Adaptive spline surface fitting with arbitrary topological control mesh

Yi-Bo Kou
School of Mathematical
Sciences, UCAS
kouyibo21@mails.ucas.ac.cn

Yi-Fei Feng
School of Mathematical
Sciences, UCAS
fengyifei15@mails.ucas.ac.cn

Li-Yong Shen
School of Mathematical
Sciences, UCAS
lyshen@ucas.ac.cn

Xin Li
School of Mathematical
Sciences, USTC
lixustc@ustc.edu.cn

Chun-Ming Yuan
KLMM, AMSS
Chinese of Academic Sciences
cmyuan@mmrc.iss.ac.cn

Abstract

Reconstructing a spline surface from a given arbitrary topological triangle mesh is a fundamental and challenging problem in computer-aided design and engineering. This paper introduces a novel surface fitting method utilizing G-NURBS capable of handling control meshes with arbitrary topologies. This method employs adaptive control point adjustment, guided by the geometric attributes of the input model, ensuring precise representation of sharp features such as edges and corners. Two primary strategies are employed: a parameter correspondence approach designed for sharp features and a control mesh iterative refinement technique that incorporates geometrical feature information. The proposed method has been tested and evaluated on various CAD models to demonstrate its effectiveness. This method can achieve higher fitting accuracy while faithfully preserving the geometrical features with fewer control points.

Keywords: Surface fitting, feature preservation, adaptive refinement, G-NURBS, arbitrary topology.

1. Introduction

Spline surfaces, particularly non-uniform rational B-spline (NURBS) surfaces, offer a precise parametric representation for 3D freeform surfaces. They possess notable advantages, including support for local modifications and numerical stability. Spline surfaces have become a significant technique extensively utilized in the realms of computer-aided design and manufacturing (CAD/CAM), animation and computer graphics. With the advancement of spline modeling technology, fitting spline surfaces to 3D objects, which are represented in the form of point clouds or triangle meshes, has become a fundamental problem in the domains of CAD and reverse engineering [18].

Surface fitting has been a subject of extensive and intensive research, leading to various approaches [22, 25, 16, 6, 3]. The primary challenge involves striking a balance between achieving high fitting precision while minimizing the number of control points through adaptive fitting [24]. Moreover, accurately fitting geometrical features, especially sharp features, within the input data remains a critical issue in surface reconstruction. Sharp features are characterized by C^0 continuity in the spline surface, encompassing corners, edges, or creases. Existing methods typically focus on either adaptive or feature-preserving spline surface fitting, but often neglecting simultaneous accomplishment of both objectives. Furthermore, there is a lack of research focusing on spline surface fitting with sharp features.

To accomplish these twin objectives, our main idea is to introduce new control points in regions exhibiting subpar fitting quality, along with additional processing of geometrical features. However, for NURBS representation, local refinement of the control mesh becomes impractical due to the tensor product structure. Moreover, the rectangular topology of NURBS control meshes can only confine global isoparametric sharp features. In contrast, T-spline [14, 15] permits the inclusion of T-junctions in the control mesh, enabling local mesh refinement and the creation of local sharp features. By using repeating knots, sharp features can be preserved on T-spline surfaces [21, 20]. However, these methods cannot handle the complex sharp features passing through EPs. In addition, T-juctions also makes the topological structure of T-spline much more complex than that for pure quadrilateral mesh, thereby partially diminishing the acceptability and applicability range of T-splines. Furthermore, another challenge in fitting with NURBS and even T-splines is dealing with extraordinary points (EPs), which are interior vertices in quadrilateral meshes shared by other than four faces [13]. A method for building AST-splines with multiple EPs per face is proposed in [23], which is mainly focusing on the applications in isogeometric analysis, such as convergence rates and linear independence of the blending functions.

A recent study has introduced a method for generating smooth spline surfaces that are compatible with NURBS even in the presence of EPs within the control mesh [2]. In this paper, we adopt the term G-NURBS as a convenient reference to this method, which stands for general NURBS and accurately reflects its broader applicability. The G-NURBS enables local refinement and supports the creation of sharp features along specified edges of the control mesh, including those crossing EPs. Moreover, when dealing with complex surface fitting using NURBS, segmentation and splicing are often required, which can compromise the overall continuity of the surface. In contrast, the G-NURBS offers a solution to this problem. These attributes collectively establish G-NURBS as an exceptional choice for fulfilling our objectives.

Due to the compatibility of G-NURBS with NURBS form, the effective active B-spline method [11, 19] can be modified to optimize the control points of G-NURBS, which is referred to as the active G-NURBS method in this paper. However, the active G-NURBS method dose not impose additional constraints on geometrical features. Therefore, even though G-NURBS can construct sharp features on surfaces, the final fitting results of active G-NURBS still cannot achieve precise recovery of sharp features, as shown in Fig. 2b.

Within this paper, we present a framework denoted as adaptive G-NURBS, which is designed for adaptive surface fitting using G-NURBS. The input is a triangular mesh of arbitrary topology with a pre-given error threshold. Based on the satisfaction of error requirements, this framework effectively achieves a reduction in the number of control points while simultaneously preserving geometrical features. Fig. 1 illustrates the workflow of the proposed adaptive G-NURBS method. This method primarily focus on fitting closed triangular meshes, which refer to meshes that are watertight. In fact, it can be extended to fitting open surfaces by specifying the boundary conditions.

Before initiating the fitting process, additional preprocessing is necessary for handling sharp features. First, the sharp features of input data are detected and identified. Subsequently, a parameter correspondence strategy is employed to establish a connection between the sharp features of the input data and those on the G-NURBS surface. This strategy addresses the limitations of the active G-NURBS method in accurately fitting sharp geometrical features. Specifically, it effectively prevents unexpected misalignment at sharp features induced by EPs, resulting in a more precise reconstruction.

After establishing the parameter correspondence for sharp features, we proceed with the subsequent steps of our adaptive surface fitting framework. Our method begins with an initial sparse quadrilateral mesh for constructing and fitting the spline surface. Typically, the preliminary fitting surface falls short of meeting accuracy requirements and fails to faithfully capture geometrical features. To address this, we employ an error assessment approach that combines geometrical feature information and implement a local quadrilateral mesh refinement algorithm. Through iterative cycles of fitting, error assessment, and control mesh refinement, a control mesh that satisfies error requirements without redundancy can be obtained.

The experiments demonstrate the effectiveness of our approach, as it is capable of generating satisfactory results adaptively even with initially lower-quality control meshes through the aforementioned iterative refinement strategy.

The main contributions of this paper are as follows:

- Introduce an adaptive surface fitting method with G-NURBS, which reduces fitting errors using fewer control points while preserving geometrical features of the input data.
- Propose a parameter correspondence strategy for precise matching of sharp features between G-NURBS surfaces and input models, improving accuracy in these regions.
- Develop an adaptive control mesh iterative refinement approach that combines geometrical features and fitting errors to guide local refinement.

The rest of this paper is arranged as follows. Section 2 introduces G-NURBS and the active G-NURBS method; Section 3 describes the parameter correspondence strategy for sharp features; Section 4 illustrates the adaptive control mesh refinement strategy. Section 5 presents the experimental results to demonstrate the effectiveness of the proposed method. Finally, the conclusion is given in Section 6.

2. Preliminaries

Before formally introducing the adaptive G-NURBS method, this section provides an introduction to G-NURBS and the active G-NURBS method.

2.1. Explanation of G-NURBS

G-NURBS is a recently proposed technique that extends the non-uniform rational B-spline (NURBS) to arbitrary topological control mesh [2]. It can generate smooth spline surfaces on control meshes containing EPs. The emphasis lies in defining basis functions near the EPs, and the main construction process is as follows. The patches adjacent to an EP is called irregular patches, and the edges connected to an EP is called spoke edges.

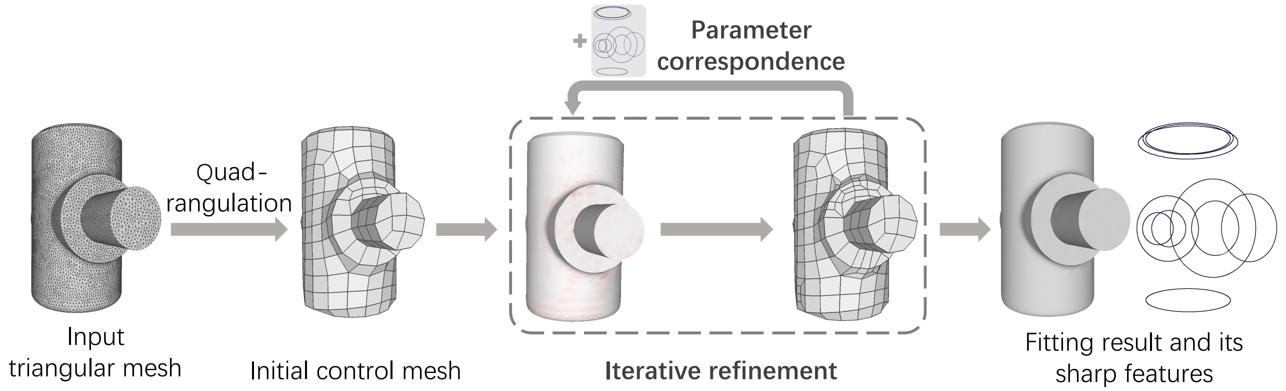
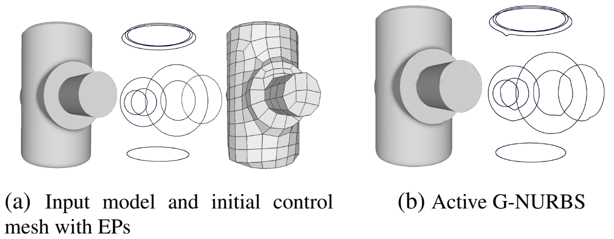


Figure 1: Overview of the proposed adaptive G-NURBS method. Given a triangular mesh with arbitrary topology as input, this method starts from a sparse control mesh and adaptively adjusts it by iterative refinement to meet the error requirements. Additionally, a parameter correspondence strategy is employed to enhance the fitting effect of sharp features. The output is a fitting surface that satisfies the error requirement, with sharp features being well-preserved.


(a) Input model and initial control mesh with EPs

(b) Active G-NURBS

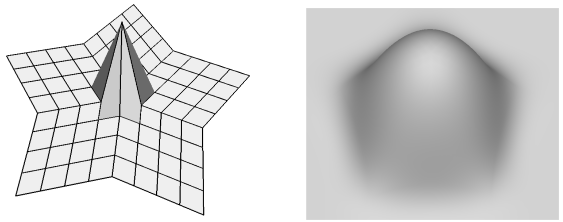
Figure 2: Fitting result of the active G-NURBS method. The sharp features on fitting surfaces are individually presented. Compared with the adaptive G-NURBS method in Fig. 1, the active G-NURBS method can not adaptively adjust the control mesh and exhibits inferior fitting performance at sharp features.

Bézier extraction. G-NURBS generates a bi-cubic Bézier patch for each control mesh face by employing and generalizing Bézier extraction techniques [1, 12]. For irregular patches, the Bézier extraction algorithm needs to be adjusted to avoid ambiguity in knot intervals. An n-degree EP is replaced by n anchors, which become the face points of irregular patches. The face points, edge points, and vertex points are computed as the linear combinations of the original control points and anchors. These points collectively form the control points of a bi-cubic Bézier patches.

Define connecting functions. The surface composed of these Bézier patches are C^0 continuous along the spoke edges and C^1 or C^2 continuous elsewhere. In order to ensure G^1 continuity between two Bézier patches, it is necessary to define the connecting functions beforehand. Based on the knot interval around EPs, the connecting functions on spoke edges are defined as functions of the angles between spoke edges.

Solve the basis functions. The irregular patches are elevated to bi-quintic Bézier patches to ensure sufficient degrees of freedom for G^1 continuity. A local optimization algorithm is employed to solve the G^1 continuous basis functions, thereby avoiding global optimization. Finally, the positions of those redundant anchors are obtained by minimizing the thin plate energy of irregular patches to make the surface smoother.

Fig. 3 shows the G-NURBS surface generated by a control mesh with an EP in the center. Compared to NURBS, this property allows G-NURBS to enable local refinement on the control mesh, as long as the refined control mesh remains a pure quadrilateral mesh.


(a) Control mesh with an EP

(b) G-NURBS surface [2]

Figure 3: The G-NURBS surface generated from a control mesh with an EP.

Sharp features are of great importance in surface modeling techniques, particularly in geometric modeling for CAD and CAM applications. By releasing the G^1 continuity constraints at the designated sharp edges, G-NURBS can create creases on the surface. G-NURBS allows sharp features to pass through EPs, a capability that was previously lacking in other methods. Fig. 4 illustrates various sharp features passing through an EP on G-NURBS surfaces. This enables G-NURBS to conveniently fit input models containing sharp features, regardless of whether the sharp features cross EPs.

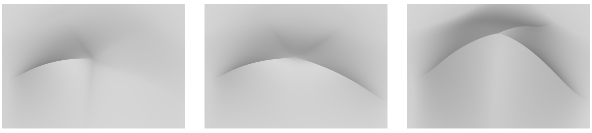

(a) Single crease (b) Double creases (c) Multiple creases

Figure 4: Sharp features passing through the EP on G-NURBS surfaces.

Due to the fact that B-spline basis can be converted into Bernstein basis, the G-NURBS surfaces are compatible with NURBS. Therefore, G-NURBS exhibit a similar expression form to NURBS as follows:

$$\boldsymbol{S}(u,v)=\sum_{k=1}^{M}\boldsymbol{c}_k G_k(u,v) \tag{1}$$

where M is the number of control points, $\boldsymbol{c}_k = (x_k, y_k, z_k) \in \mathbb{R}^3$ is the control point and G_k is the corresponding G-NURBS basis function. Let

$$B=\begin{bmatrix}G_1 & G_2 & \cdots & G_M\end{bmatrix}^{\mathrm{T}},$$

$$L=\begin{bmatrix}B & 0 & 0\\ 0 & B & 0\\ 0 & 0 & B\end{bmatrix},$$

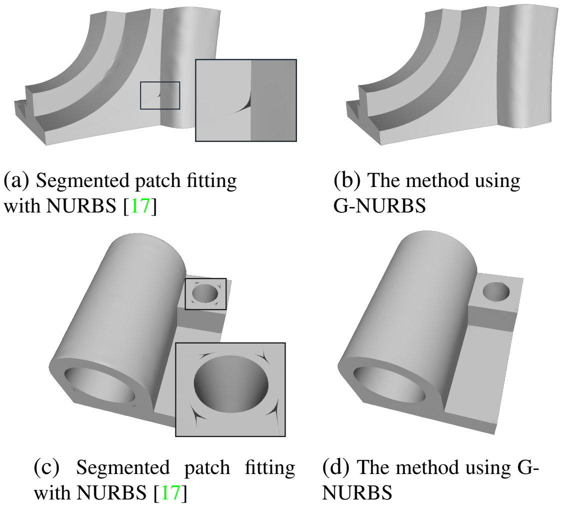
$$C=\begin{bmatrix}x_1,x_2,\cdots,x_M,y_1,y_2,\cdots,y_M,z_1,z_2,\cdots,z_M\end{bmatrix}^{\mathrm{T}}.$$

Then Eq. (1) can be expressed in the following form.

$$\boldsymbol{S}(u,v)=L^{\mathrm{T}}C. \tag{2}$$

As a result, the classical least squares based B-spline surface fitting methods can be migrated to G-NURBS, which leads to the minimization of a quadratic function in the control points.

Although the practicality of NURBS has been widely demonstrated, there are still some challenging issues when using it for surface fitting. When the input data has complex geometric shapes and topological structures, it is necessary to perform pre-segmentation before using NURBS for surface fitting to avoid redundant control points and recover sharp features. Even so, it is still not guaranteed to achieve a watertight fitting result for closed shapes, because the stitching of surface patches is not a trivial issue. The method described in [17] involves segmenting the initial control mesh into multiple patches without EPs, which can then be used as the control mesh for NURBS surfaces. This method of using NURBS surfaces for patch fitting requires a low number of EPs in the control mesh, as a high number of EPs result in an excessive number of patches. Even with additional constraints imposed on the control mesh, the watertightness of the spliced surfaces cannot be guaranteed. However, G-NURBS can ensure watertightness completely as it can generate continuous surfaces on control mesh with arbitrary topology. Fig. 5 displays a comparison of the fitting results between NURBS and G-NURBS. Due to the segmentation fitting involved, the NURBS method cannot guarantee continuity at smooth regions and overall watertightness.


(a) Segmented patch fitting with NURBS [17]

(b) The method using G-NURBS

(c) Segmented patch fitting with NURBS [17]

(d) The method using G-NURBS

Figure 5: Fitting results of NURBS and G-NURBS.

The support for local refinement of control mesh, generation of sharp features, compatibility with NURBS and the ability to ensure watertightness of the fitting result are the advantages that drive our choice to utilize G-NURBS for spline surface fitting.

2.2. Active G-NURBS Method

By simply replacing the basis of a spline surface, one can utilize the active B-spline fitting method [11, 19] for G-NURBS, referred to as the active G-NURBS method in this paper. Given the input triangular mesh, all vertices and sampling points on all faces form the set $P=\{\boldsymbol{p_i}\in\mathbb{R}^3\}_{i=1}^n$. We use $\tilde{\boldsymbol{S}}(u,v)$ and $\{\tilde{\boldsymbol{c}}_k = (\tilde{x}_k,\tilde{y}_k,\tilde{z}_k)\}_{k=1}^M$ to respectively represent the new surface and new control points obtained after fitting. The initial control mesh can be generated by quadrangulation algorithms [5, 4, 10]. The active G-NURBS is an iterative optimization process. In each iteration step, the objective is to minimize the sum of squared errors, as shown in Eq. (3), which is quadratic with respect to control points.

$$\min_{\{\tilde{\boldsymbol{c}}_k\}_{k=1}^M}\sum_{i=1}^{n}d^2(\tilde{\boldsymbol{S}},\boldsymbol{p}_i)+\lambda E_s. \tag{3}$$

Here the $d^2(\tilde{\boldsymbol{S}},\boldsymbol{p}_i)$ is estimated by the curvature-based squared distance [19]:

$$\begin{aligned} d^2(\tilde{\boldsymbol{S}},\boldsymbol{p}_i) =& \frac{d_i}{d_i+\rho_{i,1}}[(\boldsymbol{p}_i-\tilde{\boldsymbol{S}}(u_i,v_i))^{\mathrm{T}}\boldsymbol{T}_{i,1}]^2 \\ &+ \frac{d_i}{d_i+\rho_{i,2}}[(\boldsymbol{p}_i-\tilde{\boldsymbol{S}}(u_i,v_i))^{\mathrm{T}}\boldsymbol{T}_{i,2}]^2 \\ &+ [(\boldsymbol{p}_i-\tilde{\boldsymbol{S}}(u_i,v_i))^{\mathrm{T}}\boldsymbol{N}_i]^2 \end{aligned} \tag{4}$$

where $d_i = \|\boldsymbol{p}_i - \boldsymbol{S}(u_i,v_i)\|$ is the distance from the data point $\boldsymbol{p}_i$ to its foot point $\boldsymbol{S}(u_i,v_i)$ on current G-NURBS surface, $\rho_{i,1} = 1/\kappa_{i,1}$ and $\rho_{i,2} = 1/\kappa_{i,2}$ are the inverse curvatures of two principal curvatures, $\boldsymbol{T}_{i,1}$ and $\boldsymbol{T}_{i,2}$ are the unit tangent vectors along two principal directions at $\boldsymbol{S}(u_i,v_i)$, $\boldsymbol{N}_i$ is the unit normal vector at $\boldsymbol{S}(u_i,v_i)$.

E_s in Eq. (3) is the smoothing term, and we adopt the simplified thin plate energy

$$E_s = \iint (\|\tilde{\boldsymbol{S}}_{uu}\|^2 + 2\|\tilde{\boldsymbol{S}}_{uv}\|^2 + \|\tilde{\boldsymbol{S}}_{vv}\|^2)\mathrm{d}u\mathrm{d}v. \tag{5}$$

Let

$$\tilde{C} = \left[\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_M, \tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_M, \tilde{z}_1, \tilde{z}_2, \cdots, \tilde{z}_M\right]^{\mathrm{T}}$$

and$\Delta C = \tilde{C} - C$, then Eq. (3) can be rewritten as a quadratic form of ΔC:

$$\min F(\Delta C) = \frac{1}{2}\Delta C^{\mathrm{T}} A \Delta C - b^{\mathrm{T}}\Delta C \tag{6}$$

where

$$\begin{aligned} A =& \sum_{i=1}^{n} \frac{d_i}{d_i+\rho_{i,1}} L\boldsymbol{T}_{i,1}\boldsymbol{T}_{i,1}^{\mathrm{T}}L^{\mathrm{T}} \\ &+ \sum_{i=1}^{n} \frac{d_i}{d_i+\rho_{i,2}} L\boldsymbol{T}_{i,2}\boldsymbol{T}_{i,2}^{\mathrm{T}}L^{\mathrm{T}} \\ &+ \sum_{i=1}^{n} L\boldsymbol{N}_i\boldsymbol{N}_{\boldsymbol{i}}^{\mathrm{T}}L^{\mathrm{T}} + \lambda \iint (L''L''^{\mathrm{T}})\mathrm{d}u\mathrm{d}v \end{aligned}$$

and

$$\begin{aligned} b =& \sum_{i=1}^{n} \frac{d_i}{d_i+\rho_{i,1}} L\boldsymbol{T}_{i,1}\boldsymbol{T}_{i,1}^{\mathrm{T}}(\boldsymbol{p}_i - \boldsymbol{S}(u_i,v_i)) \\ &+ \sum_{i=1}^{n} \frac{d_i}{d_i+\rho_{i,2}} L\boldsymbol{T}_{i,2}\boldsymbol{T}_{i,2}^{\mathrm{T}}(\boldsymbol{p}_i - \boldsymbol{S}(u_i,v_i)) \\ &+ \sum_{i=1}^{n} L\boldsymbol{N}_i\boldsymbol{N}_{\boldsymbol{i}}^{\mathrm{T}}(\boldsymbol{p}_i - \boldsymbol{S}(u_i,v_i)) \\ &- \lambda \iint (L''L''^{\mathrm{T}})\mathrm{d}u\mathrm{d}v. \end{aligned}$$

Then the quadratic optimization problem of Eq. (3) is equivalent to solving a linear equation system

$$A\Delta C = b. \tag{7}$$

Algorithm 1: Active G-NURBS

```
Input: $\{\boldsymbol{p}_i\}_{i=1}^n$: sample points on input triangular
       mesh,
       $\epsilon$: pre-given error threshold,
       $I$: upper limit of the iteration count,
       $\mathcal{M}$: initial control mesh.
Output: $\boldsymbol{S}(u,v)$: the final G-NURBS fitting
        surface.
Current fitting surface $\boldsymbol{S}(u,v) \leftarrow$ Eq. (3)
$d_{avg} = \frac{1}{n}\sum_{i=1}^{n} \epsilon_i$
$i = 0$
while $d_{avg} > \epsilon$ and $i < I$ do
    $\boldsymbol{S}(u,v) \leftarrow$ Eq. (3)
    $d_{avg} = \frac{1}{n}\sum_{i=1}^{n} \epsilon_i$
    $i = i + 1$
end
Result: $\boldsymbol{S}(u,v)$
```

The workflow of the active G-NURBS method is presented in Algorithm 1.

Despite the widely confirmed effectiveness of this method, it is still insufficient to address the fitting issues in G-NURBS. This method optimizes the positions of control points through an iterative process, but can not adaptively insert new control points. When utilizing the active G-NURBS method, the entire surface area is uniformly fitted. However, even when the fitting error is controlled below a specified threshold, the reconstruction results in regions with sharp features may still be unsatisfactory due to the absence of additional processing. Through conducting the fitting experiments using the active G-NURBS method straightforward, when the EPs are located on sharp features, such as sharp edges and corners, the fitting performance in these regions is often unsatisfactory, as can be seen in Fig. 2. At this time, although the fitting error of the surface in these areas is sufficiently small, it fails to truly correspond the creases of the surface with the sharp features of the input model. To this end, we propose a parameter correspondence strategy by adjusting the parametrization of points on the sharp features of the input model to correspond them to the sharp features on G-NURBS surface.

Furthermore, to achieve adaptive fitting, we have extended the active G-NURBS method by an iterative refinement strategy of the control mesh, which combines geometrical feature information with fitting error.

3. Parameter Correspondence Strategy for Sharp Features

In this section, a detailed presentation of the parameter correspondence strategy for fitting sharp features is provided. By employing this strategy, the sharp features of the G-

NURBS surface can be correspondingly aligned with the sharp features of the input model, resulting in a more authentic reconstruction of the object's shape.

For smooth geometrical features, the utilization of local refinement can reduce fitting errors. However, when it comes to sharp features, the situation changes. The least squares-based fitting framework treats all sampled points equally and does not exhibit specificity when seeking the foot point for estimating squared distance. As a result, after multiple iterations of fitting, the sampled points on the sharp features of the input model tend to correspond to areas on the G-NURBS surface outside of the sharp creases. This situation becomes particularly evident when the sharp features on the G-NURBS surface pass through EPs. Even if the weight of the sampled points on the sharp features is increased, the improvement in the situation is minimal. The final fitting result presents a phenomenon where the fitting error at the sharp features is small, but in reality, there exists significant disparity between the sharp features of the surface and the input triangular mesh, and they do not establish a correspondence between each other. To address this issue, we apply a two-step parameter correspondence strategy to the sharp features. The first step is to identify the sharp features on the input triangular mesh model and locate the corresponding sharp creases on the G-NURBS surface. The second step involves modifying the active G-NURBS method by specifying the parameters of the sample points on sharp features.

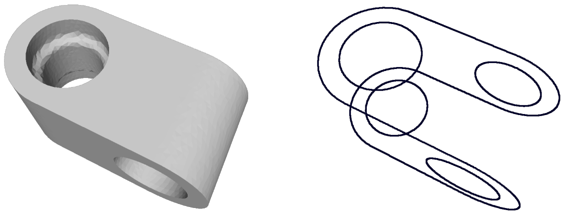
3.1. Identify Sharp Features

There are various methods for feature recognition on triangular meshes. A straightforward approach is to compute the angles of the mesh faces or the discrete curvatures of vertices to identify sharp features by setting threshold. The drawback of this method is poor adaptability. We choose to adopt a more stable approach, known as the tensor voting method [7], to identify sharp features on triangular meshes, as illustrated in Fig. 6. After identifying the sharp features, we fit G-NURBS surface to the triangular mesh model. We mark the corresponding edges of the control mesh with a sharp tag by identifying the boundaries of the Bézier patches on the G-NURBS surface that correspond to the sharp features. In this way, we identify the sharp features of the input model as well as their corresponding sharp features on the G-NURBS surface. Subsequently, the objective is not only to minimize the error between the G-NURBS surface and the input model, but also to reduce the error between their sharp features.

3.2. Sharp Feature Parametrization

In this step, we utilize the determined correspondence of sharp features from the previous step to locate the parametrization of the sampled points on the sharp features. To ensure a consistent correspondence between the sharp features of the surface and the input model, as well as to gradually minimize the error between them during the fitting process, we specify that the foot points of points on sharp features of input model must lie on the sharp feature curves of the surface. As a result, the parametrization of points on the sharp features is specified, and their parameters correspond to the boundaries of several Bézier patches on G-NURBS surface. Correspondingly, the quadratic optimization problem in Eq. (3) is modified as follows.

$$\min_{\tilde{C}} \sum_{i=1}^{n} d^2(\tilde{\boldsymbol{S}}, \boldsymbol{x}_i) + \lambda_1 \sum_{\substack{\boldsymbol{p}_i \in P_{sharp}\\ \boldsymbol{F}_i \in \tilde{\boldsymbol{S}}_{sharp}}} d^2(\boldsymbol{p}_i, \boldsymbol{F}_i) + \lambda_2 E_s. \quad (8)$$



(a) Input model

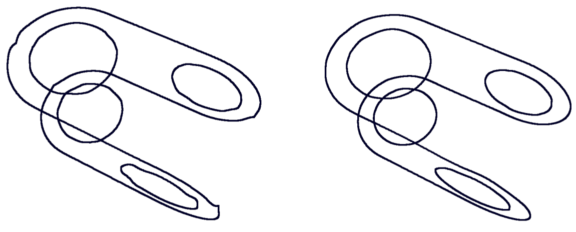
(b) Sharp features

Figure 6: Input model and the sharp features on it.

The new objective function incorporates an additional second term compared to the active G-NURBS method in Eq. (3). In the new term, P_{sharp} is the set of sample points on sharp features of input models, $\boldsymbol{F}_i$ is the foot point of those $\boldsymbol{p}_i \in P_{sharp}$ and $\tilde{\boldsymbol{S}}_{sharp}$ is the sharp features on G-NURBS surface. The second term indicates that, for $\boldsymbol{p}_i \in P_{sharp}$, its foot point $\boldsymbol{F}_i$ is searched within the G-NURBS surface feature $\tilde{\boldsymbol{S}}_{sharp}$, rather than the entire surface. Imposing the constraint that $\boldsymbol{F}_i \in \tilde{\boldsymbol{S}}_{sharp}$ is equivalent to specifying the parametrization of sample points on sharp features. By adjusting the magnitude of coefficient λ_1, it is possible to achieve different levels of fitting to sharp features.

Under the application of this parameter-corresponding strategy, G-NURBS surfaces exhibit more accurate recovery of sharp features, as shown in Fig. 7. Compared with the active G-NURBS method, our strategy can make the sharp features on G-NURBS surfaces approximate the sharp features of the input model, rather than using other smooth regions of the surface to approximate them. The latter approach can lead to actual sharp creases on the fitting surface appearing distorted and deformed, which is inconsistent with the input model.

Once the parameter correspondence of sharp features is completed, we can proceed with the formal implementation of the proposed adaptive fitting framework in this paper.



(a) Active G-NURBS method

(b) Our strategy

Figure 7: Comparison of fitting results at sharp features. (a) is the active G-NURBS method without our strategy and fail to accurately recover sharp features at EPs. (b) adopts our parameter correspondence strategy for sharp features.

4. Adaptive Refinement of the Control Mesh

In this section, we provide a detailed explanation of the control mesh adaptive refinement strategy. Our method is essentially an iterative process that involves an error assessment procedure and a quad mesh refinement algorithm, in which geometrical feature information is incorporated.

4.1. Error Assessment

4.1.1 Metric of geometrical features

Geometrical features are important for shape representation and they reflect certain properties of the model. Some methods represent geometrical features by constructing geometric descriptors. However, these methods commonly suffer from limited discriminative ability and susceptibility to scale interference. We hope that the metric of geometrical features can reflect the degree of shape variation within a certain range on the surface. The variation of normal vectors can reflect the change in surface shape, but it cannot be explicitly expressed very well. Curvature is the amount by which a surface deviates from being a plane and reflects the degree of change of normal vectors on a surface. The larger the curvature is, the more pronounced the shape variation becomes at a specific point on the surface. Therefore the regions with higher curvature are considered as geometrical feature regions and require more attention during fitting.

The reason why Gaussian curvature and average curvature, which are often used in geometric analysis, are not employed is that even when the surface shape changes in certain areas, these two measures may still be zero. Gaussian curvature is the product of the two principal curvatures, and it equals zero when the surface is developable. Mean curvature is the average of the two principal curvatures and is zero when the surface is minimal. To accurately utilize discrete curvature as a geometrical feature metric for triangular mesh models, we choose to employ absolute curvature (ABS curvature). As shown in Eq. (9), the discrete ABS curvature is defined as the sum of absolute values of two discrete principle curvatures, which is calculated by the method proposed in [9].

$$\kappa = |\kappa_1| + |\kappa_2|. \tag{9}$$

Therefore, unlike Gaussian curvature and mean curvature, when shape variations occur on the surface of a triangular mesh model, the absolute curvature of the corresponding region will, in most cases, not become zero, and thus it effectively reflects the geometrical features of input models.

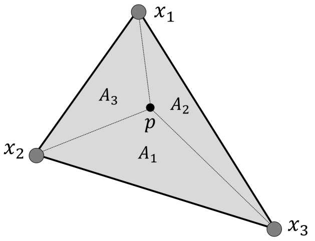


Figure 8: Calculation of discrete ABS curvature at interior point $\boldsymbol{p}$ by barycentric interpolation method.

For the vertices of a triangular mesh, we can compute their two discrete principal curvatures directly and sum them together to obtain the ABS curvature. However, for the sampling points inside the faces of a triangular mesh, there is no direct method to compute the ABS curvature. Therefore, we employ the barycentric interpolation method as shown in Eq. (10), which takes the linear combination of the discrete curvatures of the three vertices of each triangle face as the discrete ABS curvature of interior points of mesh faces.

$$\kappa_{\boldsymbol{p}} = \frac{A_1\kappa_1 + A_2\kappa_2 + A_3\kappa_3}{A_1 + A_2 + A_3}, \tag{10}$$

where κ_1, κ_2, κ_3 are the discrete ABS curvature of three vertices of triangle face, and A_1, A_2, A_3 are the areas of the three small triangles formed by interior point $\boldsymbol{p}$ and the three edges of the triangular mesh face as illustrated in Fig. 8.

4.1.2 Combination of geometrical features and errors

For regions with geometrical features, it is not sufficient to rely solely on an error threshold to assess the quality of surface fitting. Conventional fitting methods typically apply the same error threshold globally to constrain the input data. It's common to observe that, even when the final fitting result meets the error requirement, regions with more geometrical features often exhibit larger errors compared to flatter regions. In such cases, geometrical feature regions require a greater number of control points to enhance the fitting performance. However, using the error threshold as the evaluation criterion clearly fails to identify these regions

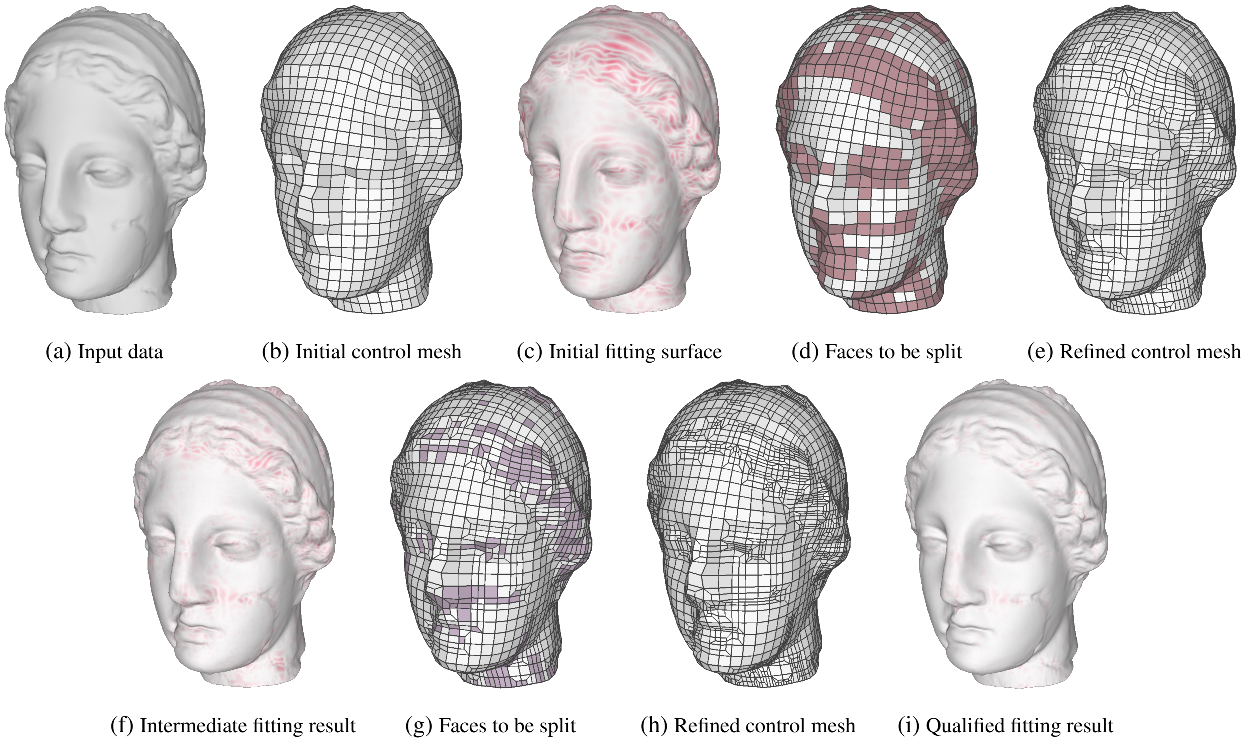

(a) Input data
(b) Initial control mesh
(c) Initial fitting surface
(d) Faces to be split
(e) Refined control mesh
(f) Intermediate fitting result
(g) Faces to be split
(h) Refined control mesh
(i) Qualified fitting result

Figure 9: The iterative refinement process. (a) is the input triangle model with non-uniform distribution of geometrical features. (b) is the initial sparse control mesh. (c) is the feature-coupled errors of the first fitting result. (d) shows the faces to be refined (faces with dark color are marked as refinement). (e) is the control mesh after refinement. (f) is the intermediate fitting result of refined control mesh (e). (g)-(i) illustrate the iterations of the aforementioned process until the final fitting result (i) meets the error requirement.

effectively. Therefore, we propose a novel strategy for evaluating the quality of fitting surfaces, which combines geometric features with errors. Compared to methods that only rely on errors as the criteria, our strategy can better identify regions with higher geometrical feature metric that require the insertion of additional control points.

After each fitting process is done, the position of control points is updated and the current fitting surface $\boldsymbol{S}(u, v)$ is generated. We evaluate the fitting quality at every vertices and sampling points inside the triangle faces on the input triangular models. For each point $\boldsymbol{p_i}$, we compute the corresponding foot point $\boldsymbol{S}(u_i, v_i)$ and the real fitting error $\epsilon_i = \|\boldsymbol{p}_i - \boldsymbol{S}(u_i, v_i)\|$. The fitting quality of each points $\boldsymbol{p}_i$ is expressed by a feature-coupled error $k_i \epsilon_i$. The coefficient k_i is a quantity related to the discrete ABS curvature, which is defined as Eq. (11).

$$k_i = \max(\eta, \frac{\alpha(\kappa_i - \kappa_{\min})}{\kappa_{\max} - \kappa_{\min}}), \tag{11}$$

where $\kappa_{\max}$ represents the maximum value of the discrete ABS curvature among all points, $\kappa_{\min}$ corresponds to the minimum value, η is typically set to 1 in order to prevent the coefficient k_i from becoming too small and α is usually set to 2. The coefficient k_i at a point $\boldsymbol{p}_i$ increases as the geometrical feature metric at that point becomes larger.

As mentioned earlier, G-NURBS generates a Bézier patch for each quad face on the control mesh. If the mean fitting error of a patch exceeds the given threshold, we consider that the corresponding regions require additional new control points to improve the fitting capability and the quad face on control mesh will be refined. The above strategy is implemented as Algorithm 2.

The purpose of the coefficient k_i is to combine geometrical features with errors, thereby increasing the difficulty for regions with higher geometrical feature metrics to pass the error assessment.

4.2. Local Refinement of Control Mesh

After the procedure of error assessment, we obtain the set of control mesh faces to be refined. We aim to refine these faces while minimizing the addition of new control points in other regions that meet the error threshold. Additionally, when performing control mesh refinement, it is necessary to ensure that the resulting mesh remains a pure quadrilateral

```
Algorithm 2: Error assessment strategy
  Input: $\mathcal{T}$: input triangle mesh,
         $\{\boldsymbol{p}_i\}_{i=1}^n$: sample points of $\mathcal{T}$,
         $\boldsymbol{S}(u,v)$: current fitting surface,
         $F$: the set of control mesh faces,
         $\epsilon$: pre-given error threshold.
  Output: $F_{split}$: the set of control mesh faces to be refined.
  $F_{split} \leftarrow \emptyset$
  for $i = 1$ to $n$ do
      $\kappa_i \leftarrow$ discrete ABS curvature of $\boldsymbol{p}_i$
      $\boldsymbol{S}(u_i, v_i) \leftarrow$ Foot point of $\boldsymbol{p}_i$
      $\epsilon_i = \|\boldsymbol{S}(u_i, v_i) - \boldsymbol{p}_i\|$
  end
  for $f \in F$ do
      Denote $n_f$ as the number of foot points on the corresponding Bézier patch $\boldsymbol{S}_f$ of $f$, compute the mean fitting error of $\boldsymbol{S}_f$:
        $\epsilon_f = \frac{1}{n_f} \sum_{\boldsymbol{S}(u_i,v_i) \in \boldsymbol{S}_f} \kappa_i \epsilon_i$
      if $\epsilon_f > \epsilon$ then
          $F_{split} \leftarrow F_{split} \cup \{f\}$
      end
  end
  Result: $F_{split}$
```

mesh. Therefore, we adopt and modify the method proposed in [8], which enables a conforming refinement (illustrated in Fig. 10), meaning that each resulting face after refinement remains a quadrilateral.

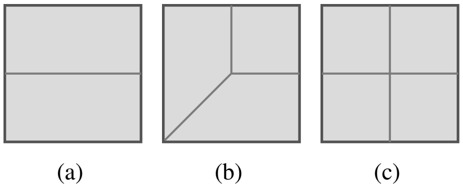


Figure 10: Conforming refinement.

Let $\mathcal{M} = (V, E, F)$ be a control mesh with vertices V, edges E and faces F. F_{split} is the set of faces to be refined. For each edge e_i in the control mesh, a binary variable s_i is defined. When $s_i = 1$, e_i is split, while when $s_i = 0$, e_i is not split. When a face f of the control mesh is marked for refinement, in order to maintain the conforming refinement, its four edges $\{e_i\}_{i=1}^4$ need to satisfy the following constraints:

$$s_1 + s_2 + s_3 + s_4 >= 2 \quad \forall f \in F_{split} \tag{12}$$

Thus, the control mesh refinement problem is transformed into a constrained binary optimization problem:

$$\begin{aligned} \min \sum_{e \in E} s \\ s.t. \quad \text{Eq. (12)} \end{aligned} \tag{13}$$

One of these conforming refinement ways introduces new EPs, called Y-configuration, as shown in Fig. 10b. However, G-NURBS is capable of handling EPs, so we have relaxed the restrictions on Y-configurations when applying this quad refinement algorithm.

4.3. Iterative Refinement Process

Below, we outline the complete process of using the error assessment strategy to drive the adaptive control mesh refinement.

Fist we implement quadrangulation algorithms [5, 4, 10] to convert the input triangular mesh into a sparse quadrilateral mesh. The quadrilateral mesh is treated as the initial control mesh and inevitably contains many EPs but G-NURBS can handle them. The G-NURBS surface $\boldsymbol{S}(u,v)$ generated by the initial control mesh can approximate the overall shape of the original triangle mesh, but cannot fully capture the subtle geometric details. After optimizing the position of control points in every iterative step, it can be considered that the current number of control points has achieved an approximation of optimal fitting effectiveness.

Subsequently, we apply the proposed error assessment strategy to the fitting surface. This strategy identifies regions that exceed the error threshold or contain substantial geometrical features. Once the error assessment is completed, we apply the modified mesh local refinement algorithm to insert new control points into those desired regions, thereby enhancing the corresponding fitting performance. We iteratively perform the aforementioned fitting, error assessment, and local refinement processes until all regions on the surface pass the error assessment. An illustration of this process is provided in Fig. 9, and the algorithm is summarized in Algorithm 3.

5. Experimental Results

In this section, we conduct several experiments to demonstrate the effectiveness of the proposed method. The input triangular mesh models possess arbitrary topological structure, with its surface exhibiting either smooth geometrical features or sharp geometrical features, such as CAD object models. By employing quadrangulation algorithms, the input triangular mesh model can be transformed into a sparse quadrilateral mesh, which serves as the initial control mesh. When solving the quadratic optimization problem for spline surface fitting, we transform it into solving a linear system and employ the bi-conjugate gradient method (BiCG) for its solution.

Algorithm 3: Adaptive G-NURBS

```
Input: $\mathcal{T}$: input triangular mesh,
       $\epsilon$: pre-given error threshold.
Output: $\boldsymbol{S}(u,v)$: the final G-NURBS fitting
        surface meeting the error requirement.
Initial control mesh $\mathcal{M} \leftarrow$ Quadrangulation of $\mathcal{T}$.
$\{\boldsymbol{p}_i\}_{i=1}^{n} \leftarrow$ Vertices and sample points of $\mathcal{T}$.
Parameter correspondence for sharp features (see
  Section 3).
Current fitting surface $\boldsymbol{S}(u,v) \leftarrow$ Eq. (8).
$d_{avg} = \frac{1}{n}\sum_{i=1}^{n} d(\boldsymbol{S}(u,v), \boldsymbol{p}_i)$.
while $d_{avg} > \epsilon$ do
    $F_{split} \leftarrow$ Error assessment in Algorithm 2.
    New control mesh $\tilde{\mathcal{M}} \leftarrow$ Eq. (13).
    Parameter correspondence for sharp features
      (see Section 3).
    $\boldsymbol{S}(u,v) \leftarrow$ Eq. (8).
    $d_{avg} = \frac{1}{n}\sum_{i=1}^{n} d(\boldsymbol{S}(u,v), \boldsymbol{p}_i)$.
end
Result: $\boldsymbol{S}(u,v)$
```

As mentioned in Section 2.1, when fitting closed surfaces with complex geometrical shapes using NURBS, it is required to have fewer EPs in the control mesh. Moreover, the watertightness of the fitting result cannot be guaranteed. Therefore, in subsequent experiments, we no longer consider active B-spline method as a comparative object but instead choose the active G-NURBS method for comparison.

In the tables appearing later in this section, $\#CPs$ denotes the number of control points, d_{max} denotes the maximum fitting error, d_{avg} denotes the mean fitting error, d^f_{max} denotes the maximum error on sharp features and d^f_{avg} denotes the mean error on sharp features. The above errors are all percentage errors, indicating the percentage between the actual error and the diagonal length of the bounding box. *Bbox diag* refers to the diagonal length of the input triangular mesh's bounding box, which is a dimensionless value. It should be noted that d_{max} represents the overall maximum error between the fitting surface and the input data, while d^f_{max} represents the maximum error between their sharp features. Therefore, there is no absolute relationship between d_{max} and d^f_{max}, and the same applies to d_{avg} and d^f_{avg}.

In the experiments, the parameter λ_1 in Eq. (8) is set to 100. In practice, for the examples tested, setting λ_1 within the range of 80 to 120 results in consistently similar outcomes. λ_2 is set to $1/2^i$, where i denotes the number of iterations in solving the optimization problem presented in Eq. (8).

5.1. Parameter Correspondence Strategy for Sharp Features

In order to validate the effectiveness of our parameter correspondence strategy in Section 3, we conducted tests on CAD workpiece models with sharp features. The experimental results indicate that our method is highly effective in accurately fitting sharp features.

The active G-NURBS method does not apply additional processing to sharp features, leading to a poor correspondence between the sharp features on fitting surface and those of the input model. As a result, even if the fitting error of the surface is small, the sharp features are not well recovered, especially at corners and EPs where the sharp features pass through. Fig. 11 shows the fitting results of the nut model. It can be observed that our method enables a better alignment of sharp creases on the G-NURBS surface with the actual sharp features of the model. In contrast, the active G-NURBS method exhibits poorer recovery results for corner points and sharp edges, leading to uncontrollable fitting outcomes near EPs. The numerical comparison of the fitting results of the nut model is shown in Tab. 1.

With the same number of control points, our method achieves smaller maximum and average errors. In this case, the error calculation follows conventional methods without explicitly considering sharp features. Additionally, we specifically calculate the error for sharp features to measure the average errors between the sharp features on the fitting surface and those of the input model. Data comparison reveals a significant enhancement in the effectiveness of our method, resulting in a remarkable 33% reduction in errors at sharp features.

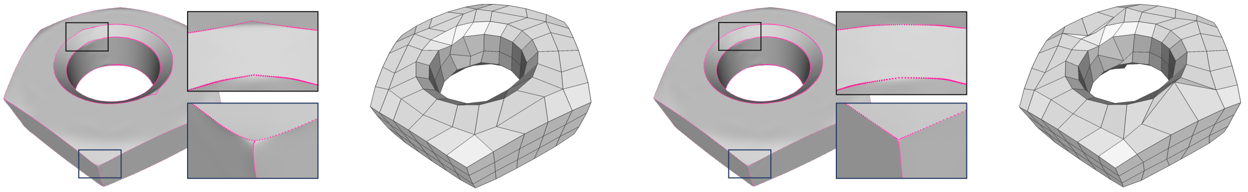
A more detailed presentation of the fitting results for the ujoint model in Fig. 7 is shown in Fig. 12 and Tab. 2. Our method achieves a significant reduction in error on sharp features while maintaining overall fitting accuracy.

Table 1: Statistics for the Fitting Result of Nut Model in Fig. 11.

	Active G-NURBS	Adaptive G-NURBS
$\#CPs$	256	256
$d_{max}(\%)$	0.193	0.156
$d_{avg}(\%)$	0.0159	0.0142
$d^f_{max}(\%)$	1.1	0.26
$d^f_{avg}(\%)$	0.11	0.08
Bbox diag	12.518	

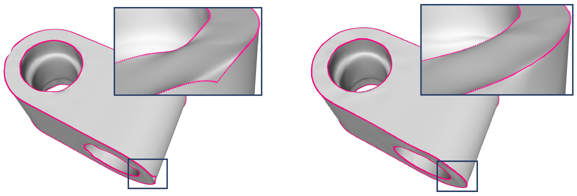
5.2. Adaptive Local Refinement of Control Mesh

To test the control mesh adaptive refinement strategy that combines geometrical features proposed in this study, we



(a) Active G-NURBS (b) Control mesh of (a) (c) Adaptive G-NURBS (d) Control mesh of (c)

Figure 11: Fitting results of nut model. Sharp features are highlighted in magenta.



(a) Active G-NURBS (b) Adaptive G-NURBS

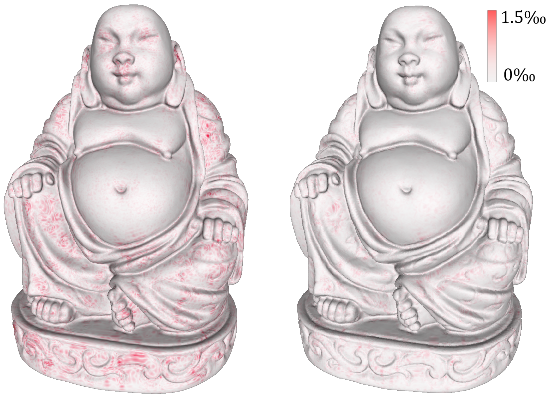
Figure 12: Fitting results of ujoint model. Sharp features are highlighted in magenta.

Table 2: Statistics for the Fitting Result of Ujoint Model in Fig. 12.

	Active G-NURBS	Adaptive G-NURBS
$\#CPs$	269	269
$d_{max}(\%)$	0.58	0.43
$d_{avg}(\%)$	0.045	0.029
$d^f_{max}(\%)$	1.27	0.39
$d^f_{avg}(\%)$	0.155	0.111
Bbox diag	36.742	

utilized G-NURBS to fit models with complex geometrical features. As mentioned in Section 2.2, we employed a sparse quadrilateral mesh as the initial control mesh of our method. Additionally, we also implement the active G-NURBS method for the purpose of comparison with our proposed approach. The active G-NURBS method directly employs a quadrilateral mesh generation algorithm to generate a dense control mesh for G-NURBS fitting, without incorporating any mesh adaptive refinement steps. Hence, the overall density distribution of this dense control mesh is relatively uniform.

The fitting results of the Buddha model is depicted in Fig. 13 and Fig. 14. It can be observed that our method exhibits better fitting performance for complex geometrical features, such as the texture details and so on. Furthermore, due to the implement of an adaptive control mesh local refinement approach, our method requires fewer control points while achieving smaller fitting errors. Tab. 3 shows the numerical comparison of the fitting results of Buddha model. Proposed adaptive refinement strategy achieves a reduction of 7% in the number of control points while simultaneously reducing the maximum and average errors by more than 50%.



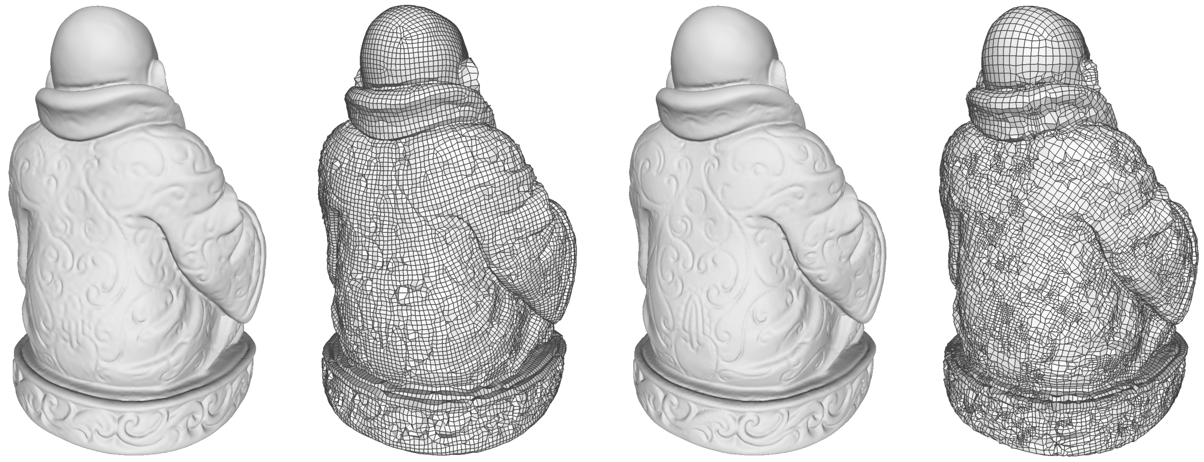
(a) Active G-NURBS (b) Adaptive G-NURBS

Figure 13: Error of Buddha model fitting results.

Table 3: Statistics for the Fitting Result of Buddha Model in Fig. 13 and Fig. 14.

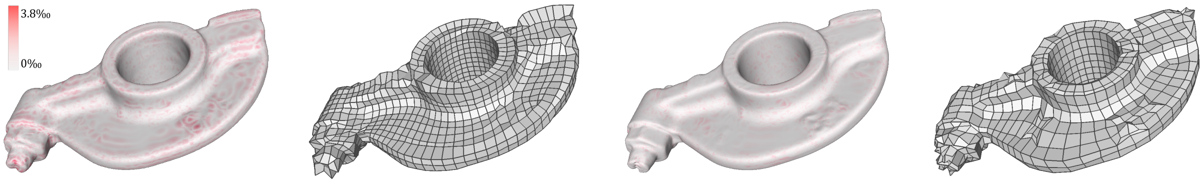
	Active G-NURBS	Adaptive G-NURBS
$\#CPs$	25433	23668
$d_{max}(\%)$	0.158	0.079
$d_{avg}(\%)$	0.0143	0.0060
Bbox diag	1.389	

For another CAD arm model, our approach starts with a sparse control mesh and progressively performs local refinements. Due to the presence of extensive smooth regions on the surface of this model, the number of newly added control vertices is relatively low at the end. The comparison of the fitting results is shown in Fig. 15. Our approach recovers detailed geometrical features more realistically. Tab. 4 shows the numerical comparison of the two methods. Our method can reduce the maximum error and



(a) Active G-NURBS (b) Control mesh of (a) (c) Adaptive G-NURBS (d) Control mesh of (c)

Figure 14: Fitting results and control meshes of Buddha model. Compared to active G-NURBS methods, our approach is capable of allocating more control points for geometrical feature regions, leading to better recovery of subtle geometrical features such as texture.



(a) Active G-NURBS (b) Control mesh of (a) (c) Adaptive G-NURBS (d) Control mesh of (c)

Figure 15: Fitting result and control mesh of arm model.

Table 4: Statistics for the Fitting Result of Arm Model in Fig. 15.

	Active G-NURBS	Adaptive G-NURBS
$\#CPs$	2090	1387
$d_{max}(\%)$	0.38	0.13
$d_{avg}(\%)$	0.026	0.011
Bbox diag	91.373	

average error by 66% and 58%, respectively, with a 33% reduction in the number of control points.

5.3. Simultaneous Utilization of Both Strategies

As previously stated, the parameter correspondence strategy for sharp features can serve as a preprocessing step for the control mesh adaptive refinement strategy, and the two are not conflicting. Therefore, in addition to conducting separate experiments for each strategy, we also present experimental results for the simultaneous utilization of these two strategies.

The experiment on the fandisk model validates the effectiveness of our adaptive fitting method, as shown in Fig. 16. The corresponding statistics comparison is presented in Tab. 5. Our method achieves significant reduction in both global fitting error and sharp features fitting error with fewer control points.

Table 5: Statistics for the Fitting Result of Fandisk Model in Fig. 16.

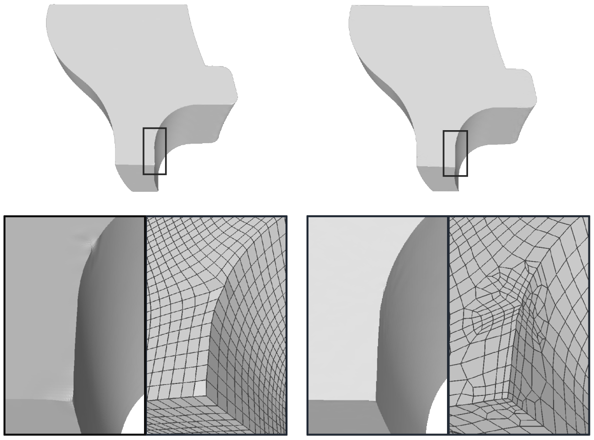
	Active G-NURBS	Adaptive G-NURBS
$\#CPs$	8098	7224
$d_{max}(\%)$	0.29	0.056
$d_{avg}(\%)$	0.032	0.015
$d^f_{max}(\%)$	0.42	0.12
$d^f_{avg}(\%)$	0.055	0.035
Bbox diag	1.454	

The fitting results of the linkage model are shown in Fig. 17. The corresponding data is compared and presented

Table 6: Statistics for the Fitting Result of Linkage Model in Fig. 17.

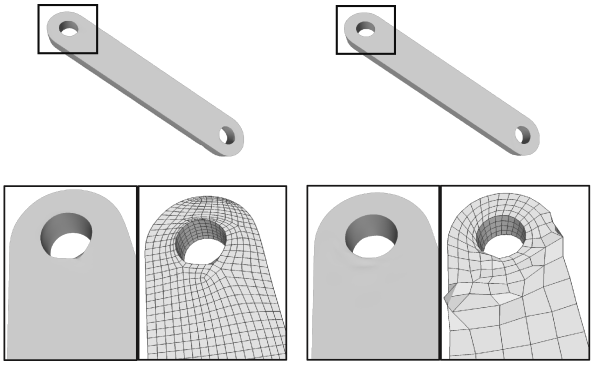
	Active G-NURBS	Adaptive G-NURBS
$\#CPs$	5105	877
$d_{max}(\%)$	0.29	0.098
$d_{avg}(\%)$	0.082	0.017
$d^f_{max}(\%)$	0.63	0.17
$d^f_{avg}(\%)$	0.092	0.05
Bbox diag	45.195	

in Tab. 6, which also verifies the effectiveness of our adaptive fitting method.



(a) Active G-NURBS (b) Adaptive G-NURBS

Figure 16: Fitting results of fandisk model. Above are the final fitting surfaces, and below are the local enlarged views of the fitting results and the corresponding control meshes.



(a) Active G-NURBS (b) Adaptive G-NURBS

Figure 17: Fitting results of linkage model. Above are the final fitting surfaces, and below are the local enlarged views of the fitting results and the corresponding control meshes.

Table 7: Time Breakdown for Experimental Results

	Total	T^1_{fit}	T^1_{refine}	T^2_{fit}	T^2_{refine}	T^3_{fit}
Nut	17.5s	17.5s				
Ujoint	12.8s	12.8s				
Buddha	936s	102s	120s	240s	120s	354s
Arm	95s	22s	26s	47s		
Fandisk	261s	24s	22s	39s	92s	84s
Linkage	42.7s	15.1s	0.5s	27.1s		

5.4. Computation Time

Tab. 7 lists the total time and breakdown of major steps for our method. T^i_{fit} and T^i_{refine} respectively represent the time spent on solving fitting problem Eq. (8) and the mesh refinement problem Eq. (13) during the i th iteration of adaptive fitting. We set an upper limit of 2 minutes for T^i_{refine}, which strikes a balance between seeking the optimal solution and computational efficiency.

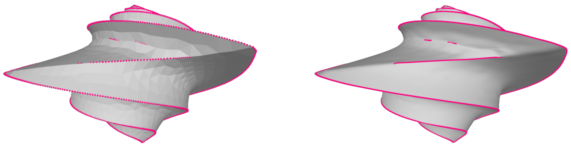
For most of the models, our method exhibits relatively short computation times. However, for models with complex surface variations, such as Buddha, the total time is longer. As the number of iterations for adaptive fitting increases, control mesh becomes more intricate, resulting in increased generation and fitting time for G-NURBS surfaces, as well as prolonged time for control mesh refinement.

5.5. Limitations

For models with sharp features, this paper employs the tensor voting method [7] to recognize these sharp features. In most cases, this method yields stable and reliable results. However, for input models with extremely uneven mesh distribution, the identified sharp features may be inaccurate, leading to redundancy or omission of sharp features on the fitting surfaces. As demonstrated in Fig. 18, when the sharp feature recognition of the input model is inaccurate, the corresponding sharp features appear on the fitting surface. However, they do not perfectly match the actual sharp features present in the input model. To address this issue, a more stable algorithm for sharp feature recognition is required.

6. Conclusion and Future Work

In this study, a method for adaptive surface fitting using G-NURBS was proposed, which can adaptively adjust the position and number of control points based on the geometrical feature distribution during the fitting process and can effectively fit sharp features. Our method consists of two key strategies. The first is the parameter correspondence for sharp features, and the second is the incorporation of geometrical feature information with errors to drive local



(a) Input model and its recognized sharp features.

(b) Fitting result

Figure 18: The fitting result obtained when the sharp features are recognized inaccurately. Sharp features are highlighted in magenta.

refinement of the control mesh. This method holds particular significance in CAD applications that demand high precision. The provided experimental results substantiate the effectiveness of the proposed method.

Similar to NURBS surfaces, the quality of G-NURBS fitting surfaces can potentially be optimized further by adjusting knot intervals and weights. This paper primarily concentrates on the influence of the quantity and position of control points, as they have the most direct impact on the spline surface. Furthermore, exploring the full potential of G-NURBS by incorporating variable weights and knot intervals remains a promising area for future work. This could involve developing algorithms specifically designed to optimize these parameters, thereby enhancing the adaptability and accuracy of G-NURBS for complex surface fitting challenges.

The quality of the control mesh is crucial for spline surface fitting, and the capability of G-NURBS to handle EPs provides a broad scope for the optimization of quadrilateral control meshes. An area for future exploration is the development of control mesh generation and optimization algorithms tailored for G-NURBS surface fitting. Existing quadrilateral meshing algorithms are not specifically designed for spline fitting tasks. It would be beneficial to consider the distribution of geometrical features and the expressive capacity of G-NURBS surfaces to devise new quadrangulation algorithms.

Furthermore, subsequent research could contemplate the development of rapid fitting algorithms for G-NURBS surfaces to optimize the overall computation time. Recognizing the significance of efficiency in practical applications, our future work will focus on algorithmic enhancements that streamline the fitting process. This includes exploring innovative approaches to reduce the complexity of mesh refinement and surface fitting, particularly for models with intricate details and control meshes with high counts of EPs. By advancing these areas, we aim to substantially improve the computational efficiency, making the G-NURBS fitting process more viable for complex and large-scale modeling tasks.

Acknowledgement

This work is partially supported by the National Key Research and Development Program of China under Grant 2020YFA0713703, Beijing Natural Science Foundation under Grant Z190004, NSFC (Nos. 12271516, 12371384 and 12201606), and Fundamental Research Funds for the Central Universities.

References

[1] M. J. Borden, M. A. Scott, J. A. Evans, and T. J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 87(1-5):15–47, 2011. 3

[2] Y.-F. Feng, L.-Y. Shen, X. Li, C.-M. Yuan, and X. Jiang. Patching Non-Uniform Extraordinary Points. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 2, 3

[3] Y.-F. Feng, L.-Y. Shen, C.-M. Yuan, and X. Li. Deep Shape Representation with Sharp Feature Preservation. *Computer-Aided Design*, 157:103468, 2023. 1

[4] J. Huang, Y. Zhou, M. Niessner, J. R. Shewchuk, and L. J. Guibas. Quadriflow: A scalable and robust method for quadrangulation. In *Computer Graphics Forum*, volume 37, pages 147–160. Wiley Online Library, 2018. 4, 9

[5] W. Jakob, M. Tarini, D. Panozzo, O. Sorkine-Hornung, et al. Instant field-aligned meshes. *ACM transactions on graphics (TOG)*, 34(6):189–1, 2015. 4, 9

[6] T. Kawasaki, P. K. Jayaraman, K. Shida, J. Zheng, and T. Maekawa. An image processing approach to feature-preserving B-spline surface fairing. *Computer-Aided Design*, 99:1–10, 2018. 1

[7] H. S. Kim, H. K. Choi, and K. H. Lee. Feature detection of triangular meshes based on tensor voting theory. *Computer-Aided Design*, 41(1):47–58, 2009. 6, 13

[8] M. Lyon, D. Bommes, and L. Kobbelt. Cost minimizing local anisotropic quad mesh refinement. In *Computer graphics forum*, volume 39, pages 163–172. Wiley Online Library, 2020. 9

[9] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003. 7

[10] N. Pietroni, S. Nuvoli, T. Alderighi, P. Cignoni, and M. Tarini. Reliable Feature-Line Driven Quad-Remeshing. *ACM transactions on graphics (TOG)*, 40(4), jul 2021. 4, 9

[11] H. Pottmann, S. Leopoldseder, and M. Hofer. Approximation with active B-spline curves and surfaces. In *10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings.*, pages 8–25. IEEE, 2002. 2, 4

[12] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, and T. J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering*, 88(2):126–156, 2011. 3

[13] M. A. Scott, R. N. Simpson, J. A. Evans, S. Lipton, S. P. Bordas, T. J. Hughes, and T. W. Sederberg. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering*, 254:197–221, 2013. 1

[14] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. *ACM transactions on graphics (TOG)*, 23(3):276–283, 2004. 1

[15] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCs. *ACM transactions on graphics (TOG)*, 22(3):477–484, 2003. 1

[16] A. Seiler, D. Großmann, and B. Jüttler. Spline surface fitting using normal data and norm-like functions. *Computer Aided Geometric Design*, 64:37–49, 2018. 1

[17] L.-Y. Shen, M.-X. Wang, H.-Y. Ma, Y.-F. Feng, and C.-M. Yuan. A framework from point clouds to workpieces. *Visual Computing for Industry, Biomedicine, and Art*, 5(1):21, 08 2022. 4

[18] T. Varady, R. R. Martin, and J. Cox. Reverse engineering of geometric models-an introduction. *Computer-aided design*, 29(4):255–268, 1997. 1

[19] W. Wang, H. Pottmann, and Y. Liu. Fitting B-spline curves to point clouds by curvature-based squared distance minimization. *ACM transactions on graphics (TOG)*, 25(2):214–238, 2006. 2, 4, 5

[20] W. Wang, Y. Zhang, M. Scott, et al. Converting an unstructured quadrilateral mesh to a standard t-spline surface. *Computational Mechanics*, 48:477–498, 2011. 1

[21] W. Wang, Y. Zhang, G. Xu, et al. Converting an unstructured quadrilateral/hexahedral mesh to a rational t-spline. *Computational Mechanics*, 50:65–84, 2012. 1

[22] Y. Wang and J. Zheng. Curvature-guided adaptive T-spline surface fitting. *Computer-Aided Design*, 45(8-9):1095–1107, 2013. 1

[23] X. Wei, X. Li, K. Qian, T. J. Hughes, Y. J. Zhang, and H. Casquero. Analysis-suitable unstructured t-splines: Multiple extraordinary points per face. *Computer Methods in Applied Mechanics and Engineering*, 391:114494, 2022. 2

[24] V. Weiss, L. Andor, G. Renner, and T. Várady. Advanced surface fitting techniques. *Computer Aided Geometric Design*, 19(1):19–42, 2002. 1

[25] Y. Zhang, J. Cao, Z. Chen, X. Li, and X.-M. Zeng. B-spline surface fitting with knot position optimization. *Computers & Graphics*, 58:73–83, 2016. 1