

안정적인 발산 제약 이동최소자승법을 이용한 연기의 난류 생성 및 이류

이수빈^o, 김종현^{*}

^o강남대학교 소프트웨어응용학부,

^{*}인하대학교 소프트웨어융합대학 디자인테크놀로지학과

e-mail: jonghyumkiun@inha.ac.kr

Turbulence Generation and Advection in Smoke Using Stable Divergence-Constrained Moving Least Squares

Subin Lee^o, Jong-Hyun Kim^{*}

^oSchool of Software Application, Kangnam University,

^{*}College of Software and Convergence (Dept. of Design Technology), Inha University

● 요약 ●

본 논문에서는 Semi-Lagrangian 이류 과정에서 역추적(Backward tracing)한 위치의 주변 속도를 Divergence-constrained MLS(Moving least squares)를 이용하여 보간하고 그 결과를 이류된 속도 데이터의 외력으로 적용해 연기 시뮬레이션의 난류 표현을 개선할 수 있는 새로운 프레임워크를 제안한다. 일반적인 MLS는 고차보간법이기 때문에 시간에 따른 연속성 보장이 안 되기 때문에 그 결과가 노이즈한 형태로 나타난다. 본 논문에서는 연기의 원본 속도와 제한하는 기법을 통해 생성된 속도 간의 각도 변화를 통해 난류를 생성하며 이를 통해 안정적이고 연기의 밀도를 이루시킨다.

키워드: 격자 기반 유체 시뮬레이션(Grid-based fluid simulation), 발산 제약 이동최소자승법(Divergence-constrained MLS), 난류(Turbulence), 나비에-스토크스 방정식(Navier-Stokes equation)

I. Introduction

물리 기반 시뮬레이션 중 유체 시뮬레이션에 주로 사용되는 Stable Fluids 기법은 세미-라그랑주(Semi-Lagrangian) 이류를 사용해 안정적인 시뮬레이션 결과를 보장한다[1]. 하지만 수치 손실로 인해 디테일한 난류를 표현하는 것이 어렵다는 단점이 있다. 이를 개선하기 위해 DCMLS(Divergence-constrained MLS)로 기존 이류 단계의 보간법을 대체해 난류를 강화한 연구가 있다[1]. 하지만, 결과가 매우 노이즈하고 불안정할 뿐만 아니라 일부 영역에서 튀는 현상이 발생하기 때문에 일반적인 환경에서 사용하기에는 충분하지 않다. 본 논문에서는 이 기법의 단점을 보완하여 고차보간법의 일종인 MLS를 물리 기반 시뮬레이션에 온전하게 통합시킬 수 있는 효율적인 프레임워크를 제안한다. 본 논문에서는 DCMLS로 보간한 속도장을 Stable Fluids의 이류 단계를 완료한 속도 데이터에 외력으로 적용하며 노이즈로 시뮬레이션 불안정성을 완화한다. 본 논문에서는 연기의 원본 속도와 제한하는 기법을 통해 생성된 속도 간의 각도 변화를 이용하여 난류로 인해 연기의 디테일을 개선한다.

II. Preliminaries

1. Related works

2.1 Underlying fluid

Stable Fluids의 시뮬레이션 과정은 크게 3단계인 이류(Advect), 확산(Diffuse), 투영(Project)으로 분류된다. 이류 단계에서 세미 라그랑주 기법을 사용하는 것이 특징이다. 주어진 입자의 위치와 속도로 Δt 만큼 이전으로 돌아간 위치에서 역추적된 속도를 근사 계산한다. 역추적한 격자 주변의 속도를 보간해 새로운 속도를 계산한다. 본 논문에서는 역추적한 위치의 속도는 2차원에서 보통 이중선형보간법(Bilinear interpolation)을 사용하여 보간한다.

2.2 Divergence-constrained MLS

MLS는 주어진 데이터 위치를 중심으로 인접 데이터들의 위치를 고려해 가중치를 적용하는 고차보간법이다. DCMLS는 물질의 양이 변하지 않는 상태인 Divergence-Free를 기반으로 MLS를 계산한다.

외삽법을 통해 유체의 질량 변화를 방지하고 운동량이 보존된다. 또한, Constrain의 속도가 수직 성분만 주어져도 0이 아닌 수평 성분의 속도가 보간 된다.

이러한 특징으로 이류 계산 과정의 보간을 DCMLS로 대체하면 Stable Fluids 보다 난류 표현이 강화된 결과를 얻을 수 있다. 하지만 Fig. 1과 같이 난류가 유체의 전체적으로 강화되어 결과가 다소 노이즈 한 경향이 있다. 실제 테스트 과정에서 일부 영역에서 속도가 튀는 현상도 종종 발생한다.

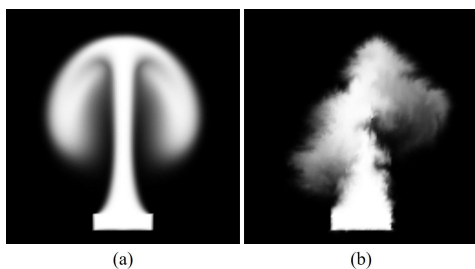


Fig. 1. Comparison of turbulence generated using the previous MLS interpolation : (a) Stable Fluids[1], (b) DCMLS interpolation[3].

III. The Proposed Scheme

3.1 Compute velocity interpolated using DCMLS

Fig 2는 본 논문에서 제안하는 새로운 보간법인 DCMLS로 속도를 보간하여 외력으로 적용 및 후처리하는 과정을 보여주는 그림이다. 먼저, 세미 라그랑주 기법을 기반으로 이전에 위치한 격자의 위치를 계산하고, 이중선형보간법과 DCMLS 보간된 속도를 각각 계산한다. DCMLS로 속도를 보간하는 알고리즘은 다음과 같다.

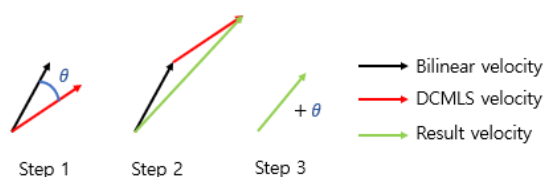


Fig. 2. An overview of the algorithm for applying DCMLS as an external force through angle changes between velocities.

각 노드에 대해서 역추적으로 이전 위치를 추적한다. 추적한 위치가 속한 격자에 대해서 Fig 3과 같이 데이터를 설정한다. 역추적한 격자의 네 꼭짓점을 제약 조건의 위치로, 주변 격자 중 회색으로 표시된 네 격자의 속도를 제약 조건의 속도로 설정한다. 나머지 격자의 속도도 함께 저장해 발산을 계산할 때 사용한다.

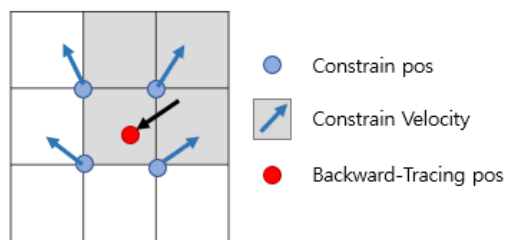


Fig. 3. DCMLS data structure.

앞에서 설명한 Constrain 속도와 위치 데이터를 기반으로 고차보간에 필요한 행렬을 구성하기 위해 이전 기법에서 제안한 DCMLS 알고리즘을 활용한다[2,3] (수식 1 참조). 좀 더 자세한 설명을 이전 논문을 읽어보길 권장한다.

$$W_{div}(x)B_{div}C_{vec}(x) = W_{div}(x)\Phi_{div} \quad (1)$$

여기서 행렬 Φ 에는 제약 조건의 속도와 발산, 행렬 B 에는 제약 조건의 위치를 미분한 데이터, 행렬 W 에는 가중치 데이터를 입력한다.

$$w(|r|) = 1 / (|r|^4 + \epsilon^2), \quad \epsilon = 0.0001 \quad (2)$$

본 논문의 모든 예시에서는 수식 2의 가중치 함수를 사용한다. 행렬 W 와 행렬 B 의 곱, 행렬 W 와 행렬 Φ 의 곱에 대해 특이값 분해(Singular value decomposition)를 통해 행렬 C 를 구한다.

3.2 Turbulence generation using DCMLS weights

이중선형보간법으로 보간된 속도 u_b 에 DCMLS로 보간된 속도 u_d 를 외력으로 적용한다 (수식 3 참조).

$$u = \frac{u_b + u_d}{\|u_b + u_d\|} * \|u_b\| \quad (3)$$

위 수식과 같이 외력이 적용된 속도를 정규화한 후 이중선형보간법으로 보간된 속도의 크기를 곱한다. 이 같이 처리한 이유는 속도 벡터의 방향만을 변화시키고 크기는 동일하게 유지해 발산을 방지하기 위함이다. 이중선형보간법으로 보간된 속도와 DCMLS로 보간된 속도간의 각도를 계산한다 (Fig. 2 참조).

수식 4의 u 는 최종적으로 계산된 속도, Δt 는 시간간격 (Time-step), $vort_d$ 는 두 속도 벡터 간의 각도를 나타낸다. 본 논문에서는 연기의 원본 속도와 제안하는 기법을 통해 생성된 속도 간의 각도 변화를 가중치로 적용해 유체의 난류가 전체적으로 강화되지 않도록 부분 별로 난류에 대한 크기를 제한한다.

$$u = u + \Delta t * vort_d \quad (4)$$

IV. Results

본 논문에서 제안한 알고리즘의 성능을 검증하기 위해 3가지 실험 환경에 대한 시뮬레이션 결과를 비교한다. Fig 4, 5, 6의 (a)는 Stable Fluids를 기반으로 한 결과이고[1], (b)는 DCMLS로 이류의 보간을 대체한 알고리즘을 실행한 결과[3], (c)는 DCMLS로 보간된 각도 변화를 외력으로 적용하여 시뮬레이션을 실행한 결과이다.

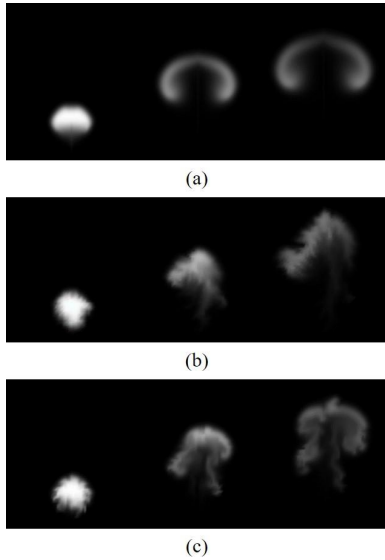


Fig. 4. Rising smoke (scene1)

Fig 4의 실험 환경은 단일 구체의 밀도가 한 번만 주입되고 부력에 의해 이루어지는 결과이며, 가장 단순한 형태의 시뮬레이션 결과를 보인다. 이전 기법은 Fig. 3b는 노이즈 한 움직임이 많이 내포되어서 CFL 조건에 따른 결과가 시뮬레이션 안정성이 달라지며, 사용자는 이 조건을 수동으로 제어해야 되기 때문에 일반적인 장면에서 활용이 어렵다. 또한 그림에서 보듯이 표면에 연기 표면에 화이트노이즈와 같은 변화만 있을 뿐 전체적인 연기의 움직임에는 큰 변화가 없다. 그에 비해 제안하는 기법인 Fig. 3b는 외력에 의한 연기의 움직임이 과장된 것이 보이며, 화이트 노이즈와 같은 자글거리는 픽셀의 변화가 없어 시뮬레이션 안정성 또한 개선되었다.

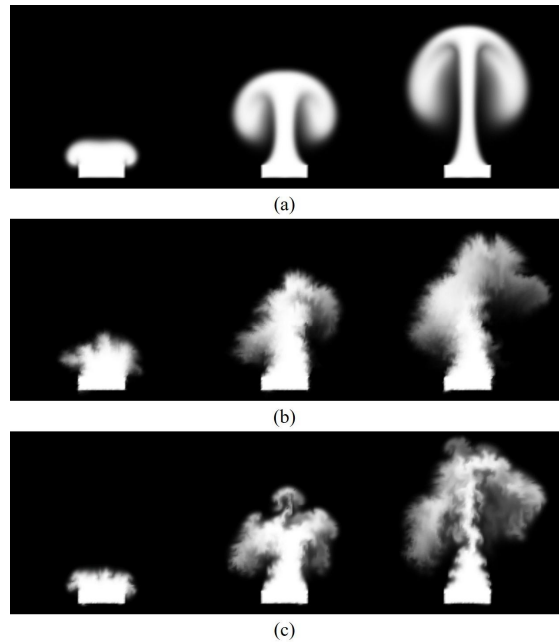


Fig. 5. Rising smoke (scene2)

Fig 5는 사각형 형태의 밀도를 지속적으로 주입하는 장면이다. 이 결과에서도 Fig. 4와 마찬가지로 역동적인 연기의 난류를 잘 표현했다.

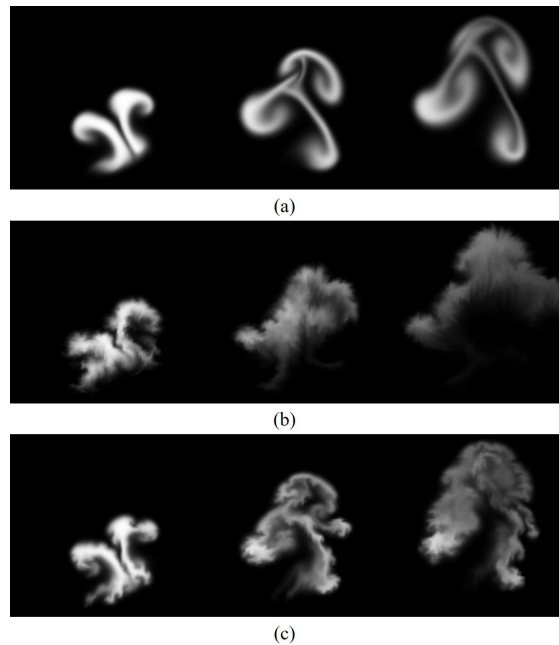


Fig. 5. Simulate smoke with two colliding densities (scene3)

Fig 6은 2개의 구형 밀도를 주입하는 장면이다. 각각의 속도는 (0.5, 0.06)와 (-0.5, -0.13)으로 정의된다. 서로 반대되는 속도를 가지는 구체 2개가 충돌할 때의 결과를 나타낸다. Fig. 6b와 Fig. 6c 모두 입력 연기 시뮬레이션보다 뚜렷하게 난류가 강화된다. DCMLS기반의 각도 변화를 통해 Fig. 6b보다 표면 노이즈가 완화되고, 전체적인 움직임에 난류 변형이 뚜렷하게 나타났다. 뿐만 아니라, 연기 표면에서 화이트 노이즈 또한 사라진 것을 볼 수 있다. 또한, DCMLS로 보간된 속도를 외력으로 적용함으로써 기존 Stable Fluids의 진행 경로를 안정적으로 유지하면서 난류가 강화되는 결과를 보인다.

V. Conclusions

본 논문에서는 DCMLS로 보간된 속도를 이중선형보간법으로 보간된 속도에 외력으로 적용하고, 두 벡터 사이의 각도 변화를 추가적인 가중치로 알고리즘을 개선했다. 그 결과, 연기 시뮬레이션이 이전 기법보다 노이즈가 완화된 난류를 통해 안정적이고 디테일한 연구 시뮬레이션을 보여주었다. MLS의 특성상 고차보간을 활용하는 이전 기법은 고차보간에 따른 움직임이 노이즈로 나타나는 경향이 있어서 CFL조건과 같은 까다로운 시간 적분법이 필요하다. 반면, 제안하는 기법은 시각적인 측면에서도 연기의 외류를 잘 표현했고, 시뮬레이션 안정성 또한 개선시켰다. 본 논문에서 제안하는 방법은 각 노드에 대한 연산이 독립적으로 이루어지기 때문에 병렬화가 용이하며, 특히 값 분해를 병렬 알고리즘으로 적용한다면 고속화가 가능하다. 따라서 밀도를 시각화 혹은 분석이 필요한 다양한 산업에서 활용될 수 있을 거라 기대한다.

REFERENCES

- [1] Jos Stam, "Stable Fluids", In Proceedings of SIGGRAPH 99, pp.121-128, 1999.
- [2] Jeong-Mo Hong, Jong-Chul Yoon, Chang-Hun Kim, "Divergence-constrained moving least squares for fluid simulation", Comp, Anim, Virtual Worlds 19, pp.469-477, 2008.
- [3] Sun-Tae Kim, Jeong-Mo Hong, "Visual simulation of turbulent fluids using MLS interpolation profiles". The Visual Computer 29, pp.1293-1302, 2013.