Computers and Artificial Intelligence；Vol.3, No.3; 2026
E-ISSN: 3006 -3728 P-ISSN：3006-371X
DOI: https://doi.org/10.70267/cai.26v3n3.2133
Published by: Zeus Press

Mesh Simplification Method Based on Implicit Geometric Constraints

Yangyang Qiu*

Chongqing Normal University, Chongqing, China

**Corresponding author: Yangyang Qiu.*

Abstract

To address the lack of continuous geometric constraints in traditional quadric error metrics (QEM)-based mesh simplification under high simplification ratios, this paper proposes a 3D mesh simplification method with implicit geometric constraints. First, TetWeave is used to perform implicit modeling on the original high-resolution mesh, from which a continuous implicit reference field is constructed, together with query modules for implicit deviation and implicit normal evaluation at candidate collapse points. Then, within the classical edge-collapse framework, the implicit geometric deviation is introduced as an additional constraint term in a normalized composite cost, while implicit normal consistency is employed as a legality criterion for collapse execution. Finally, experiments on multiple models are conducted to analyze the influence of different weight settings on simplification error and computational cost. The results show that the constructed implicit query module can reliably reflect the geometry of the original continuous surface, and that the normalized implicit constraint has a significant influence on candidate edge ranking and the final Hausdorff distance. With an appropriate weight setting, the proposed method improves the preservation of the original surface geometry, although it also introduces additional computational overhead. The proposed framework provides a feasible way to incorporate continuous implicit geometric supervision into classical edge-collapse simplification.

Keywords

mesh simplification, quadric error metrics, implicit geometric constraint, edge collapse, Hausdorff distance

1. Introduction

The goal of 3D mesh simplification is to reduce the number of vertices and triangular faces while preserving the original geometric shape and visual characteristics as much as possible, thereby lowering the costs of storage, transmission, and real-time rendering. With the rapid development of virtual reality, digital twins, 3D reconstruction, and interactive visualization, high-resolution 3D models have been widely used in industrial design, cultural heritage preservation, medical modeling, and intelligent perception. However, such models usually contain a large number of vertices and faces, and direct processing often leads to high computational and memory costs. Therefore, how to achieve effective simplification while controlling geometric error remains an important problem in geometric processing.

Among existing mesh simplification methods, edge-collapse-based frameworks have been widely adopted because of their convenient local updates, simple implementation, and good controllability over topology changes. In particular, the quadric error metrics (QEM) [1] method evaluates candidate collapses by measuring the accumulated squared distance from a candidate vertex to its adjacent local planes. Owing to its high efficiency and stable performance, QEM has become one of the most representative baseline methods for triangular mesh simplification. Its main advantage lies in its clear cost formulation and efficient local update mechanism, which make it well suited to serve as a fundamental edge-collapse framework.

However, the cost formulation of QEM mainly relies on local plane approximation of discrete triangular faces, and is therefore still a form of local geometric fitting on discrete meshes. Under high simplification ratios, the geometric constraints imposed by the original continuous surface gradually weaken. As a result, relying only on discrete local plane error to rank candidate edges may cause the simplified mesh to deviate from the original continuous surface, especially in high-curvature regions, fine-detail regions, and local contour-sensitive areas. This limitation may further affect both the geometric accuracy and the visual quality of the final simplified model. Therefore, an important problem is how to introduce additional continuous-surface constraints while preserving the efficiency and stability of the classical QEM framework.

In recent years, implicit geometric representation has attracted increasing attention in tasks such as surface reconstruction, continuous surface fitting, and differentiable geometric modeling, because it describes 3D surfaces using a continuous scalar field. For a scalar function defined in space, its zero level set can be regarded as a continuous representation of the target surface. Therefore, if the original high-resolution mesh can be converted into a continuous implicit reference field, the deviation of candidate collapse points from the implicit zero level set can be used as an additional constraint in the simplification process. Furthermore, implicit normals can also be used to evaluate directional consistency between the post-collapse local surface and the original continuous reference surface. Compared with methods that only rely on discrete plane error, this strategy has the potential to improve the preservation of the original continuous geometric shape.

Motivated by this observation, this paper proposes a 3D mesh simplification method with implicit geometric constraints. First, TetWeave is used to perform implicit modeling on the original high-resolution mesh, from which a continuous implicit reference field is constructed together with query modules for implicit deviation and implicit normal evaluation at candidate collapse points. Then, within the classical QEM edge-collapse framework, the implicit geometric deviation is introduced into a normalized composite cost to guide candidate edge ranking. Finally, implicit normal consistency is employed as a legality criterion for collapse execution, so as to further constrain local surface-direction distortion after collapse. Experiments on multiple models are conducted to analyze the influence of normalized implicit constraints under different weight settings on geometric error and computational cost, thereby verifying the feasibility and effectiveness of introducing continuous implicit geometric supervision into the classical edge-collapse framework.

2. Related Work

Geometric-error-based mesh simplification methods are among the most classical and mature approaches in 3D model simplification research. Among them, simplification frameworks centered on the edge collapse operation have been widely studied. These methods progressively reduce mesh complexity by merging edges or vertices and employ geometric error functions to evaluate the influence of each simplification operation on the original model shape.

Among various geometric error metrics, the quadric error metrics (QEM) method is regarded as one of the most representative techniques for edge-collapse-based mesh simplification due to its high computational efficiency, simple implementation, and stable performance. QEM-based simplification has been widely applied in both academic research and industrial practice, demonstrating good robustness and practicality in many application scenarios. However, the traditional QEM method mainly focuses on minimizing overall geometric error and pays limited attention to regional differences in geometric characteristics.

To address this limitation, many studies have introduced feature-enhancement strategies into the QEM framework in order to improve the sensitivity of simplification algorithms to local geometric features. Zhang et al. [2] incorporated vertex curvature and triangle density factors into the error evaluation, so that collapse operations in regions with prominent local details could be delayed, thereby preserving geometric features more effectively. Chu et al. [3] introduced a curvature-like feature metric into the QEM error and further used adjustable parameters to control feature sensitivity, so as to avoid the premature collapse of high-curvature regions. Jiang et al. [4] proposed a new vertex quadric error matrix by integrating absolute curvature, local region area, and normal-angle constraints.

In addition, some researchers have extended the QEM framework from the perspective of collapse strategy and optimization formulation. Zhou et al. [5] proposed a mesh simplification method based on triangle collapse. Their method represents the mesh using a half-edge data structure and determines the collapse position and simplification cost by minimizing the squared distance from the collapse point to adjacent triangular faces, thereby obtaining the collapse order and iteratively performing simplification. While this method maintains good computational efficiency and preserves geometric features to a certain extent, it is still essentially a serial simplification framework based on geometric error metrics. Its ability to describe complex feature regions remains limited, and detail degradation may still occur under high simplification ratios.

Liang et al. [6] proposed a mesh simplification method that combines Gaussian curvature with QEM. By improving the collapse cost function, their method enhances feature preservation, and further introduces edge split operations to reduce skinny triangles and lower approximation error. In addition, the simplification process is formulated as an optimization problem aiming at minimizing geometric error, and the WOA-DE algorithm is employed to solve the optimal operation sequence. Experimental results showed that this method outperformed conventional approaches in feature preservation, mesh quality, and geometric error, although its optimization process is relatively complex and computationally expensive.

Ungvichian et al. [7] further introduced principal curvatures and their directions into the QEM framework. By combining curvature, edge length, normal angle, and quadric error to construct the collapse cost, and by imposing shape constraints on the generated triangles, their method improves feature preservation capability. Experimental results showed that the method achieved good geometric error performance under moderate simplification ratios, but it was still inferior to conventional QEM under high simplification ratios, indicating that its robustness remains to be improved.

Overall, existing mesh simplification studies based on the QEM framework mainly focus on improving local feature awareness through enhanced error functions, curvature-related constraints, or optimization strategies. Although these methods improve geometric feature preservation to some extent, they are still primarily built upon discrete local geometric approximation and lack direct constraints from the original continuous surface. As a result, under high simplification ratios, detail degradation and shape deviation may still occur. This observation also motivates the introduction of continuous geometric information into the classical edge-collapse framework.

3. Method

3.1 Overall Framework

This paper proposes a 3D mesh simplification method with implicit geometric constraints. The core idea is to introduce continuous implicit geometric supervision into the classical QEM-based edge-collapse framework, so that the selection of candidate collapse points is influenced not only by discrete local plane errors but also by geometric constraints derived from the original continuous surface. Unlike conventional QEM, which ranks candidate edges solely according to local errors on the discrete mesh, the proposed method further constructs a continuous implicit reference field from the original high-resolution mesh and constrains the simplification process through both implicit deviation and implicit normal information, thereby improving the ability of the simplified mesh to approximate the original continuous surface.

The proposed framework consists of four stages. First, the original high-resolution triangular mesh is taken as the supervision target, and TetWeave [8] is used to construct a tetrahedron-supported piecewise linear implicit field. The zero level set extracted from this field serves as the reconstructed surface. It should be emphasized that the focus of this work is not to use the TetWeave reconstruction itself as a new simplification target, but rather to use the underlying continuous implicit representation as a geometric reference for the original model. In other words, the main role of TetWeave in this work is to provide a queryable continuous implicit field rather than simply generating a replacement mesh.

Second, an implicit query module is built on the constructed implicit field. For any candidate collapse point, the query module returns both its implicit scalar value and its local implicit normal. The scalar value is used to characterize the deviation of the candidate point from the original continuous reference surface, while the implicit normal is used to measure the directional consistency between the post-collapse local discrete surface and the continuous reference surface. Through this mechanism, candidate points in the edge-collapse process are equipped not only with discrete QEM information but also with continuous geometric information.

Third, before the actual simplification begins, a subset of candidate edges is sampled. Their QEM error terms and implicit deviation terms are computed, and the scale information of these two terms is estimated. The purpose of this step is to alleviate the inconsistency in magnitude between QEM costs and implicit deviations across different models, and to provide adaptive scale factors for the subsequent normalized composite cost. Through this treatment, the QEM term and the implicit deviation term are normalized to comparable scales, reducing the sensitivity of fixed weights across different models.

Finally, during the actual simplification process, the classical QEM cost, the deviation of the candidate collapse point from the implicit zero level set, and the normalized composite cost are computed for each candidate edge. A priority queue is then constructed according to the composite cost. Each time the current best candidate edge is extracted from the queue, the implicit normal at the candidate collapse point is further queried and compared with the normal direction of the local discrete surface after collapse. If the normal deviation exceeds a predefined threshold, the collapse is rejected because it would cause significant local surface-direction distortion; otherwise, the edge collapse is executed and the local topology, error matrices, and affected candidate costs are updated until the target simplification level is reached.

Overall, the proposed method preserves the dominant role of QEM in candidate edge ranking, while introducing additional constraints based on continuous implicit geometry. The implicit deviation term is mainly used to enhance the proximity of collapse points to the original continuous surface, whereas implicit normal consistency is employed as a legality criterion to prevent excessive local surface-direction distortion after collapse. In this way, continuous implicit geometric supervision is incorporated into the classical edge-collapse framework to improve geometric error control and surface-quality preservation.

3.2 Continuous Implicit Geometric Representation

To introduce continuous geometric reference information into the simplification process, this paper adopts a tetrahedron-supported piecewise linear implicit representation to describe the original high-resolution model. Unlike a conventional triangular mesh, which explicitly represents a surface using vertices and faces, implicit representation characterizes the target surface by a scalar field in space and uses the zero level set as a continuous geometric reference. This representation can provide candidate collapse points with both continuous surface deviation and local normal-direction information, which serves as the basis for subsequent geometric constraints in the edge-collapse process.

Let the support-point set obtained by tetrahedralizing the original high-resolution mesh be denoted by $X = x_i$, where each support point is associated with a scalar value. Within each tetrahedral element, the implicit function is defined by linear interpolation of the scalar values at the tetrahedral vertices. Suppose a query point x\mathbf{x}x lies inside a tetrahedron, and its barycentric coordinates with respect to the four vertices x_1, x_2, x_3, x_4 are $\lambda_1, \lambda_2, \lambda_3, \lambda_4$, respectively. Then the implicit function value can be expressed as

$$\phi(\mathbf{x}) = \lambda_1 s_1 + \lambda_2 s_2 + \lambda_3 s_3 + \lambda_4 s_4 \quad (1)$$

The zero level set

$$\mathcal{S} = \mathbf{x} \in \mathbb{R}^3 \mid \phi(\mathbf{x}) = 0 \quad (2)$$

can be regarded as the continuous reference surface corresponding to the original model. For any spatial point, the sign of $\phi(\mathbf{x})$ reflects its positional relationship to the reference surface, while $\phi(\mathbf{x})$ can be used to measure the deviation of the point from that surface. Therefore, during edge-collapse simplification, if the implicit function value of a candidate collapse point is closer to zero, the point is closer to the original continuous surface.

Since the implicit function is linear within each tetrahedron, it can also be written in the local form

$$\phi(x, y, z) = ax + by + cz + d \tag{3}$$

where a, b, c, and d are local coefficients associated with the current tetrahedral element. The gradient of the implicit function inside the tetrahedron is therefore constant:

$$\nabla\phi(x, y, z) = [a, b, c]^{T} \tag{4}$$

After normalization, the corresponding unit implicit normal can be obtained as

$$\mathbf{n}_{\mathrm{imp}}(\mathbf{x}) = \frac{\nabla\phi(\mathbf{x})}{\|\nabla\phi(\mathbf{x})\|} \tag{5}$$

This property implies that, under the piecewise linear implicit representation, any candidate collapse point can be queried not only for its deviation from the original continuous surface, but also for the local normal direction of that continuous surface. The former provides the basis for the implicit deviation term in the composite cost, while the latter can be used as a legality criterion to constrain the direction of the post-collapse local surface.

From the implementation perspective, this work does not directly use the TetWeave-reconstructed mesh as a new simplification target. Instead, it uses the optimized tetrahedral support points and scalar-field parameters produced by TetWeave to construct a continuous implicit reference field for the original model. In other words, the role of TetWeave in this method is to provide a queryable continuous geometric reference rather than simply generating a higher-resolution replacement mesh. All implicit deviations and implicit normal information of candidate collapse points are derived from this implicit reference field.

3.3 Quadric Error Metrics

The Quadric Error Metric (QEM) is a classical and efficient method for evaluating errors in mesh simplification. It was first proposed by Michael Garland et al. [6] and has been widely adopted in edge-collapse-based 3D mesh simplification algorithms. The QEM method approximates the deviation of a vertex from its local geometric planes, providing a computationally efficient and stable geometric error metric for candidate simplification operations.

Within the QEM framework, an error matrix is first constructed for each vertex. This matrix is obtained by accumulating the plane equations defined by the triangular faces adjacent to the vertex. Let the plane parameters associated with a triangle adjacent to vertex v_i be defined as:

$$p = [a, b, c, d]^{T} \tag{6}$$

Then, the squared distance from a vertex to this plane can be expressed as:

$$v^{T}(pp^{T})v \tag{7}$$

where $v = [x, y, z, 1]^{T}$ represents the homogeneous coordinate of the vertex. By accumulating the plane errors of all adjacent faces, the quadric error matrix of vertex v_i is given by:

$$Q_i = \sum_{p \in \mathcal{P}(v_i)} p\, p^{T} \tag{8}$$

where $\mathcal{P}(v_i)$ denotes the set of planes adjacent to vertex v_i.

When a candidate edge (v_i, v_j) is collapsed into a new vertex v, the geometric error introduced by this operation is defined as:

$$E(v) = v^{T}(Q_i + Q_j)v \tag{9}$$

Equation (9) represents the classical QEM cost function for candidate edge collapses. The core idea is to estimate the deviation of the new vertex from the original local plane set by summing the error matrices of the two endpoints, and to use this value as the priority criterion for edge collapse operations.

3.4 Implicit Geometric Deviation Constraint

In the traditional QEM framework, candidate edges are ranked only according to discrete local plane error, that is, according to how well a candidate collapse point preserves the local plane structure of the current mesh. However, this formulation mainly depends on the current discrete mesh itself and lacks direct geometric constraints from the original continuous surface. To address this limitation, this work introduces the deviation of a candidate collapse point from the continuous implicit reference surface as an additional constraint term, so as to enhance the proximity of candidate points to the original continuous surface.

Let v' be the new vertex after collapsing candidate edge(v_i, v_j). Its implicit geometric deviation is defined as

$$E_{imp}(v') = |\phi(v')| \tag{10}$$

where $\phi(v')$ is the scalar value returned by the implicit query module for the candidate collapse point. If $\phi(v') = 0$, then the point lies exactly on the implicit zero level set, i.e., on the continuous reference surface corresponding to the original high-resolution model. If $\phi(v')$ is larger, the point deviates further from the original continuous surface. Therefore, the implicit geometric deviation term provides a way to evaluate the geometric plausibility of a candidate collapse point from the perspective of the continuous reference surface.

In terms of geometric meaning, the conventional QEM cost mainly characterizes how well a candidate collapse point preserves the local discrete plane structure of the current mesh, whereas the implicit deviation term focuses on how well the candidate point approximates the original continuous reference surface. The former reflects local discrete geometric consistency, while the latter reflects global continuous geometric consistency. By introducing the implicit deviation term into candidate-edge evaluation, additional continuous geometric supervision can be imposed on the candidate collapse points while preserving the dominant role of QEM, thereby reducing the tendency of candidate points to deviate from the original continuous surface under high-ratio simplification.

It should be emphasized that the implicit geometric deviation term is treated here as a constraint cue rather than a replacement for the QEM cost. In other words, this work does not attempt to completely replace local error construction in traditional edge-collapse simplification with an implicit field. Instead, it introduces a continuous surface deviation term as an additional geometric reference for candidate-edge ranking, thereby improving the ability of the simplified mesh to preserve the original geometric shape.

3.5 Legality Constraint Based on Implicit Normal Consistency

To ensure that the local surface direction after a candidate collapse remains consistent with the original continuous reference surface, this paper further introduces an implicit normal consistency constraint. Let v' denote the candidate collapse point, and let the unit implicit normal returned by the implicit query module at v' be denoted by $\mathbf{n}_{imp}(v')$. Meanwhile, let $\mathcal{F}(v')$ denote the one-ring set of triangular faces adjacent to the post-collapse vertex v'. For any adjacent face $f \in \mathcal{F}(v')$, let its unit normal be $\hat{\mathbf{n}}_f$, and let its area be A_f. Then the local discrete normal after collapse is defined as the area-weighted average of the adjacent face normals:

$$\mathbf{n}_{mesh}(v') = \frac{\sum_{f\in\mathcal{F}(v')} A_f \hat{\mathbf{n}}_f}{\left\|\sum_{f\in\mathcal{F}(v')} A_f \hat{\mathbf{n}}_f\right\|} \tag{11}$$

Here, the local discrete normal $\mathbf{n}_{mesh}(v')$ reflects the average direction of the discrete surface in the neighborhood of the post-collapse vertex, and thus provides a good description of the local mesh orientation at that point.

The angle between the implicit normal and the local discrete normal is defined as

$$\theta(v') = \arccos\left(\mathbf{n}_{imp}(v') \cdot \mathbf{n}_{mesh}(v')\right) \tag{12}$$

If $\theta(v')$ is greater than a predefined threshold, the corresponding collapse is considered likely to cause significant local surface-direction distortion relative to the original continuous surface, and it is therefore rejected as an invalid collapse. Otherwise, the candidate collapse is regarded as satisfying the directional consistency requirement and can enter the subsequent execution stage.

Compared with using only the implicit deviation constraint, the implicit normal consistency constraint further filters candidate collapses from the perspective of surface-direction preservation. The former mainly constrains the candidate collapse point to remain close to the original continuous reference surface, whereas the latter prevents excessive local orientation shift after collapse. By combining the two, the proposed method is able to consider both positional deviation and directional consistency in the edge-collapse process, thereby improving the preservation of the original geometric shape.

3.6 Normalized Composite Cost Formulation

Since the QEM error term and the implicit deviation term may differ significantly in numerical magnitude across different models, directly combining them with a fixed coefficient may cause one term to dominate the composite cost and thus disturb candidate-edge ranking. In particular, the distribution of QEM costs is often affected by model scale, local geometric complexity, and candidate-edge composition, while the implicit deviation term may also exhibit different magnitudes depending on the distribution characteristics of the implicit field. To reduce the influence of such inter-model scale differences, this paper adopts a normalized weighted summation to construct the composite cost of candidate edges.

Let $E_{QEM}(v')$ denote the traditional QEM cost of candidate collapse point v', and let $E_{imp}(v')$ denote its implicit geometric deviation. Then the normalized composite cost is defined as

$$E(v') = \frac{E_{QEM}(v')}{s_{QEM}+\varepsilon} + \alpha \frac{E_{imp}(v')}{s_{imp}+\varepsilon} \tag{13}$$

where s_{QEM} and s_{imp} denote the scale estimates of the QEM error term and the implicit deviation term, respectively, α\alphaα is the relative weight of the implicit constraint term, and ε is a small constant introduced to avoid division by zero.

In this formulation, the first term represents the relative magnitude of the QEM error of a candidate edge with respect to the typical QEM scale of the current model, while the second term represents the relative magnitude of the implicit deviation of the candidate collapse point with respect to the typical implicit-deviation scale of the current model. In this way, cost terms originating from different sources and having different magnitudes are normalized to comparable scales, so that the composite cost becomes more adaptive across different models.

To estimate s_{QEM} and s_{imp}, a subset of candidate edges is randomly sampled before full edge-collapse execution. The QEM costs and implicit deviations of these sampled edges are computed, and the scale parameters are determined from their statistical values. Let the sampled QEM-cost set be $E_{QEM}^{(k)}$, and let the sampled implicit-deviation set be $E_{imp}^{(k)}$. This paper adopts the following scale estimation scheme:

$$s_{QEM} = \max\left(E_{QEM}^{(k)}\right) \tag{14}$$

$$s_{imp} = \max\left(E_{imp}^{(k)}\right) \tag{15}$$

The mean is adopted as the scale estimator because QEM costs are often highly concentrated, and an excessively small scale value may lead to over-amplification of the normalized QEM term, thereby disturbing the balance of the composite cost. Using the same statistical form for the implicit deviation term also helps maintain consistency between the two scale estimates. Once the scale factors are estimated from the sampled edges, they remain fixed throughout the simplification of the current model, thereby ensuring consistent candidate-edge ranking criteria.

It should also be noted that α in the normalized composite cost is not a fixed coefficient that directly reflects the absolute magnitude relationship between the two error terms. Rather, it controls the relative influence of the implicit constraint term after normalization. When α is small, the QEM term plays a stronger dominant role in the composite cost; when α is large, the implicit deviation term exerts stronger influence on candidate-edge ranking. Therefore, by comparing different values of α, the influence of implicit geometric constraints as an additional supervisory term on the final simplification error can be analyzed.

4. Experiments and Analysis

4.1 Experimental Environment and Datasets

The experimental environment used in this study is summarized in Table 1. The operating system was Windows 11, the processor was an AMD Ryzen 7 5800H with Radeon Graphics running at 3.20 GHz, the graphics card was an NVIDIA GeForce RTX 3060 Laptop GPU, and the development language was Python 3.10.

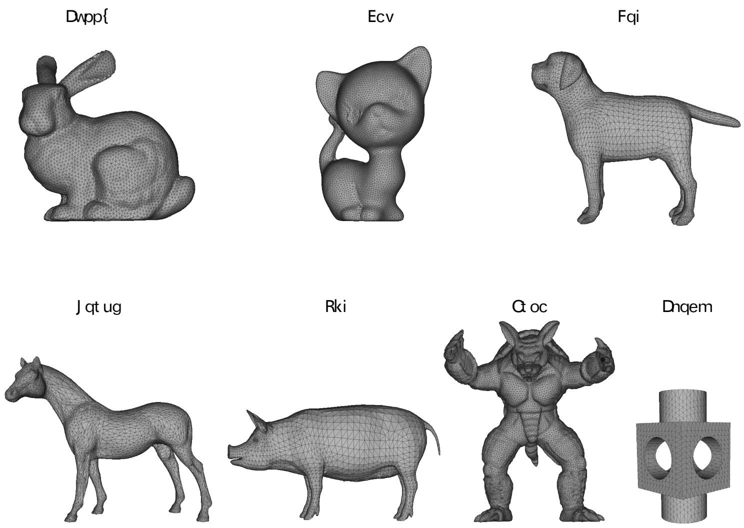
To ensure topological legality and geometric stability during simplification, the selected triangular mesh models satisfy the following basic conditions: each model is a 2-manifold structure, contains no self-intersections or degenerate faces, and has a watertight topology. These assumptions are commonly adopted in edge-collapse-based mesh simplification methods to avoid non-manifold structures or invalid geometric situations during simplification, thereby ensuring the comparability and stability of the experimental results.

The models used in the experiments include Bunny, Cat, Dog, Horse, Pig, Arma, and Block, as listed in Table 1. Specifically, Bunny has 6358 vertices, Cat has 9447 vertices, Dog has 7418 vertices, Horse has 6319 vertices, Pig has 3954 vertices, Arma has 8193 vertices, and Block has 2132 vertices. Their corresponding numbers of faces are 12712, 18894, 14977, 12607, 8011, 16382, and 4272, respectively. Figure 1 illustrates the 3D models used in the experiments.

Table 1: The 3D models used in the experiments

Module Name	Bunny	Cat	Dog	Horse	Pig	Arma	Block
Vertices	6358	9447	7418	6319	3954	8193	2132
Faces	12712	18894	14977	12607	8011	16382	4272

Figure 1: The 3D models used in the experiments



4.2 Evaluation Metrics

This section evaluates the performance of the mesh simplification methods from the perspectives of geometric error and simplification efficiency. The main evaluation metrics are defined as follows.

1) Hausdorff distance.

The Hausdorff distance is used to measure the geometric deviation between the original mesh and the simplified mesh. It is defined as

$$d_H(A,B) = \max\left\{\sup_{a\in A}\inf_{b\in B} d(a,b), \sup_{b\in B}\inf_{a\in A} d(a,b)\right\} \tag{16}$$

where $d(a,b)$ denotes the Euclidean distance between points aaa and bbb. The first term represents the maximum among the nearest distances from points on model A to model B while the second term represents the reverse distance. The Hausdorff distance takes the larger of the two directional errors and is thus used to characterize the worst-case geometric deviation between two mesh models.

2) Simplification time.

Simplification time is used to measure the computational efficiency of the mesh simplification process. It is defined as the total execution time required to complete the entire simplification procedure, measured in seconds. This metric reflects the computational performance of the algorithm under different simplification ratios and is particularly useful for evaluating the extra computational overhead introduced by implicit geometric constraints.

4.3 Experimental Results and Analysis

4.3.1 Validation of the Implicit Query Module

Before introducing continuous implicit geometric constraints into the QEM simplification framework, it is first necessary to verify whether the continuous implicit reference field constructed by TetWeave can provide reliable geometric supervision for the subsequent edge-collapse process. To this end, this work validates the consistency between the original meshes and the implicit-field reconstructions from both quantitative and qualitative perspectives. The goal here is not to directly use the reconstructed TetWeave mesh as a new simplification target, but rather to verify that the continuous implicit field learned by TetWeave is sufficiently accurate to provide reliable implicit deviation and implicit normal queries for candidate collapse points.

For quantitative evaluation, seven models—Bunny, Cat, Dog, Horse, Pig, Arma, and Block—were used to analyze the outputs of TetWeave. The evaluation metrics include Chamfer Distance (cd), normal consistency (nc), F1 score (f1), edge Chamfer Distance (ecd), and edge F1 score (ef1). Here, cd measures the overall geometric positional deviation between the reconstructed surface and the original surface, nc evaluates the average normal consistency between them, f1 reflects the overall reconstruction quality, while ecd and ef1 assess geometric deviation and feature preservation in edge-related regions.

As shown in Table 2, the cd values of all seven models are on the order of 10^{-6}, indicating that the implicit fields constructed by TetWeave are highly consistent with the original meshes in terms of overall geometric position. Specifically, Pig achieves the smallest cd of 2.3287×10^{-6}, while Dog and Horse also obtain relatively small values of $3.5702 \times 10^{-6}$and 3.7356×10^{-6}, respectively. Although Bunny, Cat, and Block exhibit slightly larger cd values, they still remain within the same order of magnitude. These results indicate that, from the perspective of overall surface approximation, the continuous implicit reference field produced by TetWeave can accurately fit the original surfaces of models with different complexities.

From the normal consistency metric nc, all models achieve values higher than 0.98, among which Dog reaches 0.9977, Bunny 0.9962, Cat 0.9953, and Block 0.9946. This indicates that the reconstructed surfaces maintain high consistency with the original input meshes in local normal direction, demonstrating that the continuous surface orientation obtained from the implicit field is reliable. Since the proposed simplification method further queries the implicit normal at each candidate collapse point and uses it as the legality criterion for collapse execution, the high nc values provide direct support for the later introduction of implicit normal consistency constraints.

In terms of the f1 metric, some differences can be observed across models. Pig yields the highest f1 value of 0.5759, while Dog and Horse obtain 0.4251 and 0.3987, respectively. Bunny and Cat reach 0.3240 and 0.3308, whereas Block attains 0.2965. Although the overall reconstruction matching ratio varies across models under strict thresholds, the values remain at an acceptable level. This suggests that TetWeave reconstruction exhibits relatively stable global geometric recovery performance, while differences in local structural complexity and surface-detail distribution may influence the strict-threshold matching ratio.

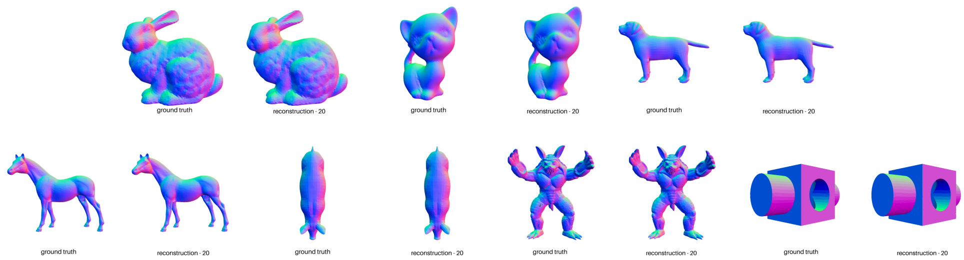
For edge-related metrics, more pronounced differences can be observed among models. Bunny and Cat achieve ef1 values of 0.8066 and 0.7929, indicating favorable edge reconstruction performance. Block further attains an ef1 value of 0.9330, while its ecd is only 4.2365×10^{-5}, suggesting that its boundaries and linear structural features are reconstructed very accurately. In contrast, Dog and Horse obtain relatively low ef1 values of 0.2255 and 0.2390, while Arma reaches 0.2464, indicating weaker feature-edge reconstruction quality on these models. Nevertheless, the overall trend shows that the principal contours and salient structures are still well preserved in the reconstructions.

For qualitative validation, Figure 2 presents visual comparisons between the original meshes and the zero-level-set reconstructions extracted from the implicit fields. From the visual results, the reconstructed surfaces can recover the overall contours, major structures, and principal local features of the original models. This indicates that the continuous implicit reference fields constructed by TetWeave are not only numerically consistent with the original meshes but also visually effective in shape reconstruction. Taken together, the quantitative and qualitative results demonstrate that the implicit fields learned by TetWeave can serve as reliable continuous geometric references, providing a solid basis for the implicit deviation and implicit normal queries of candidate collapse points in the subsequent simplification stage.

Table 2: Evaluation results of implicit reconstruction for different models

Model	cd	nc	f1	ecd	ef1
Bunny	4.9680e-06	0.9962	0.3240	0.0454	0.8066
Cat	5.0575e-06	0.9953	0.3308	0.0471	0.7929
Dog	3.5702e-06	0.9977	0.4251	0.1081	0.2255
Horse	3.7356e-06	0.9831	0.3987	0.0237	0.2390
Pig	2.3287e-06	0.9928	0.5759	0.0137	0.5358
Arma	3.7879e-06	0.9837	0.3935	0.0243	0.2464
Block	5.6431e-06	0.9946	0.2965	4.2365e-05	0.9330

Figure 2: Comparison diagrams of the original meshes of different models and the zero isosurface reconstruction results extracted from the implicit field



4.3.2 Simplification Error Analysis

Table 3: The influence of the weight of implicit geometric constraint terms on the simplification results

α	Hausdorff Distance
0.05	0.00792
0.1	0.00688
0.15	0.00621
0.2	0.00647
0.25	0.00708
0.3	0.00776

Table 3 presents the influence of the weight α of the implicit geometric constraint on the simplification result. To analyze the effect of the normalized implicit constraint term, Bunny was used as an example under an 80% simplification ratio, and the corresponding Hausdorff distances were measured under different values of α.

The results show that as α increases from 0.05 to 0.15, the Hausdorff distance gradually decreases, indicating that an appropriate increase in the implicit geometric constraint can improve the ability of the simplified mesh to approximate the original continuous surface. When α=0.15, the Hausdorff distance reaches the minimum value, suggesting that the normalized implicit deviation term and the traditional QEM term achieve a relatively good balance at this point.

However, when α is further increased to 0.20, 0.25, and 0.30, the Hausdorff distance rises again. This indicates that although the implicit geometric constraint can positively influence candidate-edge ranking, an excessively large weight will strongly disturb the local geometric preservation mechanism of the original QEM framework, thereby increasing the final geometric error. Therefore, under the normalized composite cost framework, the influence of the implicit geometric constraint on simplification error is not monotonically increasing; instead, there exists a relatively appropriate range of weight values.

Overall, the results indicate that, under the current experimental setting, α=0.15 yields a relatively favorable simplification error. This suggests that, while preserving the dominant role of QEM, a proper choice of the normalized implicit-constraint weight can improve the ability of the simplified mesh to preserve the original continuous surface.

Table 4 further compares the Hausdorff distances of different simplification methods under an 80% simplification ratio. The results show that the traditional QEM method produces relatively large geometric deviations on all test models. After introducing the implicit geometric deviation constraint, the "QEM + Implicit" method reduces the Hausdorff distance on all models, indicating that continuous implicit geometric information can effectively improve the consistency between the simplified mesh and the original geometry.

On this basis, the method proposed in this chapter, which further introduces implicit normal consistency, achieves the smallest Hausdorff distance on all test models. This indicates that collapse legality checking based on implicit normals can effectively suppress geometric distortion in high-curvature regions and thus achieve more accurate geometric approximation under high-ratio simplification.

Table 4: The Hausdorff distance under the 80% simplification condition for different simplification methods

Model	QEM	QEM +Implicit	Ours (Implicit+Normal)
Bunny	0.0128	0.0096	**0.0089**
Cat	0.0154	0.0117	**0.0108**
Dog	0.0179	0.0132	**0.0121**
Horse	0.0224	0.0198	**0.0195**
Pig	0.0168	0.0132	**0.0126**
Arma	0.0213	0.0165	**0.0152**
Block	0.0198	0.0156	**0.0134**

4.3.3 Time-Cost Analysis

Figure 3: Comparison of initialization time and iteration time among different methods on the Bunny model

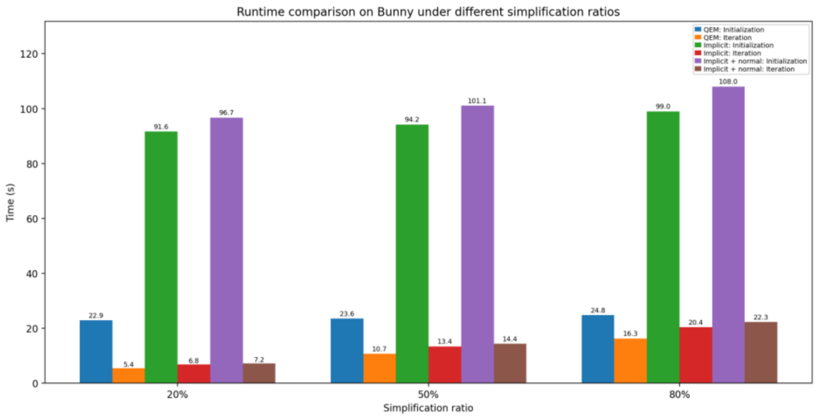


Figure 3 shows the comparison of initialization time and iteration time among different methods on the Bunny model under simplification ratios of 20%, 50%, and 80%. Overall, the initialization time of all three methods is significantly larger than the corresponding iteration time, indicating that preprocessing remains the major source of computational cost.

Among the three methods, the traditional QEM method has the lowest initialization cost, with approximate values of 22.90 s, 23.55 s, and 24.75 s under the three simplification ratios, respectively. After introducing implicit geometric constraints, the initialization time increases significantly to approximately 91.60 s, 94.20 s, and 99.00 s. When implicit normal consistency is further incorporated, the initialization time increases again to 96.73 s, 101.08 s, and 108.01 s. These results show that implicit-field loading, query-module construction, and the additional processing required by the normal-consistency constraint all contribute considerable preprocessing overhead.

From the perspective of iteration time, all three methods exhibit increasing runtime as the simplification ratio rises from 20% to 80%, indicating that the iteration-stage cost becomes more significant when more collapse operations must be performed. The traditional QEM method takes approximately 5.44 s, 10.72 s, and 16.32 s under the three simplification ratios, respectively. After introducing implicit geometric constraints, the iteration time increases to 6.80 s, 13.40 s, and 20.40 s. When implicit normal consistency is further added, the iteration time increases again to 7.18 s, 14.38 s, and 22.26 s. Compared with the traditional QEM baseline, the implicit geometric-constraint method shows higher iteration costs at all simplification ratios, while the introduction of implicit normal consistency further increases the computational overhead by about 5%–10% on top of the implicit-constraint version.

Overall, Figure 3 reveals two main observations. First, continuous implicit geometric constraints provide additional geometric supervision for the edge-collapse process, but they significantly increase both initialization and iteration costs. Second, although the implicit normal consistency constraint further strengthens geometric control during simplification, it also inevitably introduces additional computational overhead. Therefore, while the proposed method improves geometric supervision capability, subsequent work still needs to consider query acceleration and local update optimization to improve computational efficiency.

5. Conclusion

To address the lack of continuous geometric constraints in traditional QEM-based mesh simplification under high simplification ratios, this paper proposed a 3D mesh simplification method with implicit geometric constraints. The method uses TetWeave to perform implicit modeling on the original high-resolution mesh, constructs a continuous implicit reference field, and introduces both implicit geometric deviation and implicit normal consistency into the classical edge-collapse framework. Specifically, the implicit geometric deviation is used to measure the offset of a candidate collapse point from the original continuous reference surface, while implicit normal consistency is employed as a legality criterion to suppress excessive local surface-direction distortion after collapse. To reduce the influence of scale differences between different error terms across models, a normalized composite cost is further adopted so that the conventional QEM term and the implicit deviation term can be evaluated in a unified framework.

The experimental results show that the continuous implicit reference field constructed by TetWeave maintains high consistency with the original mesh in both overall geometric position and average normal direction, providing a reliable basis for implicit deviation and implicit normal queries during simplification. On this basis, the normalized implicit geometric constraint has a clear influence on candidate edge ranking and can improve the preservation of the original continuous surface under appropriate weight settings. After further introducing implicit normal consistency, the method shows stronger control over local surface orientation under high-ratio simplification, which helps reduce local geometric distortion and feature drift. At the same time, the experiments also indicate that implicit field loading, query-module construction, and candidate-point implicit queries introduce additional computational overhead, suggesting that continuous implicit geometric supervision improves simplification quality at the cost of higher runtime.

In summary, this work incorporates continuous implicit geometric supervision into both candidate edge ranking and collapse legality checking while preserving the basic structure of the classical QEM edge-collapse framework. It provides a feasible way to improve geometric preservation under high-ratio mesh simplification. Future work will focus on accelerating the implicit query module, designing adaptive parameter selection strategies, and conducting more systematic experiments on implicit normal consistency, so as to further improve the stability and practicality of the proposed method on more complex models and larger-scale datasets.

References

[1] Garland M, Heckbert P S. Surface simplification using quadric error metrics[C]. Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, 1997: 209-216.

[2] Zhang X J, Yu Y, Liu J F. Model simplification method based on visual element preservation[J]. Mechatronics, 2017, 23(1): 44–47.

[3] Chu S R, Niu Z X, Song C H, Niu B N. Progressive mesh simplification algorithm for mobile terminals[J]. Journal of Computer Applications, 2020, 40(3): 806–811.

[4] Jiang Z, Li M, Sun C, Liu J C, Wang P N. Research on model simplification algorithm considering absolute curvature and area-angle constraints[J]. Science of Surveying and Mapping, 2024, 49(7): 76–88.

[5] Zhou G, Yuan S, Luo S. Mesh simplification algorithm based on the quadratic error metric and triangle collapse[J]. IEEE Access, 2020, 8: 196341-196350.

[6] Liang Y, He F, Zeng X. 3D mesh simplification with feature preservation based on whale optimization algorithm and differential evolution[J]. Integrated Computer-Aided Engineering, 2020, 27(4): 417-435.

[7] Ungvichian V, Kanongchaiyos P. Mesh simplification method using principal curvatures and directions[J]. Computer Modeling in Engineering & Sciences (CMES), 2011, 77(3-4): 201-219.

[8] Binninger A, Wiersma R, Herholz P, et al. TetWeave: Isosurface Extraction using On-The-Fly Delaunay Tetrahedral Grids for Gradient-Based Mesh Optimization[J]. ACM Transactions on Graphics (TOG), 2025, 44(4): 1-19.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Copyrights