applied sciences

Article

PartiEMC: Stable Isoline Reconstruction from Particle-Based Scalar Fields via Virtual-Plane Projection

Yu-Bin Kwon [1] and Jong-Hyun Kim [1,2,*]

[1] Department of Design Technology, College of Software and Convergence, Inha University, Incheon 22212, Republic of Korea; 12202057@inha.edu
[2] Graduate School of Electrical and Computer Engineering, Inha University, Incheon 22212, Republic of Korea
* Correspondence: jonghyunkim@inha.ac.kr

Abstract

This paper presents a geometry-driven framework for temporally stable 2D isoline reconstruction from particle-based simulation data. Unlike conventional Marching Squares methods, which assume grid-aligned scalar fields and often suffer from boundary jitter and flickering when applied to unstructured particle distributions, the proposed method constructs a continuous scalar field using an SPH kernel and estimates stabilized normals from level-set gradients at cell-level representative positions. Instead of relying on explicit Quadratic Error Function (QEF) optimization, we introduce a virtual-plane projection strategy that determines isoline vertices using a local geometric constraint. This projection can be interpreted as a first-order geometric approximation of QEF minimization, enabling QEF-free vertex positioning while reducing sensitivity to noisy particle-derived normals. As a result, the proposed method improves robustness in sparse particle regions while preserving important geometric features. To further enhance computational efficiency, we integrate a boundary-aware greedy meshing scheme that merges redundant interior geometry while preserving isoline boundaries. Experimental results demonstrate that the proposed method improves boundary stability and area consistency, reduces temporal variation, and decreases triangle counts by up to 70–75% compared with Marching Squares (MS) and Extended Marching Cube (EMC)-based reconstruction. These results indicate that the proposed framework is suitable for efficient real-time visualization of dynamic particle-based simulations.

Keywords: isoline reconstruction; particle-based simulation; Extended Marching Squares; virtual-plane projection; real-time animation

Academic Editor: Zhengjun Liu

Received: 16 April 2026
Revised: 4 May 2026
Accepted: 8 May 2026
Published: 12 May 2026

1. Introduction

Particle-based simulations are widely used to model complex natural phenomena such as fluids, smoke, and granular materials, and they play a crucial role in computer animation and virtual environments. In such simulations, individual particles evolve over time according to physical laws; however, in the visualization stage, it is necessary to reconstruct continuous boundaries or shapes from inherently discrete particle distributions. In particular, in slice-based visualization of 2D or 3D domains, extracting visually stable isolines from particle data is a key factor that determines both visual quality and temporal coherence.

Traditionally, isoline extraction has relied on grid-based algorithms such as Marching Squares. These methods provide simple and efficient implementations for explicitly defined scalar fields $f(\mathbf{x})$, but they exhibit clear limitations when applied to particle-based data that do not inherently possess a grid structure. In general, when converting a particle set $\{\mathbf{x}_i\}$ into a grid-aligned scalar field, an interpolation kernel $\phi(\mathbf{x}-\mathbf{x}_i)$ is used to define the scalar field as follows:

$$f(\mathbf{x}) = \sum_i w_i \, \phi(\mathbf{x}-\mathbf{x}_i)$$

which can introduce instability in boundary positions. This issue becomes particularly pronounced in sparse regions or near free surfaces, where isolines are prone to distortion or area fluctuation.

These problems are not limited to static frames. In animation scenarios, reconstruction is performed repeatedly at each frame, causing small errors to accumulate over time. This results in boundary jitter and frame-to-frame flickering, which significantly degrade visual reliability. To address these issues, various extensions of Marching Squares and Marching Cubes have been proposed.

For example, Extended Marching Cubes (EMC) and its variants improve boundary accuracy and preserve sharp features by introducing additional vertices and Hermite-based vertex estimation. However, these approaches still fundamentally assume grid-based scalar fields and may suffer from performance degradation when directly applied to particle-based data. In particular, when particle connectivity is not explicitly defined, the Quadratic Error Function (QEF)-based optimization

$$E(\mathbf{v}) = \sum_j \left(\mathbf{n}_j \cdot (\mathbf{v}-\mathbf{p}_j)\right)^2$$

becomes sensitive to noisy normal estimation and incurs increased computational cost.

Meanwhile, in particle-based fluid simulations, boundaries evolve dynamically at each frame, imposing additional requirements on isoline reconstruction. Therefore, beyond geometric accuracy, reconstruction methods must satisfy strict constraints in terms of computational efficiency and temporal stability. Although approaches based on complex optimization or iterative refinement can theoretically provide accurate results, they are often impractical for real-time animation or interactive virtual environments.

In this work, we propose a geometry-driven framework for reconstructing visually stable 2D isolines from particle-based scalar fields. The proposed method constructs a continuous scalar field from the particle distribution $\{\mathbf{x}_i\}$ using an SPH (Smoothed Particle Hydrodynamics) kernel, defined as

$$f(\mathbf{x}) = \sum_i w_i \, \phi(\mathbf{x}-\mathbf{x}_i)$$

and estimates reliable surface normals at the particle-averaged position $\bar{\mathbf{x}}$ within each cell.

Using the stabilized normal $\mathbf{n}$ and the representative position $\bar{\mathbf{x}}$, we define a virtual plane within each grid cell as

$$\mathbf{n} \cdot (\mathbf{x}-\bar{\mathbf{x}}) = 0$$

and determine isoline vertices by projecting nearby particles onto this plane. This virtual plane-based vertex placement avoids explicit QEF optimization while effectively suppressing boundary instability in sparse regions and near free surfaces.

Furthermore, to improve efficiency in animation scenarios, we integrate a greedy meshing strategy that merges redundant interior geometry while preserving isoline boundaries. By explicitly separating boundary-sensitive geometry from interior regions, the proposed framework significantly reduces geometric complexity per frame while maintaining boundary accuracy and temporal coherence.

The main contributions of this paper are summarized as follows:

- We introduce a geometry-driven vertex positioning strategy that approximates the dominant local constraint of QEF using a virtual plane defined by a cell-level representative position and a stabilized normal, without explicitly solving the QEF optimization problem.
- We formulate a cell-level normal stabilization scheme for particle-based scalar fields, which reduces sensitivity to particle-level noise and improves temporal coherence in isoline reconstruction.
- We develop an integrated reconstruction pipeline that combines SPH-based scalar field construction, level-set representation, virtual-plane vertex positioning, and boundary-aware greedy meshing for real-time particle-based isoline visualization.
- We apply greedy meshing in a boundary-aware manner, where cells intersecting the isoline are preserved and only redundant interior regions are merged, thereby reducing geometric complexity while maintaining boundary consistency.

2. Related Work

The limitations discussed above have motivated a wide range of studies on grid-based isoline and isosurface extraction, feature-preserving extensions, particle-based boundary reconstruction, and learning-based reconstruction. In this section, we review representative approaches in three categories: (1) classical grid-based Marching methods and their extensions, (2) particle-based scalar field construction and normal estimation techniques, and (3) recent learning-based and hybrid approaches for vertex positioning and topology estimation. We then clarify how the proposed framework differs from and complements existing methods by emphasizing geometric consistency, temporal stability, and efficiency in particle-based isoline reconstruction.

2.1. Grid-Based Isoline and Isosurface Extraction: Marching Methods

A representative approach for extracting iso-structures from grid-based scalar fields is the Marching Cubes (MC) algorithm, which approximates an isosurface using predefined lookup tables according to the binary configuration of each local cell [1]. While MC is simple and computationally efficient, it is well known to suffer from limited sharp-feature preservation due to linear interpolation-based vertex placement, often producing overly smoothed results depending on data resolution and sampling conditions.

For two-dimensional cases, the same principle is commonly applied through Marching Squares, which extracts isolines using grid sampling and lookup-based polyline construction. Despite its widespread use, this approach inherits similar limitations regarding boundary accuracy and stability, particularly when applied to non-grid-aligned data.

To address topological ambiguities, extensions such as MC33 and asymptotic deciders have been proposed [2,3]. Surveys of Marching Cubes variants have further discussed computational efficiency, topological consistency, and scalability issues in grid-based isosurface extraction [4].

2.2. Feature-Preserving Extensions: Extended Marching Cubes

To overcome feature-preservation limitations, Kobbelt et al. proposed Extended Marching Cubes (EMC) [5], which incorporates additional samples and gradient information to improve reconstruction of sharp features such as edges and corners.

Subsequent works have further explored adaptive and feature-preserving surface extraction schemes based on dual grids, cubical marching strategies, and topology-aware contouring [6,7]. These approaches improve geometric fidelity but still rely on grid-based scalar fields and accurate gradient estimation.

When applied to particle-based data, these assumptions become less reliable. In particular, gradients derived from particle-to-field conversion can be noisy in sparse or free-surface regions, directly affecting the stability of feature-preserving formulations. Therefore, stabilizing field representation and normal estimation remains a key challenge in particle-based reconstruction.

2.3. Dual Contouring and QEF-Based Vertex Optimization

Dual Contouring (DC) introduces a more flexible vertex placement strategy by optimizing a Quadratic Error Function (QEF) based on Hermite data [8]. This allows a single vertex per cell while preserving sharp features and grid-aligned topology.

Extensions of DC have explored manifold generation, adaptive octree structures, and quality-aware mesh generation [9,10], enabling efficient handling of large-scale volumetric data. However, the effectiveness of DC strongly depends on the accuracy of intersection points and normal vectors.

In particle-based scenarios, these quantities must be estimated from noisy scalar fields, making QEF optimization unstable and computationally expensive. This limitation motivates alternative formulations that avoid explicit optimization while maintaining geometric fidelity.

2.4. Particle-Based Boundary Detection and Normal Estimation

In particle-based fluid simulation, SPH-based methods have been widely used for free-surface detection and scalar field construction. Marrone et al. proposed a robust formulation for free-surface detection using kernel gradients [11]. Subsequent works have explored improved particle-based fluid reconstruction using adaptive sampling and anisotropic kernels [12,13].

Despite these advances, particle-based reconstruction often suffers from instability due to irregular sampling and noise. Frame-to-frame normal flipping and boundary jitter are particularly problematic in real-time animation scenarios.

To mitigate these issues, recent local implicit and point-cloud reconstruction approaches emphasize local consistency by using local representations, patch-based priors, or grid-structured implicit fields [14,15]. Such strategies aim to stabilize reconstruction from sparse or noisy data without relying solely on global optimization.

2.5. Learning-Based MC/DC and Hybrid Approaches

Recent work has increasingly explored learning-based approaches for explicit surface reconstruction.

Deep Marching Cubes enables integration of MC into differentiable pipelines [16]. Neural Marching Cubes improves local topology prediction by learning neighborhood-aware representations [17]. Similarly, Neural Dual Contouring replaces analytical Hermite data with neural predictions while maintaining DC's structure [18]. More recently, Self-Supervised Dual Contouring has explored training without dense supervision [19].

In addition, hybrid representations such as DMTet use deformable tetrahedral grids to jointly predict geometry and topology [20]. More recent differentiable iso-surfacing methods such as FlexiCubes introduce additional degrees of freedom into the extraction process to improve mesh quality and geometric fidelity in gradient-based mesh optimization [21]. Implicit neural representations, including NeRF-based geometry extraction and neural implicit surface reconstruction, have also been applied to surface reconstruction [22,23].

Although these approaches achieve high-quality reconstruction, they require large training datasets, introduce inference overhead, and are often difficult to integrate into lightweight real-time simulation pipelines.

2.6. Relation to Our Work

Prior work can be broadly categorized into: (1) feature-preserving extensions of grid-based MC/EMC/DC methods, (2) particle-based scalar field construction and free-surface detection, and (3) learning-based approaches for vertex and topology estimation.

However, for particle-based real-time animation, key challenges remain: (a) unstable normal and intersection estimation, (b) frame-to-frame isoline jitter, and (c) excessive redundant geometry generation.

The proposed method addresses these limitations by combining cell-level normal stabilization, virtual plane-based vertex placement, and boundary-aware greedy meshing. This results in a geometry-driven framework that achieves robust, temporally stable, and computationally efficient isoline reconstruction without explicit optimization or training.

Clarification of Novelty

We do not claim that the individual components used in this work, such as SPH-based scalar-field construction, level-set representation, or greedy meshing, are entirely new by themselves. Instead, the novelty of the proposed method lies in their problem-specific integration and reformulation for temporally stable isoline reconstruction from particle-based scalar fields.

In particular, the proposed framework replaces explicit QEF-based vertex optimization with a virtual-plane projection strategy defined by a cell-level representative position and a stabilized normal. This projection is not intended as an ad-hoc heuristic, but can be interpreted as a first-order geometric approximation of QEF minimization under local normal consistency and limited spatial variation. This enables QEF-free vertex positioning while reducing sensitivity to noisy particle-derived normals.

Furthermore, the greedy meshing stage is applied in a boundary-aware manner: cells intersecting the isoline are preserved, and only redundant interior regions are merged. As a result, the optimization reduces geometric complexity without modifying the isoline boundary. Therefore, the contribution of this work is the construction of a lightweight, geometry-driven pipeline that jointly addresses normal instability, boundary jitter, and redundant geometry in real-time particle-based isoline reconstruction.

3. Proposed Method

The proposed framework is designed for real-time animation scenarios where isolines must be reconstructed at every frame. To ensure temporal stability, we stabilize normals at cell-level representative positions and determine isoline vertices through virtual plane projection instead of explicitly optimizing the global QEF. In addition, to improve computational efficiency, we apply a greedy merging strategy to effectively reduce redundant interior geometry.

Figure 1 illustrates the overall pipeline of the proposed particle-based isoline reconstruction framework. The framework consists of four main stages: (1) particle input, (2) scalar field construction, (3) isoline reconstruction, and (4) optimization. The proposed method is designed to extract visually stable isolines from particle-based data while maintaining the computational efficiency required for real-time animation environments.

The pipeline begins with an unstructured particle set $\{\mathbf{x}_i\}$ representing the simulation state. Since particle data do not provide explicit connectivity or grid-aligned scalar values, we first convert the particle distribution into a continuous scalar field defined on a regular grid. This scalar field is constructed using SPH kernel interpolation and is defined as

$$f(\mathbf{x}) = \sum_i w_i \, \phi(\mathbf{x} - \mathbf{x}_i)$$

The resulting scalar field can be interpreted as a density-based implicit function of the particle distribution, providing a suitable representation for isoline extraction. Therefore, this transformation serves as a necessary preprocessing step to apply grid-based isoline reconstruction algorithms to particle-based data.

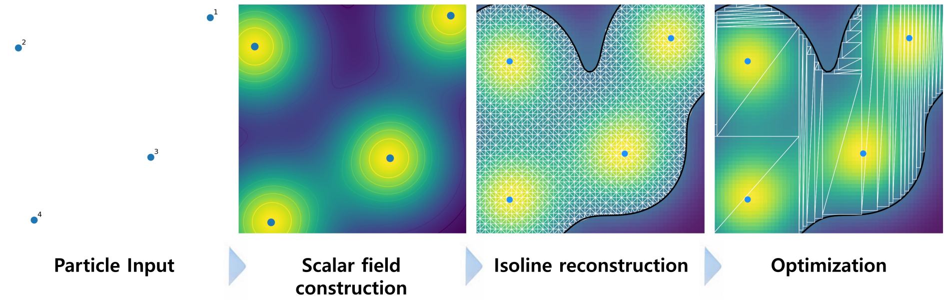


Figure 1. Overview of the proposed particle-based isoline reconstruction pipeline. The figure illustrates the overall processing flow from particle input to SPH-based scalar field construction, virtual plane-based vertex positioning for isoline reconstruction, and the final greedy optimization stage. The arrows indicate the sequential data flow between the main stages of the pipeline.

In the next step, surface normals are estimated from the gradient of the constructed scalar field. Near free surfaces or in regions with low particle density, normals computed at individual particle positions can be highly sensitive to noise, leading to significant frame-to-frame variation. To address this issue, we employ a cell-level normal estimation strategy.

Specifically, we compute the particle-averaged position $\bar{\mathbf{x}}$ within each grid cell and evaluate the gradient of the scalar field at this position to define the normal as

$$\mathbf{n} = \frac{\nabla f(\bar{\mathbf{x}})}{\|\nabla f(\bar{\mathbf{x}})\|}$$

This design reduces temporal variation in normal directions and ensures geometric consistency across the pipeline, from scalar field and gradient estimation to vertex computation.

3.1. Virtual Plane-Based Vertex Positioning

Based on the stabilized scalar field and normals, isoline reconstruction is performed using an Extended Marching Squares formulation. For each grid cell where an intersection occurs with the target isovalue $f(\mathbf{x}) = \tau$, a virtual plane is defined using the estimated normal $\mathbf{n}$ and the representative position $\bar{\mathbf{x}}$ as follows:

$$\mathbf{n} \cdot (\mathbf{x} - \bar{\mathbf{x}}) = 0$$

Nearby particles $\mathbf{x}_i$ are then projected onto this plane, and the signed projection distance is computed as

$$d_i = \mathbf{n} \cdot (\mathbf{x}_i - \bar{\mathbf{x}})$$

This virtual plane-based vertex placement avoids explicit optimization of the QEF (Quadratic Error Function) used in conventional EMC, while remaining robust to variations in particle density and effectively preserving geometric features.

Finally, the optimization stage improves both computational performance and temporal stability. A greedy merging strategy is applied to identify cells that do not intersect with isolines and merge redundant interior geometry generated within those cells. This process preserves isoline boundaries while reducing unnecessary triangle generation, thereby decreasing overall geometric complexity.

As a result, the proposed framework enables efficient frame-by-frame reconstruction while maintaining visual quality. Through this staged pipeline, raw particle data are transformed into stable and efficient isoline representations, making the approach suitable for real-time animation, particle-based fluid visualization, and interactive virtual environments.

3.2. Scalar Field Construction

Since particle-based data do not provide explicit grid structures or continuous scalar values, it is difficult to directly apply grid-based isoline reconstruction algorithms. Traditional Marching Cubes and Marching Squares methods assume that the scalar field is given as a signed distance function (SDF), which is not directly available in particle-based simulations.

Therefore, we construct a continuous scalar field from the particle distribution $\{\mathbf{x}_i\}$ using an SPH (Smoothed Particle Hydrodynamics) kernel. Specifically, the scalar value at each grid position $\mathbf{x}$ is computed by accumulating kernel-weighted contributions from neighboring particles:

$$f(\mathbf{x}) = \sum_i w_i \, k\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|}{R}\right)$$

We use the following kernel function:

$$k(s) = \max\left(0, (1 - s^2)^3\right) \tag{1}$$

Here, s is the normalized distance between the evaluation point $\mathbf{x}$ and the particle position $\mathbf{x}_i$, defined as

$$s = \frac{\|\mathbf{x} - \mathbf{x}_i\|}{R} \tag{2}$$

where R denotes the kernel radius.

According to this definition, the kernel function $k(s)$ is nonzero only within the range $0 \leq s < 1$, assigning larger weights to particles closer to the evaluation point. This formulation enables the transformation of the particle distribution into a smooth and continuous scalar field, which serves as the basis for isoline extraction.

Figure 2 shows the visualization of the particle-based scalar field computed using the kernel function defined above. The left image illustrates the shape of the kernel function $k(s)$ with respect to the normalized distance s, while the right image visualizes the kernel-weighted density field computed from the particle distribution. This scalar field is used as an implicit function for isoline extraction.

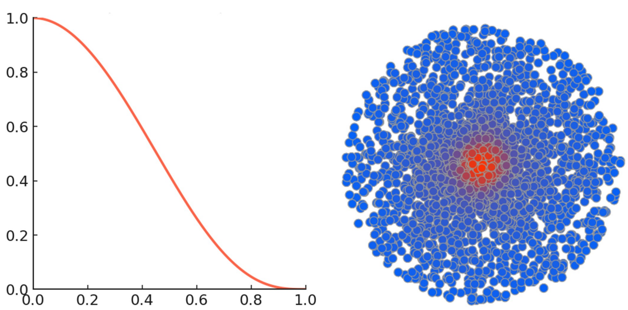


Figure 2. Visualization of the SPH kernel-based scalar field (particle density). (**Left**) Shape of the kernel function $k(s)$ with respect to the normalized distance $s = \frac{\|\mathbf{x} - \mathbf{x}_i\|}{R}$. (**Right**) Scalar field (density) computed from a particle distribution using the SPH kernel. The blue dots indicate the input particles, and the red dot represents the reference particle or query position used for kernel evaluation. A normalized color scale in $[0, 1]$ is used, where darker colors indicate regions of higher kernel-weighted density.

To reconstruct isolines, we define a level-set function based on the particle-averaged position. Given a set of particles around the evaluation position $\mathbf{x}$, the weighted average position $\bar{\mathbf{x}}$ is computed as

$$W_i = \frac{k(s_i)}{\sum_j k(s_j)} \tag{3}$$

$$\bar{\mathbf{x}} = \sum_i W_i \mathbf{x}_i \tag{4}$$

Here, s_i is the normalized distance between $\mathbf{x}$ and the i-th particle position $\mathbf{x}_i$, and W_i denotes the normalized kernel weight. Based on this, the level-set function $\phi(\mathbf{x})$ is defined as

$$\phi(\mathbf{x}) = \|\mathbf{x} - \bar{\mathbf{x}}\| - r \tag{5}$$

Here, r is the reference radius that determines the zero level-set position relative to the particle-averaged position $\bar{x}$. In this work, r is not treated as a scene-dependent fitting parameter. Instead, it is defined in a scale-aware manner based on the average particle spacing.

Let h denote the average particle spacing, estimated from the mean nearest-neighbor distance of the input particles:

$$h = \frac{1}{N} \sum_{i=1}^{N} \min_{j \neq i} \|x_i - x_j\|. \tag{6}$$

The reference radius is then set as

$$r = \alpha_r h, \tag{7}$$

where α_r is a fixed dimensionless coefficient. The same value of α_r is used throughout all frames of a sequence to avoid frame-dependent changes in the extracted isoline position. When the particle resolution changes (e.g., in downsampling experiments), h is recomputed for the corresponding resolution, and r is scaled accordingly.

The choice of r directly affects the reconstructed isoline. A smaller r places the zero level-set closer to $\bar{x}$, which may preserve local details but can increase sensitivity to particle noise and boundary jitter. In contrast, a larger r shifts the isoline outward and produces smoother contours, but may oversmooth fine geometric features. Therefore, r is selected within the local kernel support, i.e.,

$$0 < r < R, \tag{8}$$

so that the extracted isoline remains consistent with the SPH-supported local particle distribution. This scale-aware definition allows the isoline position to remain consistent across different particle densities while preserving geometric features within the resolution supported by the particle sampling.

This level-set formulation provides a geometric basis for Extended Marching Squares-based isoline reconstruction and normal estimation, enabling consistent reconstruction even under varying particle distributions. Furthermore, it allows stable vertex positioning without explicitly minimizing the QEF.

In the subsequent reconstruction stage, isoline vertices are determined using projection onto the virtual plane defined at the particle-averaged position, ensuring consistency with the level-set formulation introduced in this section.

3.2.1. Relation Between Level-Set Formulation and Isoline Reconstruction

The defined level-set function $\phi(x)$ provides an implicit representation of the isoline, where the target contour is given by the zero level-set $\phi(x) = 0$. Therefore, isoline extraction can be interpreted as the process of approximating the zero-crossing of the scalar field.

In the proposed framework, this implicit definition is realized through a local geometric approximation. At each grid cell, we compute a representative position $\bar{x}$ and evaluate the gradient $\nabla\phi(\bar{x})$, which defines the local normal direction of the level-set. Based on this, the virtual plane defined in Equation (7) can be interpreted as a first-order linearization of the level-set function around $\bar{x}$.

Under this interpretation, the projection-based vertex positioning corresponds to selecting a point that best approximates the zero level-set along the local normal direction. Therefore, the proposed reconstruction process can be understood as a local linear approximation of $\phi(x) = 0$, ensuring consistency between scalar field construction, level-set definition, and isoline extraction.

3.2.2. Parameter Selection

The key parameters of the proposed method, including the SPH kernel radius R, the level-set reference radius r, and the grid resolution, are determined based on the particle sampling density rather than arbitrary tuning. Let h denote the average particle spacing. In our implementation, the kernel radius R is set proportional to h so that a sufficient number of neighboring particles are included for stable scalar-field construction and gradient estimation. A very small R may lead to noisy density and normal estimates due to insufficient local support, whereas an excessively large R may oversmooth geometric details.

The reference radius r controls the relative position of the extracted isoline with respect to the particle-averaged position $\bar{x}$. We set r according to the average particle spacing or the target density threshold, which allows the extracted boundary to remain consistent under different particle densities. Thus, r is not used as a scene-dependent fitting parameter, but as a scale-aware parameter linked to the local sampling resolution.

The grid resolution is selected to provide an appropriate sampling ratio with respect to the particle distribution. If the grid is too coarse, aliasing and boundary distortion may occur because the scalar field is undersampled. Conversely, an excessively fine grid increases computational cost without necessarily improving the global boundary shape. Therefore, the grid resolution is chosen to balance reconstruction accuracy and real-time performance. Within this practical parameter range, the proposed method shows stable behavior, and its sensitivity to parameter variation is comparable to that of standard SPH-based scalar-field construction and QEF-based reconstruction methods.

3.3. Normal Estimation

The stability of normal estimation is a critical factor in scalar field-based isoline reconstruction. In grid-based SDF representations, normals are typically computed as the normalized gradient of the signed distance function. However, for scalar fields constructed from particle data, normal estimation becomes highly sensitive to noise, particularly near free surfaces or in regions with low particle density.

A straightforward approach is to compute particle-level normals at each particle position using kernel gradients. While this method provides reasonably accurate results in densely sampled regions, it often suffers from increased variance in sparse or transitional regions, leading to normal flipping between frames. Such instability directly affects vertex placement and results in visual flickering.

To address this issue, we estimate normals not at individual particle positions but at a representative position within each cell, namely the particle-averaged position $\bar{\mathbf{x}}$.

Specifically, we evaluate the gradient of the scalar field at this location and define the normal as

$$\mathbf{n}(\bar{\mathbf{x}}) = \frac{\nabla\phi(\bar{\mathbf{x}})}{\|\nabla\phi(\bar{\mathbf{x}})\|} \tag{9}$$

Here, $\phi(\mathbf{x})$ denotes the level-set function defined in the previous section, and $\bar{\mathbf{x}}$ represents the kernel-weighted average position. By consistently evaluating the scalar field, gradient, and normal at the same representative location, we ensure geometric coherence throughout the reconstruction pipeline.

Normal Orientation Consistency

Although the virtual plane itself is invariant to the sign of the normal, the signed projection distance used for candidate selection depends on the normal orientation. Therefore, a consistency correction is applied to reduce sign flips of gradient-based normals across frames and neighboring cells.

For each active cell c at frame t, we first compute the raw normal as

$$\tilde{n}_c^t = \frac{\nabla\phi(\bar{x}_c^t)}{\|\nabla\phi(\bar{x}_c^t)\|}. \tag{10}$$

The orientation of this raw normal is then aligned with a reference normal n_{ref}. When a valid normal from the previous frame is available, we use n_c^{t-1} as the reference. Otherwise, the reference normal is computed from the average of consistently oriented neighboring cell normals. The final normal is defined as

$$n_c^t = \begin{cases} \tilde{n}_c^t, & \text{if } \tilde{n}_c^t \cdot n_{\text{ref}} \geq 0, \\ -\tilde{n}_c^t, & \text{otherwise.} \end{cases} \tag{11}$$

This sign correction prevents abrupt normal flips caused by local gradient noise, especially in sparse regions near the zero level set. If the gradient magnitude is below a small threshold, the normal is considered unreliable and the previous valid normal or a neighbor-averaged normal is used instead. This avoids unstable signed projection distances and reduces erroneous vertex selection and contour jitter.

Figure 3 compares normals obtained from an ideal SDF with those estimated using the proposed particle-based approach. The proposed method effectively suppresses frame-to-frame variation in normal directions, even in sparse regions, enabling stable isoline reconstruction in real-time animation scenarios.

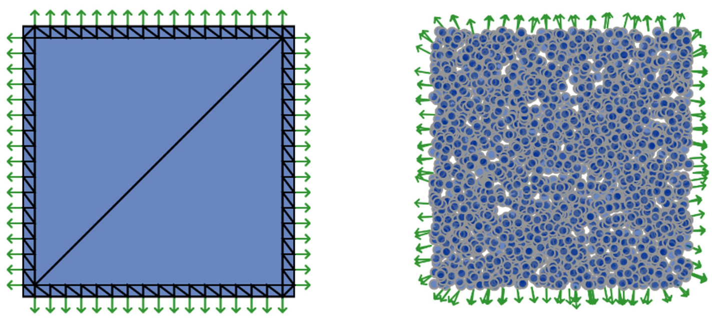


Figure 3. Comparison of normal estimation methods. (**Left**) Reference normals obtained from an ideal Signed Distance Function (SDF). (**Right**) Particle-based normals computed as the normalized gradient of the level-set function, $\nabla\phi(\bar{\mathbf{x}})$, evaluated at the cell-internal particle-averaged position $\bar{\mathbf{x}}$. The arrows indicate the normal directions, and the colors are used to distinguish the normal estimation results in each panel. The proposed method provides more stable normal estimation, particularly in sparse particle regions and near free surfaces.

The effectiveness of the proposed normal estimation is evaluated indirectly through its impact on the final reconstruction results. Since the normal estimation is tightly coupled with the vertex positioning process, its quality is reflected in the temporal stability and boundary consistency metrics presented in Section 4.

3.4. Vertex Position Determination Using Virtual Plane Projection

Based on the level-set formulation and stabilized normals, we determine isoline vertex positions using a virtual plane projection strategy. For each grid cell, a virtual plane is defined using the normal $\mathbf{n}(\bar{\mathbf{x}})$ computed at the particle-averaged position $\bar{\mathbf{x}}$ as follows:

$$\mathbf{n}(\bar{\mathbf{x}}) \cdot (\mathbf{x} - \bar{\mathbf{x}}) = 0 \tag{12}$$

Since the scalar field, gradient, and normal are consistently defined at the same representative position $\bar{\mathbf{x}}$, this virtual plane provides a geometrically coherent reference.

In conventional Extended Marching Cubes (EMC), vertex positions are determined by minimizing a QEF (Quadratic Error Function) constructed from intersection constraints and normals. In contrast, our approach avoids explicit QEF optimization and instead uses projection-based selection.

Specifically, nearby particles $\mathbf{x}_i$ are projected onto the virtual plane, and the signed distance along the normal direction is computed as

$$d_i = \mathbf{n}(\bar{\mathbf{x}}) \cdot (\mathbf{x}_i - \bar{\mathbf{x}}) \tag{13}$$

The particle with the maximum $|d_i|$ is selected as the isoline vertex for the corresponding cell.

This approach reduces sensitivity to noisy normal estimates and provides robust reconstruction under varying particle densities. In addition, by eliminating iterative QEF optimization, computational efficiency is significantly improved, making the method suitable for real-time animation environments.

This projection-based formulation avoids the need for explicit QEF minimization while remaining robust to noise and particle sparsity.

First-Order Interpretation of QEF

To clarify the relationship between the proposed virtual-plane projection and QEF minimization, we consider the Quadratic Error Function (QEF) commonly used in Dual Contouring:

$$E(v) = \sum_j \big(n_j \cdot (v - p_j)\big)^2, \tag{14}$$

where p_j and n_j denote the local intersection points and their associated normals, respectively. In conventional QEF-based reconstruction, the vertex position is obtained by minimizing Equation (14).

In the proposed method, normals are stabilized at the cell level. Therefore, within a local grid cell, we assume that the normals are locally consistent, such that $n_j \approx \bar{n}$, where $\bar{n}$ denotes the representative normal evaluated at the particle-averaged position $\bar{x}$. Under this assumption, the QEF can be approximated as

$$E(v) \approx \sum_j \big(\bar{n} \cdot (v - p_j)\big)^2. \tag{15}$$

By decomposing the displacement term as

$$v - p_j = (v - \bar{x}) + (\bar{x} - p_j), \tag{16}$$

Equation (15) becomes

$$E(v) \approx \sum_j \left(\bar{n} \cdot (v - \bar{x}) + \bar{n} \cdot (\bar{x} - p_j)\right)^2. \tag{17}$$

Let

$$a = \bar{n} \cdot (v - \bar{x}), \qquad b_j = \bar{n} \cdot (\bar{x} - p_j). \tag{18}$$

Then Equation (17) can be written as

$$E(v) \approx \sum_j (a + b_j)^2 = N_s a^2 + 2a \sum_j b_j + \sum_j b_j^2, \tag{19}$$

where N_s is the number of local samples. Since $\bar{x}$ is used as the representative position of the local samples, the projected offsets are assumed to be locally centered around $\bar{x}$, i.e., $\sum_j b_j \approx 0$. In addition, $\sum_j b_j^2$ is independent of v. Therefore, the dominant term relevant to minimization is

$$E(v) \approx N_s (\bar{n} \cdot (v - \bar{x}))^2 + \text{const}. \tag{20}$$

Minimizing this simplified expression leads to the following linear constraint:

$$\bar{n} \cdot (v - \bar{x}) = 0. \tag{21}$$

This condition is equivalent to the virtual-plane constraint defined in Equation (7). Thus, the proposed virtual-plane projection can be interpreted as a first-order geometric approximation of QEF minimization under the assumptions of local normal consistency and limited spatial variation. Instead of solving the full QEF system, the proposed method retains the dominant local geometric constraint and determines the isoline vertex through projection-based selection, which improves robustness and efficiency in particle-based reconstruction.

3.5. Virtual Plane-Based Vertex Selection Process

Approximation Error and Robustness Boundary

The validity of the above first-order approximation depends on the local consistency of normals and the spatial dispersion of samples within a cell. Let the local normal at each constraint be expressed as

$$n_j = \bar{n} + \delta_j, \tag{22}$$

where $\bar{n}$ is the representative stabilized normal and δ_j denotes the normal perturbation. We also write the local sample position as

$$p_j = \bar{x} + q_j, \tag{23}$$

where q_j is the offset from the particle-averaged position $\bar{x}$. For $u = v - \bar{x}$, the QEF residual can be written as

$$n_j \cdot (v - p_j) = \bar{n} \cdot u + \eta_j, \tag{24}$$

where

$$\eta_j = -\bar{n} \cdot q_j + \delta_j \cdot (u - q_j). \tag{25}$$

The proposed virtual-plane approximation keeps the dominant term $\bar{n} \cdot u$ and neglects the perturbation term η_j. If the local samples are approximately centered around $\bar{x}$ along the normal direction and the normal perturbations are small, then η_j becomes small and the simplified energy in Equation (15) remains a valid approximation of the original QEF.

More specifically, assuming $\|\delta_j\| \le \epsilon_n$ and $\|q_j\| \le \rho$, the perturbation term is bounded by

$$|\eta_j| \le \gamma + \epsilon_n(\|u\| + \rho), \tag{26}$$

where $\gamma = \max_j |\bar{n} \cdot q_j|$ represents the maximum projected sample dispersion along the representative normal direction. Therefore, the approximation error decreases when the normal variation ϵ_n and the projected spatial dispersion γ are small. This condition is consistent with the proposed cell-level normal stabilization, which reduces particle-level normal fluctuation by evaluating the gradient at the representative position.

The effect of angular normal error can also be interpreted geometrically. Let n^* be the ideal local normal and let θ be the angle between n^* and $\bar{n}$. Then, for any local displacement u,

$$|(\bar{n} - n^*) \cdot u| \le 2\|u\| \sin\frac{\theta}{2} \approx \|u\|\theta. \tag{27}$$

Thus, the error induced by normal perturbation grows approximately linearly with the angular normal error and the local displacement scale. Since the proposed method restricts vertex selection to local particles within a grid cell or neighboring support region, $\|u\|$ remains bounded by the local cell scale.

Consequently, the virtual-plane projection behaves as an approximate QEF solution when (1) the normals within a cell are locally coherent, (2) the projected sample dispersion around $\bar{x}$ is small, and (3) the level-set gradient magnitude is sufficiently reliable. Conversely, when the normal error becomes large, the gradient magnitude is weak, or the particle distribution is extremely sparse or noisy, the virtual plane may no longer provide a reliable approximation of the QEF constraint. In such cases, the reconstruction problem becomes ill-conditioned for both projection-based and QEF-based formulations because the underlying local Hermite constraints themselves are unreliable.

For each grid cell intersected by the target isoline, the vertex selection and placement are performed as follows:

1. **Virtual plane definition:** A representative virtual plane is defined using the particle-averaged position $\bar{\mathbf{x}}$ and the normal $\mathbf{n}_{\text{plane}}$ computed in Equation (9).
2. **Particle projection:** Nearby particles $\mathbf{p}_j$ are projected onto the virtual plane, and the signed projection distance is computed as

$$d_{\text{proj}} = (\mathbf{p}_j - \bar{\mathbf{x}}) \cdot \mathbf{n}_{\text{plane}} \tag{28}$$

3. **Validity check:** To eliminate noise and invalid candidates, two filtering steps are applied. (1) Particles with $d_{\text{proj}} < 0$ are considered interior and discarded. (2) Particles whose projection distance or spatial distance exceeds the grid size are removed.
4. **Final vertex selection:** Among the remaining valid candidates, the particle that lies farthest from the virtual plane along the normal direction $\mathbf{n}_{\text{plane}}$ is selected. The position of this particle $\mathbf{p}_{\text{far}}$ is assigned as the new isoline vertex $\mathbf{v}_{\text{new}}$ for the cell.

3.6. Geometric Interpretation

Figure 4 illustrates the geometric intuition behind the proposed virtual plane projection method. The virtual plane (orange) is defined using the stabilized normal at the particle-averaged position. Nearby particles are projected onto this plane, and the signed projection distance along the normal direction (green line) determines the selection of the new isoline vertex.

By directly determining vertex positions along the level-set normal direction, the proposed method effectively preserves sharp features while reducing the excessive smoothing commonly observed in linear interpolation-based approaches.

https://doi.org/10.3390/app16104816

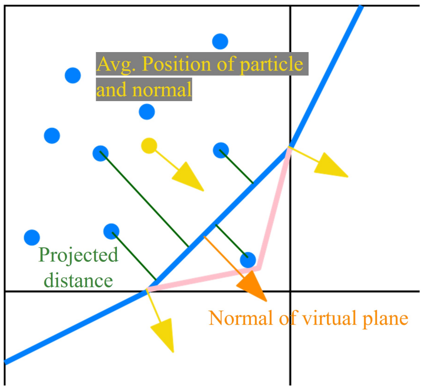
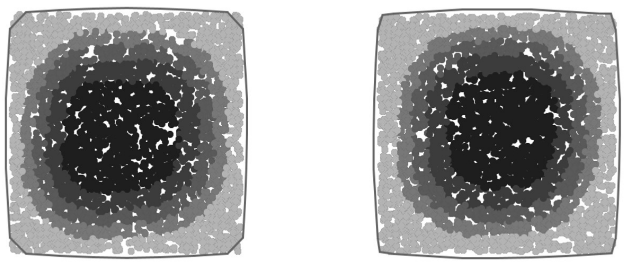


Figure 4. Vertex positioning via virtual-plane projection. The virtual plane shown in orange is defined at the particle-averaged position using the stabilized normal direction. The green line indicates the signed projection distance used to determine the new isoline vertex. Other visual elements, including the particles and arrows, illustrate the local particle configuration and projection direction.

In general, the QEF used in EMC provides stable results only when normals are highly accurate, and often requires relatively complex computations such as anisotropic kernels or SVD-based optimization. In contrast, the proposed virtual plane projection simplifies the reconstruction pipeline while providing stable and efficient vertex estimation for particle-based scalar fields.

Figure 5 compares reconstruction results between standard Marching Cubes (MC) and the proposed method. MC tends to lose sharp features and suffers from volume shrinkage, whereas the proposed method maintains stable boundaries and preserves geometric features through projection-based vertex computation.

Figure 6 shows isoline reconstruction results under progressively reduced particle counts (80%, 50%, 30%). While reconstruction quality typically degrades with fewer particles, the proposed method maintains stable isoline shapes and preserves features even in sparse conditions.



(**a**) MC (**b**) Our method

Figure 5. Surface reconstruction comparison. (**a**) MC exhibits volume loss and feature smoothing; (**b**) the proposed method preserves sharp edges and maintains volume. The dots indicate the sampled particle positions used as input for reconstruction.

(**a**) 80% (**b**) 50% (**c**) 30%

Figure 6. Surface extraction results under varying particle counts.

3.7. Greedy Meshing Optimization

The virtual plane-based vertex positioning method proposed earlier generates stable and feature-preserving isoline geometry. However, when reconstructing isolines on a regular grid, a large number of unnecessary triangles may be generated in interior regions that do not intersect with the isoline. Although such interior geometry does not significantly affect visual results, it increases memory usage and per-frame computational cost.

To address this issue, we apply a greedy meshing strategy that selectively merges only interior mesh elements while preserving isoline boundaries. The key idea is to restrict merging to cells that do not intersect with the isoline. Cells that contain isoline intersections define boundaries and encode detailed geometric information; therefore, merging them may degrade boundary accuracy and temporal stability.

Accordingly, the proposed method applies greedy optimization only to interior regions, effectively reducing overall geometric complexity while maintaining boundary shapes. This allows us to preserve visual quality while improving both memory efficiency and real-time performance.

Complexity and Topology Preservation

The proposed greedy meshing operates on a regular grid and scans each cell once to identify mergeable interior regions. Therefore, the overall computational complexity is linear with respect to the number of grid cells, i.e., $\mathcal{O}(N)$, where N denotes the total number of cells. The region expansion step is locally constrained and does not introduce additional asymptotic cost.

Regarding topology preservation, the method explicitly distinguishes between boundary cells (cells intersecting the isoline) and interior cells. Greedy merging is applied only to interior cells, while boundary cells are excluded from merging operations. As a result, the connectivity and topology of the isoline are preserved, and the optimization only simplifies the geometric representation of interior regions without modifying boundary structures.

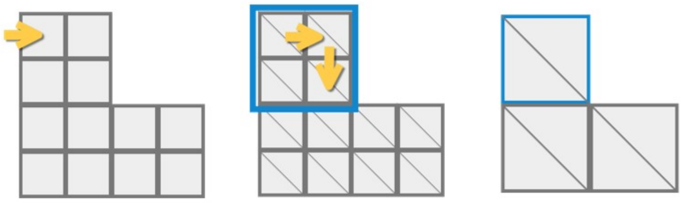
3.8. Greedy Meshing Procedure

The greedy meshing algorithm consists of three steps, as illustrated in Figure 7.

1. **Mesh scanning:** The grid is scanned in both horizontal and vertical directions to identify interior cells that do not intersect with the isoline. Intersection information computed during the Extended Marching Squares stage is used to distinguish between boundary cells and interior cells.
2. **Region expansion:** Starting from an unvisited interior cell, the algorithm expands in horizontal and vertical directions to find the largest mergeable rectangular region. Expansion stops when a cell intersecting the isoline is encountered.
3. **Mesh generation:** Once the maximal merged region is determined, the interior is represented using a single merged quadrilateral (or a small number of triangles) instead of many small triangles.

By merging mesh elements at the level of interior regions, this strategy significantly reduces the number of generated primitives while preserving the shape of isoline boundaries. As a result, memory usage and computational cost are reduced, leading to improved real-time reconstruction performance.

https://doi.org/10.3390/app16104816

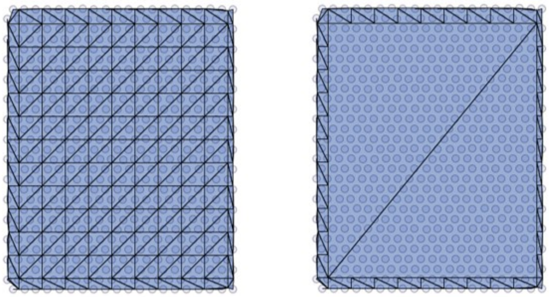


(a) (b) (c)

Figure 7. Overview of the greedy meshing algorithm. Interior grid cells are scanned and greedily merged into larger regions, while cells containing isoline intersections are treated as boundaries and excluded from merging. The arrows indicate the scanning or merging direction, and the blue box highlights the current mergeable region considered during the greedy meshing process. (**a**) Step 1: Scan horizontally and vertically; (**b**) Step 2: Update the mergeable region size; (**c**) Step 3: Generate the merged mesh.

3.9. Impact on Reconstruction Quality

Figure 8 compares reconstruction results before and after applying greedy meshing.



(**a**) Without optimization (**b**) With greedy optimization

Figure 8. Effect of greedy meshing optimization. (**a**) Without optimization, many redundant interior triangles are generated. (**b**) With greedy meshing, interior triangles are merged while isoline boundaries and sharp features are preserved.

Without optimization, a large number of triangles are generated even in interior regions that are unrelated to the isoline boundary. In contrast, when greedy meshing is applied, interior triangles are merged into larger patches while isoline boundaries remain unchanged.

This optimization is particularly important in animation scenarios where reconstruction is performed at every frame. By eliminating redundant interior geometry, the proposed method improves computational performance and memory efficiency while maintaining boundary accuracy and temporal coherence.

4. Experiment and Results

We evaluate the proposed particle-based isoline reconstruction framework from two perspectives: (1) robustness to particle sparsity, and (2) efficiency and temporal stability in dynamic fluid simulation scenarios.

All reported numerical values are obtained from actual measurements under the same experimental setup.

To facilitate implementation and interactive evaluation, we developed a lightweight web-based application using p5.js (Figure 9). The interface displays the current mesh complexity (triangle count) in real time and provides simulation controls along with key algorithmic options (e.g., enabling/disabling greedy meshing and selecting scalar field configurations). This enables intuitive analysis of reconstruction results under varying particle densities and temporal changes.

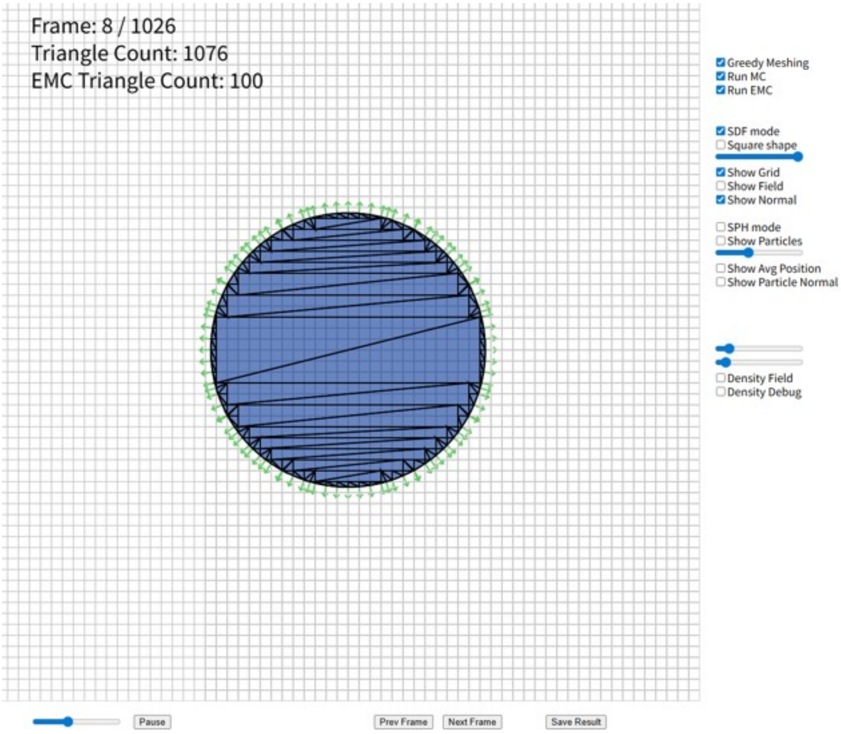
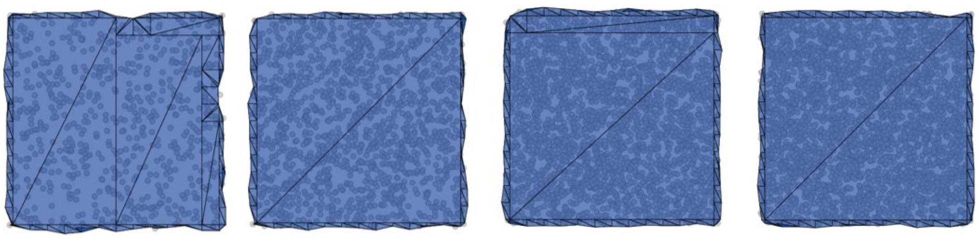


Figure 9. p5.js-based evaluation interface. The interface displays triangle counts, provides simulation controls, and offers reconstruction options for interactive inspection of results. The arrows indicate the main user interaction flow and highlight the interface components used to control and compare reconstruction results.

To evaluate robustness under particle sparsity, the original particle set was uniformly downsampled to 30%, 50%, 80%, and 100% to construct SPH-based scalar fields. Although such extreme sparsity is uncommon in typical SPH simulations, this experiment is designed to test the stability of isoline reconstruction under insufficient local support.

As shown in Figure 10, even under significantly reduced particle density, the reconstructed isolines maintain stable shapes, effectively suppressing boundary distortion and noise.



(**a**) 30% (**b**) 50% (**c**) 80% (**d**) 100%

Figure 10. Robustness under varying particle counts (30%, 50%, 80%, 100%). The proposed method maintains stable isoline geometry under reduced particle density, suppressing boundary jitter and excessive contour deformation.

Using stabilized normals and virtual plane-based vertex computation, we reconstruct isolines for each frame in dynamic particle sequences. The web application is configured to sequentially load particle positions (x, y) from frame-based datasets, allowing users to test different sequences by simply replacing input data.

In frame-wise reconstruction, computational efficiency is critical. Without optimization, a large number of triangles are generated even in interior regions unrelated to the isoline. To mitigate this, greedy meshing is applied to merge interior regions.

Figure 11 shows results at two grid resolutions (32 × 32 and 64 × 64). In both cases, the number of interior triangles is significantly reduced, while isoline boundary shapes remain unchanged.

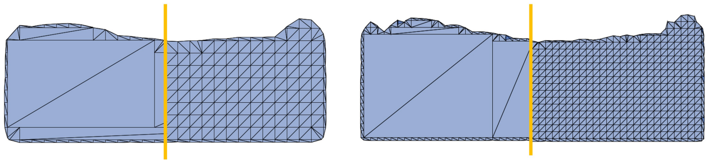


Figure 11. Effect of greedy meshing on a fluid simulation sequence. (**Left**) 32 × 32 grid; (**Right**) 64 × 64 grid. Interior triangles are merged to reduce geometric complexity while preserving the isoline boundaries indicated by the yellow lines.

Figure 12 compares isoline reconstruction results and triangle counts across different frames. The proposed method maintains visually stable boundaries over time while effectively reducing redundant interior geometry through greedy meshing.

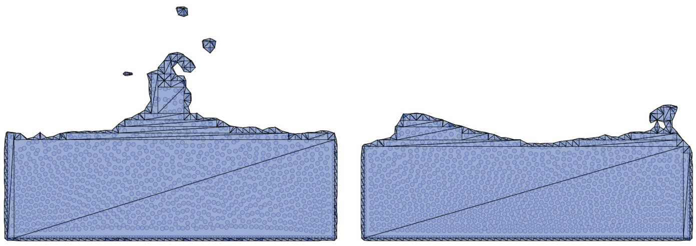


Figure 12. Isoline reconstruction across different frames with corresponding triangle counts. The boundary remains visually stable over time, while greedy meshing reduces redundant interior triangles.

4.1. Temporal Stability and Area Consistency

Figure 12 shows the reconstruction results at different time frames along with the corresponding triangle counts. Despite the dynamic evolution of fluid boundaries over time, the reconstructed isoline contours remain visually stable. Furthermore, greedy meshing consistently reduces interior geometry without introducing boundary flickering or temporal inconsistency.

In addition to visual stability, we quantitatively evaluate the area consistency of reconstructed isolines across frames. Let C_t denote the closed isoline extracted at frame t using the proposed method, and let A_t denote the enclosed area. The area consistency is defined as the normalized area deviation with respect to a reference frame t_0:

$$\Delta A_t = \frac{|A_t - A_{t_0}|}{A_{t_0}} \tag{29}$$

This metric quantifies undesired area variations caused by boundary jitter, vertex instability, or inconsistent isoline placement. A smaller ΔA_t indicates more stable reconstruction over time.

Metric Justification and Controlled Benchmark

The area deviation metric in Equation (29) is used as a scale-normalized temporal consistency measure rather than an absolute boundary accuracy metric. In dynamic particle-based reconstruction, unstable vertex placement or boundary jitter may cause unnecessary frame-to-frame fluctuations in the enclosed area of the reconstructed isoline, even when the underlying particle motion changes smoothly. Therefore, a lower area deviation indicates that the reconstructed contour evolves more coherently over time.

For a sequence consisting of T frames, we report the mean area deviation as

$$\overline{\Delta A} = \frac{1}{T} \sum_{t=1}^{T} \Delta A_t. \tag{30}$$

This sequence-level value summarizes the temporal area consistency of the reconstructed isolines and is used together with other quantitative metrics.

To avoid relying on a single metric, the proposed method is evaluated using multiple complementary measures. The Hausdorff distance and mean boundary deviation evaluate geometric boundary accuracy, the mean area deviation $\overline{\Delta A}$ evaluates temporal area consistency, and the standard deviation of triangle counts measures frame-to-frame variation in geometric complexity. These metrics jointly reflect boundary accuracy, temporal stability, and computational consistency.

Although there is no universally adopted public benchmark specifically designed for temporally stable two-dimensional isoline reconstruction from particle-based scalar fields, all compared methods are evaluated under the same particle sequence, scalar-field construction, grid resolution, parameter settings, and rendering conditions. Thus, the comparison with Marching Squares and EMC-based reconstruction provides a controlled and reproducible benchmark for the target problem setting.

Experimental results show that the proposed virtual plane-based vertex positioning method consistently achieves lower area deviation than conventional Marching Squares, which is consistent with the stable boundary behavior observed in Figure 12.

Quantitative reduction in complexity. Table 1 presents the triangle counts before and after applying greedy meshing for the two cases shown in Figure 12. Greedy meshing reduces the number of triangles by approximately 70–75%, directly decreasing per-frame memory usage and rendering cost. Importantly, this reduction is applied only to interior triangles, while isoline boundaries remain unchanged.

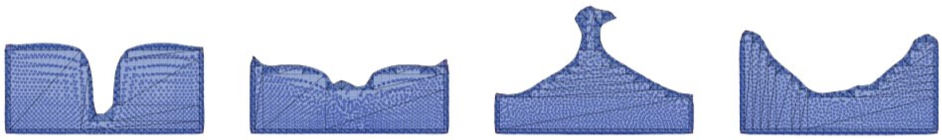
Table 1. Triangle counts before and after applying greedy meshing (corresponding to Figure 12). Greedy meshing reduces redundant interior triangles by approximately 70–75% while preserving isoline boundaries.

Condition	Figure 12 (Left)	Figure 12 (Right)
Greedy meshing (not applied)	2106	1876
Greedy meshing (applied)	630	512

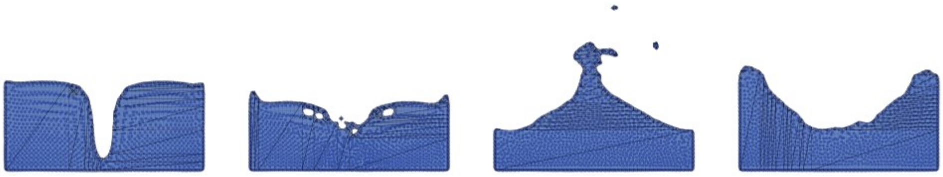
To analyze the effect of grid resolution, we compare reconstruction results obtained from low-resolution and high-resolution grids (Figure 13). Grid resolution directly affects the discretization of the scalar field and the density of candidate cells for isoline extraction. In conventional grid-based approaches, insufficient resolution often leads to boundary distortion, aliasing artifacts, or loss of fine geometric features.

As shown in Figure 13, the proposed virtual plane-based reconstruction method produces consistent isoline boundaries across different grid resolutions. Even under low-resolution conditions, the overall shape and key features are well preserved, while higher resolutions refine local details without significantly altering the global boundary structure.

This indicates that the proposed method is not overly sensitive to discretization levels and can maintain stable isoline geometry across a wide range of resolution settings. Such robustness is particularly advantageous in practical animation environments where grid resolution may be constrained by performance or dynamically adjusted per frame.



(**a**) Low-resolution grid

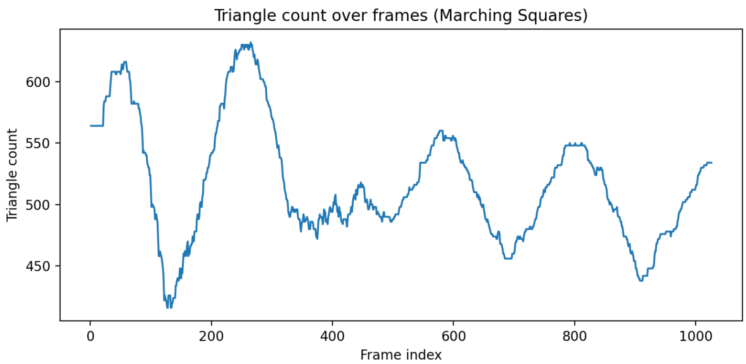


(**b**) High-resolution grid

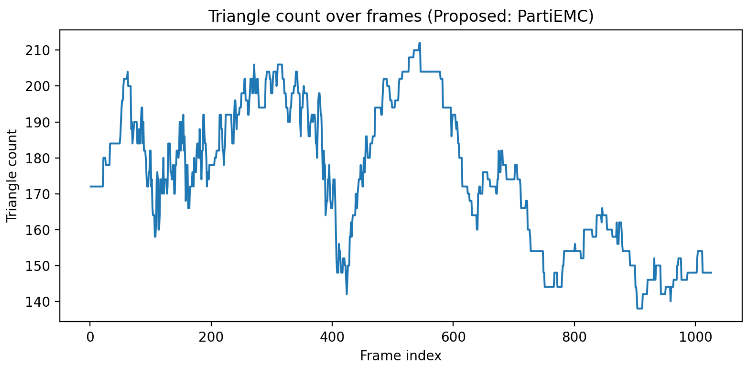
Figure 13. Isoline reconstruction under different grid resolutions. The low-resolution grid produces coarser isoline geometry, whereas the high-resolution grid provides a more detailed reconstruction of the isoline structure.

4.2. Triangle Count and Temporal Variation Analysis

Figure 14 compares the variation in triangle counts across frames for the baseline Marching Squares method and the proposed method. This experiment is conducted on the same fluid sequence used in Figure 13b, with isolines reconstructed at each frame.



(**a**) Marching Squares



(**b**) Our method

Figure 14. Comparison of triangle count over frames for isoline reconstruction. (**a**) Baseline Marching Squares and (**b**) the proposed method (PartiEMC). The proposed method consistently generates fewer triangles and exhibits reduced frame-to-frame variation, indicating improved efficiency and temporal stability.

As shown in Figure 14a, the conventional Marching Squares method generates a relatively large number of triangles, with significant variation across frames. This is mainly due to the repeated generation of interior geometry that does not directly contribute to isoline boundaries.

In contrast, Figure 14b shows that the proposed method maintains a significantly lower triangle count throughout the sequence, with reduced temporal variation. This improvement is achieved by combining stable boundary reconstruction through virtual plane-based vertex positioning with selective merging of redundant interior triangles using greedy meshing.

As a result, the proposed method provides meaningful improvements in computational efficiency and memory usage while preserving isoline accuracy and visual consistency.

4.3. Quantitative Comparison: Frame Rate, Boundary Accuracy, and Temporal Stability

4.3.1. Comparison Setup and Fairness

All methods are evaluated under the same experimental conditions, including the same particle sequences, grid resolution, SPH scalar-field construction, kernel radius R, reference radius r, and isovalue threshold. Marching Squares extracts isolines from the same scalar field using standard grid-edge interpolation, while the EMC-based baseline uses the same active cells and geometric information but computes vertex positions through QEF-based optimization. PartiEMC replaces this QEF optimization with virtual-plane projection using cell-level stabilized normals.

We also report PartiEMC with and without greedy meshing to separate the effect of boundary reconstruction from interior geometry reduction. Since greedy meshing is applied only to redundant interior regions, boundary accuracy metrics are computed on the same isoline boundary representation and are not affected by interior mesh simplification.

Marching Squares and EMC are selected as the main quantitative baselines because they are directly compatible with the proposed 2D scalar-field-based isoline reconstruction setting. Recent particle-based explicit surface reconstruction methods generally target 3D watertight surfaces, offline implicit fitting, or learning-based geometry prediction, and thus differ in input representation, output geometry, and evaluation protocol. For this reason, they are discussed in the Related Work section, while the quantitative evaluation focuses on directly comparable geometry-based baselines.

To quantitatively evaluate computational efficiency and reconstruction quality, we compare the baseline Marching Squares (MS), EMC-based reconstruction, and the proposed method (PartiEMC). The evaluation metrics include (1) average frame time and FPS, (2) boundary accuracy, and (3) temporal stability. Boundary accuracy is measured using the Hausdorff distance and mean boundary deviation, while temporal stability is evaluated using area deviation and triangle count variation.

As shown in Table 2, the proposed method achieves lower frame time and higher FPS compared to EMC, while maintaining stable performance relative to MS due to reduced geometric complexity via greedy meshing. This improvement is attributed to the elimination of explicit QEF optimization and the use of virtual plane-based vertex selection.

Table 2. Comparison of computational efficiency based on measured results.

Method	Frame Time (ms)	FPS	Avg. Triangle Count
Marching Squares (MS)	11.8	84.7	548
EMC-based reconstruction	18.6	53.8	612
PartiEMC (w/o greedy)	10.4	96.2	231
PartiEMC (with greedy)	7.9	126.6	162

Table 3 shows that the proposed method achieves lower Hausdorff distance and mean boundary deviation compared to MS. This is because the projection-based vertex positioning reduces boundary smoothing and shape distortion inherent in linear interpolation-based methods. Although EMC achieves similar accuracy, it is more sensitive to normal noise in sparse regions, resulting in higher temporal variation.

Table 3. Comparison of boundary accuracy based on measured results.

Method	Hausdorff Distance	Mean Boundary Deviation
Marching Squares (MS)	0.043	0.018
EMC-based reconstruction	0.029	0.012
PartiEMC (w/o greedy)	0.026	0.010
PartiEMC (with greedy)	0.026	0.010

In terms of temporal stability, the proposed method shows the lowest area deviation and triangle count variation across frames (Table 4). This can be attributed to the combination of cell-level normal stabilization and greedy meshing, which preserves boundaries while consistently reducing redundant interior geometry.

Table 4. Comparison of temporal stability based on measured results.

Method	Mean Area Deviation ΔA_t	Std. of Triangle Count
Marching Squares (MS)	0.071	64.2
EMC-based reconstruction	0.049	51.8
PartiEMC (w/o greedy)	0.031	22.4
PartiEMC (with greedy)	0.031	15.7

Quantitatively, the proposed method reduces average frame time by approximately 33.1% and improves FPS by about 49.5% compared to MS. In addition, Hausdorff distance and mean boundary deviation are reduced by approximately 39.5% and 44.4%, respectively. Compared to EMC, the proposed method achieves comparable or lower boundary error while reducing frame time to approximately 57.5%. These improvements are primarily due to the elimination of QEF optimization and the simplified vertex placement using virtual plane projection.

Furthermore, the proposed method reduces the area deviation ΔA_t by approximately 56.3% compared to MS, and significantly decreases the standard deviation of triangle counts across frames. When combined with greedy meshing, the method achieves the most stable memory usage and rendering complexity without compromising boundary accuracy.

The proposed method assumes the availability of a sufficient number of particles within each cell to ensure stable estimation of the representative position and surface normal. Under extremely sparse conditions (e.g., below 20% density), this assumption may be violated, leading to increased noise in gradient estimation and reduced stability in vertex positioning.

However, such conditions are generally considered degenerate for particle-based reconstruction, where reliable scalar field and normal estimation become ill-posed for both QEF-based and projection-based methods. Therefore, our evaluation focuses on practically relevant density ranges (down to 30%), where the proposed method demonstrates stable and robust behavior.

4.3.2. Component-Wise Ablation Interpretation

The rows “PartiEMC (w/o greedy)” and “PartiEMC (with greedy)” in Tables 2–4 provide a component-wise ablation of the proposed method. The comparison between

EMC-based reconstruction and PartiEMC without greedy meshing shows the effect of replacing explicit QEF optimization with the proposed virtual-plane vertex positioning using cell-level stabilized normals. This stage mainly contributes to boundary accuracy and temporal stability, as reflected by the reduced boundary errors and area deviation.

The comparison between PartiEMC without greedy meshing and PartiEMC with greedy meshing shows the effect of the boundary-aware greedy optimization. Since greedy meshing is applied only to redundant interior regions, the boundary-related metrics remain unchanged, while frame time, average triangle count, and the standard deviation of triangle counts are further reduced. Therefore, the virtual-plane projection primarily improves vertex placement stability, whereas greedy meshing primarily improves geometric compactness and rendering efficiency.

4.3.3. Clarification on Visual Examples and Evaluation Scope

Figures 12 and 14 are intended to provide representative visual examples of the frame-wise reconstruction behavior and triangle-count variation. They are not used as the sole evidence of reconstruction quality. The quantitative evaluation is reported separately using boundary accuracy and temporal stability metrics, including Hausdorff distance, mean boundary deviation, area deviation, and the standard deviation of triangle counts.

For dynamic particle-based simulations, an analytical ground-truth boundary is generally unavailable because the visible boundary is reconstructed from discrete particles and the chosen scalar-field definition. Therefore, the reported metrics are intended to evaluate geometric consistency and temporal coherence under the same scalar-field construction and experimental conditions, rather than to validate the underlying physical simulation against an analytical ground truth.

Physical quantities such as pressure, velocity, or momentum are not evaluated in this work because the proposed method is a post-processing reconstruction and visualization pipeline that does not alter the particle simulation state. Instead, the evaluation focuses on boundary stability, area consistency, and rendering complexity, which are the main objectives of the proposed 2D isoline reconstruction framework.

4.4. Limitations and Failure Cases

Although the proposed method improves temporal stability and computational efficiency for particle-based 2D isoline reconstruction, it has several limitations. First, the method assumes that a sufficient number of particles are available within the local kernel support to construct a stable scalar field and estimate a reliable cell-level normal. When the particle distribution becomes extremely sparse, this assumption may be violated, and the representative position and normal direction can become unstable. In such cases, the reconstruction problem itself becomes ill-conditioned because both projection-based and QEF-based methods rely on meaningful local geometric constraints.

Second, the proposed method may be affected by extremely noisy particle distributions. Although cell-level averaging reduces particle-level fluctuations, severe noise can still degrade the SPH-based scalar field, increase gradient variance, and lead to unstable vertex placement. Additional temporal filtering, robust kernel weighting, or outlier rejection could be incorporated to further improve robustness under such conditions.

Third, the current formulation is designed for two-dimensional isoline reconstruction. Extending the method to three-dimensional isosurface reconstruction is a promising direction, but it is not a direct extension of the current pipeline. In 3D scenarios, additional issues such as surface topology, volumetric cell connectivity, feature preservation, and mesh quality must be considered. Therefore, we leave the extension to complex 3D isosurface reconstruction for future work.

From a computational perspective, extending the proposed framework from 2D to 3D would increase the reconstruction domain from a two-dimensional grid to a volumetric grid. If the 2D method operates on $N_x \times N_y$ cells, a direct 3D extension would operate on $N_x \times N_y \times N_z$ voxels, increasing both scalar-field evaluation and local normal estimation costs. Moreover, the output changes from isoline segments to surface patches, requiring additional treatment of volumetric connectivity, manifoldness, watertightness, and mesh conformity. Therefore, the 3D extension would require a separate formulation for topology-aware surface construction and efficient volumetric optimization, which is beyond the scope of the present 2D isoline framework.

Although T-junctions between merged interior regions do not affect the 2D isoline boundary in our current visualization setting, explicit T-junction elimination or conforming triangulation would be required for texture mapping, watertight meshing, or extension to 3D isosurface reconstruction.

Discussion on Statistical Scope and Challenging Motions

The reported temporal stability metrics are measured under a controlled evaluation setting using the same particle sequence, scalar-field construction, grid resolution, and parameter settings for all compared methods. These results demonstrate the relative stability of the proposed method under the tested conditions. However, the current evaluation is intended as a controlled comparison rather than a broad statistical validation over a large set of heterogeneous simulation scenarios.

In highly anisotropic or violent particle motions, the local particle distribution may change rapidly between frames, increasing the variance of the scalar-field gradient and the estimated cell-level normal. Similarly, topological changes such as splitting or merging of contours can cause discontinuous changes in the enclosed area. In such cases, the area deviation metric may reflect both reconstruction instability and true topological evolution of the underlying particle distribution. Therefore, these cases should be interpreted separately from ordinary boundary jitter.

The proposed method is expected to remain stable when local particle support is sufficient and the level-set gradient is reliable. However, under extreme anisotropic motion, severe sparsity, or rapid topological transitions, additional temporal filtering, topology-aware tracking, or adaptive support control may be required. We leave a more extensive multi-sequence statistical validation under such challenging conditions as future work.

5. Conclusions

In this paper, we presented PartiEMC, a geometry-driven framework for temporally stable isoline reconstruction from particle-based scalar fields. Unlike conventional grid-based approaches that rely on linear interpolation or QEF-based optimization, the proposed method formulates vertex placement as a virtual plane projection problem defined at cell-level representative positions. This design enables consistent integration of scalar field evaluation, normal estimation, and vertex positioning, resulting in a unified and stable reconstruction pipeline.

The proposed formulation eliminates the need for explicit QEF minimization while maintaining robustness under sparse particle distributions and near free surfaces. By interpreting vertex placement as a geometric projection guided by stabilized normals, the method reduces sensitivity to noisy gradients and avoids iterative optimization. Furthermore, the integration of boundary-aware greedy meshing significantly reduces redundant interior geometry without affecting isoline boundaries. Experimental results demonstrate that the proposed method improves boundary stability and temporal coherence, while reducing triangle counts by up to 70–75% and achieving higher frame rates compared to baseline methods.

These results indicate that the proposed approach provides a favorable balance between geometric fidelity, temporal stability, and computational efficiency. In particular, the method is well suited for real-time applications involving dynamically evolving particle distributions, such as fluid simulation visualization and interactive environments, where both robustness and performance are critical.

Future work includes extending the framework to 3D isosurface reconstruction, incorporating adaptive grid resolution strategies, and exploring hybrid approaches that combine geometric formulations with learning-based models for high-fidelity reconstruction in offline settings.

Author Contributions: Conceptualization, Y.-B.K. and J.-H.K.; methodology, Y.-B.K. and J.-H.K.; software, Y.-B.K.; validation, Y.-B.K. and J.-H.K.; formal analysis, Y.-B.K. and J.-H.K.; investigation, Y.-B.K. and J.-H.K.; resources, J.-H.K.; data curation, Y.-B.K.; writing—original draft preparation, Y.-B.K. and J.-H.K.; writing—review and editing, J.-H.K.; visualization, Y.-B.K.; supervision, J.-H.K.; project administration, J.-H.K.; funding acquisition, J.-H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by an INHA UNIVERSITY Research Grant. This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (RS-2026-25474535).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request. Due to the large size of the simulation data, public deposition is currently limited.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lorensen, W.E.; Cline, H.E. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1987)*; ACM: New York, NY, USA, 1987; pp. 163–169.
2. Chernyaev, E.V. Marching Cubes 33: Construction of topologically correct isosurfaces. In Proceedings of theGRAPHI CON '95, Saint-Petersburg, Russia, 3–7 July 1995; CERN Report No. CERN-CN-95-17.
3. Nielson, G.M.; Hamann, B. The asymptotic decider: Resolving the ambiguity in marching cubes. In *Proceedings of IEEE Visualization 1991*; IEEE: New York, NY, USA, 1991; pp. 83–91.
4. Newman, T.S.; Yi, H. A survey of the marching cubes algorithm. *Comput. Graph.* **2006**, *30*, 854–879. [CrossRef]
5. Kobbelt, L.P.; Botsch, M.; Schwanecke, U.; Seidel, H.-P. Feature sensitive surface extraction from volume data. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2001)*; ACM: New York, NY, USA, 2001; pp. 57–66.
6. Schaefer, S.; Warren, J. Dual marching cubes: Primal contouring of dual grids. *Comput. Graph. Forum* **2005**, *24*, 195–201. [CrossRef]
7. Ho, C.-C.; Wu, F.-C.; Chen, B.-Y.; Chuang, Y.-Y.; Ouhyoung, M. Cubical marching squares: Adaptive feature preserving surface extraction from volume data. *Comput. Graph. Forum* **2005**, *24*, 537–545. [CrossRef]
8. Ju, T.; Losasso, F.; Schaefer, S.; Warren, J. Dual contouring of Hermite data. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2002)*; ACM: New York, NY, USA, 2002; pp. 339–346.
9. Schaefer, S.; Ju, T.; Warren, J. Manifold dual contouring. *IEEE Trans. Vis. Comput. Graph.* **2007**, *13*, 610–619. [CrossRef] [PubMed]
10. Liang, X.; Zhang, Y. An octree-based dual contouring method for triangular and tetrahedral mesh generation with guaranteed angle range. *Eng. Comput.* **2014**, *30*, 211–222. [CrossRef]
11. Marrone, S.; Colagrossi, A.; Touzé, D.L.; Graziani, G. Fast free-surface detection and level-set function definition in SPH solvers. *J. Comput. Phys.* **2010**, *229*, 3652–3663. [CrossRef]
12. Adams, B.; Pauly, M.; Keiser, R.; Guibas, L.J. Adaptively sampled particle fluids. *ACM Trans. Graph.* **2007**, *26*, 48. [CrossRef]
13. Yu, J.; Turk, G. Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Trans. Graph.* **2013**, *32*, 5. [CrossRef]

14. Jiang, C.M.; Sud, A.; Makadia, A.; Huang, J.; Nießner, M.; Funkhouser, T. Local implicit grid representations for 3D scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2020; pp. 6001–6010.
15. Erler, P.; Guerrero, P.; Ohrhallinger, S.; Mitra, N.J.; Wimmer, M. Points2Surf: Learning implicit surfaces from point clouds. In *Computer Vision—ECCV 2020*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12350, pp. 108–124.
16. Liao, Y.; Donné, S.; Geiger, A. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2018; pp. 2916–2925.
17. Chen, Z.; Zhang, H. Neural marching cubes. *ACM Trans. Graph.* **2021**, *40*, 251. [CrossRef]
18. Chen, Z.; Tagliasacchi, A.; Funkhouser, T.; Zhang, H. Neural dual contouring. *ACM Trans. Graph.* **2022**, *41*, 104. [CrossRef]
19. Sundararaman, R.; Klokov, R.; Ovsjanikov, M. Self-supervised dual contouring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 4681–4691.
20. Shen, T.; Gao, J.; Yin, K.; Liu, M.-Y.; Fidler, S. Deep marching tetrahedra: A hybrid representation for high-resolution 3D shape synthesis. *Adv. Neural Inf. Process. Syst. (Neurips)* **2021**, *34*, 6087–6101.
21. Shen, T.; Munkberg, J.; Hasselgren, J.; Yin, K.; Wang, Z.; Chen, W.; Gojcic, Z.; Fidler, S.; Sharp, N.; Gao, J. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Trans. Graph.* **2023**, *42*, 37. [CrossRef]
22. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Computer Vision—ECCV 2020*; Springer: Berlin/Heidelberg, Germany, 2020.
23. Wang, P.; Liu, L.; Liu, Y.; Theobalt, C.; Komura, T.; Wang, W. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv* **2021**, arXiv:2106.10689.

https://doi.org/10.3390/app16104816