

GPU 최적화를 이용한 물리 기반 옷감과 액체의 상호작용

문성혁^o, 김종현^{*}

^o강남대학교 소프트웨어응용학부,

^{*}강남대학교 소프트웨어응용학부

e-mail: jonghyunkim@kangnam.ac.kr

Physics-Based Cloth and Liquid Interaction using GPU Optimization

Seong-Hyeok Moon^o, Jong-Hyun Kim^{*}

^oSchool of Software Application, Kangnam University,

^{*}School of Software Application, Kangnam University

● 요약 ●

본 논문에서는 물리 기반 옷감 시뮬레이션과 SPH(Smoothed particle hydrodynamics) 기반의 유체 시뮬레이션 간의 상호작용에서 표현되는 다양한 물리적 효과를 GPU 기반으로 빠르게 표현할 수 있는 프레임워크를 제안한다. 기존 기법과는 다르게 수치적 안정성을 개선하기 위해 CCD(Continuous collision detection)를 활용하였으며, 모든 연산이 GPU에서 동작하기 때문에 매우 빠르게 옷감과 유체의 상호작용 장면인 다공성 재질, 기공 흐름, 흡수, 방사, 확산을 모델링할 수 있다.

키워드: 유체 시뮬레이션(Fluid simulation), 옷감 시뮬레이션(Cloth simulation), 상호작용(Interaction), GPU(Graphics Processing Unit), 다공성 재질(Porosity material), 기공 흐름(Pore flow), 흡수(Absorption), 방사(Emission), 확산(Diffusion)

I. Introduction

본 논문에서는 물리 기반 유체 시뮬레이션과 옷감의 상호작용에 의해 표현되는 다양한 물리적 특성을 GPU 프레임워크에서 온전히 표현할 수 있는 프레임워크를 제안한다. 특정 현상만을 모델링할 수 있는 기존 기법과는 다르게 본 논문에서 제안하는 방법은 유체와 옷감의 상호작용 시 표현되는 다양한 물리적 특성을 하나의 GPU 프레임워크에서 표현할 수 있는 통합형 시뮬레이션 기법이며, 사용자는 이를 통해 빠르게 3차원 장면을 모델링 및 제작할 수 있다.

2.2 Notation

자세한 내용을 설명하기 앞서 표기법을 설명하려 한다. 옷감의 node는 n , SPH 파티클은 f , 다공성 파티클은 p 로 표기한다. V_i 는 바운더리 파티클일 경우에는 [3]Akinci가 제안한 방법으로 계산된 파티클 부피이고 SPH 파티클일 경우에는 m_i/ρ_i 를 의미한다.

ϕ_i 는 다공성 물질 내 고체의 공극률이고 s_i 는 다공성 물질 내 유체 중 액체의 공극률이다. 남은 표기법은 Table. 1에서 나와있다.

II. The Proposed Scheme

2.1 Boundary particle sampling

옷감과 강체의 파티클 샘플링과 SPH 파티클과의 상호작용은 [2]에서 제안된 기법으로 구현했다. 옷감에 샘플링 된 파티클은 다공성 파티클(Pore particle)이라 부르겠다.

Table 1. Summary of notation used in this paper.

Symbol	Location	Meaning
h_i	particle	support radius
r_i	particle	radius
x_i	-	position
v_i	-	velocity
$v_{i,f}$	pore particle	pore flow
V_i	particle	volume
m_i	-	total mass
$m_{i,f}$	node, pore particle	fluid mass
$m_{i,s}$	node, pore particle	solid mass
w_i	pore particle	barycentric coordinate
ρ_i	SPH, pore particle	density
ρ_i^0	-	rest density
Φ_i	node, pore particle	solid fraction
s_i	node, pore particle	saturation

2.3 Porosity modeling

Node의 포화도는 이웃 face에 샘플링 된 다공성 파티클 i로부터 다음과 같이 계산된다.

$$m_{n,f\max} = \sum_i w_i (1 - \Phi_i) V_i \rho_f^0 \quad (1)$$

$$s_n = \frac{m_{n,f}}{m_{n,f\max}} \quad (2)$$

위 수식에서 $m_{n,f\max}$ 는 node에서 흡수할 수 있는 최대 액체 질량을 의미한다. $(1 - \Phi_i) V_i$ 는 다공성 파티클 내 유체의 부피이고 ρ_f^0 를 곱하게 되면 다공성 파티클에서 흡수할 수 있는 최대 액체 질량이 된다. 여기에 node에 대한 i의 무게중심 좌표인 w_i 를 곱하고 모두 더해지면서 node에서 흡수할 수 있는 최대 액체 질량을 계산한다. 이를 node의 액체 질량에 나누면 node의 포화도가 된다.

다공성 파티클의 포화도는 node의 포화도를 계산한 뒤 선형보간법을 통해 계산한다.

$$s_p = s_0 w_0 + s_1 w_1 + s_2 w_2 \quad (3)$$

여기서 0~2는 파티클이 샘플링 된 face의 3개의 node를 의미한다.

2.4 Pore flow

[6]에서는 다르시 법칙에 따른 다공성 물질 내 유속을 중력과 압력을 통해 계산한다. 하지만 SPH기법을 통해 계산 된 유체 파티클의 압력은 불안정하다. 본 논문에서는 실제 파티클의 압력을 대신할 가상의 압력을 이용하여 다공성 물질 내 유체흐름의 방향만을 계산하여 안전성을 높였다.

다공성 파티클에서 가상의 압력은 다음과 같이 계산한다.

$$P' = k^p \sum_i V_i \nabla W_{p,i} \quad (4)$$

위 수식에서 i는 다공성 파티클의 이웃 SPH 및 다공성 파티클을 제외한 바운더리 파티클이고 k^p 는 사용자 지정 계수이다. 이 압력으로 유체의 흐름은 다음과 같이 계산된다.

$$v_{p,f} = \frac{(P' + g)}{\|P' + g\| + \epsilon} \quad (5)$$

분모에 ϵ 을 더해준 이유는 $\|P' + g\|$ 값이 0에 가까울 때는 $v_{p,f}$ 를 0벡터로 만들기 위해서다. $v_{p,f}$ 값이 0벡터일 때는 다공성 파티클과 SPH 파티클 간의 흐름이 없다는 것을 나타낸다. 이렇게 구해진 유체 흐름은 물의 흡수에 영향을 준다.

2.5 Absorption

SPH 파티클이 다공성 파티클에 접촉하면 다공성 파티클에서 SPH 파티클의 질량을 흡수하게 된다. 이때 SPH 파티클 i에서 이웃 다공성 파티클 j로 흡수되는 질량은 다음과 같이 계산된다.

$$\Delta m_i = \Delta t \sum_j k_j^a (s_{j,\max} - s_j + (s_{j,\max} - 1) \cos \theta_{ij}) V_j \nabla^2 W_{i,j} \quad (4)$$

여기서 k_j^a (0~1)는 흡수 정도를 조절하는 상수이고 $s_{j,\max}$ (1~2)는 흡수를 할 수 있는 최대 포화도를 의미한다. 이를 조절하여 유체가 옷감을 통과하는 정도를 조절할 수 있다. $(s_{j,\max} - 1) \cos \theta_{ij}$ 는 다공성 파티클 내 유체 흐름에 대한 영향을 의미하고 여기서 θ_{ij} 는 $x_i - x_j$ 와 $v_{j,f}$ 가 이루는 각이다. SPH 파티클이 유체흐름의 반대 방향에 위치한 다면 다공성 파티클의 포화도가 1까지만 흡수되기 때문에 통과하지 못하게 된다. 반대로 유체흐름의 방향에 맞게 위치해 있다면 $1 - s_{j,\max}$ 만큼 더 흡수된다.

질량 변화량 Δm_i 값이 $m_i - m_{f,\min}$ 보다 크거나 같다면 변화량은 Δm_i 값 대신 m_i 가 된다. m_i 에 Δm_i 를 빼주고 m_i 값이 0이 되면 SPH 파티클을 시뮬레이션에서 제거하고 아니면 SPH 파티클의 h_i 와 r_i 를 $m_i/m_{f,\max}$ 비율로 수정해준다. 여기서 $m_{f,\min}$ 와 $m_{f,\max}$ 은 최초, 최대 SPH 파티클 질량이다.

계산된 Δm_i 만큼 j로 전달하는데 수식에서 계산했을 때 Δm_i 값에 기여한 비율을 대로 나눠준다. 다공성 파티클로 전달된 Δm_i 는 옷감의 node로 다음과 같이 전달된다.

$$\Delta m_n = \sum_j w_j \Delta m_j \quad (5)$$

전달받은 node에서는 Δm_n 만큼 $m_{n,f}$ 를 더해주고 동시에 포화도 도 수정해준다.

2.6 Emission

Node의 포화도가 1을 넘게 된다면 SPH 파티클을 방출하게 된다. 먼저 node에서 방출해야 하는 유체 질량 $\Delta m_n = m_{n,f} - m_{n,f\max}$ 을 샘플링 된 다공성 파티클 i에 다음과 같이 전달한다.

$$w'_n = \sum_j w_j \quad (6)$$

$$\Delta m_i = w_0 \Delta m_0 / w'_0 + w_1 \Delta m_1 / w'_1 + w_2 \Delta m_2 / w'_2 \quad (7)$$

수식 6에서 j 는 node의 이웃 face에 샘플링 된 다공성 파티클이다. Δm_i 를 구한 후 다공성 파티클 i 와 이웃 SPH 파티클 j 의 상대적 위치와 $v_{i,j}$ 가 이루는 각이 θ 보다 작고 $m_j > m_{j,max}$ 보다 작으면 j 에 $m_{j,max} - m_j$ 보다 적게 질량변화량을 전달한다. 그리고 Δm_i 값이 $m_{j,min}$ 보다 크면 남은 질량 크기의 SPH 파티클을 $v_{i,j}$ 방향으로 방출한다.

이때 방출 방향과 이웃 바운더리 파티클 j 의 상대적 위치가 이루는 각이 θ 보다 작으면 방출하지 않는다. 바운더리 파티클이 다공성 파티클이라면 남은 유체 변화량을 해당 파티클의 질량변화량으로 전달한다. 최종적으로 질량변화량은 3.3절처럼 node로 전달한다. SPH 파티클이 방출되었을 때 고밀도 상태가 될 수 있다. 이 문제를 해결하기 위해 [7]에서 제안한 방법대로 일정 프레임동안 밀도와 속도를 제한하여 시뮬레이션을 안정시켰다.

2.7 Diffusion

확산 계산순서는 다음과 같다. 먼저 수식 3을 통해 흡수와 방출이 된 후 다공성 파티클의 포화도를 계산한다. 같은 방법으로 node의 $m_{n,j}$ 를 다공성 파티클로 전달한다. 마지막으로 SPH 메소드를 이용한 확산공식을 통해 확산을 계산한다.

다공성 파티클 i 가 이웃 다공성 파티클 j 로부터 확산으로 인한 질량 변화량 계산 공식을 다음과 같이 정의한다.

$$\Delta m_i = -\Delta t m_{i,j} k_j^d (s_i - s_j + \eta^d \cos \theta_{ij}) \nabla_j^2 W_{i,j} \quad (8)$$

위 수식에서 $k_j^d(0\sim1)$ 는 확산 정도를 조절하는 상수이고 $\eta^d(0\sim1)$ 는 중력의 영향을 받는 정도를 조절하는 상수이다. θ_{ij} 는 $x_i - x_j$ 와 중력이 이루는 각이다. Δm_i 는 j 로부터 가져온 질량이기 때문에 $\Delta m_j = -\Delta m_i$ 가 된다. 또한 i 로부터 j 로 이동하는 질량 또한 계산해야 하기 때문에 최종적으로 확산공식은 다음과 같이 정리된다.

$$\Delta m_i = -\Delta t (m_{i,j} k_j^d \nabla_j + m_{j,i} k_i^d \nabla_i) (s_i - s_j + \eta^d \cos \theta_{ij}) \nabla^2 W_{i,j} \quad (9)$$

계산된 질량변화량은 3.3절과 같이 node에 전달된다. Δt 가 너무 크거나 질량 변화량이 너무 크면 node의 질량이 음수가 되는 등 시뮬레이션이 불안정해진다. 따라서 n 번 $\Delta t/n$ 의 타임스텝으로 나누어 반복적으로 계산했다.

III. Result

본 논문에서 유체 시뮬레이션은 DFSPH 기법을 이용하였고 support radius 크기가 다른 파티클들은 [1]에서 제안한 기법을 사용했다. 옷감 시뮬레이션은 Jacobian Projective Dynamics 기법을 이용했고 충돌처리는 [5] 기법을 통해 구현했다. 옷감이 찢었을 때 옷감끼리 붙는 기능은 [6]에서 제안한 모델을 이용하여 구현했다.

적용형 타임스텝을 사용하면 시뮬레이션이 불안정해져 하나의 타임스텝으로 계산했다. 그렇기 때문에 SPH만 계산했을 때는 뚫리는 문제가 발생하기 때문에 CCD계산을 통해 이러한 문제를 방지했다.

모든 테스트에서 사용하는 상수 값 k^p 는 16.8, k^v 는 0.01, k_j^d 는 0.1, s_{max} 는 1.18, θ^c 는 80.0° , η^d 는 0.01로 설정했다.

Fig. 1과 2는 옷감위에 물방울을 떨어뜨리고 던지는 장면이다. 옷감의 node개수는 18,199개, SPH 파티클 개수는 74,048개다.

Fig. 3는 구 위에 옷감을 떨어뜨리고 그 위에 물방울을 떨어뜨리는 장면이다. 옷감의 node개수는 18,199개고 SPH 파티클 개수는 개다.

Table 2. Performance.

	# of Node.	# of SPH	frame
Fig. 1	14,569	74,088	0.3 sec
Fig. 2	14,569	74,088	0.3 sec
Fig. 3	18,199	405,224	1.2 sec

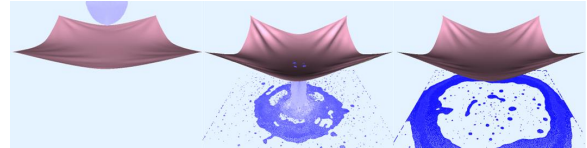


Fig. 1. Falling water on cloth.



Fig. 2. Falling cloth on sphere and falling water.

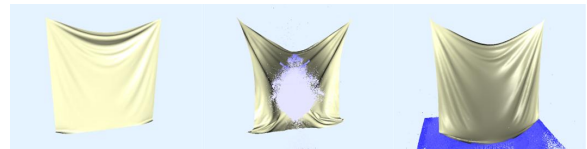


Fig. 3. Water injected from the side.

IV. Conclusions

본 논문에서는 물리 기반 유체 시뮬레이션 옷감의 상호작용 시 발생할 수 있는 다양한 물리효과를 하나의 GPU 통합형 프레임워크를 통해 효율적으로 표현해냈다. 향후, 이 알고리즘을 개선하여 흡수로 인해 옷감이 찢어지는 효과까지 포함하여 보다 복잡한 상호작용 현상을 GPU에서 표현할 수 있도록 확장할 계획이다.

REFERENCES

- [1] Adams, Bart, Mark Pauly, Richard Keiser, and Leonidas J. Guibas. "Adaptively sampled particle fluids." In ACM SIGGRAPH 2007 papers, pp. 48-es. 2007.
- [2] Akinci, Nadir, Jens Cornelis, Gizem Akinci, and Matthias Teschner. "Coupling elastic solids with smoothed particle hydrodynamics fluids." *Computer Animation and Virtual Worlds* 24, no. 3-4 (2013): 195-203.
- [3] Akinci, Nadir, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, and Matthias Teschner. "Versatile rigid-fluid coupling for incompressible SPH." *ACM Transactions on Graphics (TOG)* 31, no. 4 (2012): 1-8.
- [4] Akinci, Nadir, Gizem Akinci, and Matthias Teschner. "Versatile surface tension and adhesion for SPH fluids." *ACM Transactions on Graphics (TOG)* 32, no. 6 (2013): 1-8.
- [5] Bridson, Robert, Ronald Fedkiw, and John Anderson. "Robust treatment of collisions, contact and friction for cloth animation." In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 594-603. 2002.
- [6] Bender, J., C. Duriez, F. Jaillet, and G. Zachmann. "Coupling Hair with Smoothed Particle Hydrodynamics Fluids."
- [7] Solenthaler, Barbara, and Markus Gross. "Two-scale particle simulation." In *ACM SIGGRAPH 2011 papers*, pp. 1-8. 2011.