 되므로 다시 Perlin turbulence function을 사용해 사실성을 추가하였다. 애니메이션 과정에서도 파도의 기본 구조와 작은 스케일의 동요(perturbation)를 분리해서 기본 구조는 시간에 따라 위상 변화를 통해 애니메이션을 하고 두드러지지 않는 동요는 간단히 변위 변화로 구현하였다. 이들 연구는 새로운 파도 모델을 제시했으나 깊은 바다에 제한된 연구 결과를 보였다.

지금까지 설명한 논문들은 주로 해수면과 해안선의 사실적 모델링 및 시뮬레이션에 중점을 두었던 반면 최근에 발표된 논문들은 인터랙티브한 응용을 목적으로 성능 개선에 초점을 맞추고 있다. 2002년 Hinsinger 등은 실제 해수면 모델을 계산하는 과정은 앞서 설명한 Thon과 유사한 방법을 사용하였으나 화면 상의 격자를 역으로 해수면에 투사하여 필요한 만큼의 메시만 생성하는 방법으로 해수면 시뮬레이션의 성능을 개선하였다[13]. 렌더링은 환경맵과 하드웨어 가속 기법을 이용하여 50×50 크기의 격자에 대해 초당 20~30프레임의 영상을 생성하였다. 이들 역시 (그림 7)과 같이 해안에서의 굴절 등은 고려하지 않고 먼 바다의 외형 변형에 대해서만 다루었다.

2003년 Jeschke 등은 파도의 모양뿐만 아니라 그 움직임까지도 일련의 함수로 모델링되는 파도 모



(그림 7) Hinsinger 등의 수면 모델

델을 제시함으로써 빠른 시간에 파도의 부서짐까지도 표현할 수 있었다[14]. 파도는 코사인, 지수, 회전, 스케일 4개의 함수의 조합으로 표현되었고 함수가 가지는 연속성으로 인해 이전 단계에서의 정보 없이도 파도상 어느 지점의 정보든 계산이 가능하다. 렌더링은 Hinsinger 등과 마찬가지로 영상 기반의 접근법을 채용하여 1024×768 영상에 대해 4픽셀 단위로 샘플링하여 초당 35프레임의 영상을 생성하였다. 이 연구는 부서지는 파도 등을 표현하기 위해 새로운 모델링 방법을 제안했으나 속도 개선을 위해서는 Hinsinger의 방법을 그대로 이용하는 데 그쳤다.

Loviscach는 파동의 모든 지점에서 새로운 원형 파동이 발생한다는 Huygen의 원리를 이용해 돌이 떨어지는 갑작스런 교란이나 배가 지나가는 것과 같은 지속적인 교란에 반응하는 물 표면을 빠르게 시뮬레이션 하였다[15]. 렌더링 과정에서는 수면을 구성하는 다각형 중 일부만을 실제로 변형하고 나머지는 텍스처 매핑을 이용해 표현함으로써 성능의 개선을 도모하였다. 또한 텍스처 셰이더를 이용한 하드웨어 축적(accumulation) 버퍼의 에물레이션을 통해 커스틱스(caustics)를 재현하여 사실성을 증대시켰다. 하지만 깊이에 따른 굴절이나 경계 처리 등이 고려되지 않아 아직 단순한 경우에만 적용이 가능하다.

지금까지 살펴 보았듯이 현재까지 발표된 정적인 대용량의 물에 대한 연구는 단순화된 상황에 대한 표현 기법들에 치중되었고 외부 물체와의 상호작용이나 기후와 같은 외부 환경 변화에 대한 시뮬레이션 등 복잡한 상황에 대한 고려는 이루어지지 않고 있다. 또한 파도나 외부 물체에 의한 포말 등 사실성을 높이기 위한 요소들도 고려되어야 할 것이다.

### III. 동적 액체 및 연기, 불 표현

이 장에서 다루고자 하는 동적 액체는 물론 제한된 공간에 고여 있는 상태와 같이 정적인 표현도 가능하지만, 불과 연기처럼 움직임의 주체가 되는 액

체를 의미한다. 즉, 흐르는 물, 폭포 등과 같이 사용자의 의도에 따라 동선을 갖고 움직이는 액체를 말한다.

불이나 화염과 같은 유체를 표현하고자 하는 처음 시도는 앞서 말했듯이 1983년 Reeves에 의해 영화의 한 장면을 완성하기 위해 이루어졌다[3]. Reeves에 의해 제안된 파티클 시스템은 유체를 일련의 파티클의 집합으로 구성하고 각각의 파티클이 시간, 온도 등의 매개변수에 의해 탄생, 소멸해가며 전체의 모습을 구성하는 기술로서 현재까지도 영화 및 게임 분야에서 특수효과 연출을 위해 이용되고 있다. 이 시기의 파티클 시스템은 파티클의 생사를 프로그래밍하여 원하는 효과를 생성할 수는 있었으나 유체의 사실적 시뮬레이션의 결과와 결합되어 사용되지는 못하였다.

1990년대에 들어서기까지는 기술의 커다란 변화가 없다가 1993년 Jos Stam이 대류와 확산 과정을 통해 가스 형태의 유체를 표현하기 위한 기술을 제안하였다[16]. Stam은 가스의 움직임에 영향을 미치는 바람 장(wind field)을 거시적 움직임과 가스 자체의 확산 과정에서 발생하는 난류를 반영하기 위한 미세 움직임의 합으로 생각하였다. 거시적 움직임은 사용자 지정에 의해 결정되며 미세 움직임은 주파수 도메인에서 에너지 스펙트럼 함수를 적용하여 표현하였다. 사용자는 이 스펙트럼 함수의 모양을 지정함으로써 난류의 질을 조정할 수 있다. 계산된 바람 장에서 움직이는 가스는 일련의 파티클로 구성되며, 각 파티클은 밀도 분산 함수(density distribution function)를 가져 가스를 표현한다. 렌더링은 시선에서 나간 광선이 어떤 파티클 블롭(blob)들과 만나는지 체크하고 광선을 여러 개의 겹치지 않는 조각으로 나눈 후 각 교차점에서 셰이딩

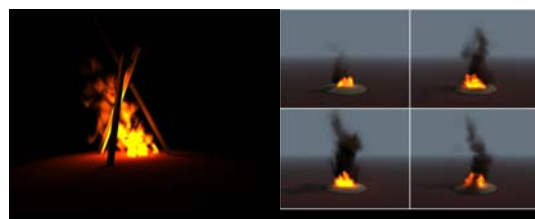
하여 투명도를 고려해 축적해 감으로써 이루어진다. 이 연구는 (그림 8)과 같이 보기에 그럴듯한 결과를 생성하였으나 정확한 시뮬레이션 모델은 아니었고 미세한 움직임은 거시적 움직임에 영향을 전혀 받지 않아 사실성이 미흡하였다.

1995년 Stam은 다시 이전에 제안한 방법보다 더 적은 수의 프리미티브로 다양한 가스 현상을 애니메이션 할 수 있는 방법을 제안하였다[17]. 가스의 움직임을 제어하기 위한 바람 장은 이전과 마찬가지로 사용자 지정의 거시적 움직임과 통계적 매개변수를 이용한 미세 움직임으로 구성하였으나 가스의 밀도 및 온도는 advection-diffusion 프로세스를 이용하여 업데이트 하였다. 삼차원 그리드에서 advection-diffusion 식을 계산한 후 각 지점에서의 밀도 및 온도를 계산하는 것은 너무 많은 계산이 필요하므로 이 방법에서는 특정 개수의 파티클에서 계산한 후 인접한 파티클의 값을 이용해 임의의 지점에서 밀도 및 온도 값을 얻는 방법을 사용하였다. 또한 온도에 따른 상태의 변화를 이용해 연료에 의한 불의 전이 등을 애니메이션 할 수 있었다. 렌더링에는 가스에 의한 빛의 분산, 흡수, 발산 과정을 반영하여 (그림 9)와 같은 사실적인 영상을 얻었다. Stam의 연구는 유체 표현의 가시적인 사실성을 한 단계 높였으나 아직 물리적인 사실성에서 여지를 남겨두고 있었다.

1996년에 Nick Foster와 Dimitri Metaxas는 유체의 움직임을 설명하는 Navier-Stokes 식을 유한차분(finite difference) 방법을 이용해 계산하여 적용하였다[18]. 이들은 장애물 및 대기를 모두 일종의 유체로 보고 전체 환경을 사각 그리드(grid)로 나눈 후 각 셀을 구성하는 육 면의 중심에서 속도를 정

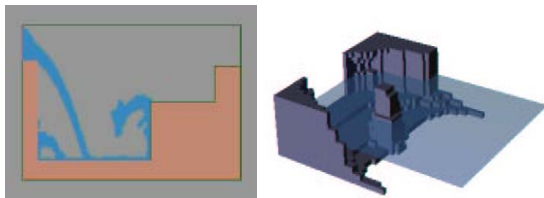


(그림 8) Stam의 1993년 연구 결과



(그림 9) Stam의 1995년 연구 결과

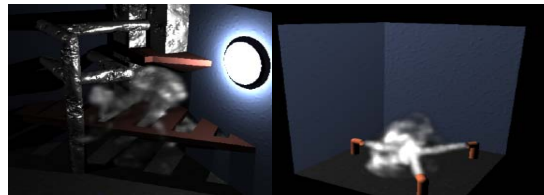
의하였다. 이후 Navier-Stokes 식과 질량 보존의 법칙을 이용해 각 셀의 속도와 압력을 업데이트하고 이웃한 셀 사이의 질량 보존을 만족시키기 위한 보정 과정을 반복한다. 장애물과 유체가 만나는 셀 면의 속도는 0으로, 접선 방향의 속도는 장애물과 유체의 속성을 고려해 정함으로써 물체와의 상호작용도 고려하였다. 또한 특정 셀의 속도를 지정하고 유지함으로써 유체의 유입과 유출도 표현 가능하였다. 물의 경우는 기타 가스 및 불과 달리 수면을 추적하는 과정을 필요로 한다. 이들은 유체를 따라 움직이는 파티클들을 이용해 각 셀이 유체에 속하는지 아닌지를 판단하고 이로부터 수면을 구성하였다. 이 논문은 처음으로 Navier-Stokes 식을 적용해 시뮬레이션한 것으로 이전의 연구들과 이후의 연구들을 구분 짓는 큰 의미를 갖는다. 그러나 속도와 압력을 구하기 위한 반복 과정에서 빠르게 수렴하지 않을 가능성이 있어 불안정한 요소를 갖고 있다. (그림 10)은 이들의 연구 결과를 보이고 있다.



(그림 10) Navier-Stokes 식을 적용한 Foster의 연구 결과

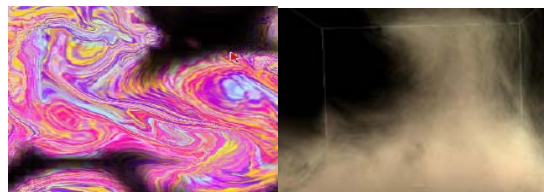
1997년 이들은 다시 유사한 방법으로 뜨거운 가스가 다른 매질이나 물체를 만났을 때 발생하는 난류성 회전 운동을 표현하기 위한 애니메이션 방법을 제안한다[19]. 비압축성 유체를 표현하기 위한 Navier-Stokes 식에 주위와의 온도 차로 인한 속도 변화 항을 추가해 가스의 세밀한 동작을 표현하였다. 식에 나타나는 미분 항은 테일러 시리즈(Taylor Series)로 근사하고 유한 차분 방법으로 해를 구하였고, 셀의 크기에 따라 발생할 수 있는 오차를 제거하기 위해 보정 단계를 반복적으로 적용하였다. 이 방법은 이전 논문에서도 지적되었던 불안정성 문제

를 여전히 내포하고 있어 가스의 속도가 빠르거나 시뮬레이션 간격이 커질 경우 문제가 발생할 수 있다. 이들은 (그림 11)과 같이 물체와의 경계면 등에서 경계 조건을 다양하게 적용하여 여러 가지 특별한 효과를 생성할 수 있음을 보였다.



(그림 11) Foster의 난류성 가스 표현

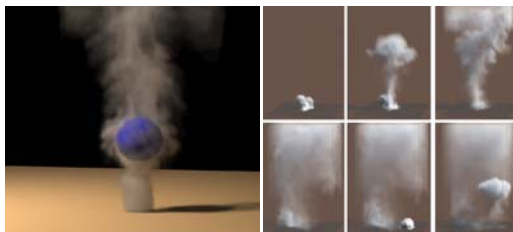
1999년 Stam은 이전 Foster의 논문들에서 보였던 불안정성 문제를 해결하고 좀 더 긴 시간 간격으로 시뮬레이션이 가능한 새로운 방법을 제안하였다 [20]. Stam은 Navier-Stokes 식을 구성하는 질량 보존의 식과 에너지 보존의 식을 Helmholtz-Hodge 분해 방법을 이용해 속도에 대한 하나의 식으로 표현을 하고 외부 힘, 이동, 확산에 의해 계산된 값을 발산을 막기 위해 프로젝션, 즉 보정하는 방식으로 시뮬레이션을 수행하였다. Stam의 방법이 이전의 방법들에 비해 안정적인 이유는 단순히 이웃한 셀의 값으로부터 현재 셀의 속도를 계산하는 방식이 아니라, 현재 셀에 위치한 파티클을 역추적해 이전에 있었던 셀로부터 속도를 가져오는 semi-Lagrangian 방식을 사용했기 때문이다. 특히 시뮬레이션 공간의 외부 경계 조건이 주기적인 경우는 푸리에 영역으로 옮겨 계산을 수행함으로써 빠른 성능을 보였다. 하지만 이 방법은 애니메이션 측면의 실용성을 위해 정확도를 어느 정도 포기한 방법으로



(그림 12) Stam의 Stable Fluids

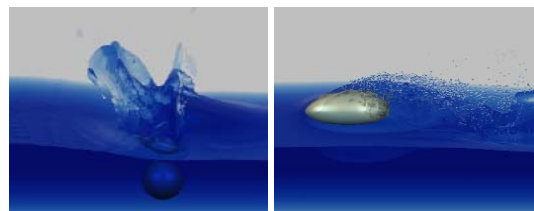
실제에 비해 빠르게 수렴하는 경향을 보이고 물과 같이 자유 경계를 갖는 것에 대해서는 다루지 않았다. (그림 12)는 이들의 연구 결과를 보인다.

Stam의 안정적인 유체 시뮬레이션 방법에서 발생했던 빠르게 잠잠해지는 문제는 2001년 Fedkiw 등과의 연구에서 개선이 이루어진다[21]. 이 논문은 계산량이 많은 점성 Navier-Stokes 식 대신 가스 표현에 적합한 비점성의 유체 표현을 위한 Euler 식을 사용함으로써 기존 방법들보다 좀 더 넓은 간격의 그리드에서 빠르고 효율적으로 연기를 표현할 수 있는 수치적 방법을 제안하였다. 또한 성긴 그리드에서 표현하기 어려운 작은 크기의 회전 특성을 모델링하기 위해 물리적으로 옳은 소용돌이 제한 항인 vorticity confinement를 적용함으로써 이전 Stam의 연구에서 발생한 문제를 해결하였다. 전체 시뮬레이션 과정은 먼저 그리드 상에서 부력과 vorticity confinement를 포함하는 외부 힘을 계산하고 Euler 식으로부터 속도를 계산한 후 끝으로 계산된 값이 질량을 보존하도록 보정을 가한다. 속도를 계산하는 과정에서는 이전 Stam의 방법과 마찬가지로 semi-Lagrangian 방법을 사용하였다. 렌더링 방법으로는 삼차원 텍스처를 이용한 하드웨어 기반의 고속 렌더링 기법과 photon mapping을 이용한 고품질 렌더링 기법 두 가지를 제안하여 목적에 맞게 사용할 수 있도록 하였다. (그림 13)은 vorticity confinement를 이용한 시뮬레이션 결과를 photon mapping 방식으로 렌더링한 결과이다. 이 논문의 결과는 이전 Stam의 논문과 함께 이후에 발표되는 유체 관련 논문의 주요 시뮬레이션 방법으로 이용되고 있다.



(그림 13) Vorticity Confinement 적용과 Photon Mapping 결과

2000년대에 들어서면서부터는 물의 사실적 표현에 관한 연구가 기존 기체성 유체 표현에 관한 연구와 함께 큰 주류를 형성하게 된다. 물과 기타 유체의 큰 차이점은 물이 표면을 형성한다는 점에서 찾을 수 있다. 2001년 Foster와 Fedkiw는 Stam의 semi-Lagrangian 방법을 이용해 시뮬레이션을 하되 이 결과를 물 표현에 직접 적용하기에는 질량의 확산이 과다해 가스 등의 표현에 적합하므로 비관성(inertialless)의 파티클과 level set을 이용하여 물을 표현하였다[22]. 먼저 semi-Lagrangian 방법과 혼용된 유한 차분을 이용해 속도 벡터를 계산하고 움직이는 물체에 의한 제약 조건을 반영한 후 액체의 비압축성이 보장되도록 보정을 수행한다. 속도 벡터의 계산이 완료되면 초기에 계산되었던 파티클의 위치와 level set으로 표현되는 수면을 업데이트한 후 수면과 파티클의 관계를 고려해 파티클을 제거 또는 추가한 후 level set도 수정을 가한다. 이 방법은 implicit surface만을 사용할 경우 발생할 수 있는 체적의 손실이나 지나친 부드러움을 파티클로 보정하고, 수면 외부의 파티클을 물 튀김 현상 표현에 사용함으로써 (그림 14)와 같은 사실적인 수면 표현이 가능하였다. 특히 동적으로 움직이는 객체와의 상호작용을 고려하여 애니메이션에서의 활용 가능성을 극대화하였다.



(그림 14) Foster의 물 및 상호작용 표현

이어서 2002년에는 Enright, Marschner와 Fedkiw이 앞서 설명한 Foster의 결과를 개선한 연구를 발표하였다[2]. 이들은 Foster의 연구는 파티클을 이용하여 수면이 안쪽으로 수축하여 체적을 손실하는 것은 막았으나 수면이 외부로 팽창해가는 것은 막을 수 없음을 보이고 해결책으로 외부에도 일련의



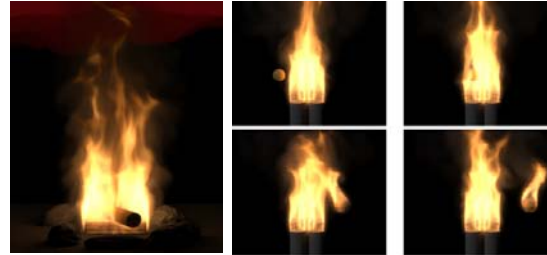


(그림 15) Particle Level Set Method

파티클을 분포시켜 수면을 정확히 추적할 수 있는 Particle Level Set Method를 제안하였다. 또한 시뮬레이션 된 결과를 photon mapping이라는 전역 조명 렌더링 알고리즘에 적용하여 (그림 15)와 같이 더욱 사실적인 영상을 생성하였다. 하지만 처음 시뮬레이션 방법을 제안했던 Stam의 의도와는 달리 너무 많은 시간이 소요되는 단점을 안고 있다.

이 외에도 2002년 Nguyen, Fedkiw와 Jensen은 불을 모델링하고 렌더링하기 위한 물리적 모델을 제안하였다[23]. 기화된 연료와 뜨거운 가스를 모델링하기 위해 Navier-Stokes 식을 사용하였고 기화된 연료가 가스를 형성할 때 생기는 팽창과 고체 연료가 가스 상태로 기화할 때 생기는 팽창을 표현하기 위한 모델도 제안하였다. 이들은 가스 형태의 연료가 가스 화합물로 변하는 경계면인 blue core를 level set 방법을 이용해 추적하고 가스 연료와 가스 화합물의 시뮬레이션은 Stam의 방법에 vorticity confinement를 혼합하여 적용하였다. 불의 렌더링은 불 속 매개물에서 빛을 흩뿌리는 현상을 일련의 위상 함수로 표현을 하고 렌더링하기 위한 광선을 여러 개의 조각으로 나눈 후 각 지점에서의 에너지를 계산하여 합하는 방식을 사용하였다. 이 연구는 고체 또는 기체 연료로부터 사실적인 난류성 화염을 생성하는 데 이용이 가능하며 (그림 16)과 같이 불과 외부 물체와의 상호작용도 표현이 가능하다.

지금까지 설명한 기술을 정점으로 물 및 유체 표현을 위한 CG 기술은 시각적 사실성뿐만 아니라 물리적 사실성에서도 만족할 만한 완성도를 보였고, 이후의 기술들은 이들을 바탕으로 속도의 개선, 자유로운 유체의 조정, 세부 현상 표현의 극대화라는



(그림 16) 물리 기반의 사실적 불 표현

방향으로 전개되기 시작하였다. 다음 장에서는 최근에 발표된 기술들을 설명하고 이들로부터 기술 개발의 동향에 대해 논한다.

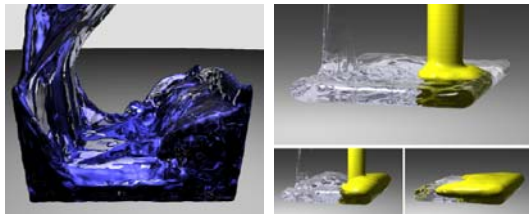
## IV. 최근 기술 동향

### 1. 파티클 기반의 시뮬레이션 기술

유체를 시뮬레이션 하는 방법은 그 방식에 따라 삼차원 공간 상의 특정 위치 즉, 그리드 상에서 속도를 계산하는 Eulerian 접근법과 공간 상의 임의의 위치 즉, 파티클의 이동을 따라가며 속도를 계산하는 Lagrangian 접근법으로 구분이 된다. 앞서 설명한 대부분의 방법들이 삼차원 그리드 상에서 계산된 속도 벡터를 이용해 파티클을 이동하고 수면을 추적하는 Eulerian 접근법을 사용한 반면, 최근 들어서는 Lagrangian 방법을 사용한 논문들이 발표되기 시작하였다.

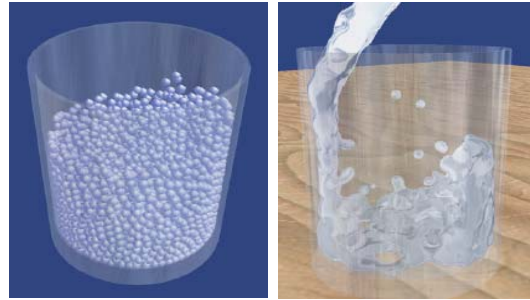
2003년 Premoze 등은 유체 표현에 사용되는 질량, 에너지, 운동량 보존의 식을 파티클 사이의 상호작용 식으로 표현하는 방법을 사용해 유체를 시뮬레이션 하였다[24]. 각 파티클의 속도와 위치를 초기화한 후, 매 시뮬레이션 시간 간격에 계산된 힘에 따라 파티클의 새 속도와 위치를 얻고 비압축성 유체를 위해 Poisson 압력 방정식을 풀어 다시 속도와 위치의 보정 과정을 거치는 MPS 방식으로 시뮬레이션이 진행된다. 하지만 이 방법은 유체의 유입과 유출을 처리할 수 없고 필요에 따른 지역적 해상도 증대가 어려우므로 다시 그리드가 필요 없는 복합 접근법을 제시하였다. 먼저 앞서 설명한 MPS 방식

으로 파티클의 위치 및 속도를 계산하고, 고정 경계에 해당하는 파티클을 원위치 시키며 필요에 따라 파티클을 추가하는 재설정 단계를 거친 후, 다시 1차원 그리드에서 파티클의 이동을 계산하는 방식이다. 이 방법은 (그림 17)과 같이 서로 다른 유체 간의 혼합이나 유체와 물체의 상호작용, 그리고 유체 경계면의 깨어짐 등의 현상을 표현할 수 있다. 이들도 수면의 추적을 위해서는 level-set 방법을 적용하였다.



(그림 17) Premoze 등의 연구 결과

Lagrangian 접근법을 사용한 또 다른 연구로는 같은 해 Muller 등에 의해 발표된 연구가 있다[25]. 이들은 빠른 응답 시간을 요구하는 응용을 위해 SPH 방법에 기반하여 유체의 자유 표면 시뮬레이션을 빠르게 수행하는 방법을 제안하였다. SPH는 공간에 이산적으로 분포된 파티클에 정의된 값으로부터 smoothing kernel을 이용해 임의의 지점에서 값을 얻는 방법으로 특정 지점에서의 값은 모든 파티클의 영향을 가중치를 두어 합함으로써 계산된다. 각 파티클은 압력, 중력 등의 외부 힘, 점성, 표면 장력에 의해 결정되는 힘으로부터 가속도를 계산하고 이로부터 새로운 속도를 계산한다. 이들은 그리드 대신 파티클에서 시뮬레이션을 수행하므로 Navier-Stokes 식 중 질량 보존 제한과 대류 항을 생략할 수 있어 빠른 계산이 가능하였다. SPH의 안정성은 smoothing kernel에 의해 결정된다. 수면의 추적과 렌더링을 위한 법선 벡터의 계산은 파티클이 위치한 곳은 1, 없는 곳은 0으로 표시되는 color field를 이용해 이루어졌으며 영상은 (그림 18)과 같이 점 기반 렌더링을 통해 생성되었다. 결과적으로 GeForce4 그래픽 카드를 갖춘 Pentium 4 1.8GHz

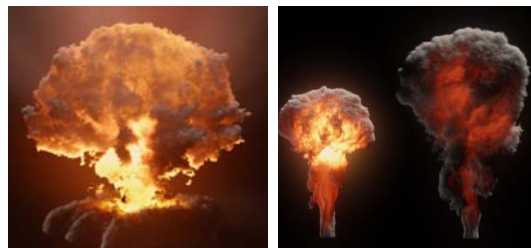


(그림 18) SPH와 점 기반 렌더링

PC에서 2200개의 파티클에 대해 초당 20프레임의 영상을 생성하는 성능을 보였다.

## 2. 시뮬레이션 성능의 개선

영화 및 애니메이션에 유체 기술이 활용되는 사례는 늘어가고 있으나 컴퓨팅 환경의 발전은 이들의 요구를 만족시키기에는 아직 한계가 있어 이전에 발표된 연구 결과들은 작은 규모의 시뮬레이션에서만 실용성이 있었다. 하지만 2003년 Rasmussen 등은 (그림 19)의 핵 폭발과 같이 큰 규모의 장면을 시뮬레이션 하기 위한 효율적 방법을 제안하였다[26]. 이들은 이전의 방법들과 마찬가지로 Stam과 Fedkiw의 vorticity confinement를 이용한 시뮬레이션 방법을 사용했으나 여러 개의 고해상도 이차원 속도장(velocity fields)를 구한 뒤 이를 보간(interpolation)해 삼차원 속도 장을 구함으로써 성능을 개선하였다. 특정 지점에서 속도를 구하기 위해서는 인접한 두 평면 상의 가장 가까운 두 점을 찾아 보간하여 구한다. 또한 이차원 접근법으로 인한 시각적 문제를 해결하기 위해 Kolmogorov 에너지 스펙트럼



(그림 19) 대규모 폭발 장면의 표현

을 이용해 삼차원 난류 요소를 추가하였다. 끝으로 계산된 속도 벡터에 따라 파티클을 이동시킴으로써 최종 결과를 얻는다. 렌더링 시에는 이동시킨 파티클을 삼차원 그리드로 매핑시켜 렌더링하는 것은 메모리 사용이 과다하므로 카메라의 뷰 프러스텀(view frustum)과 평행한 피라미드 형태의 그리드를 사용하여 렌더링함으로써 성능을 향상시켰다. 이 방법은 2000×2000 크기의 이차원 그리드를 사용하여 시뮬레이션하는 데 약 2분이 소요되었고 이 결과에 따라 6백만 개의 파티클을 이동시키는 데 추가로 2분 정도가 소요되었다. 렌더링에는 일반적으로 5~10분이 소요된다.

2004년 Lossaso 등은 기존의 균일한 그리드에서 수행하던 시뮬레이션 방식을 탈피해 수면, 물체와의 경계 등 3차원 환경의 중요도에 따라 옥트리(octree)를 이용해 공간을 적응적으로 분할하고 이를 기반으로 시뮬레이션을 수행하는 방법을 발표하였다[27]. 이는 기존의 방법들이 불필요한 부분까지 자세한 시뮬레이션을 수행하느라 소비하던 시간을 중요한 부분의 세밀한 시뮬레이션에 사용할 수 있어 같은 시간에 더 나은 결과를 얻을 수 있었다. 이들은 512×512×512 크기의 공간에 대해서 한 프레임 당 4~5분 정도의 시뮬레이션을 수행하는 성능을 보였다.

이 외에도 1990년대에 유체 표현 기술의 발전을 주도했던 Stam은 기존 연구의 실시간성에 초점을 맞추어 게임 등에서의 활용을 위한 연구를 꾸준히 진행하고 있다[28],[29]. 이 방법은 유체 시뮬레이션을 위한 공간의 경계 조건이 반복적이라는 가정 하에 주파수 도메인으로 문제를 옮겨 빠르게 계산하는 방법으로 실제 경계조건이 반복되는 경우는 실 세계에서 보기 힘든 경우이나 게임과 같이 사실성보다는 빠른 계산 결과를 요구하는 분야에서는 실용성이 있을 것으로 보인다.

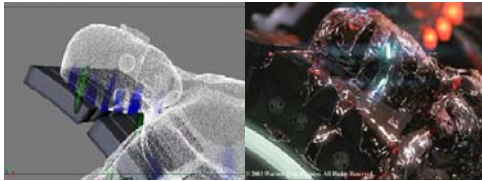
### 3. 유체 컨트롤 기술

유체 시뮬레이션 기술은 사용자에게 자동으로 사실적인 애니메이션을 생성할 수 있는 방법을 제공하

였으나 시뮬레이션에 전적으로 의존하다 보니 사용자의 의도대로 유체의 움직임을 조정하는 데에는 어려움이 있다. 2003년 Treuille 등은 기존 애니메이션에서 널리 사용되는 키프레임(keyframe) 방식을 유체 애니메이션에도 적용하고자 하는 연구를 수행하였다[30]. 사용자에게 의해 지정된 밀도와 속도 등에 대한 키프레임이 시뮬레이션에 영향을 미치도록 매개 변수화된 힘  $u$ 를 정의하고 최적화 과정을 이용해 시뮬레이션의 상태가 사용자가 정의한 상태에 가까워지도록  $u$ 를 계산해 간다. 시뮬레이션과 키프레임이 얼마나 일치하는가를 판단하기 위해 목적 함수(objective function)을 정의해 사용하는데 이 함수는 시뮬레이션의 결과가 최대한 키프레임에 가깝되 최소한의 힘을 적용해 근접시켜야 한다는 목적을 만족시켜야 한다. 이 논문에서는 이 방법을 해결하기 위해 그래디언트(gradient) 기반의 방법을 사용하였다. 시뮬레이션 자체는 마찬가지로 Stam과 Fedkiw의 방법을 사용하였고 시뮬레이션의 길이가 길어질수록 복잡해지는 미분 계산과 지역적 최소 해에 도달하는 문제를 해결하기 위해 문제를 작은 단위로 나누어 계산하고 그 결과를 전이시키는 다중 사격법(multiple shooting scheme)을 사용하였으며 30×30×30 크기의 그리드에서 최적화에 2시간이 소요되었다. 이 방법은 문제의 도메인이 큰 경우 계산량이 과다해지고 키프레임이 너무 떨어져 있거나 서로 경쟁하는 경우 지역적 최소 해에 빠질 수 있다는 단점을 갖고 있다. 그러나 (그림 20)의 결과와 같이 처음으로 유체 시뮬레이션을 위한 사용자 편의의 조정 방법을 연구했다는 데 의의가 있다. 이 연구를 필두로 2004년에는 마찬가지로 Fattal과 McNamara에



(그림 20) Keyframe을 이용한 유체 컨트롤



(그림 21) 파티클을 이용한 유체 컨트롤

의해 키프레임을 활용한 유체의 컨트롤 방법들이 발표되었다[31],[32].

또 다른 유체 컨트롤 기술로는 2004년 Rasmusen 등이 발표한 파티클 기반의 방법을 들 수 있다 [33]. 이들은 (그림 21)과 같이 사용자의 의도에 따라 움직이는 파티클들을 키프레임 방식을 이용해 애니메이션되도록 지정해두고 실제 시뮬레이션 결과 계산 시 이들과의 보간을 통해서 적절한 애니메이션이 생성되도록 하였다. 이 기술은 실제 프로덕션 제작에 적용하기 위해 개발된 것으로 파티클의 키프레이밍을 Maya에서 수행하였다.

지금까지 살펴본 것과 같이 최근에는 사실성을 바탕으로 사용자 편의의 조작성, 대규모 장면 연출을 위한 효율성 등으로 그 연구 방향이 확대되어 가고 있음을 볼 수 있다. 또한 2005년 SIGGRAPH에 발표될 예정인 논문들을 보면 특수한 상황의 유체 표현을 통한 사실성의 극대화가 최근의 이슈임을 알 수 있다.

## V. 관련 연구 과제

II~IV장을 통해 설명한 기술 동향 분석을 바탕으로 한국전자통신연구원에서도 유체 관련 기술 개발을 위해 연구개발 과제를 수행하고 있다.

2004년 말부터 한국전자통신연구원(ETRI)의 디지털콘텐츠연구단은 호주의 국가 연구소인 CSIRO의 Mathematical and Information 부서와 “영상 특수효과용 유체 시뮬레이션 기술 개발” 국제공동연구 과제를 수행하고 있다. CSIRO MIS의 Computation Fluid Dynamics 팀은 파티클 기반 유체 시뮬레이션의 대표적 기술인 SPH 기술을 오랜 기간 연

<표 1> 그리드 방식과 파티클 방식의 비교

|    | 그리드 기반 방식  | 파티클 기반 방식  |
|----|--|--|
| 장점 | <ul style="list-style-type: none"> <li>- 유한 차분법을 이용한 효율적 계산</li> <li>- CG 분야에 보편적</li> <li>- 구현 용이</li> <li>- 세밀한 유체 표현</li> <li>- Level Set과의 연동 통한 부드러운 유체 표현</li> </ul> | <ul style="list-style-type: none"> <li>- 공간에 비 의존적이어서 동적, 확장성 유체 표현</li> <li>- 한 프레임워크에서 입자성 현상 표현 가능</li> <li>- 외부 물체와의 상호작용 시 입자와 물체간의 상호작용으로 간주, 세밀한 표현</li> <li>- 가변 파티클 및 다중 해상도 이용 성능 조정 가능</li> </ul> |
| 단점 | <ul style="list-style-type: none"> <li>- 제한 공간내 시뮬레이션</li> <li>- 고해상도 시뮬레이션 시 과도한 계산량</li> <li>- 외부 물체와의 상호작용 시 해상도에 영향 받음</li> <li>- 입자성 액체 표현 시 별도 수작업 필요</li> </ul>     | <ul style="list-style-type: none"> <li>- 파티클의 효율적 관리 기술 필요</li> <li>- 부드러운 수면 표현 위해 다수의 파티클이 필요</li> </ul>   |

구해 온 곳으로 디지털콘텐츠연구단이 보유한 영상 콘텐츠 제작용 CG 기술과의 연계를 통해 시너지 효과를 기대하고 있다.

앞서 설명한 바 있는 파티클 기반의 유체 기술은 기존의 CG 분야에서 사용되어 온 그리드 기반의 기술과 비교해 볼 때 <표 1>과 같이 장단점을 가지고 있다.

현 상태에서 각각의 기술이 어느 것이 더 낫다고 판단하기는 어려우며 표현하고자 하는 장면의 특성에 따라 더 적합한 기술을 선택해 사용하는 것이 적절하리라 판단된다. 어느 정도 성숙 단계에 들어서 있는 그리드 기반의 시뮬레이션 개발에 비해 파티클 기반의 시뮬레이션 기술은 CG 분야에 많이 활용되지 않은 편이어서 지속적인 연구 개발이 이루어진다면 세계적인 수준의 결과를 낼 수 있으리라 본다.

본 과제에서는 상용화와 실제 프로덕션에의 직접 적용을 목표로 3차원 상용 소프트웨어인 Maya의 플러그인 형태로 개발되고 있으며 매년 아웃소싱을 이용한 시범 콘텐츠 제작을 통해 기술을 검증하고 기술의 적용 능력을 확대해 갈 계획이다.

또한 호주와의 공동연구는 영화 및 특수효과와 포스트 프로덕션 메카로 발돋움하고 있는 현지 산업

계와의 공조를 통해 기술 개발 이상의 효과를 가져다 줄 것으로 기대하고 있다.

## VI. 결론

본 논문에서는 유체 표현을 위한 컴퓨터 그래픽스 기술의 연구 동향에 대해 살펴보았다. 유체를 크게 바다와 같이 정적인 대규모 액체와 불, 연기, 폭포와 같은 동적인 유체로 구분을 지었고 각각에 대해 시간의 순서에 따라 기술 동향을 기술하였다.

정적인 대규모 액체를 구성하는 기술적 요소는 해수면의 모델링, 지면과의 거리에 따른 굴절, 파도의 부서짐 표현, 해저 및 해상에서의 렌더링 등이 있다. 오랜 기간의 연구 기간에도 불구하고 해수면을 표현하고자 하는 기술은 꾸준히 연구되어 왔으나 1980년대 후반이나 현재나 비슷한 경향을 보이고 있는데, 이는 영화 및 애니메이션을 제작하는 현장으로부터의 기술적 개선 요구가 기타 유체 표현 기술에 비해 적기 때문으로 보인다. 최근에는 전체적인 그래픽스 기술의 추세에 따라 게임 등 빠른 응답 시간을 요구하는 응용을 위해 실시간에 시뮬레이션으로 연구가 진행되고 있다. 이와 더불어 단순한 해수면의 모델링 및 시뮬레이션뿐만 아니라 그 안에서 동작하는 다른 객체와의 상호작용, 기후 등의 외부 요인에 의한 변화 등 다양한 효과 연출을 위한 방향으로 연구가 진행되어야 할 것이다.

동적인 액체 또는 불, 연기 등의 유체에 대한 연구는 1996년 Foster 등이 유체 시뮬레이션을 위해 삼차원 Navier-Stokes 식을 도입한 이래 물리적 사실성을 바탕으로 꾸준한 발전을 이루어 왔다. 이 분야의 특이할 점은 최근에 발표된 주요 연구결과물이 주로 Stanford 대학과 ILM[34], Alias[35], DreamWorks SKG[36]의 구성원에 의한 것이라는 점이다. 즉, 기술의 생산자인 연구자와 수요자인 영화 및 애니메이션 제작 업체들이 밀접한 연관을 갖고 프로젝트를 진행해 온 것이다. 이를 볼 때, 디지털 위주의 우리 나라 스튜디오도 연구 개발에 적

극 투자하거나 학계 및 연구계의 연구 결과를 적극 활용하고 연구자도 이들의 요구 사항을 연구 방향에 적용함으로써 서로 상승의 효과를 얻을 수 있을 것으로 본다.

## 약어 정리

|       |  |
|-------|--|
| CFD   | Computational Fluid Dynamics                                 |
| CG    | Computer Graphics  |
| CSIRO | Commonwealth Scientific and Industrial Research Organization |
| MPS   | Moving Particle Semi-Implicit                                |
| SPH   | Smoothed Particle Hydrodynamics                              |

## 참고 문헌

- [1] Shrek, <http://www.shrek.com>
- [2] D. Enright, S. Marschner, and R. Fedkiw, "Animation and Rendering of Complex Water Surfaces," in *Proc. of SIGGRAPH 2002*, July 2002, pp.736-744.
- [3] W.T. Reeves, "Particle System - A Technique for Modeling a Class of Fuzzy Objects," in *Proc. of SIGGRAPH '83*, July 1983, pp.359-376.
- [4] J.D. Anderson Jr., "Computational Fluid Dynamics: The Basic and Applications," McGraw-Hill, 1995.
- [5] A. Fournier and W. Reeves, "A Simple Model of Ocean Waves," in *Proc. of SIGGRAPH '86*, Aug. 1986, pp.75-84.
- [6] D. Peachey, "Modeling Waves and Surfs," in *Proc. of SIGGRAPH '86*, Aug. 1986, pp.65-74.
- [7] P. Ts'O and B. Barsky, "Modeling and Rendering Waves: Wavetracing Using Betasplines and Reflective and Refractive Texture Mapping," *ACM Transactions on Graphics*, Vol.6, No.3, 1987, pp.191-214.
- [8] M. Kass and G. Miller, "Rapid, Stable Fluid Dynamics for Computer Graphics," in *Proc. of SIGGRAPH '90*, 1990, pp.49-57.
- [9] J. Chen and N. Lobo, "Toward Interactive Simulation of Fluids with Moving Obstacles Using Navier-Stokes Equations," *Graphical Models and Image Proc.*, Vol.57, No.2, Mar. 1995, pp.107-116.

- [10] J.C. Gonzato and B.L. Saec, "On Modelling and Rendering Ocean Scenes," *The Journal of Visualization and Computer Animation*, Vol.11, No.1, 2000, pp.27-37.
- [11] S. Premoze and M. Ashikhmin, "Rendering Natural Waters," in *Proc. of Pacific Graphics 2000*, 2000, pp.23-30.
- [12] S. Thon, J. Dischler, and D. Ghazanfarpour, "Ocean Waves Synthesis Using a Spectrum-Based Turbulence Function," in *Proc. of the Int'l Conf. on Computer Graphics*, June 2000, p.65.
- [13] D. Hinsinger, F. Neyret, and M. Cani, "Interactive Animation of Ocean Waves," in *Proc. of the 2002 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, 2002, pp.161-166.
- [14] S. Jeschke, H. Birkholz, and H. Schumann, "A Procedural Model for Interactive Animation of Breaking Ocean Waves," in *Proc. of WSCG*, 2003.
- [15] J. Loviscach, "Complex Water Effects at Interactive Frame Rates," in *Proc. of WSCG*, 2003.
- [16] J. Stam and E. Fiume, "Turbulent Wind Fields for Gaseous Phenomena," in *Proc. of SIGGRAPH '93*, Aug. 1993, pp.369-376.
- [17] J. Stam and E. Fiume, "Depicting Fire and Other Gaseous Phenomena Using Diffusion Processes," in *Proc. of SIGGRAPH '95*, 1995, pp.129-136.
- [18] N. Foster and D. Metaxas, "Realistic Animation of Liquids," *Graphical Models and Image Proc.*, Vol.58, No.5, 1996, pp.471-483.
- [19] N. Foster and D. Metaxas, "Modeling the Motion of Hot, Turbulent Gas," in *Proc. of SIGGRAPH '97*, 1997, pp.181-188.
- [20] J. Stam, "Stable Fluids," in *Proc. of SIGGRAPH '99*, 1999, pp.121-128.
- [21] R. Fedkiw, J. Stam, and H.W. Jensen, "Visual Simulation of Smoke," in *Proc. of SIGGRAPH 2001*, 2001, pp.15-22.
- [22] N. Foster and R. Fedkiw, "Practical Animation of Liquids," in *Proc. of SIGGRAPH 2001*, 2001, pp.23-30.
- [23] D. Nguyen, R. Fedkiw, and H.W. Jensen, "Physically Based Modeling and Animation of Fire," in *Proc. of SIGGRAPH 2002*, 2002, pp.721-728.
- [24] S. Premoze, T. Tasdizen, J. Bigler, A. Lefohn, and R. Whitaker, "Particle-Based Simulation of Fluids," *Computer Graphics Forum*, Vol.22, No.3, 2003, pp.401-410.
- [25] M. Muller, D. Charypar, and M. Gross, "Particle-based Fluid Simulation for Interactive Applications," in *Proc. of the 2002 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, 2003, pp.154-159.
- [26] N. Rasmussen, D. Nguyen, W. Geiger, and R. Fedkiw, "Smoke Simulation for Large Scale Phenomena," in *Proc. of SIGGRAPH 2003*, 2003, pp.703-707.
- [27] Frank Losasso, Frederic Gibou, and Ron Fedkiw, "Simulating Water and Smoke with an Octree Data Structure," in *Proc. of SIGGRAPH 2004*, 2004.
- [28] J. Stam, "A Simple Fluid Solver Based on the FFT," *Journal of Graphics Tools*, No.2, 2001, pp.43-52.
- [29] J. Stam, "Real-Time Fluid Dynamics for Games," in *Proc. of the Game Developer Conf.*, Mar. 2003.
- [30] A. Treuille, A. McNamara, Z. Popovic, and J. Stam, "Keyframe Control of Smoke Simulations," in *Proc. of SIGGRAPH 2003*, July 2003, pp.716-723.
- [31] Raanan Fattal and Dani Lischinski, "Target Driven Smoke Animation," in *Proc. of SIGGRAPH 2004*, 2004.
- [32] Antoine McNamara, Adrien Treuille, Zoran Popovic, and Jos Stam, "Fluid Control Using the Adjoint Method," in *Proc. of SIGGRAPH 2004*, 2004.
- [33] N. Rasmussen, D. Enright, D. Nguyen, S. Marino, N. Sumner, W. Geiger, S. Hoon, and R. Fedkiw, "Directable Photorealistic Liquids," in *Proc. of Symp. on Computer Animation(SCA) 2004*, 2004.
- [34] Industrial Light and Magic, <http://www.ilm.com>
- [35] Alias, <http://www.alias.com>
- [36] Dreamworks SKG, <http://www.pdi.com>