점 렌더링 기법을 사용한 입자 기반 물 데이터의 렌더링*

이재학[1], 차득현[2], 장병준[3], 임인성[4], 김장희[5], 구본기[6]
서강대학교 컴퓨터학과[1234]
한국전자통신연구원 디지털콘텐츠연구단[56]
{rrhak0000[1],seaboy7[2],jbj[3]}@grmanet.sogang.ac.kr
ihm[4]@sogang.ac.kr
{rolldice[5],bkkoo[6]}@etri.re.kr

Rendering of Particle-Based Water Data Using Point Rendering Method

Jaehak Lee[1], Deukhyun Cha[2], Byungjoon Chang[3], Insung Ihm[4],
Janghee Kim[5], Bonki Koo[6]
Dept. of Computer Science, Sogang University[1234]
Electronics and Telecommunications Research Institute[56]

요 약

사실적인 물 애니메이션을 위한 격자 기반 시뮬레이션 기법은 자연스러운 물의 움직임뿐만 아니라 부드러운 물의 표면을 잘 표현해주는 장점이 있다. 이러한 격자 기반 방법과 함께 상대적으로 적은 계산으로 안정적인 결과를 산출해주는 입자 기반의 액체 시뮬레이션 기법이 최근 애니메이션 분야에 적용되기 시작했고, 그로 인하여 입자로 이루어진 시뮬레이션 데이터에 특화된 효과적인 렌더링 기술의 개발이 요구되고 있다. 본 논문에서는 주로 3 차원 스캔 데이터와 같이 물체 표면을 샘플링 하여 얻어진 점 집합에 대한 렌더링 기법을 확장하여, 위상 변화가 크고 점 집합에 의해 내부까지 표현되는 물 데이터의 특성에 적합한 렌더링 기법을 제안한다. 본 기법에서는 시뮬레이션을 통하여 얻은 입자 데이터로부터 물의 표면을 표현해주는 새로운 점 집합을 생성하고, 시뮬레이션 된 데이터의 특성을 잘 반영하도록 각 점에 대한 법선 벡터와 반지름을 결정한다. 특히 가공된 점 집합 데이터에 대하여 확장된 점 집합 렌더링 기법을 적용함으로써 입자 데이터가 표현해주는 세밀한 부분들을 보존하면서, 부드러운 물의 표면을 가시화할 수 있도록 하였다.

Keywords: Liquid simulation, particle, point rendering, surface extraction, smoothing.

*본 연구는 2005 년도 서강대학교 교내 연구비 지원에 의하여 이루어졌음.

1. 서론

최근 컴퓨터 그래픽스 분야에서는 물이나 연기, 불과 같은 자연 현상을 사실적으로 표현하기 위한 시뮬레이션 기법에 대한 연구가 활발하게 진행되고 있다. 특히 유체 역학 분야에서 널리 사용되어 오던 대표적 방정식인 Navier-Stokes 방정식을 이용하여 유체의 흐름을 수치적으로 계산 할 수 있는 다양한 방법들이 제시되고 있다. 이러한 수치 엔진을 구현 함으로서 사실적인 유체의 움직임에 대한 정보의 획득을 자동화 할 수 있다.

이때 기체와 같은 유체와는 달리 액체는 그 표면의 경계가 명확하기 때문에 표면을 찾아내는 방법이 정교하지 못하면 사실적인 액체의 형태를 얻어 내기가 힘들다. 이 문제를 해결하기 위한 방법 중 하나가 등위 집합 기법(level set method)이다. 등위 집합 기법은 부호 거리(signed distance)함수를 사용해 3 차원 공간에 정의된 이산화된 격자에서 물체의 표면으로부터 떨어져 있는 거리를 저장하고 이 값들이 시간에 따라 어떻게 변화하는지를 얻는 방법이다[1,2]. 이 등위집합 방법을 사용하면 부드러운 액체의 표면을 매우 잘 표현할 수 있다.

하지만 이러한 방법은 3 차원에서 정의된 격자에서 속도와 등위 집합에 대한 계산이 이루어져야 하므로 해상도에 따르는 계산 시간의 증가가 크다. 뿐만 아니라 등위 집합에 대한 계산이 진행됨에 따라 오차가 누적이 되어 처음에 가지고 있던 액체의 체적을 지속적으로 잃어버리는 단점이 있다. 또한 액체의 표면이 가지고 있는 상세함을 표현하기 위해서는 높은 해상도의 시뮬레이션이 필요하게 되고 이는 많은 계산 시간과 큰 데이터를 저장하기 위한 자원을 요하게 된다. 등위 집합 기법을 사용하였을 때 액체의 체적을 잃어 버리는 문제를 해결하기 위해 제안된 것이 파티클 등위 집합 기법이다[3]. 이 방법은 기존의 등위 집합 기법이 가지고 있는 단점을 파티클을 사용해 극복하는 방법이다. 즉, 등위 집합 기법을 사용할 때 잃어버리는 체적과 상세함을 입자들이 표현하고 있는 정보를 이용해 정확한 표면을 복원하는 방법이다. 이를 통해 등위 집합 자체의 문제점은 해결 되었지만 상세한 유체를 표현하기 위한 계산 시간과 메모리에 대한 문제는 적절하게 해결되지 못했다.

최근 3 차원 격자상에서 파티클 등위 집합 기법을 이용한 액체 시뮬레이션 기법에 반하여 상대적으로 적은 계산량과 함께 안정적인 결과를 얻을 수 있는 파티클 기반의 액체 시뮬레이션에 관한 연구가 활발히 진행 되고 있다[4,5,6]. 파티클 기반의 액체 시뮬레이션은 렌더링 시에 파티클 데이터만으로는 액체의 표면을 부드럽게 표현하는 것이 어려운 단점이 있다. 이런 단점이 렌더링 단계에서 해결 될 경우 파티클 기반의 액체 시뮬레이션의 결과들이 더 많은 응용에 사용되어 질 수 있으며 고품질의 실시간 액체 렌더링도 가능해 질 것이다.

본 논문에서 제안한 렌더링 기법에서는 SPH(Smoothed Particle Hydrodynamcis) 기법[7,8,9]에 의해 시뮬레이션 된 파티클 데이터로부터 보다 정확한 액체의 표면을 찾아내기 위해, 파티클 정보로부터 밀도장을 생성하고 마칭 큐브 알고리즘을 이용하여 액체의 표면을 표현해주는 등가면에 대하여 다면체 모델을 생성한다. 그리고 액체의 표면에서 벗어난 파티클들과 같이 곡률이 높은 부분에 대한 형태를 유지하면서 부드러운 표면을 렌더링하기 위하여 3 차원 스캔 데이터에 대한 렌더링 방법인 점 렌더링 알고리즘을 액체 데이터에 적합하도록 확장 및 변형하였다. 이러한 본 논문의 기법을 사용하여 본래의 상세함 잃어 버리지 않고 부드러운 액체의 표면을 얻을 수 있다.

2. 액체 표면의 다면체 모델화

SPH 역학 기법을 통하여 산출한 파티클 데이터는 그림 1 과 같다. 여기서 관찰할 수 있듯이 이 데이터에는 단순히 물 표면뿐만 아니라 물 내부에 대해서도 파티클들이 존재하며, 시뮬레이션 데이터를 렌더링 하는데 있어

우선적으로 해결해야 하는 문제점 중의 하나는 물 표면에 대한 정보를 추출하는 것이다.

본 논문에서는 SPH 기법에 대한 응용 연구 중 하나인 [5]의 연구에서 제안한 밀도장 생성 방법을 이용하여 파티클 데이터에 대한 밀도장을 생성하고 이 밀도장으로부터 마칭 큐브 알고리즘[13]을 이용하여 액체의 표면의 등가면에 대한 다면체 모델을 생성한다. 생성된 다면체 모델로부터 점 집합 만으로는 얻기 힘든 표면에 대한 정보들을 얻을 수 있다.

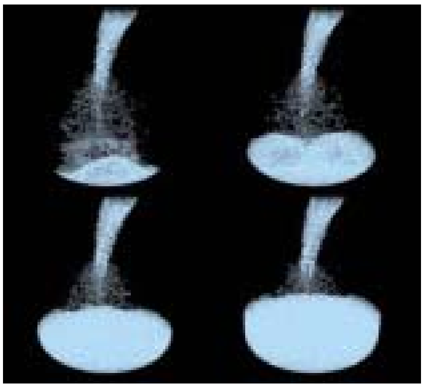

그림 1. SPH 기법에 의해 산출된 액체 시뮬레이션 데이터

SPH 기법에 따르면 공간 상의 한 지점 r 에 대한 스칼라량(scalar quantity) A 는 식(1)과 같이 공간상의 모든 각 각의 파티클의 질량(m_j)과 위치(r_j), 밀도(ρ_j)와 스칼라량(A_j)의 가중합을 보간함으로써 계산된다.

$$A_s(r) = \sum_j m_j \frac{A_j}{\rho_j} W(r - r_j, h) \qquad (1)$$

함수 $W(r,h)$ 는 r 의 값이 사용자가 설정하는 h 라는 반경 이내에 들어올 경우 적절한 가중치를 부여하기 위한 함수로서 식(2)와 같다.

$$W(r,h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3, & 0 \le r \le h \\ 0, & otherwise \end{cases} \qquad (2)$$

각 파티클 j 의 부피는 $V_j = \frac{m_j}{\rho_j}$ 이고 질량인 m_j 는 모든 파티클에 대해 동일한 상수로서 사용되어 진다. 따라서 한 공간상의 지점 r 에서의 밀도는 식(3)과 같다.

$$\rho_s(r) = \sum_j m_j W(r - r_j, h) \qquad (3)$$

밀도장으로부터 마칭 큐브 알고리즘과 같은 등가면 추출법을 사용하여 이 등가면을 삼각형의 메쉬 데이터로 구성한 후 광선 추적법을 수행한 결과는 그림 2와 3과 같다.

그림 2. (좌)파티클 데이터, (우)다면체 모델

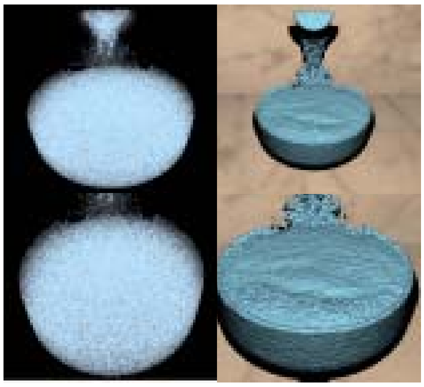

그림 3. (좌)파티클 데이터, (우)다면체 모델

이 두 그림에서와 같이 다면체 모델은 파티클 데이터의 표면을 원형에 가깝게 복원하지만 표면을 부드럽게 표현하지 못한다는 문제점이 있다.

3. 점 집합 렌더링 기법의 적용

3-1. 점 집합 렌더링 기법

점 집합 렌더링 기법은 3 차원 스캔 데이터에 대한 렌더링 기법으로서 물체의 표면에 대한 점

데이터를 이용하여 물체를 렌더링 하는 기법으로서 물체를 표현하는데 있어 좋은 결과를 도출해내고 있다[10]. 본 논문에서는 이러한 점 집합 렌더링 기법을 액체 시뮬레이션 데이터를 렌더링 하는데 있어서 적용하였다.

점 집합 렌더링 기법은 점의 위치(p_i), 법선 벡터(n_i), 반지름(r_i)의 정보를 이용해서 각 점의 반경에 대한 가중 함수를 정의하고 이렇게 정의된 가중 함수를 이용해 음함수를 정의한다. 가중 함수는 식(4)와 같다.

$$w_i(x) = W\left(\frac{\|x - p_i\|}{r_i}\right) \quad (4)$$

식(4)에서 $W(r)$ 은 감소 가중 함수로서 식(5)와 같다.

$$W(r) = \begin{cases} 1 - r, & r < 1 \\ 0, & r \geq 1 \end{cases} \quad (5)$$

공간 상의 한 지점 x 에 대해 이 x 를 둘러싼 파티클들의 위치와 법선 벡터에 대한 가중 평균값은 식(6)과 (7)과 같이 정의 된다.

$$\overline{p}(x) = \frac{\sum w_i(x) p_i}{\sum w_i(x)} \quad (6)$$

$$\overline{n}(x) = \frac{\sum w_i(x) n_i}{\sum w_i(x)} \quad (7)$$

파티클들의 위치에 대한 가중 평균 값인 $\overline{p}$ 와 파티클들의 법선 벡터에 대한 가중 평균 값인 $\overline{n}$ 에 의해서 지역적인 한 평면이 결정되고 이를 통해 음함수가 식 (8)과 같이 정의 된다.

$$f(x) = (x - \overline{p}(x)) \cdot \overline{n}(x) \quad (8)$$

광선 추적법에서 광선을 $R(t)$ 라 하였을 때 식 (8)의 x 에 $R(t)$ 를 대입 하면 광선과 지역적인 평면과의 부호 거리(signed distance)가 결정된다. 이 광선 $R(t)$ 를 k 만큼 분할하여 $t_0, t_1, ..., t_k$ 에 대한 지역적인 평면과의 부호 거리를 계산하는 경우에 $sign(f(t_i)) \neq sign(f(t_{i+1}))$ 인 결과가 발생하면 t_i 와 t_{i+1} 사이에 광선과 지역적인 평면의 교점이 존재하는 것이다. 따라서 $f(t_i)$ 와 $f(t_{i+1})$ 로서 t_i 와 t_{i+1} 를 선형 보간 하여 광선과 지역적인 평면의 교점을 계산하고 퐁 쉐이딩 기법을 통하여 교점에서의 색깔을 계산한다.

3-2. 점 집합 렌더링을 위한 기하 정보 추출

점 집합 렌더링 기법은 점의 위치, 법선 벡터, 반지름의 정보를 이용하는 렌더링 기법이다. 따라서 유연한 파티클 유체 역학 기법에 의해 산출된 액체 시뮬레이션 데이터에 대한 액체 표면의 점의 위치, 법선 벡터, 반지름의 기하 정보 추출이 필요하다. 액체 표면의 점의 위치는 2 절에서 제안한 액체 표면의 다면체 모델로부터 추출하고 법선 벡터는 Max 등에 의해 소개된 방법으로 추출된 점이 속한 면들 중에서 보다 작은 면에 대해 더 큰 가중치를 할당해 주어 주변에 연결되어 있는 면들의 크기 차이를 고려하여 법선 벡터를 구하는 방법을 이용한다[11]. 파티클의 반지름을 설정하는 방법은 Wu 와 Kobbelt 가 제안한 최소 자승 평면(least square plane)알고리즘을 이용한다[12]. 최소 자승 평면 알고리즘은 점 p_i 의 반지름을 추정하기 위해 점 p_i 를 지나고 점 p_i 의 법선 벡터인 n_i 를 법선 벡터로서 갖는 평면(L_i)을 정의한다. 이렇게 정의된 평면에 점 p_i 의 주위의 점들 $p_j (1 \leq j \leq n)$ 을 투영 시킨 후 식 (9)에 의해 투영된 거리(h_j)가 정의되고 이 값이 사용자가 정의한 ε 값 이상일 경우 식 (10)에 의해 점 p_i 의 반지름(r_i)이 결정 된다.

$$h_j = n_i^T (p_j - p_i) \quad (9)$$

$$r_i = \left\| (p_j - p_i) - n_i^T (p_j - p_i) n_i \right\| \quad (10)$$

3-3. 점 집합 렌더링 수행 결과

그림 4 는 2 절의 다면체 모델과 최소 자승 평면 알고리즘의 ε 의 값이 0.5 일 때 점 집합 렌더링을 수행한 결과를 비교한 것으로서 파티클 데이터의 원형을 보존하면서 액체의 표면이 부드럽

게 표현되는 것을 볼 수 있다.

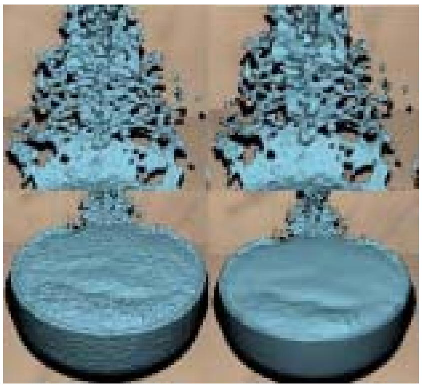

그림 4. (좌)다면체 모델, (우)점 집합 렌더링

4. 최소 자승 평면 알고리즘의 확장

4-1. 최소 자승 평면 알고리즘의 문제점

그림 5 에서처럼 액체 시뮬레이션 데이터에서의 파티클의 형태 및 분포가 복잡한 경우 최소 자승 평면 알고리즘을 이용하여 파티클의 반지름을 생성하고 점 집합 렌더링 알고리즘을 수행하면 그림 6 과 같이 물방울과 같이 표현되어야 할 파티클 데이터가 잘못 표현되는 경우가 발생한다.

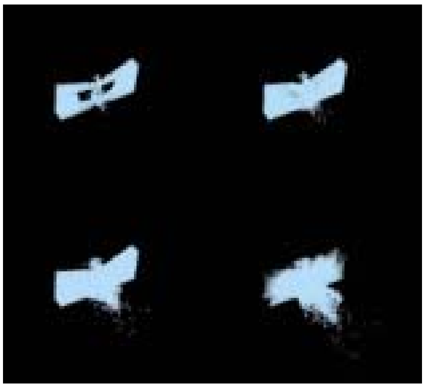

그림 5. 파티클의 형태 및 분포가 복잡한 물 데이터

이런 문제가 발생하는 원인은 최소 자승 평면 알고리즘이 3 차원 스캔 데이터를 기준으로 파티클의 반지름을 설정하는 알고리즘인데 반해 액체 시뮬레이션 데이터의 형태 및 분포가 3 차원 스캔 데이터가 가지는 노이즈(noise)에 비해 복잡하기 때문이다.

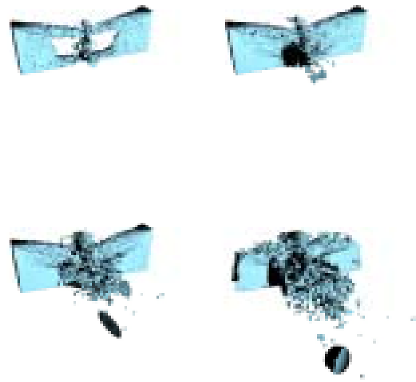

그림 6. 최소 자승 평면 알고리즘의 문제점

4-2. 최소 자승 평면 알고리즘의 확장

최소 자승 평면 알고리즘은 작은 액체 덩어리로서 물방울과 같은 모습으로 표현되어야 하는 부분에 있어서 적합하지 않은 결과를 도출한다. 따라서 본 논문에서는 최소 자승 평면 알고리즘의 확장을 통해 이러한 문제를 해결하는 방법을 제안한다. 그림 7 과 같이 파티클이 조밀하게 분포 되어 있지 않을 때 파티클 p_i 의 반지름을 계산하기 위해 p_i 의 법선 벡터를 법선 벡터로 갖는 평면에 주위의 점들 p_j 를 투영 시킬 때 부가적인 조건을 부여한다. 즉, p_i 의 주위의 점들의 거리가 ε' 보다 큰 경우 p_i 의 반지름은 p_i 의 곡률을 이용하여 결정한다.

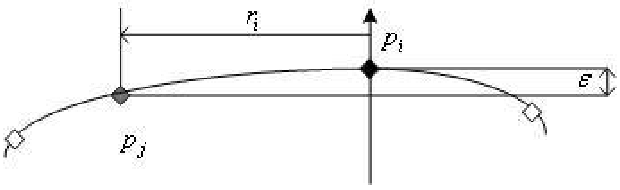

그림 7. 파티클 데이터가 조밀하게 분포되지 않은 경우

곡률은 평면 위에 어떤 곡선이 주어졌을 때 그 곡선의 굽은 정도를 의미하며 곡면에 대해서는 그 곡면 위에 놓인 곡선들의 곡률을 이용하여 곡면의 휘어짐 정도를 나타낸다. 곡선의 곡률이란,

곡면 위의 한 점 p 를 지나면서 점 p 에서의 법선 벡터를 포함하는 평면이 주어진 곡면과 만나 하나의 평면 곡선을 이루는데, 이러한 곡선을 곡면의 곡률이라고 하며 곡선의 곡률은 단위 길이당 곡선에 접한 접선의 방향이 바뀌는 비율로서 정의된다. 이 때, 곡선에 접하는 원이 접평면의 어느 쪽에 있느냐에 따라서 그 부호가 정해진다. 그림 8 에서와 같이 곡선의 길이가 x 인 곡선이 반지름이 r 인 원의 원둘레 중 일부라고 한다면 곡선 양쪽 끝의 접선의 사이 각은 θ 이고 곡선의 길이 x 는 $x = r\theta$ 이므로, 이 곡선의 곡률은 식 (11)과 같다.

$$\frac{\theta}{x} = \frac{1}{r} \tag{11}$$

즉, 곡선을 만드는 원의 반지름의 역수가 바로 그 곡선의 곡률과 같으며 이러한 반지름을 곡률 반경이라 한다. 그리고 곡선 곡률들 중에는 최대 곡률을 갖는 곡선과 최소 곡률을 갖는 곡선이 있는데 최대, 최소인 두 곡률을 주곡률(principal curvature)이 라 한다. 그리고, 주곡률의 곱을 가우스 곡률 또는 전곡률(total curvature)이라 하며, 주곡률의 평균값을 평균 곡률이라고 한다. 곡면 상에서의 무한히 많은 곡률을 하나의 숫자로 표현하는 접근 방법 중에서 가장 많이 사용되는 방법이 가우스 곡률(Gaussian curvature)과 평균 곡률(mean curvature)이다.

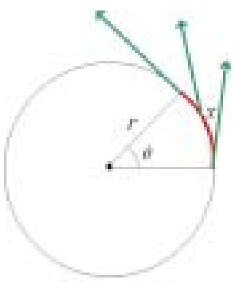

그림 8. 곡선의 곡률

곡률을 이용하여 파티클의 반지름을 결정하기 위해서 본 논문에서는 최대 곡률(K_1)과 최소 곡률(K_2)의 절대값의 기하 평균을 이용한다. 기하 평균은 n 개의 양수 $a_1, a_2, ..., a_n$ 이 있을 때, 이들 수의 n 제곱근이다. 따라서 최대 곡률과 최소 곡률의 기하 평균은 $\sqrt{|K_1||K_2|}$ 이고 이 기하 평균의 역수는 곡률 반경을 의미하며 식 (12)와 같다. 이 식을 통해서 파티클의 반지름을 결정한다. 이때 식 (10)에서 $|K_1K_2|$ 의 값은 최대 곡률과 최소 곡률의 곱으로서 즉, 가우스 곡률로서 표현된다.

$$r = \frac{1}{\sqrt{|K_1||K_2|}} = \frac{1}{\sqrt{|K_1K_2|}} \tag{12}$$

가우스 곡률은 2 절의 다면체 모델로부터 계산된다. 한 점의 가우스 곡률을 구하는 방법은 점에 인접한 삼각형들의 면적과 점과 각각의 삼각형들이 이루는 각에 의해서 정의된다. 다면체 내부의 한 점에 대한 가우스 곡률은 식 (13)과 같고 다면체 경계선 위의 한 점에 대한 가우스 곡률은 식(14)와 같다.

$$K_g = \frac{2\pi - \sum_{i=1}^{n} \alpha_i}{\frac{1}{3}A} \tag{13}$$

$$K_g = \frac{\pi - \sum_{i=1}^{n} \alpha_i}{\frac{1}{3}A} \tag{14}$$

A 는 곡률을 구하고자 하는 점에 인접한 삼각형들의 면적의 합으로서 삼각형의 면적이 T_i 라고 할 때 $A = \sum_{i=1}^{n} T_i$ 이고 α_i 는 삼각형의 인접한 변 사이의 각을 나타낸다. 이렇게 파티클의 가우스 곡률($(K_g)_i$)을 계산하여 식 (12)에 대입하면 파티클의 반지름은 식 (15)에 의해 결정되며 식 (10)은 식(16)와 같이 변형된다.

$$r_i = \frac{1}{\sqrt{|(K_g)_i|}} \tag{15}$$

$$r_i = \begin{cases} \left\| (p_j - p_i) - n_i^T (p_j - p_i) \cdot n_i \right\|, & if\ |p_j - p_i| < \varepsilon' \\ \dfrac{1}{\sqrt{|(K_g)_i|}}, & otherwise \end{cases} \tag{16}$$

최소 자승 평면 알고리즘의 확장을 통한 점 집합 렌더링 결과는 그림 9 와 같다.

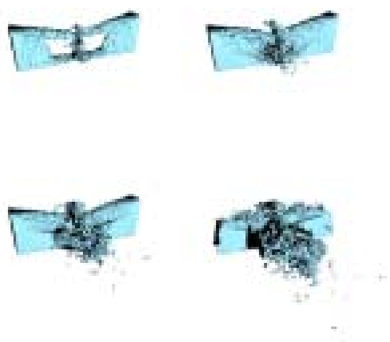

그림 9. 최소 자승 평면 알고리즘 확장 결과

5. 점 집합 렌더링 알고리즘의 변형

5-1. 점 집합 렌더링 알고리즘의 문제점

최소 자승 평면 알고리즘을 이용하여 파티클의 반지름 생성시 액체의 표면을 부드럽게 표현하기 위하여 ε 의 값을 증가시켜 반지름을 크게 설정하면 그림 10 과 같은 액체 데이터의 경우 문제가 발생한다.

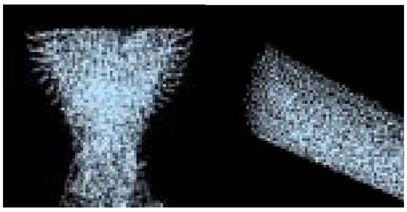

그림 10. 점 집합 렌더링 알고리즘 수행시 문제가 발생하는 액체 데이터

그림 11 은 최소 자승 평면 알고리즘을 사용하여 ε 의 값이 0.1, 0.3, 0.5 일 때 반지름을 생성한 이후 점 집합 렌더링을 수행한 결과로서 ε 의 값이 증가할수록 검은 부분이 생기는 문제가 발생한다. 이러한 문제가 발생하는 이유는 파티클의 반지름이 증가 할수록 광선과 물체와의 교점 계산시 식(4)에 따라서 그 파티클의 반지름이 작을 경우 $w_i(x)$ 의 값이 0 이 되지만 파티클의 반지름이 큰 경우 0 이 아닌 값을 갖게 되어 광선과 물체의 교점 계산에 영향을 미치게 된다. 이런 경우 원래 광원에서 보이지 않는 파티클이 교점 계산시에 반영이 되어 법선 벡터의 가중 평균 값이 광원에서 보이지 않는 법선 벡터의 값에 가까워져 색깔 계산이 잘못 되어 검은 부분이 생기게 된다.

그림 11. (좌) 0.1 (중) 0.3 (우) 0.5

5-2. 점 집합 렌더링 알고리즘의 변형

그림 11 과 같이 ε 의 값이 커짐에 따라 검은 부분이 생기는 문제를 해결하기 위해 ε 의 값을 작게 하여 파티클의 반지름을 생성 하면 액체의 부드러운 표면을 얻을 수 없다. 따라서 본 논문에서는 점 집합 렌더링 알고리즘을 변형하여 ε 의 값이 커짐에 따라 잘못된 파티클을 포함하는 문제를 해결하는 기법을 제안한다.

임의의 파티클이 광선과 물체와의 교점 계산시에 반영이 될 경우 그 파티클이 광원에서 보이는 파티클 인지 아닌지를 결정한다. 이는 한 파티클에서 광원까지의 벡터(l_i)와 한 파티클의 법선 벡터(n_i)와의 내적을 통해서 이 값이 음수이면 광원에서 보이지 않는 파티클이며 양수이면 광원에서 보이는 파티클이 된다. 단, l_i 벡터와 n_i 벡터는 정규화 되어 있다고 가정한다. 식(4)를 식(17)과 같이 변형하여 $l_i \cdot n_i$ 의 값에 따라 파티클이 반영되는 가중치에 0 에서 1 사이의 α 값을 곱하여 광원에서 보이지 않는 파티클의 위치와 법선 벡터의 가중값을 줄여 색깔이 잘못되어 검은 부분이 생기는 문제를 해결한다.

$$w_i(x) = \begin{cases} W(\frac{\|x-p_i\|}{r_i}), & if\ n_i \cdot l_i \ge 0 \\ W(\frac{\|x-p_i\|}{r_i}) \times \alpha, & if\ n_i \cdot l_i < 0 \end{cases} \quad (17)$$

그림 12 는 ε 의 값이 0.5 일 때 α 값에 따른 점 집합 렌더링 알고리즘의 변형 결과이다.

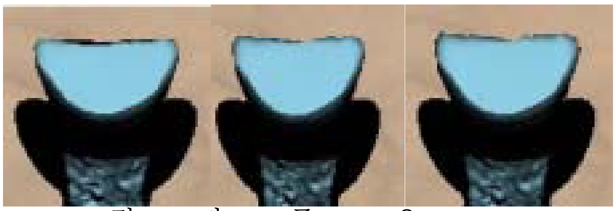

그림 12. (좌) 0.1 (중) 0.01 (우) 0.001

6. 실험 결과

본 논문에서 제안한 기법이 파티클 기반의 액체 시뮬레이션 데이터를 렌더링 하는데 얼마나 효과적으로 동작하는지를 보여주기 위하여 기본적인 광선 추적법을 기반으로 본 논문에서 제안한 렌더링 기법을 적용한 렌더러를 사용하여 이미지를 생성하였다. 그림 13 은 컵에 물을 따르는 장면에 대한 파티클 데이터에 대한 렌더링 결과이다. 전체 195 프레임의 데이터에 대해여 512×512 해상도의 렌더링을 수행하였다. 파티클 정보가 가지고 있는 세밀함을 잘 보존하면서 부드러운 물의 표면을 찾고 있으므로 빛의 반사와 굴절이 깨끗하게 잘 일어나는 것을 볼 수 있다.

양쪽 복도에서 물이 쏟아져 나오는 장면에 대한 시뮬레이션 결과인 그림 14 에서도 복잡하게 튀어나가는 물의 형태가 보존 되면서 부드러운 물의 표면이 렌더링 되는 것을 확인 할 수 있다. 본 논문에서 제안한 기존 렌더링 기법들의 확장을 통해 파티클로 이루어진 물 데이터에 적합하게 적용되어 기존의 문제점들이 해결 되었음을 알 수 있다. 역시 부드러운 표면으로부터 투명한 물이 깨끗하게 렌더링 됨을 볼 수 있다.

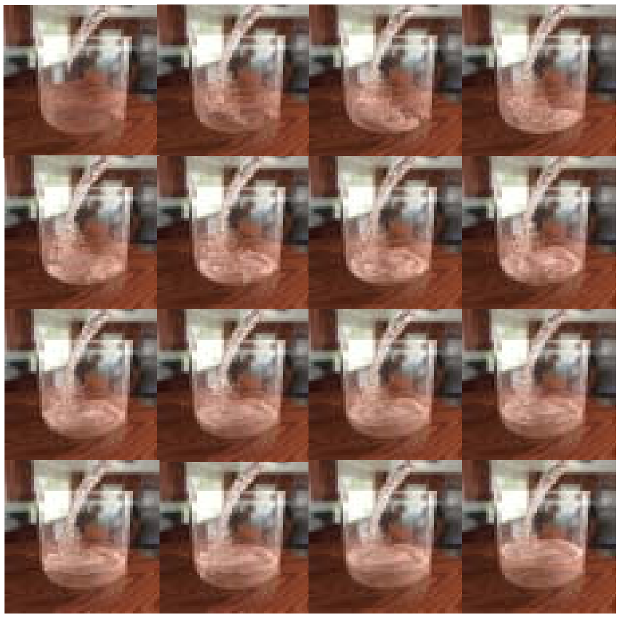

그림 13. 컵에 물을 따르는 장면

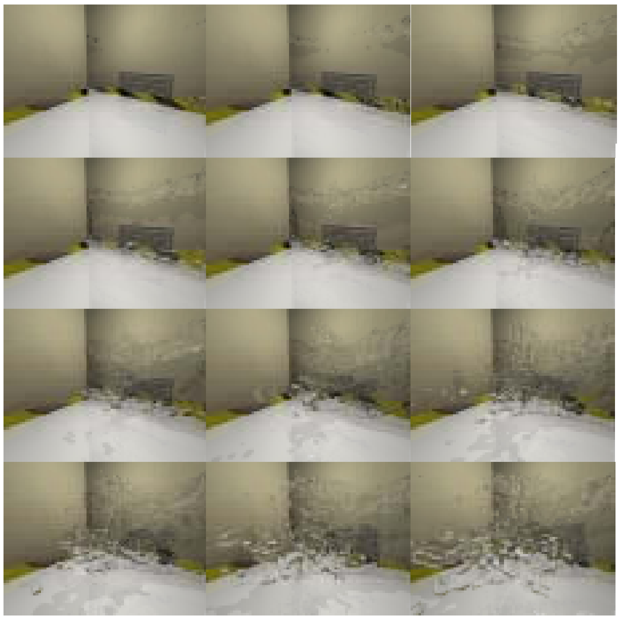

그림 14. 물이 복도를 지나는 장면

7. 결론

본 논문에서는 유연한 파티클 유체 역학 기법에 의한 액체 시뮬레이션 데이터로부터 사실적인 액체를 렌더링하기 위한 기법을 제안하였다. 파티클로 시뮬레이션 된 데이터는 액체의 내부를 채우고 있어 사실적인 렌더링을 위해서 파티클 정보로부터 정확한 액체의 표면을 찾아내고 액체의 복잡한 모양을 유지하면서 부드러운 표면을 표현하기 위해 점 렌더링 기법을

변형 및 확장하였다. 그리고 몇 가지 파티클 기반의 액체 시뮬레이션 데이터를 렌더링 하는데 적용해 봄으로써 본 논문에서 제안한 렌더링 기법이 효율적으로 작용 할 수 있음을 볼 수 있었다.

본 논문에서 제안한 점 집합 렌더링에 필요한 기하 정보를 만드는데 있어서 많은 변수가 존재하고 이러한 변수는 액체 시뮬레이터에 의해 산출되는 파티클 데이터에 모두 동일하게 적용 될 수 없다. 즉, 실험을 통해서 적절한 변수를 설정함으로써 원하는 결과를 도출해내야 하는 어려움이 존재한다. 향후 이러한 변수에 대한 설정을 자동화하는 지속적인 연구가 필요하다. 또한 점 집합 렌더링 알고리즘이 3 차원 스캔 데이터에 기반하였기 때문에 파티클 기반의 액체 시뮬레이션 데이터에 대한 렌더링을 수행하는데 있어서 많은 시간이 걸린다. 따라서 액체와 같은 유체를 렌더링 하는데 있어서 보다 최적화된 자료 구조가 필요하다.

참고문헌

[1] J. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 2nd edition, 1999.

[2] S. Osher and R. Fedkiw. *The Level Set Method and Dynamics Implicit Surfaces*. *Springer-Verlag*, 2002

[3] 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, pages 82-116, 2002.

[4] S. Premoze, T. Tasdizen, J. Bigler, A. Lefohn and R. Whitaker. Particle based simulation of fluids. *In Proc. of Eurograhpics 2003*, page 401-410, 2003.

[5] M. Mller, D. Charypar and M. Gross. Particle-based fluid simulation for interactive applications. *In Proc. of the Eurographics/ACM SIGGRAPH Symposium on Comptuter Animation 2003*, pages 154-159, 2003.

[6] N. Bell, Y. Yu and P.J. Mucha. Particle-based simulation of granular materials. *In Proc. of ACM SIGGRAPH 2005*, pages 956-972, 2005.

[7] M. Desbrun and M.P. Cani. Smoothed particles: A new paradigm for animating highly deformable bodies. *In Computer Animation and Simulation '96(Proceedings of EG Workshop on Animation and Simulation)*, pages 61-76. Springer-Verlag, Aug 1996.

[8] L.B. Lucy. A numerical approach to the testing of fission hypothesis. *The Astronomical Journal*, vol. 82, pages 1013-1024, 1997.

[9] R.A. Gingold and J.J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, vol. 181, pages 375-398, 1997.

[10] I. Wald and H. Seidel. Interactive ray tracing of point-based models. *In Proc. of Eurographics Symposium on Point-Based Graphics*, pages 1-8, 2005.

[11] N. Max. Weight for computing vertex normals from facet normals. *Journal of Graphics Tools*, vol. 4, no. 2, pages 1-6, 1999.

[12] J. Wu and L. Kobbelt. Optimized sub-sampling of point sets for surface splatting. *In Proc. of Eurographics 2004*, pages 643-652, 2004.

[13] W. Lorensen and H. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *In Proc. of ACM SIGGRAPH 1987*, pages 163-169, 1987.