

# 몬테카를로 방법 기반의 이동최소제곱을 이용한 밀도 데이터의 벡터장 시각화

김종현\*

인하대학교\*

jonghyunkim@inha.ac.kr

## Visualization of Vector Fields from Density Data Using Moving Least Squares Based on Monte Carlo Method

Jong-Hyun Kim\*

Inha University\*

### 요약

본 논문에서는 밀도 데이터로부터 다양한 벡터장 패턴을 시각화하는 새로운 방법을 제안한다. 이를 위해 물리 기반 시뮬레이션과 기하학적 처리에서 사용되는 이동최소제곱(Moving least squares, MLS)을 이용한다. 하지만 일반적인 MLS는 벡터 기반의 제약조건을 통해 고차 보간되기 때문에 밀도의 특성을 고려하지 못한다. 본 논문에서는 입력 데이터에 내포되어 있는 밀도의 특성을 효율적으로 고려하기 위해 몬테카를로 기반의 가중치를 MLS에 통합하여 다양한 형태의 벡터장을 표현할 수 있도록 알고리즘을 설계한다. 결과적으로 일반적인 MLS와 발산제약 기반의 MLS 같은 기존 기법으로는 표현하기 힘든 디테일한 벡터장을 실험을 통해 보여준다.

### Abstract

In this paper, we propose a new method to visualize different vector field patterns from density data. We use moving least squares (MLS), which is used in physics-based simulations and geometric processing. However, typical MLS does not take into account the nature of density, as it is interpolated to a higher order through vector-based constraints. In this paper, we design an algorithm that incorporates Monte Carlo-based weights into the MLS to efficiently account for the density characteristics implicit in the input data, allowing the algorithm to represent different forms of white noise. As a result, we experimentally demonstrate detailed vector fields that are difficult to represent using existing techniques such as naive MLS and divergence-constrained MLS.

**키워드:** 몬테카를로법, 이동최소제곱, 벡터장, 시각화, 물리 기반 시뮬레이션

**Keywords:** Monte Carlo Method, Moving least squares, Vector field, Visualization, Physically based simulations

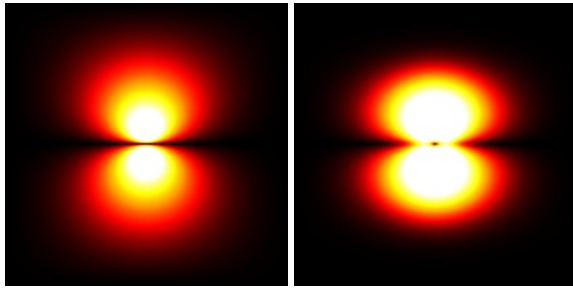
## 1 서론

MLS는 분산된 데이터를 부드럽게 근사하거나 고차 보간하기 위해 사용되는 방법이다. MLS를 확률 밀도 함수에 적용하면 보다 정확한 확률 분포를 계산할 수 있기 때문에 유체 [1, 2], 변형체 [3], 메쉬화 [4, 5] 등의 분야에서 활용되고 있다. 다항식 기반의 보간은 샘플링 지점 사이에서 데이터를 검색하는 데 활용되기

때문에 연속적인 재질의 동적 움직임을 표현하는 유체 시뮬레이션이나 캐릭터 모델링에서 많이 사용되며, 이 경우 적절한 보간 방법을 선택하는 것은 품질에 영향을 주기 때문에 매우 중요하다 [6, 7, 8].

보간을 통해 샘플링 데이터를 얻는 과정은 물리 기반 시뮬레이션에서 중요한 과정이다. 기하학적 처리에서는 포인트 클라우드 데이터로부터 메쉬표면을 근사 생성하는데 활용되며, 포인트

\*corresponding author: Jong-Hyun Kim/ Inha University (jonghyunkim@inha.ac.kr)



(a) Naive MLS

(b) Divergence-constrained MLS [6]

Figure 1: Comparing standard MLS vector interpolation to pseudo-divergence-free vector interpolation [6].

클라우드의 연결정보가 없기 때문에 고차 수치 방법을 통해 암시적으로 삼각형 기반으로 구성된 메쉬를 만들어 낸다 [4]. 또한 물리 기반 시뮬레이션 분야에서는 시뮬레이션 정확도를 개선하기 위해 고차 수치 보간법을 사용한다 [9]. 연기 시뮬레이션의 이류항에 활용되는 선형보간 기반의 세미 라그랑주 방식을 [10] 개선하기 위해 BFECC(Back and forth error compensation and correction) [11], CIP(Constrained interpolation profile) [12], USCIP(Unsplit semi-Lagrangian CIP) [13], 스트림 기반 연기 [14], 주파스 기반 연기 [15], 적응형 와류 기법 [16], Polynomial PIC [17] 등 기법들이 제안되었다.

Feldman et al.은 적응형 격자에서 부드러운 벡터장을 얻기 위해 속도의 정규 성분(Normal component)을 기반으로 MLS를 계산했다 [18]. MLS는 고차 보간을 위한 기술이며, 앞에서 언급했듯이 다양한 분야에 활용되지만 MLS의 연속성 조건이나 물리적 제약조건을 만족시키는 것이 어렵기 때문에 시뮬레이션에서는 적극적으로 활용되지 못했다. Hong et al.은 발산제약 기반 MLS 기법을 제안하여 푸아송 방정식을 풀지 않아도 유체와 유사한 움직임을 표현할 수 있는 벡터장 생성 기법을 제안했으며 [6], 이를 연기 시뮬레이션에 적용하여 복잡한 난류의 움직임을 표현했다 [7].

본 논문에서는 밀도를 가진 데이터에서 벡터장을 표현할 수 있는 방법을 몬테카를로 가중치 기반 MLS 기법을 통해 계산한다. Figure 1은 일반적인 MLS 기법과 Hong et al.이 제안한 발산제약 기반 MLS의 벡터보간 차이를 보여주는 그림이다 [6]. Figure 1a는 수직으로 마주 보는 두 벡터를 제약조건으로 MLS를 계산한 결과이며, Figure 1b는 발산제약 기반 MLS를 통해 계산된 벡터장이다. 발산제약 기반 MLS는 유체 시뮬레이션에 적용하기 위해 비압축성을 보장하는 보간기법이며 (Figure 1b 참조), 그 결과는 일반적인 MLS와 비교했을 때 명확하게 차이가 나타난다 (Figure 1a 참조). 특히 이 결과는 벡터형태의 제약을 발산 조건에 따라 다르게 표현할 수 있기 때문에 벡터장을 다양한 형태로 디자인하거나 물리 기반 시뮬레이션에서 활용할 수 있다. 하지만, 제약조건이 벡터로만 표현되기 때문에 밀도를 고려하지 못한다.

밀도는 곧 연기 시뮬레이션에서 중요한 물리적 특성이기 때문에 반드시 고려해야 될 부분이다. 본 논문에서는 몬테카를로법을 활용하여 밀도가 분포되어 있는 위치를 추정하고, 추정된 위치와 주변 데이터 간의 거리를 MLS의 추가적인 가중치로 활용하여 밀도를 고려한 벡터장을 시각화할 수 있게한다.

## 2 관련 연구

유체 시뮬레이션 분야에서는 난류의 시각적인 디테일을 개선하기 위해 다양한 종류의 이류 및 외력 기법들이 제안되어 왔다 [19, 20, 10]. 세미 라그랑주 기법은 안정적인 유체를 시뮬레이션하기 위해 제안되었으며 [10], 와류 제한 기법(Vorticity confinement method)은 격자기반에서 난류를 표현하기 위한 와류 힘을 추정했으며 [21], 향후 라그랑지안 입자를 와류 운반체로 활용하여 와류 입자 방법(Vortex particle method)으로 개선되었다 [22]. 그 외에도 BFECC [11], CIP [12], USCIP [13], 스트림 기반 연기 [14], 주파스 기반 연기 [15], 적응형 와류 기법 [16], Polynomial PIC [17] 등 기법들이 제안되었다.

많은 기법들이 입자 시뮬레이션을 통해 서브그리드 디테일을 강화하려고 노력해왔다. Kim et al. [23]과 Losasso et al. [24]은 서브그리드 스플래시를 해결하기 위해 SPH(Smoothed particle hydrodynamics)모델을 사용했다. Greenwood와 House [25], Hong et al. [26], Thuerey et al. [27]은 다상 유체를 보다 세밀하게 표현하기 위해 입자를 활용한 혼합 접근법을 이용했으며, Gao et al.은 라그랑지안 입자를 사용하여 기체의 고속 특징을 포착했다 [28]. 서브그리드 역학에 물리 모델을 도입하면 다상 인터페이스 [29], 화염 [30, 31], 액체의 디테일을 한 층 더 개선시킬 수 있다 [32]. Hong et al.은 푸아송 방정식(Poisson's equation)을 풀지 않고도 유체의 움직임 표현할 수 있는 발산제약 MLS를 기법을 제안했다 [6]. 이 기법을 확장하여 연기 시뮬레이션의 이류 과정에 통합 시킴으로써 난류를 표현했다 [7]. 하지만, MLS 기반 난류는 디테일보다는 노이즈한 움직임이 크기 때문에 시뮬레이션을 제어하기 어렵게 만들며, 고차 방정식으로 인한 계산량이 매우 크기 때문에 산업에서 활용하기 어려운 접근법이다.

이 외에도 후처리로 유동의 디테일을 향상시키는 접근법들도 있다. 그 중 일부는 웨이블릿 노이즈 [33], 켄 노이즈를 사용하여 저해상도 시뮬레이션을 위한 난류 움직임을 합성했다 [34, 35]. Chu와 Thuerey는 합성곱 신경망을 사용하여 저해상도 유동과 고해상도 유동간의 관계를 학습했다 [36]. Sato et al.은 패치 기반 텍스처 합성을 이용하여 저해상도 흐름에 디테일 난류를 전송하는 기법을 제안했다 [37]. 최근에 Chen et al.은 연속체 역학을 기반으로 하는 시뮬레이션과 MLS를 이용하여 종기와 유체의 상호작용과 그로인해 찢김 현상을 표현할 수 있는 통합형 기법을 제시했다 [38]. 이러한 접근법은 그럴듯한 디테일을 합성하지만 노이즈 함수를 사용하면 고해상도 시뮬레이션의 흐름보다 덜 사실적인 유동이 생성되는 문제가 있다.

본 논문에서 MLS 가중치로 사용하는 몬테카를로 기법을 사

용하고 있지만, 일반적으로 몬테카를로 방법은 다양한 분야에서 샘플링으로 활용되고 있다. Sawhney와 Crane는 격자없이 기하학적 모델 기반에서 편미분 방정식을 풀 수 있는 몬테카를로 기법을 제시했다 [39]. 최근에는 인공지능을 통해 장면을 3차원으로 복원하기 위한 접근법에 몬테카를로 기법을 사용되기도 한다 [40].

### 3 제안하는 프레임워크

본 논문에서 제안하는 몬테카를로 가중치를 이용한 발전제한 기반 MLS를 설명하기 위해 기본적인 스칼라와 벡터 보간을 위한 MLS를 먼저 설명한다. 그런 다음 ‘몬테카를로 가중치 기반 MLS’ 장에서 몬테카를로 방법을 통해 밀도의 위치를 추정하고 그로부터 MLS 가중치를 계산할 수 있는 방법을 제안하여 이로부터 벡터장을 계산한다.

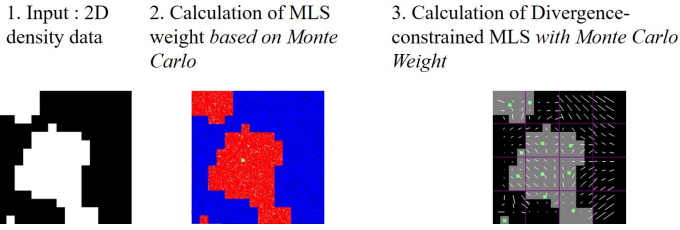


Figure 2: Algorithm overview of our method.

Figure 2는 본 논문에서 제안하는 알고리즘 개요를 보여주는 그림이다. 본 논문에서는 밀도 데이터를 고려하여 벡터장을 시각화하기 위해 몬테카를로 방법을 이용해 MLS의 가중치를 조절하는 새로운 기법을 제안한다. 기존의 MLS 방식이 벡터 기반의 제약조건만 고려하는 한계를 극복하고, 밀도 정보를 활용하여 보다 디테일한 벡터장을 표현할 수 있다는 점이 본 논문의 장점이다.

#### 3.1 이동최소제곱

##### 3.1.1 스칼라 보간

만약  $N$ 개의 제어점  $p_i$ 가 특정 영역 내에 위치한다면, 이를 활용한 다항식은 아래와 같이 정의된다 (수식 1 참조).

$$f(\mathbf{x}) = \mathbf{b}^T(\mathbf{x})\mathbf{c} \quad (1)$$

제어점인  $p_i$ 에서의 근사값인  $\phi^i$ 는 아래와 같다 (수식 2 참조).

$$f(\mathbf{p}_i) = \mathbf{b}^T(\mathbf{p}_i)\mathbf{c} = \phi^i, \quad (2)$$

여기서 샘플링 위치  $\mathbf{x}$ 는 2차원에서는  $[x, y]$ 이고, 3차원에서는  $[x, y, z]$ 로 표현되며,  $\mathbf{b}(\mathbf{x}) = [b_1(\mathbf{x}), \dots, b_k(\mathbf{x})]$ 는 다항식의 기저함수이며,  $\mathbf{c} = [c_1, \dots, c_k]$ 는 계수로 미지수 벡터이다. 일반적으로,  $\mathbf{b}(\mathbf{x})$ 내에 있는 요소들의 개수는  $k = \frac{(d+m)!}{m!d!}$ 로 주어지며, 여기서  $f$ 는  $\prod_d^m$ 로 부터 주어지고, 그것은 공간적 차원인  $m$ 와  $d$ 를

이용한 다항식의 공간이다. 2차원( $d=2$ )일 때 아래와 같이 쓸 수 있으며 (수식 3 참조),

$$\begin{aligned} \mathbf{b}(\mathbf{x}) &= [1] \text{ if } m = 0 \\ \mathbf{b}(\mathbf{x}) &= [1, x, y] \text{ if } m = 1 \\ \mathbf{b}(\mathbf{x}) &= [1, x, y, x^2, y^2, xy] \text{ if } m = 2 \end{aligned} \quad (3)$$

3차원( $d=3$ )일 때는 다음과 같이 쓸 수 있다 (수식 4 참조).

$$\begin{aligned} \mathbf{b}(\mathbf{x}) &= [1] \text{ if } m = 0 \\ \mathbf{b}(\mathbf{x}) &= [1, x, y, z] \text{ if } m = 1 \\ \mathbf{b}(\mathbf{x}) &= [1, x, y, z, x^2, y^2, z^2, xy, yz, xz] \text{ if } m = 2 \end{aligned} \quad (4)$$

MLS는 수식 2에서 설명했듯이 모든 값에 제약조건을 곱하여 얻을 수 있다. 제약조건에 따른 가중치는  $\omega(|\mathbf{r}_i|)$ 이며 (여기서,  $\mathbf{r}_i = \mathbf{p}_i - \mathbf{x}$ ), 이때 위 수식은 다음과 같이 하나의 방정식으로 수식화 될 수 있다 (수식 5 참조).

$$\begin{bmatrix} \omega(|\mathbf{r}_1|) \\ \vdots \\ \omega(|\mathbf{r}_N|) \end{bmatrix} \begin{bmatrix} \mathbf{b}^T(|\mathbf{p}_1|) \\ \vdots \\ \mathbf{b}^T(|\mathbf{p}_N|) \end{bmatrix} \mathbf{c} = \begin{bmatrix} \omega(|\mathbf{r}_1|) \\ \vdots \\ \omega(|\mathbf{r}_N|) \end{bmatrix} \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_N \end{bmatrix} \quad (5)$$

이것은 다음과 같이 줄여 쓸 수 있다 (수식 6 참조).

$$\mathbf{W}(\mathbf{x}) \mathbf{B} \mathbf{c}(\mathbf{x}) = \mathbf{W}(\mathbf{x}) \Phi, \quad (6)$$

여기서  $\mathbf{W}$ 는  $N \times N$ 인 대각행렬이고,  $\mathbf{B}$ 는  $N \times k$ 인 행렬,  $\Phi$ 는  $\mathbb{R}^N$ 차원 벡터이다. 이 방정식에 정규방정식(Normal equation)을 적용하면 아래와 같은 수식을 얻을 수 있다.

$$\mathbf{B}^T (\mathbf{W}(\mathbf{x}))^2 \mathbf{B} \mathbf{c}(\mathbf{x}) = \mathbf{B}^T (\mathbf{W}(\mathbf{x}))^2 \Phi, \quad (7)$$

해당 방정식을 선형 시스템으로 해결하면  $\mathbf{c}(\mathbf{x})$ 를 얻을 수 있다. 이를 통해 위치  $\mathbf{x}$ 에서의  $f$ 값은 수식 1을 통해 계산한다. 본 논문에서는 특이값 분해(Singular value decomposition)를 활용하여 계산했다.

##### 3.1.2 벡터 보간

이전 장에서는 스칼라 보간을 벡터장  $\mathbf{u} = [u, v, w]^T$ 으로 확장하는 방법에 대해 설명했다. 각 요소를 단순히 벡터로 대체함으로써  $f$ 에 대한  $\mathbf{u}$ 의 값을 계산할 수 있으며, 이를 통해 수식 8과 같은 제약조건을 얻을 수 있다.

$$\mathbf{u}(\mathbf{p}_i) = \begin{bmatrix} u(\mathbf{p}_i) \\ v(\mathbf{p}_i) \\ w(\mathbf{p}_i) \end{bmatrix} = \begin{bmatrix} \mathbf{b}^T(\mathbf{p}_i \mathbf{c}_x) \\ \mathbf{b}^T(\mathbf{p}_i \mathbf{c}_y) \\ \mathbf{b}^T(\mathbf{p}_i \mathbf{c}_z) \end{bmatrix} = \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} = \mathbf{u}_i. \quad (8)$$

일반적으로 벡터 보간은 각 요소를 독립적으로 보간함으로써

계산할 수 있다. 그러나 요소들 간에 상호 제한이 있고 발산 제약을 충족해야 할 때는 단일 벡터 보간을 재구성해야 한다. 이에 따라, 벡터를 보간하기 위해 수식 2를 다음과 같이 다시 작성한다 (수식 9 참조).

$$\mathbf{W}_{vec}(\mathbf{x}) \mathbf{B}_{vec} \mathbf{c}_{vec}(\mathbf{x}) = \mathbf{W}_{vec}(\mathbf{x}) \Phi_{vec}, \quad (9)$$

여기서

$$\mathbf{W}_{vec} = \begin{bmatrix} \omega(|\mathbf{r}_1|) \\ \omega(|\mathbf{r}_1|) \\ \omega(|\mathbf{r}_1|) \\ \ddots \\ \omega(|\mathbf{r}_N|) \\ \omega(|\mathbf{r}_N|) \\ \omega(|\mathbf{r}_N|) \end{bmatrix}, \mathbf{B}_{vec} = \begin{bmatrix} \mathbf{b}(\mathbf{p}_1) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{b}(\mathbf{p}_1) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{b}(\mathbf{p}_1) \\ \vdots & & \\ \mathbf{b}(\mathbf{p}_N) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{b}(\mathbf{p}_N) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{b}(\mathbf{p}_N) \end{bmatrix} \quad (10)$$

$\mathbf{c}_{vec}$ 는  $[\mathbf{c}_x, \mathbf{c}_y, \mathbf{c}_z]$ 이고,  $\Phi_{vec}$ 는  $[\mathbf{u}_1, \dots, \mathbf{u}_N]$ 이다.  $\mathbf{W}_{vec}$ 는  $dN \times dN$  대각행렬이고,  $\mathbf{B}_{vec}$ 는  $dN \times dk$ 행렬,  $\Phi$ 는  $\mathbb{R}^{dN}$ 차원 벡터이다. 본 논문에서는 스칼라 보간에 대한 수식 6을 풀 때와 같은 방식으로 수식 9을 풀어 벡터 보간을 계산한다.

## 3.2 몬테카를로 가중치를 이용한 발산제약 기반 이 동최소제곱

### 3.2.1 확산 미분

이번 장에서는 확산 도함수(Diffuse derivatives)와 몬테카를로 가중치 기반의 MLS 기법에 대해 소개한다. 수식 1을 미분하면  $f$ 의 정확한 미분 값을 얻을 수 있다 (수식 11 참조).

$$\frac{\partial f}{\partial \mathbf{x}} = \frac{\partial \mathbf{b}^T}{\partial \mathbf{x}} \mathbf{c} + \mathbf{b}^T \frac{\partial \mathbf{c}}{\partial \mathbf{x}} \quad (11)$$

위 식에서 오른쪽 두번째 항은 간단하게 계산될 수 있지만, MLS에 통합하기에는 계산이 복잡해진다. 이 항을 확산 도함수로 교체함으로써 계산 양이 큰 정확한 미분값에 근사치를 효율적으로 얻을 수 있다 [6] (수식 12 참조).

$$\frac{\partial f}{\partial \mathbf{x}} = \frac{\partial \mathbf{b}^T}{\partial \mathbf{x}} \mathbf{c} + \mathbf{b}^T \frac{\partial \mathbf{c}}{\partial \mathbf{x}} \quad (12)$$

확산 도함수를 사용하면 벡터장인  $\nabla \cdot \mathbf{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}$ 의 발산은 다음과 같이 근사할 수 있다 (수식 13 참조).

$$\nabla \cdot \mathbf{u} \simeq \frac{\partial \mathbf{b}^T}{\partial x} \mathbf{c}_x + \frac{\partial \mathbf{b}^T}{\partial y} \mathbf{c}_y + \frac{\partial \mathbf{b}^T}{\partial z} \mathbf{c}_z \quad (13)$$

### 3.2.2 발산제약 조건

Huerta et al.은 발산제약 조건인  $\nabla \cdot \mathbf{u} = \sigma$ 를 사용하여  $\sigma = 0$ 으로 비압축성에 대한 정확도를 개선했다 [41]. 본 논문에서는  $\sigma$

가 0이 아닌 값을 가질 수 있도록 허용하여 속도 보간에서 발산을 제어했다.  $\mathbf{B}_{vec}$ 에 발산제약 조건을 추가하려면 수식 13을 다음과 같이 다시 작성해야 한다 (수식 14 참조).

$$\nabla \cdot \mathbf{u} \simeq \mathbf{b}_{div}^T(\mathbf{x}) \cdot \mathbf{c} \quad (14)$$

여기서  $\mathbf{b}_{div}$ 는 다음과 같이 계산한다 (수식 15 참조).

$$\begin{aligned} \mathbf{b}_{div}(\mathbf{x}) &= \frac{\partial [\mathbf{b}^T(\mathbf{x}) \ 0 \ 0]}{\partial x} + \frac{\partial [0 \ \mathbf{b}^T(\mathbf{x}) \ 0]}{\partial y} + \frac{\partial [0 \ 0 \ \mathbf{b}^T(\mathbf{x})]}{\partial z} \\ &= \left[ \frac{\partial \mathbf{b}^T(\mathbf{x})}{\partial x} \ 0 \ 0 \right] + \left[ 0 \ \frac{\partial \mathbf{b}^T(\mathbf{x})}{\partial x} \ 0 \right] + \left[ 0 \ 0 \ \frac{\partial \mathbf{b}^T(\mathbf{x})}{\partial z} \right] \\ &= \left[ \frac{\partial \mathbf{b}^T(\mathbf{x})}{\partial x} \ \frac{\partial \mathbf{b}^T(\mathbf{x})}{\partial y} \ \frac{\partial \mathbf{b}^T(\mathbf{x})}{\partial z} \right] \end{aligned} \quad (15)$$

$\mathbf{b}$ 의 편미분은  $\mathbf{b}$ 의 각 요소를 개별적으로 미분하여 계산한다. 예를 들어,  $d = 3$ 이고  $m = 2$ 인 경우는 다음과 같다 (수식 16 참조).

$$\begin{aligned} \frac{\partial \mathbf{b}(\mathbf{x})}{\partial x} &= [0, 1, 0, 0, 2x, 0, 0, y, 0, z] \\ \frac{\partial \mathbf{b}(\mathbf{x})}{\partial y} &= [0, 0, 1, 0, 0, 2y, 0, x, z, 0] \\ \frac{\partial \mathbf{b}(\mathbf{x})}{\partial z} &= [0, 0, 0, 1, 0, 0, 2z, 0, y, x] \end{aligned} \quad (16)$$

수식 9에 발산제약 조건인  $\mathbf{b}_{div}^T \cdot \mathbf{c} = \sigma$ 를 넣으면 발산제약 기반 MLS를 계산할 수 있다 (수식 17 참조).

$$\mathbf{W}_{div}(\mathbf{x}) \mathbf{B}_{div} \mathbf{c}_{vec}(\mathbf{x}) = \mathbf{W}_{div}(\mathbf{x}) \Phi_{div} \quad (17)$$

이 방정식은 앞에서 설명한 MLS 방법을 사용하여 계산할 수 있다. 본 논문에서 사용하는 MLS 제약은 다음과 같다:  $\mathbf{W}_{div} = \begin{bmatrix} \mathbf{W}_{vec} \\ \omega_{div} \end{bmatrix}$ ,  $\mathbf{B}_{div} = \begin{bmatrix} \mathbf{b}_{vec} \\ \omega_{div} \end{bmatrix}$ ,  $\Phi_{div} = [\Phi, \sigma]$ .  $\mathbf{W}_{div}$ 는  $(dN + 1) \times (dN + 1)$ 크기의 대각행렬이며,  $\mathbf{B}_{div}$ 는  $(dN + 1) \times dk$ 크기의 행렬이고,  $\Phi_{div}$ 는  $\mathbb{R}^{dN+1}$ 크기의 벡터이다.

### 3.2.3 몬테카를로 기반 가중치

몬테카를로법은 문제를 수치 계산으로 해결하는 대신, 확률적인 난수를 활용하여 근사적으로 해결하는 방법이다. 난수 발생 횟수의 증가와 균일한 분포 여부에 따라 정확도가 향상되며, 몬테카를로법을 이용하여 원주율을 계산하는 방법은 가장 간단하고 널리 알려진 예시 중 하나이다.

우선, 하나의 정사각형과 해당 정사각형의 한 변 길이를  $\frac{1}{4}$ 로 하는 반지름을 가진 원을 그리고, 정사각형 내부에  $N$ 개의 난수를 생성한다 (Figure 3 참조). 이후 원의 중심과 각 난수 사이의 거리를 계산하여 원 내부에 속하는지 여부를 판별한다. 난수가 충분히 생성되었다면, 수식 18과 같은 관계식이 성립한다. 이때  $\alpha$ 는 원 내부에 속하는 난수의 개수이며,  $N$ 은 전체 난수의 개수이다.



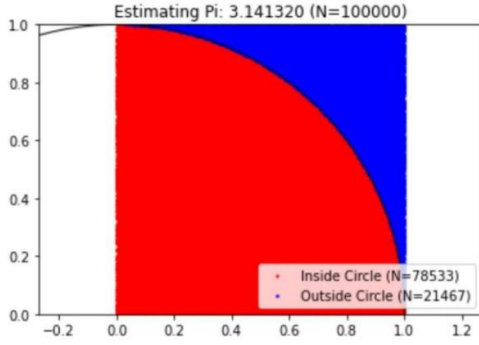


Figure 3: Calculate  $\pi$  with Monte Carlo method.

$$\frac{\pi}{4} : 1 = a : N, \pi = \frac{4a}{N} \quad (18)$$

본 논문에서는 이미지 내 밀도와 밀집된 위치를 추정하기 위해 몬테카를로법을 사용하며, 이는 원주율을 계산하는 원리와 유사하다. 수식 19에서  $d$ 는 추정하려는 밀도이며,  $a$ 는 임계값 이상의 밀도를 갖는 영역에 해당하는 난수의 개수를 나타내고,  $N$ 은 전체 난수의 개수이다.

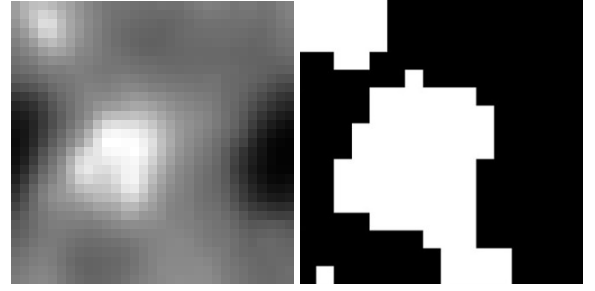
$$d : 1 = a : N, d = \frac{a}{N} \quad (19)$$

본 논문에서는 효율적인 밀도 검출을 위해 밀도를 먼저 이진화한다 (이 과정은 선택이며 생략이 가능하다). Figure 4a는 원본 밀도 데이터이며, Figure 4b는 임계값을 0.5로 설정하여 이진화한 결과이다. Figure 4c와 Figure 4d는 각각 1,000개와 100,000개의 난수 샘플을 사용해 밀도를 몬테카를로법으로 추정한 결과이다. 원본 밀도의 평균이 0.3476인 경우, Figure 4c의 추정값은 0.324로 측정되었고, Figure 4d의 추정값은 0.3452로 계산되었으며, 난수 샘플 수가 많은 Figure 4d가 더 정확한 근사한 값을 추론하였다. 이를 바탕으로 이미지 내에서 밀도가 밀집한 위치 추정된다. Figure 4c와 Figure 4d에 나타난 녹색 지점은 추정된 밀도의 밀집 위치를 나타낸다. Figure 4c의 경우 밀집 지역은 각 시뮬레이션마다 약간씩 다르게 나타났지만, Figure 4d는 거의 일정한 결과를 보여주었다.

몬테카를로법으로 계산된 밀도의 밀집 위치와 각 제약 노드 간의 거리를 계산하고, 제약 노드의 결과를 0~1 범위로 정규화한다. 몬테카를로법을 통해 MLS 가중치인  $\omega^*(|\mathbf{r}_i|)$ 를 구성하는 방법은 다음과 같다 (수식 20 참조).

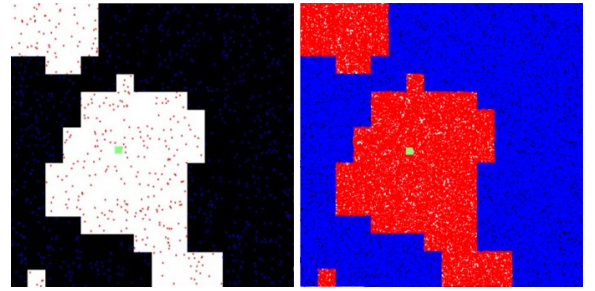
$$\omega^*(|\mathbf{r}_i|) = \omega(|\mathbf{r}_i|) (\mathbf{q}_i - \mathbf{x}) \quad (20)$$

여기서  $\omega(|\mathbf{r}_i|)$ 는 노드와 제약 노드 간의 거리를 계산한 가중치이고,  $(\mathbf{q}_i - \mathbf{x})$ 는 노드와 몬테카를로법으로 추정된 밀도의 밀집 위치 사이의 거리이며, 이 값을 통해 가중치 행렬인  $\mathbf{W}$ 를 구성하여 MLS를 계산한다.



(a) Original density map

(b) Discretized density



(c) Predicted density location

(d) Sampling position with Monte Carlo estimation

Figure 4: Calculate densely location with Monte Carlo method.

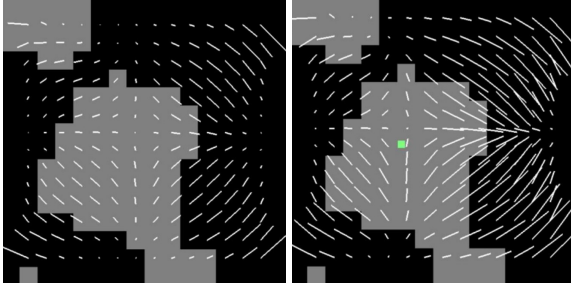
## 4 실험 결과

본 연구의 결과들을 만들기 위해 실험한 환경은 Intel Core i7-7700K CPU, 32GB RAM, Geforce GTX 1080Ti GPU가 탑재된 컴퓨터를 이용하였다. 본 논문에서 보여주는 결과는 2차원 밀도 맵을 가진 데이터를 통해 벡터장을 표현했다.

Figure 5는 제약 노드를 가지는 장면에서 벡터장을 추출한 결과이다. Figure 5a는 4개의 각 코너에 제약노드를 가지며 ( $p1(0.0558, -0.9984), p2(-0.4041, 0.9147), p3(-0.6049, -0.7962), p4(-0.7164, 0.6976)$ ), 이전 기법인 발산제약 MLS를 통해 계산된 벡터장이다. 이전 기법에서는 밀도 데이터의 특성을 고려하지 못하고, 제약 노드의 방향만 고려한 벡터장이 나타나는 것을 볼 수 있다. Figures 5b는 앞의 결과와 같은 조건에서 몬테카를로 가중치를 추가하여 MLS를 계산한 결과이다. 같은 제약 노드 환경임에도 불구하고 Figures 5a의 결과와는 다르게 제약 노드의 형태를 유지하면서 밀도의 밀집 위치를 기반으로 벡터장의 흐름이 디테일하게 표현되었다.

Figure 6은 Figure 5와는 다른 제약 노드로 실험한 결과이다 ( $p1(0.9177, 0.397), p2(0.9386, -0.3448), p3(0.2805, -0.9598), p4(0.9249, 0.3802)$ ). 이전 기법에서는 제약 노드에 영향을 받기 때문에 발산제약 MLS임에도 불구하고 벡터장의 크기가 작아 거의 표현되지 않는 경우도 있지만, 제안하는 방법은 밀도가 큰 영역을 중심으로 와류가 소실되지 않고 명확하게 표현되었다.

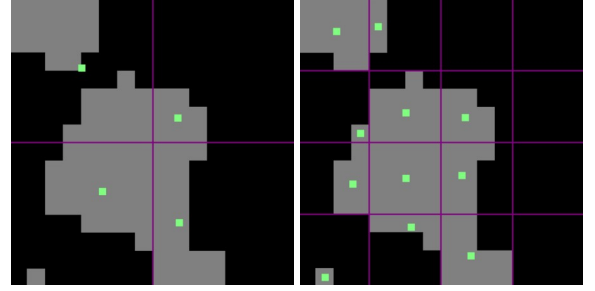
앞에서 보여준 결과들은 밀도의 밀집 위치가 하나로 표현되기



(a) Previous method [6]

(b) Our method

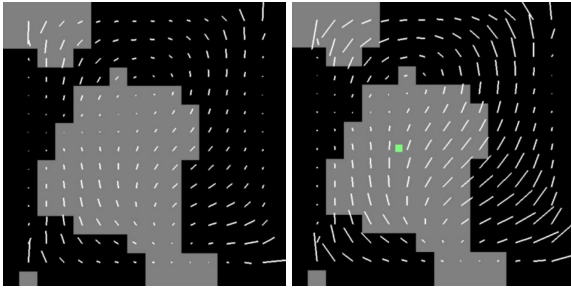
Figure 5: Comparison of vector field results generated by the previous method and our method.



(a)  $2 \times 2$

(b)  $4 \times 4$

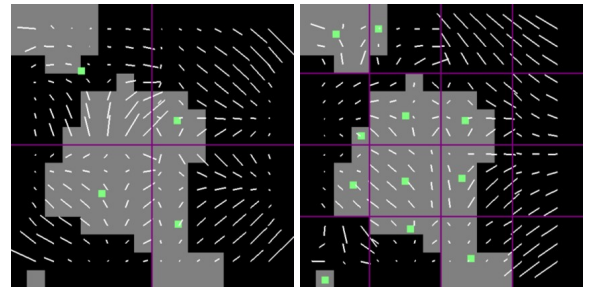
Figure 7: Grid partitioning scheme for Monte Carlo weight control.



(a) Previous method [6]

(b) Our method

Figure 6: Comparison of vector field results generated by the previous method and our method (constrained nodes).



(a) Our method ( $2 \times 2$ )

(b) Our method ( $4 \times 4$ )

Figure 8: Generation of vector field with our method.

때문에 밀도의 특징을 온전히 고려하기에는 부족하다. 이를 효율적으로 처리하기 위해 본 논문에서는 밀도를 다루는 공간을 격자로 분할하여 격자마다 몬테카를로 가중치를 계산하도록 했다. Figure 7은 분할된 격자마다 계산된 밀도의 밀집 위치를 보여주고 있다. 앞에서 보여준 결과와 다르게 격자 분할 해상도가 높을수록 밀도의 밀집 위치가 정확하게 추론되는 것을 볼 수 있다.

Figure 8은 다양한 격자 해상도를 통해 몬테카를로 가중치 MLS를 계산한 결과이다 ( $p1(0.0558, -0.9984)$ ,  $p2(-0.4041, 0.9147)$ ,  $p3(-0.6049, -0.7962)$ ,  $p4(-0.7164, 0.6976)$ ). 몬테카를로 부터 밀도의 밀집 위치가 여러 개가 나올 수 있고, MLS에 필요한 가중치는 노드 위치에 해당하는 밀도의 밀집 위치를 이용하였다. 같은 장면인 Figure 5와 비교했을 때 제안하는 기법은 밀도의 형태에 따라 와류 및 벡터장의 흐름을 디테일하게 잘 표현했다.

## 5 결론 및 향후 연구

본 논문에서는 밀도의 분포를 고려하여 MLS기반으로 벡터장을 생성하는 새로운 방식을 제안했다. 격자 내 밀도 분포를 효율적으로 추론하기 위해 입자 기반의 몬테카를로 시뮬레이션 기법을 도입했으며, 이로 인해 격자 해상도를 증가시키지 않고도 디테일

한 벡터장을 계산했다. 하지만 정규격자로 분할하여 밀도의 밀집 위치를 계산하기 때문에 밀도가 고르게 분포되어 있지 않을 경우에는 약간의 오차가 발생한다. 향후 이 문제를 효율적으로 해결하기 위해 쿼드트리(Quadtree) 기반의 적응형 공간 분할 기법을 이용하여 알고리즘을 개선할 계획이다. 또한 격자 내 표현되는 서브그리드(Subgrid) 디테일을 개선하기 위해 세미 라그랑주 방식을 몬테카를로 기반 MLS와 통합하여 디테일하게 연기와 불의 비주얼 시뮬레이션을 개선시킬 계획이다. 본 논문에서는 입력 데이터가 밀도 이미지이지만, 시뮬레이션에서는 연속적인 밀도 데이터를 이용하기 때문에 시간에 따른 연속성이 보장되어야 하며, 그렇지 않을 경우 깜빡이는 노이즈 문제가 발생할 수 있다. 향후 위에서 언급한 문제들을 개선시킬 수 있도록 연구를 확장할 계획이다.

## References

- [1] C. Felter, J. H. Walther, and C. Henriksen, "Moving least squares simulation of free surface flows," *Computers & Fluids*, vol. 91, pp. 47–56, 2014.
- [2] L. White, R. Panchadhara, and D. Tenev, "Flow simulation in heterogeneous porous media with the moving least-squares

- method,” *SIAM Journal on Scientific Computing*, vol. 39, no. 2, pp. B323–B351, 2017.
- [3] Y. Zhu and S. J. Gortler, “3d deformation using moving least squares,” 2007.
- [4] S. Fleishman, D. Cohen-Or, and C. T. Silva, “Robust moving least-squares fitting with sharp features,” *ACM transactions on graphics (TOG)*, vol. 24, no. 3, pp. 544–552, 2005.
- [5] C. Kang, T. Lu, M. Zong, F. Wang, and Y. Cheng, “Point cloud smooth sampling and surface reconstruction based on moving least squares,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, pp. 145–151, 2020.
- [6] J.-M. Hong, J.-C. Yoon, and C.-H. Kim, “Divergence-constrained moving least squares for fluid simulation,” *Computer Animation and Virtual Worlds*, vol. 19, no. 3-4, pp. 469–477, 2008.
- [7] S.-T. Kim and J.-M. Hong, “Visual simulation of turbulent fluids using mls interpolation profiles,” *The Visual Computer*, vol. 29, pp. 1293–1302, 2013.
- [8] A. Cuno, C. Esperança, A. Oliveira, and P. R. Cavalcanti, “3d as-rigid-as-possible deformations using mls,” in *Proceedings of the 27th computer graphics international conference*. Citeseer, 2007, pp. 115–122.
- [9] T. Sato, T. Igarashi, C. Batty, and R. Ando, “A long-term semi-lagrangian method for accurate velocity advection,” in *SIGGRAPH Asia 2017 Technical Briefs*, 2017, pp. 1–4.
- [10] J. Stam, “Stable fluids,” in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH, no. 8, 1999, pp. 121–128.
- [11] B. Kim, Y. Liu, I. Llamas, and J. Rossignac, “Advections with significantly reduced dissipation and diffusion,” *IEEE transactions on visualization and computer graphics*, vol. 13, no. 1, pp. 135–144, 2006.
- [12] O.-Y. Song, H. Shin, and H.-S. Ko, “Stable but nondissipative water,” *ACM Transactions on Graphics (TOG)*, vol. 24, no. 1, pp. 81–97, 2005.
- [13] D. Kim, O.-y. Song, and H.-S. Ko, “A semi-lagrangian cip fluid solver without dimensional splitting,” in *Computer Graphics Forum*, vol. 27, no. 2. Wiley Online Library, 2008, pp. 467–475.
- [14] S. Sato, Y. Dobashi, and T. Kim, “Stream-guided smoke simulations,” *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–7, 2021.
- [15] Z. Forootaninia and R. Narain, “Frequency-domain smoke guiding,” *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–10, 2020.
- [16] S. He, H.-C. Wong, and U.-H. Wong, “An efficient adaptive vortex particle method for real-time smoke simulation,” in *2011 12th International Conference on Computer-Aided Design and Computer Graphics*. IEEE, 2011, pp. 317–324.
- [17] C. Fu, Q. Guo, T. Gast, C. Jiang, and J. Teran, “A polynomial particle-in-cell method,” *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, pp. 1–12, 2017.
- [18] B. E. Feldman, J. F. O’Brien, and B. M. Klingner, “Animating gases with hybrid meshes,” *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 904–909, 2005.
- [19] N. Foster and D. Metaxas, “Realistic animation of liquids,” *Graphical models and image processing*, vol. 58, no. 5, pp. 471–483, 1996.
- [20] —, “Modeling the motion of a hot, turbulent gas,” in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997, pp. 181–188.
- [21] R. Fedkiw, J. Stam, and H. W. Jensen, “Visual simulation of smoke,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 15–22.
- [22] A. Selle, N. Rasmussen, and R. Fedkiw, “A vortex particle method for smoke, water and explosions,” in *ACM SIGGRAPH 2005 Papers*, 2005, pp. 910–914.
- [23] J. Kim, D. Cha, B. Chang, B. Koo, and I. Ihm, “Practical animation of turbulent splashing water,” in *Symposium on Computer Animation*, 2006, pp. 335–344.
- [24] 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.
- [25] S. T. Greenwood and D. H. House, “Better with bubbles: enhancing the visual realism of simulated fluid,” in *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2004, pp. 287–296.

- [26] J.-M. Hong, H.-Y. Lee, J.-C. Yoon, and C.-H. Kim, “Bubbles alive,” *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, pp. 1–4, 2008.
- [27] N. Thürey, F. Sadlo, S. Schirm, M. Müller-Fischer, and M. Gross, “Real-time simulations of bubbles and foam within a shallow water framework,” in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2007, pp. 191–198.
- [28] Y. Gao, C.-F. Li, S.-M. Hu, and B. A. Barsky, “Simulating gaseous fluids with low and high speeds,” in *Computer Graphics Forum*, vol. 28, no. 7. Wiley Online Library, 2009, pp. 1845–1852.
- [29] J.-M. Hong and C.-H. Kim, “Discontinuous fluids,” *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 915–920, 2005.
- [30] J.-M. Hong, T. Shinar, and R. Fedkiw, “Wrinkled flames and cellular patterns,” *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, pp. 47–es, 2007.
- [31] D. Q. Nguyen, R. Fedkiw, and H. W. Jensen, “Physically based modeling and animation of fire,” in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 2002, pp. 721–728.
- [32] D. Kim, O.-y. Song, and H.-S. Ko, “Stretching and wiggling liquids,” in *ACM SIGGRAPH Asia 2009 papers*, 2009, pp. 1–7.
- [33] T. Kim, N. Thürey, D. James, and M. Gross, “Wavelet turbulence for fluid simulation,” *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, pp. 1–6, 2008.
- [34] R. Narain, J. Sewall, M. Carlson, and M. C. Lin, “Fast animation of turbulence using energy transport and procedural synthesis,” *ACM Transactions on Graphics (TOG)*, vol. 27, no. 5, pp. 1–8, 2008.
- [35] H. Schechter and R. Bridson, “Evolving sub-grid turbulence for smoke animation,” in *Proceedings of the 2008 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2008, pp. 1–7.
- [36] M. Chu and N. Thuerey, “Data-driven synthesis of smoke flows with cnn-based feature descriptors,” *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–14, 2017.
- [37] S. Sato, Y. Dobashi, T. Kim, and T. Nishita, “Example-based turbulence style transfer,” *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–9, 2018.
- [38] X.-S. Chen, C.-F. Li, G.-C. Cao, Y.-T. Jiang, and S.-M. Hu, “A moving least square reproducing kernel particle method for unified multiphase continuum simulation,” *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–15, 2020.
- [39] R. Sawhney and K. Crane, “Monte carlo geometry processing: A grid-free approach to pde-based methods on volumetric domains,” *ACM Transactions on Graphics*, vol. 39, no. 4, 2020.
- [40] Y. Xu, T. Fan, Y. Yuan, and G. Singh, “Ladybird: Quasi-monte carlo sampling for deep implicit field based 3d reconstruction with symmetry,” in *European Conference on Computer Vision*. Springer, 2020, pp. 248–263.
- [41] A. Huerta, Y. Vidal, and P. Villon, “Pseudo-divergence-free element free galerkin method for incompressible fluid flow,” *Computer methods in applied mechanics and engineering*, vol. 193, no. 12-14, pp. 1119–1136, 2004.



## 〈 저 자 소 개 〉



김 종 현

- 2008년 세종대학교 컴퓨터학과 학사
- 2010년 고려대학교 컴퓨터학과 석사
- 2016년 고려대학교 컴퓨터학과 박사
- 2013년~2016년 (주)텐일레븐 선임연구원
- 2023년~현재 인하대학교  
소프트웨어융합대학(디자인테크놀로지학과)  
부교수
- 관심분야 : 물리 기반 시뮬레이션, 가상/  
증강현실, 지오메트리 프로세싱, 게임 물리, 게임  
AI
- <https://orcid.org/0000-0003-1603-2675>