

# 발산 제약 이동최소제곱법을 이용한 평활화된 입자 유체역학 기반 벡터장 설계

김종현\*

인하대학교\*

jonghyunkim@inha.ac.kr

## Designing Vector Fields on Smoothed Particle Hydrodynamics Using Divergence-Constrained Moving Least Squares

Jong-Hyun Kim\*

Inha University\*

### 요약

본 연구에서는 SPH(Smoothed particle hydrodynamics) 기반의 입자 데이터를 활용하여 다양한 벡터장 패턴을 효과적으로 시각화할 수 있는 새로운 기법을 제안한다. 제안된 방법은 물리 기반 시뮬레이션에서 널리 사용되는 다항식 보간 방식인 이동최소제곱(Moving Least Squares, MLS)을 응용한다. 그러나 기존의 MLS는 격자 공간을 전제로 고차 보간을 수행하며, 벡터 제약 조건을 포함하기 때문에 SPH와 같은 비정형 입자 데이터에 직접 적용하기에는 한계가 있다. 이를 해결하기 위해 본 논문에서는 입자들이 공간상에 불규칙하게 분포되어 있다는 특성을 고려하여, SPH 커널 기반의 비등방성 가중치를 활용해 입자 밀도를 추정하고 이를 MLS 프레임워크에 통합하였다. 제안된 알고리즘은 다양한 형태의 벡터장을 입자 기반 데이터에서도 정밀하게 재현할 수 있도록 설계되었으며, 기존 격자 기반 MLS나 발산 제약형 MLS로는 포착하기 어려운 미세한 벡터장 구조들을 시각화 실험을 통해 효과적으로 표현함을 보여준다.

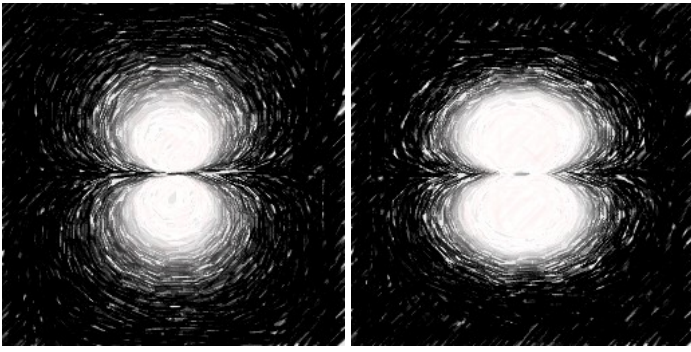
### Abstract

This study proposes a novel method for effectively visualizing various vector field patterns based on Smoothed Particle Hydrodynamics (SPH) particle data. The proposed approach utilizes the Moving Least Squares (MLS) technique, a polynomial interpolation method widely used in physics-based simulations. However, conventional MLS assumes a structured grid and relies on vector-based constraints for higher-order interpolation, making it difficult to apply directly to unstructured particle data such as SPH. To address this limitation, we incorporate SPH kernel-based anisotropic weighting to estimate particle density and integrate it into the MLS framework, taking into account the irregular spatial distribution of particles. The resulting algorithm is designed to accurately represent a wide range of vector field structures within particle-based datasets. Experimental results demonstrate that our method effectively captures fine vector field details that are challenging to express using traditional grid-based MLS or divergence-constrained MLS techniques.

**키워드:** 벡터장 설계, 발산 제약 이동최소제곱, 비등방성 가중치, 시각화, 평활화된 입자 유체역학

**Keywords:** Designing vector fields, Divergence-constrained moving least squares, anisotropic weighting, Visualization, Smoothed particle hydrodynamics

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



(a) Naive MLS (b) Divergence-constrained MLS [6]

Figure 1: In comparison to the standard MLS vector interpolation, the pseudo-divergence-free vector interpolation [6].

## 1 서론

이동최소제곱법(Moving Least Squares, MLS)은 흩어진 데이터를 부드럽게 근사하거나 고차로 보간하는 데 널리 사용되는 기법이다. 확률 밀도 함수에 MLS를 적용하면 보다 정확한 분포 추정이 가능해지기 때문에, 유체 시뮬레이션 [1, 2], 변형체 모델링 [3], 메시화 [4, 5] 등 다양한 분야에서 활용되고 있다. 다항식 보간은 샘플 지점 간의 값을 추정하는 데 사용되며, 연속적인 물질의 움직임을 표현해야 하는 유체 시뮬레이션이나 캐릭터 모델링에서 특히 중요하다. 이때 적절한 보간 방식의 선택은 결과 품질에 직접적인 영향을 미친다 [6, 7, 8].

물리 기반 시뮬레이션에서 보간을 통한 샘플링 데이터 추정 은 핵심적인 과정이다. 기하 처리 분야에서는 점군(Point Cloud) 데이터로부터 메시 표면을 근사 생성하는 데 이 기법이 사용된다. 점군 데이터는 연결 정보가 없기 때문에, 고차 수치 기법을 통해 암시적으로 삼각형 기반 메시를 생성해야 한다 [4]. 물리 시뮬레이션에서도 시뮬레이션 정확도를 향상시키기 위해 고차 보간 기법이 자주 사용된다 [9]. 예를 들어, 연기 시뮬레이션에서 이류(advection) 항에 사용되는 선형 보간 기반의 세미-라그랑주(Semi-Lagrangian) 방법 [10]을 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], 주파수 기반 난류 [15], 적응형 와류 기법 [16], Polynomial PIC [17] 등의 다양한 방법이 제안되었다.

Feldman 등은 적응형 격자 구조에서 부드러운 벡터장을 얻기 위해 속도의 수직 성분(normal component)을 기반으로 MLS를 계산하였다 [18]. 이처럼 MLS는 고차 보간을 위한 유용한 도구이지만, 연속성 조건이나 물리적 제약조건을 충족시키기 어려워 실제 시뮬레이션에서는 활용에 제약이 있다. 이를 해결하고자, Hong 등은 발산 제약 기반 MLS(Divergence-Constrained MLS)를

제안하였으며, 이는 포아송 방정식을 푸는 과정을 생략하면서도 유체와 유사한 움직임을 갖는 벡터장을 생성할 수 있게 한다 [6]. 이 기법은 연기 시뮬레이션에 적용되어 복잡한 난류 현상을 효과적으로 재현하였다 [7].

본 연구에서는 밀도 기반 입자 데이터로부터 벡터장을 시각화하기 위한 새로운 접근으로 몬테카를로 가중치를 적용한 MLS 기법을 제안한다. Figure 1은 기존 MLS와 Hong et al.이 제안한 발산 제약 MLS 간의 벡터 보간 차이를 보여준다 [6]. Figure 1a는 서로 반대 방향의 벡터 두 개를 제약 조건으로 사용하여 MLS를 적용한 결과이며, Figure 1b는 발산 제약 기반 MLS를 활용해 생성한 벡터장이다. 발산 제약 기반 MLS는 비압축성을 만족시키는 보간 기법으로, 유체 시뮬레이션에 적합한 특징을 지닌다 (Figure 1b 참조). 일반적인 MLS와 비교했을 때 벡터장 구조에서 뚜렷한 차이를 보이며(Figure 1a 참조), 발산 조건을 통해 다양한 형태의 벡터장을 설계할 수 있다는 장점이 있다.

하지만, 이 방식은 격자 공간에서 벡터 제약만을 고려하기 때문에 오일러(Eulerian) 기반 시뮬레이션에만 적용이 가능하다는 한계가 있다. 이를 보완하기 위해 본 논문에서는 SPH 접근법을 통해 입자 밀도가 분포하는 위치를 추정하고, 이 위치와 주변 입자 간의 거리를 MLS 내 추가 가중치로 반영함으로써 SPH 입자 데이터의 분포를 고려한 벡터장 시각화를 가능하게 한다.

## 2 관련 연구

유체 시뮬레이션 분야에서는 난류의 시각적 세부 묘사를 향상시키기 위해 다양한 이류(Advection)와 외력(External force) 기법들이 활발히 연구되어 왔다 [19, 20, 10]. 대표적으로 세미 라그랑지안 방법은 유체의 안정적인 시뮬레이션을 위해 고안되었으며 [10], 와류 제한(Vorticity confinement) 기법은 격자 기반 환경에서 난류의 와류 힘을 효과적으로 추정하여 적용하였다 [21]. 이후 라그랑지안 입자를 와류 전달 매개체로 사용하는 와류 입자법으로 발전하는 과정을 거쳤다 [22]. 그 외에도 BFECC [11], CIP [12], USCIP [13], 스트림 기반 연기 표현 [14], 주파수 기반 연기 [15], 적응형 와류 기법 [16], 그리고 Polynomial PIC [17] 등 여러 기법들이 제안되어 왔다.

특히 입자 기반 시뮬레이션을 통해 서브그리드(Subgrid) 수준의 유체 디테일을 강화하려는 시도도 많았다. Kim et al.[23]과 Losasso et al.[24]는 SPH(Smoothed Particle Hydrodynamics) 모델을 도입하여 서브그리드 스플래시 현상을 효과적으로 처리하였다. 다상 유체의 정밀한 표현을 위해 입자 혼합 모델을 제안한 연구로는 Greenwood와 House [25], Hong et al.[26], Thuerey et al.[27]의 사례가 있으며, Gao et al.은 라그랑지안 입자를 활용해 기체 흐름의 고속 특성을 포착하는 연구를 수행하였다 [28]. 또한, 서브그리드 역학에 물리 모델을 결합함으로써 다상 인터페이스 [29], 화염의 주름 현상 [30, 31], 그리고 액체 표면의 세밀한 디테일을 더욱 향상시키는 연구가 이어졌다 [32]. Hong et al.은 발산 제한 MLS 방법을 개발하여 포아송 방정식의 계산 없이도

유체 움직임은 효과적으로 표현하는 방식을 제안하였다 [6]. 이 기법은 이후 연기 시뮬레이션 내 이류 과정에 통합되어 난류 표현에 활용되었으나 [7], MLS 기반 난류는 디테일보다는 노이즈가 두드러져 시뮬레이션 제어가 어렵고, 고차 방정식으로 인한 계산 비용이 커 산업적 적용에는 제한이 따른다.

이와는 별개로, 후처리 단계에서 유체 흐름의 디테일을 보강하는 방법들도 연구되고 있다. 웨이블릿 노이즈 [33]와 쉘 노이즈 [34, 35] 같은 노이즈 함수를 활용해 저해상도 시뮬레이션에 난류 움직임을 합성하는 접근이 대표적이다. 최근에는 Chu와 Thuerey가 합성곱 신경망을 이용해 저해상도와 고해상도 유동 사이의 관계를 학습하는 연구를 진행했다 [36]. 또한 Sato et al.은 패치 기반 텍스처 합성 기법을 통해 저해상도 유체 흐름에 난류 디테일을 효과적으로 전달하는 방법을 제안하였다 [37]. Chen et al.은 연속체 역학과 MLS를 결합하여 종이와 유체의 상호작용 및 찢김 현상을 동시에 시뮬레이션하는 통합적인 프레임워크를 개발하였다 [38]. 그러나 이러한 후처리 기법들은 노이즈 함수 기반 특성상 실제 고해상도 시뮬레이션보다 유동의 사실감이 다소 떨어지는 한계가 있다.

MLS를 SPH 기반 유체 시뮬레이션의 경계 처리에 직접 적용한 연구도 있다 [39]. SPH 시뮬레이션에서는 일반적으로 경계 입자 (boundary particle)를 활용해 경계를 처리하지만, 평면에 가까운 경계에서는 노이즈가 섞인 불규칙한 움직임이 발생하는 문제가 있었다. 이를 개선하기 위해 MLS를 적용하여 노이즈 없이 보다 부드럽게 경계를 처리했으나, MLS를 경계 처리에만 사용했기 때문에 다양한 벡터장을 직접 생성하거나 설계하는 데는 한계가 있다.

본 연구에서는 MLS 가중치 산출에 SPH 기법을 활용하고 있으나, SPH가 다양한 분야에서 널리 사용되는 샘플링 기법임을 밝힌다. Yu et al.이 제안한 비등방성 형태의 SPH 커널을 계산하여 이를 MLS에 통합함으로써 [40], SPH 입자의 속도가 수치적으로 소실되거나 과도하게 스무딩되는 문제를 완화할 수 있다. 이로 인해 본 방법은 SPH 기반 속도장 설계뿐만 아니라, 물리 기반 시뮬레이션에서 나타나는 난류 모델링에도 효과적으로 적용할 수 있다.

### 3 제안하는 프레임워크

본 논문에서는 제안하는 발산 제약 기반의 MLS 기법을 소개하기에 앞서, 먼저 스칼라 및 벡터 보간을 위한 기본적인 MLS 방법에 대해 설명한다. 이후, ‘SPH 비등방성 가중치 기반 MLS’ 장에서는 SPH 기법을 활용하여 비등방성 가중치를 계산하고, 이를 MLS 프레임워크에 통합하는 방식을 제안한다. 이 과정을 통해 최종적으로 벡터장을 효과적으로 구성할 수 있는 방법론을 제시한다.

Figure 2는 본 논문에서 제안하는 SPH 입자 기반 방법으로 계산된 벡터장을 시각화한 것이다. 그림의 왼쪽에 있는 빨간색 점은 제약 입자를 나타내며, 제약 벡터가 주어졌을 때 제안하는 기법을 통해 해당 벡터장 전체가 자동으로 생성되는 과정을 보여준다.

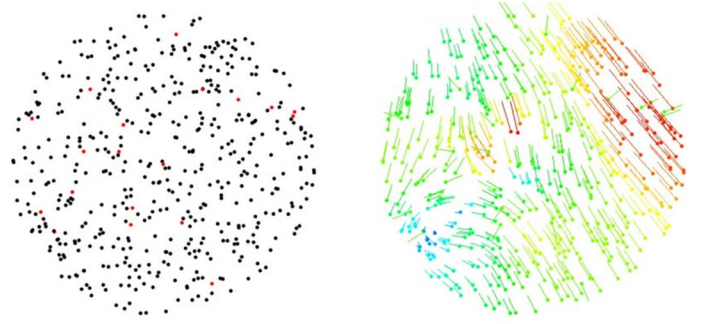


Figure 2: Algorithm overview of our method.

일반적으로 SPH 방식은 커널 함수로 인해 주변 입자의 속도가 소실되거나 과도하게 스무딩되는 경향이 있으나, 본 방법은 주변 입자의 속도를 제약 벡터로 설정함으로써 보다 난류적인 형태를 효과적으로 표현할 수 있다.

### 3.1 이동최소제곱

#### 3.1.1 스칼라 보간

특정 영역 내에  $N$ 개의 제어점  $p_i$ 가 분포해 있을 경우, 이들을 기반으로 구성된 다항식 표현은 다음과 같다 (식 1 참조).

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

이때, 각 제어점  $p_i$ 에서 함수  $f$ 는 주어진 근사값  $\phi^i$ 를 만족해야 하며, 이는 아래와 같이 표현된다 (식 2 참조).

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

여기서 위치  $\mathbf{x}$ 는 2차원 공간에서는  $[x, y]$ , 3차원에서는  $[x, y, z]$ 로 나타나며,  $\mathbf{b}(\mathbf{x}) = [b_1(\mathbf{x}), \dots, b_k(\mathbf{x})]$ 는 다항식 기저함수 벡터이고,  $\mathbf{c} = [c_1, \dots, c_k]^T$ 는 결정해야 할 계수 벡터이다. 기저 함수의 개수  $k$ 는 다음과 같이 정의된다:  $k = \frac{(d+m)!}{m!d!}$ , 여기서  $d$ 는 공간 차원,  $m$ 은 다항식 차수를 의미한다. 예를 들어 2차원 공간에서는 다음과 같은 형태를 갖는다 (식 3 참조):

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

3차원의 경우는 아래와 같이 구성된다 (식 4 참조):

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

Moving Least Squares(MLS) 보간은 이러한 기저함수에 가중치를 부여하여 이루어지며, 각각의 제어점에 대한 오차를 최소화하는 방향으로 계수  $\mathbf{c}$ 를 구하게 된다. 가중치는  $\omega(|\mathbf{r}_i|)$ 로 표현되

며,  $\mathbf{r}_i = \mathbf{p}_i - \mathbf{x}$ 는 제어점과 평가 지점 간의 거리이다. 이를 통해 다음과 같은 선형 방정식 시스템이 구성된다 (식 5 참조):

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

이를 보다 간결하게 행렬 형식으로 표현하면 다음과 같다 (식 6):

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

여기서  $\mathbf{W}(\mathbf{x})$ 는 대각 원소로 가중치가 구성된  $N \times N$  행렬이며,  $\mathbf{B}$ 는  $N \times k$  차원의 기저행렬,  $\Phi$ 는 관측값 벡터이다. 이 선형 시스템에 정규방정식을 적용하면 아래와 같은 형태로 계수를 계산할 수 있다:

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

위의 선형 시스템을 해석함으로써 계수 벡터  $\mathbf{c}(\mathbf{x})$ 를 구할 수 있으며, 이를 다시 식 1에 대입함으로써 임의의 위치  $\mathbf{x}$ 에서의 함수 값  $f(\mathbf{x})$ 를 계산할 수 있다. 본 연구에서는 이 연산을 효율적으로 수행하기 위해 특이값 분해(Singular Value Decomposition, SVD)를 활용하였다.

### 3.1.2 벡터 보간

앞선 절에서는 스칼라 함수  $f$ 에 대한 보간 기법을 벡터장  $\mathbf{u} = [u, v, w]^T$ 에 확장하는 절차를 기술하였다. 스칼라 함수 대신 벡터장의 각 성분에 동일한 방식의 보간을 적용함으로써, 제어점  $\mathbf{p}_i$ 에서의 벡터장 값을 다음과 같이 구할 수 있다 (식 8 참조).

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

벡터 보간은 각 성분에 대해 개별적인 보간을 수행함으로써 구성할 수 있다. 하지만, 성분 간의 물리적 또는 수학적 제약(예: 발산 제로 조건 등)이 존재하는 경우, 이들 보간을 별도로 처리하는 대신 전체 벡터장을 일관되게 재구성해야 한다. 이러한 목적을 위해 스칼라 보간에 사용된 기본 식 (식 2)을 벡터장의 경우로 확장하여 다음과 같이 표현할 수 있다 (식 9 참조).

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

여기서 각각의 행렬과 벡터는 다음과 같이 구성된다 (식 10 참조):

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

이때, 계수 벡터는  $\mathbf{c}_{\mathbf{vec}} = [\mathbf{c}_x^T, \mathbf{c}_y^T, \mathbf{c}_z^T]^T$ 이며, 관측된 벡터장 값은  $\Phi_{\mathbf{vec}} = [\mathbf{u}_1^T, \dots, \mathbf{u}_N^T]^T$ 로 구성된다. 가중치 행렬  $\mathbf{W}_{\mathbf{vec}}$ 은  $dN \times dN$ 의 대각행렬이고, 보간 기저 행렬  $\mathbf{B}_{\mathbf{vec}}$ 은  $dN \times dk$  형태를 가지며,  $\Phi_{\mathbf{vec}} \in \mathbb{R}^{dN}$ 이다.

본 논문에서는 스칼라 보간과 동일하게 정규 방정식을 통해 위 선형 시스템을 풀고,  $\mathbf{c}_{\mathbf{vec}}$ 을 구하여 최종적으로 벡터장 보간 결과를 도출하였다.

## 3.2 SPH의 비등방성 가중치를 이용한 발산제약 기반 이동최소제곱

### 3.2.1 확산 미분

이번 절에서는 확산 도함수(diffuse derivative) 개념과 몬테카를로 기반 가중치를 사용하는 MLS(Moving Least Squares) 기법을 중심으로 설명한다. 우선 함수  $f$ 의 정확한 미분 값을 얻기 위해 수식 1을 미분하면 다음과 같은 결과를 얻을 수 있다 (수식 11 참고).

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

위 표현에서 첫 번째 항은 비교적 간단하게 계산 가능하지만, 두 번째 항은  $\mathbf{c}$ 의 위치 의존성을 고려해야 하므로 계산 복잡도가 급격히 증가한다. 따라서 이를 실질적인 MLS 프레임워크에 통합하기 어려운 측면이 있다. 이러한 복잡성을 회피하기 위해, 두 번째 항을 확산 도함수로 대체하면 근사적으로 충분한 정확도의 미분 값을 계산할 수 있으며, 전체 계산의 효율성을 높일 수 있다 [6] (수식 12).

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

확산 도함수를 활용하면 벡터장  $\mathbf{u}$ 에 대해 발산 연산자인  $\nabla \cdot \mathbf{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}$ 을 보다 간단한 방식으로 근사할 수 있으며, 그 결과는 다음과 같다 (수식 13).

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

### 3.2.2 Divergence 제약 조건

Huerta 등은  $\nabla \cdot \mathbf{u} = \sigma$  형태의 발산 제약식을 활용하여, 특히  $\sigma = 0$ 일 때 비압축성 유동에 대한 해석 정확도를 향상시켰다 [41]. 본 연구에서는  $\sigma$ 를 비제로 값으로 설정할 수 있도록 하여, 속도장의 보간에서 발산 정도를 정밀하게 조절할 수 있게 하였다. 이를 위해 기존의 벡터 보간 행렬인  $\mathbf{B}_{vec}$ 에 발산 제약 항을 결합하며, 이 과정에서 수식 13을 다음과 같이 재구성한다 (수식 14).

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

여기서  $\mathbf{b}_{div}$ 는 다음 방식으로 구성된다 (수식 15).

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

이때  $\mathbf{b}$ 의 편미분은 각 기저함수 항에 대해 독립적으로 계산되며, 예를 들어  $d = 3, m = 2$ 일 경우 다음과 같은 형태를 갖는다 (수식 16).

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

이제 위의 발산 조건  $\mathbf{b}_{div}^T \cdot \mathbf{c} = \sigma$ 를 기존 MLS 시스템(수식 9)에 결합하면, 발산 제약을 반영한 MLS 방정식을 얻을 수 있다 (수식 17).

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

이 방정식 역시 앞서 소개한 최소자승 기법을 통해 안정적으로 풀 수 있으며, 본 논문에서 사용하는 전체 시스템의 구성은 다음과 같다:

$\mathbf{W}_{div} = \begin{bmatrix} \mathbf{W}_{vec} \\ \omega_{div} \end{bmatrix}$ ,  $\mathbf{B}_{div} = \begin{bmatrix} \mathbf{b}_{vec} \\ \omega_{div} \end{bmatrix}$ ,  $\Phi_{div} = [\Phi, \sigma]$ . 여기서  $\mathbf{W}_{div}$ 는  $(dN + 1) \times (dN + 1)$ 이고,  $\mathbf{W}_{div} \in \mathbb{R}^{(dN+1) \times (dN+1)}$ 는 가중 대각 행렬이다. 또한  $\mathbf{B}_{div} \in \mathbb{R}^{(dN+1) \times dk}$ 는 계수 행렬이며,  $\Phi_{div} \in \mathbb{R}^{dN+1}$ 는 보간 타겟 값을 포함하는 벡터이다.

### 3.2.3 평활화된 입자 유체역학 기반 비등방성 가중치

우리는 얇은 시트를 감지하기 위해 비등방성 커널 방법을 활용한다 [40]. 이를 위해 먼저 입자들의 가중 평균 공분산  $C$ 를 다음과 같이 계산한다 (수식 18).

$$C_i = \frac{\sum_j (\mathbf{x}_j - \mathbf{x}_i^w)(\mathbf{x}_j - \mathbf{x}_i^w)^T W(\mathbf{x}_j - \mathbf{x}_i^w, h)}{\sum_j W(\mathbf{x}_j - \mathbf{x}_i^w, h)} \quad (18)$$

여기서  $\mathbf{x}_i$ 는 타겟 입자, 그리고  $\mathbf{x}_j$ 는 인접 입자들이며,  $W$ 는 SPH 시뮬레이션에서 일반적으로 사용하는 등방성 커널 함수이다. 이 값들을 기반으로  $\mathbf{x}_i^w$ 는 다음과 같이 계산한다 (수식 19).

$$\mathbf{x}_i^w = \frac{\sum_j \mathbf{x}_j W(\mathbf{x}_j - \mathbf{x}_i, h)}{\sum_j W(\mathbf{x}_j - \mathbf{x}_i, h)} \quad (19)$$

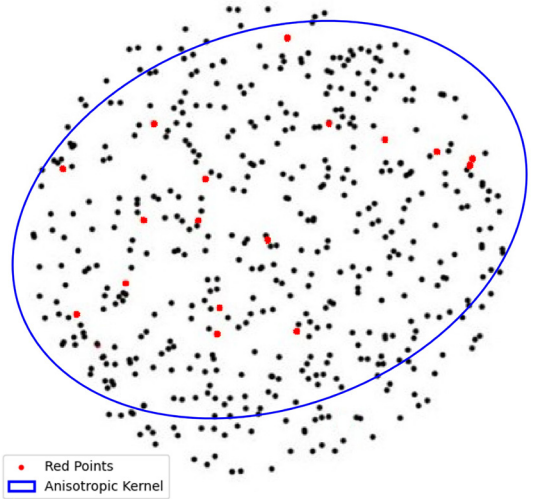


Figure 3: Anisotropic ellipse fitting to red points.

연관된  $C_i$ 에 대해 특이값 분해(Singular Value Decomposition, SVD)를 수행하면, 주변 입자 위치의 방향성과 신장 정도를 각각 고유벡터와 고유값의 형태로 얻을 수 있다 (수식 20).

$$C_i = \mathbf{C} = \mathbf{R} \mathbf{\Sigma} \mathbf{R}^T = \begin{bmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \\ \mathbf{e}_3^T \end{bmatrix}^T \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \begin{bmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \\ \mathbf{e}_3^T \end{bmatrix} \quad (20)$$

여기서  $\mathbf{e}_n$ 과  $\sigma_n$ 은 각각 분산의 크기와 신장 정도에 따라 정렬된 주축과 고유값을 의미한다. SVD는  $\alpha_1 \rho_0 < \rho(\mathbf{x}_i) < \alpha_2 \rho_0$ 인 경우에만 적용되며, 그 외의 경우에는  $\text{diag}(\sigma_1, \sigma_2, \sigma_3) = I$ 를 사용한다. 본 논문에서는  $\alpha_1$ 은 0.05,  $\alpha_2$ 는 0.7로 설정하여 사용했으며, 이 값은 실험을 통해 얻어진 값이다. 이후, 우리의 기법은  $\mathbf{G}_i$ 를 다음과 같은 형태의 대칭 행렬로 구성한다 (수식 21). Figure 3은 무작위로 배치된 SPH 입자들에서 비등방성 가중치를 타원체 형태로 시각화한 결과이다. 본 논문에서 사용하는 발산 제약 기반 MLS에서는 제약 벡터의 영향이 크기 때문에, 이를 가중치에 반영하기 위해 그림에 붉은색 점으로 표시된 제약 입자들만을 이용해 비등방성 가중치를 계산하였다.

$$\mathbf{G}_i = \frac{1}{h} \mathbf{R} \mathbf{\Sigma}^{-1} \mathbf{R}^T \quad (21)$$

이렇게 계산된 선형 변환 행렬  $\mathbf{G}$ 를 이용하여 비등방성 커널을 계산한다 (수식 22).

$$\Gamma(\mathbf{x}_i) = \min_i (\|\mathbf{G}_i(\mathbf{x}_j - \mathbf{x}_i)\|) \quad (22)$$

SPH 기반 비등방성 가중치를 통해 MLS 가중치인  $\omega^*(|\mathbf{r}_i|)$ 를 구성하는 방법은 다음과 같다 (수식 23).

$$\omega^*(|\mathbf{r}_i|) = \omega(|\mathbf{r}_i|) \Gamma(\mathbf{x}_i) \quad (23)$$

여기서  $\omega(|\mathbf{r}_i|)$ 는 노드와 제약 노드 간의 거리를 계산한 가중치이고,  $\Gamma(\mathbf{x}_i)$ 는 입자 간의 비등방성 가중치이며, 이 값들을 통해 가중치 행렬인  $\mathbf{W}$ 를 구성하여 MLS를 계산한다.

## 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 SPH 입자들을 가진 데이터를 통해 벡터장을 표현했다.

Figure 4는 2차원 공간에 SPH 입자들을 무작위로 배치한 후, 제약 노드를 포함한 장면에서 벡터장을 추출한 결과이다. 제약 노드에 해당하는 입자는 매 프레임마다 무작위로 선택되도록 설정했으며, 제약 벡터 또한 임의로 결정되도록 실험을 구성하였다. 이는 본 논문에서 제안하는 방법이 특정 장면에만 국한되지 않고, 다양한 무작위 환경에서도 안정적으로 작동함을 입증하기 위함이다. 실험 결과, 모든 장면에서 안정적인 벡터 흐름이 형성되었으며, 이에 따른 벡터장도 자연스럽게 생성되는 양상을 확인할 수 있었다.

## 5 정성적 평가

본 논문에서는 SPH 기반 입자 데이터를 활용하여 다양한 벡터장 패턴을 생성하는 기법을 제안했다. 특히 발산 제약 조건을 포함한 MLS 프레임워크에 SPH의 비등방성 커널을 통합함으로써, 기존 격자 기반 방법이나 일반 MLS 보간 방식으로는 표현하기 어려운 세밀한 벡터 흐름을 효과적으로 시각화할 수 있음을 보였다.

Figure 4에서는 본 연구에서 제안한 기법을 통해 생성된 다양한 벡터장의 형태가 시각적으로 제시되어 있다. 실험은 2차원 공간에 SPH 입자들을 무작위로 배치한 후, 각 프레임마다 제약 노드를 임의로 선택하고 제약 벡터 또한 랜덤하게 설정하여 수행되었다. 이러한 구성은 특정한 조건이나 장면에 종속되지 않고, 다양한 초기 상태에서도 제안 기법이 안정적으로 작동할 수 있는지를 평가하기 위한 목적을 지닌다.

실험 결과, 제약 조건이 무작위로 주어졌음에도 불구하고 생성된 벡터장들은 전체적으로 부드럽고 자연스러운 흐름을 유지하였으며, 입자 분포의 불균일성과 관계없이 구조적인 일관성을

갖춘 형태로 표현되었다. 이는 제안된 방법이 특정 장면에만 국한되어 작동하는 것이 아니라, 다양한 환경에서도 일관된 결과를 생성할 수 있는 범용성과 알고리즘적 안정성을 확보하고 있음을 정성적으로 입증하는 결과라 할 수 있다.

## 6 결론 및 향후 연구

본 논문에서는 SPH의 분산된 입자 데이터를 고려하여, MLS 기반으로 벡터장을 생성하는 새로운 방식을 제안하였다. 제약 조건만 입력으로 주어지면 다양한 패턴의 벡터장을 표현할 수 있기 때문에, 본 방법은 벡터 디자인이나 속도장이 요구되는 모델링 및 시뮬레이션 분야에 활용될 수 있다.

현재 결과는 난류가 고려된 벡터장을 표현할 수 있는 가능성을 보여주고 있지만, 이를 실제 물리 기반 시뮬레이션에 적용하기 위해서는 연속성을 갖는 벡터장으로 생성되도록 알고리즘의 개선이 필요하다. 단순히 생성된 벡터장을 외력으로 적용할 경우 노이즈처럼 작용할 가능성이 높으며, CFL 조건을 조절함으로써 일정 부분 해결할 수는 있으나, 이는 결과적으로 힘의 크기를 줄이는 방식이기 때문에 근본적인 해결책이 되기 어렵다. 향후에는 이러한 문제들을 개선할 수 있도록 연구를 지속적으로 확장해 나갈 계획이다.

## 감사의 글

The National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2023-00254695, Contribution Rate : 50%). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2022-00155915, Artificial Intelligence Convergence Innovation Human Resources Development (Inha University), Contribution Rate : 50%)

## 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.

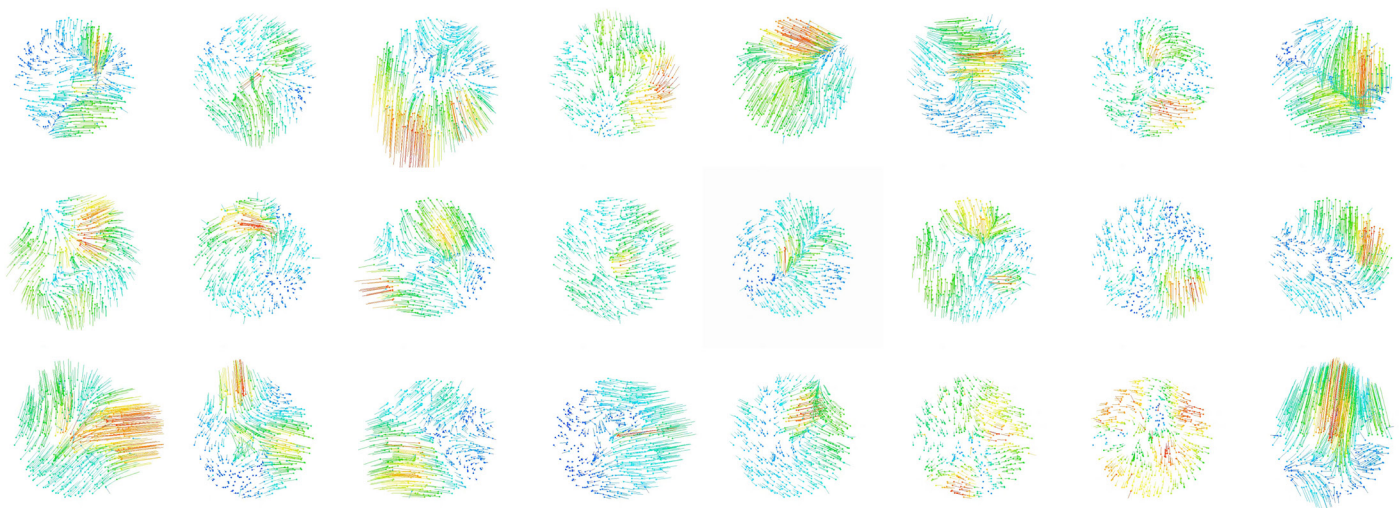


Figure 4: Various types of vector fields generated using our method based on SPH particle simulations.

- [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] S. Band, C. Gissler, and M. Teschner, “Moving least squares boundaries for sph fluids,” in *VRIPHYS*, 2017, pp. 21–28.
- [40] J. Yu and G. Turk, “Reconstructing surfaces of particle-based fluids using anisotropic kernels,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 1, pp. 1–12, 2013.
- [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.

## 〈 저자 소개 〉



### 김 중 현

- 2008년 세종대학교 컴퓨터학과 학사
- 2010년 고려대학교 컴퓨터학과 석사
- 2016년 고려대학교 컴퓨터학과 박사
- 2013년~2016년 (주)텐일레븐 선임연구원
- 2017년~2022년 강남대학교 가상현실학과 부교수
- 2023년~현재 인하대학교 디자인테크놀로지학과 부교수
- 2025년~현재 인하대학교 인공지능공학과 부교수
- 2025년~현재 인하대학교 대학원 전기컴퓨터공학과 부교수
- 관심분야 : 물리 기반 시뮬레이션, 가상/증강현실, 지오메트리 프로세싱, 게임 물리, 게임 AI
- <https://orcid.org/0000-0003-1603-2675>