Patching Non-Uniform Extraordinary Points

Yi-Fei Feng
UCAS
Beijing, China
fengyifei15@mails.ucas.ac.cn

Li-Yong Shen
UCAS
Beijing, China
lyshen@ucas.ac.cn

Xin Li
USTC
Hefei, Anhui, China
lixustc@ustc.edu.cn

Chun-Ming Yuan
AMSS, CAS
Beijing, China
cmyuan@mmrc.iss.ac.cn

Xing Jiang
Beihang University
Beijing, China
jiangxin@buaa.edu.cn

Abstract

Smooth surfaces from an arbitrary topological control grid have been widely studied, which are mostly generalized from splines with uniform knot intervals. These methods fail to work well on extraordinary points (EPs) whose edges have varying knot intervals. This paper presents a patching solution for arbitrary topological 2-manifold control grid with non-uniform knots that defines one bi-cubic Bézier patch per control grid face except those faces with EPs. Experimental results demonstrate that the new solution can improve the surface quality for non-uniform parameterization. Applications in surface reconstruction, arbitrary sharp features on the complex surface and tool path planning for spline representation are also provided in the paper.

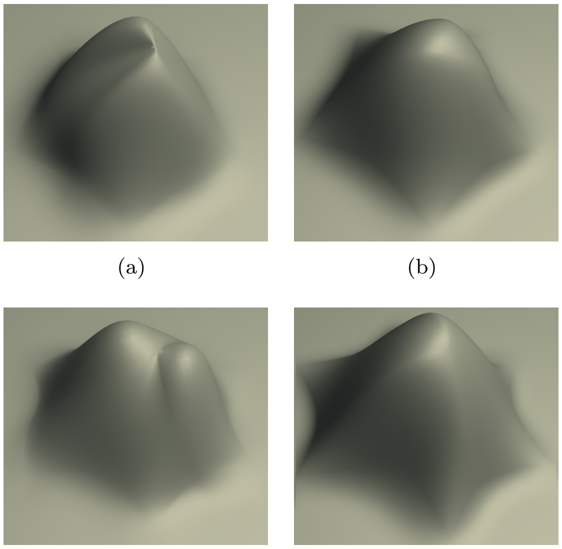
Keywords: NURBS, T-Splines, Extraordinary Points, Capping.

1. Introduction

Catmull-Clark surfaces [3] are ubiquitous in animation, while the CAD industry is dominated by non-uniform rational B-spline (NURBS). Each has its own advantage: Catmull-Clark surfaces are superior in their ability to create smooth surfaces of arbitrary topology and in their ease of use for animation, while NURBS can be refined by insertion of one row or column control points and are preferred for high-precision engineering models [8]. Hence, several surface types have been developed that generalize both Catmull-Clark surfaces and NURBS surfaces [35, 2, 27, 26, 16, 19, 20]. Each such surface expresses knot information by assigning a knot interval to the control mesh edge. Each can replicate NURBS if there are no extraordinary points (EPs, an interior vertex in a quadrilateral mesh which is shared by other than four faces), and Catmull-Clark surfaces if all knot intervals are one. However, these approaches have one common disadvantage: NURBS incompatibility. The CAD-analysis-manufacture workflow usually involves passing models between numerous software packages and most such software is currently NURBS based. Unfortunately, the above methods are not backward compatible with NURBS because these methods produce an infinite sequence of bi-cubic patches near EPs, of which NURBS software can only import a finite truncation [22].

Backward incompatibility is avoided in patch-based methods such as [28, 32] that replace the infinite sequence of patches near an EP with a small number of patches. Most previous patch-based papers [28, 13] only address uniform knot intervals. For more details, please refer to the survey paper [29] and some new constructions for uniform knot intervals [?, 14, ?, ?]. The support for non-uniform parameterization is a necessary step for forward compatibility with NURBS, where a NURBS surface can be exactly represented with the new representation. Although some modifications can be used to handle non-uniform knots [32], unfortunately, as illustrated in Figure 1 for the blending functions and Figure 2 for the real model, the resulting surfaces often exhibit a vexing problem near EPs with non-uniform knot intervals.

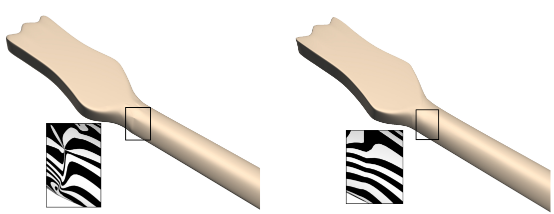
This paper presents a framework for producing acceptable surfaces near EPs with non-uniform knot intervals. The framework is patch based and it is backward and forward compatible with NURBS. The surfaces of our solution are illustrated in Figures 1 (b) and (d). We refer to the patches adjacent to an EP as irregular patches, and the boundary curves of patches that touch an EP as spoke curves. The method pro-



(a) (b) (c) (d)

Figure 1. Blending functions for valences 5 and 6 EP, where the knot intervals are randomly defined with a ratio no greater than 5. (a) and (c) are the results produced by [32] while (b) and (d) are surfaces produced by the present method.

duces one Bézier patch per control grid face. The irregular patches are G^1 with each other, and C^1 with the neighboring regular patches. The method involves two main steps. First, each face of the control mesh is replaced with a bi-cubic Bézier patch. The surface defined by these patches is C^0 along spoke edges and C^2 or C^1 elsewhere. Second, all irregular patches are degree elevated to bi-quintic and adjusted to provide G^1 continuity across spoke edges.



(a) Surface by [32] (b) Our result

Figure 2. A real model with different patching methods.

Although the problem can be formalized as a constrained optimization problem [28, 32], the construction cannot work well for non-uniform parameterization. The best implementation we know (currently used in the Autodesk T-Splines products) also produces ugly results similar to those in Figures 1 and 2. The present paper has two main key ideas to improve the construction in [32] and Autodesk T-Splines products. Firstly, we formulize the connecting functions as functions of an angle configuration for an EP and optimize the configuration to define the connecting functions. Secondly, the use of auxiliary control points near an EP is another idea to improve the surface quality, and avoid ambiguous computations for the Bézier extraction. Our discussion assumes a control grid of four-sided faces with equal knot intervals on the opposing edges of each face. Additionally our method only discusses the generalization from bi-cubic NURBS surface, which can be extended to arbitrary degrees if we can perform a similar Bézier extraction process as in Section 3.2.

The rest of the paper is organized as follows. Section 2 provides the background. The detailed construction of the blending functions is discussed in Section 3. The applications of the current construction are provided in Section 4. The last section draws conclusions and suggests future work.

2. Background

Geometric modeling is fundamental to CAD and isogeometric analysis (IGA) [11]. To provide a more flexible geometry modeling kernel, the study of EPs has been one of the most active research directions because they are inevitable in complex watertight geometric representations.

Complex geometric representations can be divided into two main categories: subdivision-based and patch-based representations. Most subdivision schemes are defined for uniform knots [3, 30]. For the subdivision scheme supporting non-uniform knot intervals, [35] first introduced non-uniform Catmull-Clark surfaces (NURCCs). And then, there are many extensions. For example, extended subdivision surfaces [27, 26] defined the scheme by forcing all knot intervals for spoke edges to be equal. [2] constructed a new rule by performing one directional refinement near EPs until the ratio of the largest and smallest knot intervals was less than two. [16] defined a subdivision rule for analysis-suitable T-splines [21] with a new non-uniform rule for EPs. [19] provided a subdivision rule with the help of eigen-polyhedron and [20] constructed a hybrid rule with proven G^1 continuity around EPs.

Patch-based methods are well studied in many papers, such as the patch-based methods from Peters et al. [28, 13, 29, 14], degenerated Bézier construction [38, 36] and manifold-based construction [23]. The manifold-based construction is not NURBS compatible and most of the other patch-based methods only address uniform knot intervals except that in [32], which made some modifications from [28] to support non-uniform knot intervals.

3. Define the Blending functions

This section provides detailed blending function construction for given control grid and knot intervals. The blending function construction for control grid with sharp edges will be discussed in Section 4.

Given an arbitrary topological control grid with predefined knot intervals, we need to define a blending function for each anchor such that it specializes to NURBS for the regular faces (grey faces in Figure 3). As shown in Figure 3, d_i, and a_i are knot intervals and can be any non-negative real numbers. Our discussion assumes that the control grid is a regular manifold grid and all the faces are quadrilaterals and knot intervals on the opposite edges are the same.

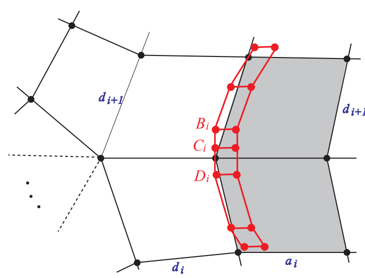


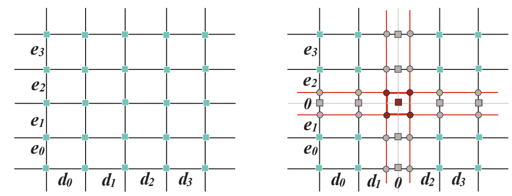
Figure 3. Notations and descriptions of the present problem.

As the representations for regular faces are bi-cubic B-splines, we can represent them in Bézier form, as shown with red points in Figure 3. Suppose B_i, C_i and D_i are the Bézier control points, then vectors $\overrightarrow{B_iC_i}$ and $\overrightarrow{C_iD_i}$ are parallel. Let $p_i = \frac{\overrightarrow{B_iC_i}}{\overrightarrow{C_iD_i}}$. Then in this paper, we assume that p_i can be different for different i. However, the patching approaches for uniform knots must have $p_i = 1$ for all i. The different p_i values raise the main challenge for this problem. We will discuss the main ideas to solve the problem in the following.

3.1. Framework for blending construction

The blending function construction and representation can be standardized using Bézier extraction operators, i.e., each blending function is defined as the contribution from the corresponding control point or anchor to the Bézier elements. While it is possible to obtain G^1 continuity with bi-cubic patches, the resulting surface is quite rigid and does not allow inflection points in spoke edges [28]. The problem is even worse for the non-uniform case. To obtain a less rigid solution, we refine the irregular patches to be bi-quintic, thereby providing additional degrees of freedom. We use most of these new degrees of freedom to enable a solution to the constraint equations that assure G^1 continuity.

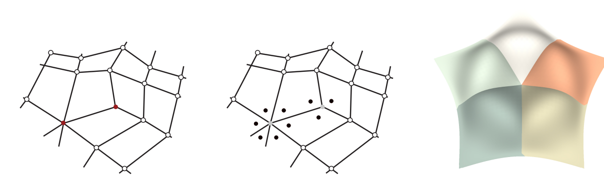
Reducing the continuity for bi-cubic NURBS can be achieved by inserting double knots. Referring to Figure 4, if one inserts a zero knot interval in both directions for the NURBS control grid, then one row and column of control points will be replaced by two rows and columns of control points. Thus, the red control point is replaced by four red points if we insert double knots.



(a) NURBS control grid (b) Grid with double knots

Figure 4. Insertion of double knots in a NURBS control grid.

We can naturally generalize the idea of double knots in NURBS to control grids of arbitrary topology because the surface has reduced continuity for the spoke edges. Thus in our construction, we replace a valence n EP with n anchors. The new idea can avoid ambiguous computations for face points and improve the surface quality. For the control grid in Figure 5 (a), the result of applying this procedure to obtain a different control grid topology is shown in Figure 5 (b). The idea of adding additional anchors is related to so-called splines allowing T-junction [34, 33, 4, 9, 6]. However, our approach is different in that we do not consider the T-junctions topological structures but rather we consider topological structures with only quadrilateral meshes and the new anchors are associated with control grid. The blending function constructions are totally different from those for splines with T-junctions.



(a) Given control grid (b) New anchors (c) Blending functions

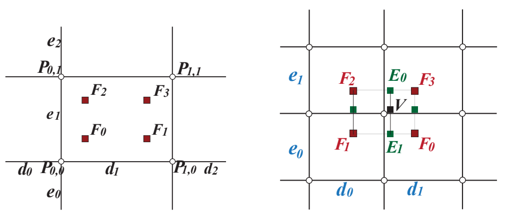
Figure 5. The control grid, auxiliary points and blending function construction.

In summary, for irregular patches, the blending functions are defined using a Bézier extraction and a G^1 perturbation procedure,

- Generate bi-cubic Bézier extraction as described in Section 3.2;
- Perturb the extraction operators to be a G^1-continuous blending as discussed in Section 3.4.

3.2. Bézier extraction

We use Bézier extraction to obtain the original C^0 blending functions, which compute the face, edge and vertex points as a linear combination of control points or anchors. If we refine a bi-cubic NURBS control grid by inserting one zero knot interval between each pair of non-zero knot intervals, control points on the refined grid are face points of the original grid.



(a) Face point (b) Edge and vertex points

Figure 6. Inserting zero knot intervals to compute the face points.

In Figure 6 (a), $P_{i,j}$ are the original control points and e_i, d_i are knot intervals. Then face points

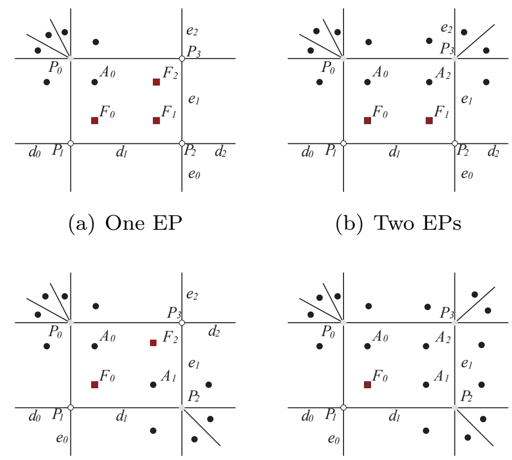
$$F_{2i+j} = (1-\alpha_i)[(1-\gamma_j)P_{0,0}+\gamma_j P_{0,1}]+\alpha_i[(1-\gamma_j)P_{1,0}+\gamma_j P_{1,1}]$$

where $\alpha_0 = \frac{d_0}{d_0+d_1+d_2}$, $\alpha_1 = \frac{d_0+d_1}{d_0+d_1+d_2}$, $\gamma_0 = \frac{e_0}{e_0+e_1+e_2}$, and $\gamma_1 = \frac{e_0+e_1}{e_0+e_1+e_2}$. If we now refine the grid by inserting a zero knot interval next to each existing zero knot interval, then we will obtain the edge and vertex points, as shown in Figure 6 (b). We refer to this process as Bézier extraction. Here

$$E_1 = \frac{d_1 F_1 + d_0 F_0}{d_0 + d_1}, \quad V = \frac{e_1 E_1 + e_0 E_0}{e_0 + e_1}. \tag{1}$$

However, the Bézier extraction process encounters difficulties for control grid with EPs and non-uniform knot intervals because the face point computation requires the neighbor knot intervals in both directions. If the vertex is an EP, the definition of the neighbor knot intervals is ambiguous. With the new anchors, Bézier extraction for irregular faces is slightly different from that for the regular faces. All the equations can be derived from double knots insertion of bi-cubic NURBS [8].

The EP is replaced by n anchors, which serve as face points for irregular faces. If the vertices of an irregular face are all EPs, then we replace the face points with anchors. The other cases are shown in Figure 7. If the irregular face only has one EP, as shown in Figure 7



(a) One EP (b) Two EPs

(c) Two EPs (d) Three EPs

Figure 7. Bezier extraction on irregular faces with different EPs configurations.

(a), one face point is replaced with anchor A_0 and face point F_0 is a linear combination of A_0, P_1 and P_2, F_2 is a linear combination of A_0, P_2 and P_3, and F_1 is a linear combination of A_0, P_1, P_2 and P_3, given by

$$\begin{aligned}
F_0 =& \frac{e_0}{e_0+e_1}A_0 + \frac{e_1}{e_0+e_1}\left(\frac{d_0+d_1}{d_0+d_1+d_2}P_1 + \frac{d_2}{d_0+d_1+d_2}P_2\right),\\
F_2 =& \frac{d_2}{d_1+d_2}A_0 + \frac{d_1}{d_1+d_2}\left(\frac{e_2}{e_0+e_1+e_2}P_2 + \frac{e_0+e_1}{e_0+e_1+e_2}P_3\right),\\
F_1 =& \frac{d_2}{d_1+d_2}\frac{e_0}{e_0+e_1}A_0 + \frac{d_0}{d_0+d_1+d_2}\frac{e_1}{e_0+e_1}P_1 + \gamma P_2 +\\
& \frac{e_2}{e_0+e_1+e_2}\frac{d_1}{d_1+d_2}P_3,
\end{aligned}$$

where $\gamma = 1 - \frac{d_2}{d_1+d_2}\frac{e_0}{e_0+e_1} - \frac{d_0}{d_0+d_1+d_2}\frac{e_1}{e_0+e_1} - \frac{e_2}{e_0+e_1+e_2}\frac{d_1}{d_1+d_2}$.

There are two cases when the irregular face has two EPs. If two EPs are connected, as shown in Figure 7 (b), two face points are replaced with anchors A_0, A_2. The face point F_0 is a linear combination of A_0, P_1 and P_2 and F_1 is a linear combination of A_2, P_1 and P_2, which is the same as the equation above. Another case is when the two EPs are not connected, as shown in Figure 7 (c), then the two face points are replaced with anchors A_0 and A_1. The face point F_0 is linear combination of P_1, A_0, while A_1, F_2 are linear combinations of P_3, A_0 and A_1.

$$\begin{aligned}
F_0 =& \frac{e_0}{2(e_0+e_1)}A_0 + \frac{d_0}{2(d_0+d_1)}A_1 + (1-\gamma_1)P_1,\\
F_2 =& \frac{e_2}{2(e_2+e_1)}A_1 + \frac{d_2}{2(d_2+d_1)}A_0 + (1-\gamma_2)P_1,
\end{aligned} \tag{2}$$

where $\gamma_1 = \frac{e_0}{2(e_0+e_1)} - \frac{d_0}{2(d_0+d_1)}$ and $\gamma_2 = \frac{e_2}{2(e_2+e_1)} - \frac{d_2}{2(d_2+d_1)}$.

If the irregular face has three EPs, as shown in Figure 7 (d), three face points are replaced with anchors

A_0, A_1 and A_2. The face point F_0 is a linear combination of P_1, A_0, A_1 and A_2, which is the same as equation (2).

After computing the face points, the edge points can be defined as in equation (1) by replacing some face points with anchors. The vertex point V for a valence n EP is computed as the linear combination of neighboring anchors,

$$V = \frac{\sum_{i=1}^{n} \omega_i A_i}{\sum_{i=1}^{n} \omega_i}, \quad \omega_i = d_{i-1}d_{i+2}. \tag{3}$$

3.3. Constraints for G^1 continuity

It is well known that G^1 continuity between two patches depends on the first partial derivatives only. The G^1 continuity of two Beźier patches is characterized by connecting functions.

Lemma 1. Let two Beźier patches $A(s,t)$ and $B(s,t)$ share a common boundary $R(s) = A(s,0) = B(s,0)$, see Figure 8 for an illustration, then $A(s,t)$ and $B(s,t)$ are G^1 continuous along the common boundary curve $R(s)$ if and only if there are $\alpha(s)$, $\beta(s)$, $\gamma(s)$ such that

$$\gamma(s)A_t(s,0) + \alpha(s)B_t(s,0) - \beta(s)A_s(s,0) = 0. \tag{4}$$

where $\alpha(s), \gamma(s) \geq 0$. Here $\alpha(s)$, $\beta(s)$ and $\gamma(s)$ are called connecting functions.

If two patches $A(s,t)$ and $B(s,t)$ share a common boundary $R(s) = A(s,0) = B(s,0)$ and are G^1 continuous along $R(s)$, then the connecting functions should satisfy equation (4). For any parameter s, since the vectors $A_t(s,0)$, $B_t(s,0)$ and $A_s(s,0)$ lie in the same plane, the equation can be solved explicitly as

$$\begin{aligned}
\alpha(s) &= |A_s(s,0) \times A_t(s,0)|c(s),\\
\beta(s) &= |A_t(s,0) \times B_t(s,0)|c(s),\\
\gamma(s) &= |B_t(s,0) \times A_s(s,0)|c(s),
\end{aligned}$$

where $|v_1 \times v_2|$ denotes the directed area of two vectors v_1 and v_2, and $c(s)$ should be appropriately chosen to ensure that $\alpha(s)$ and $\gamma(s)$ satisfy the requirement.

Now we are ready to study the G^1-continuity for the spoke edges. For a valence n EP, suppose the connecting functions for the i-th spoke edge are $\alpha_i(s)$, $\beta_i(s)$ and $\gamma_i(s)$. Although we can increase the degrees of the polynomials $\alpha_i(s)$, $\beta_i(s)$ and $\gamma_i(s)$, the G^1 constraints will be difficult to solve and it is hard to control the shape under these constraints. Thus, in the present paper, we set $\alpha_i(s) = d_i d_{i-1}$, $\gamma_i(s) = d_i d_{i+1}$ and $\beta_i(s) = b_i(1-s)^2$ if the other vertex of the spoke edge is not an EP and $\beta_i(s) = b_i(1-s) - c_i s$ if the spoke edge connects two EPs, where c_i is the constant

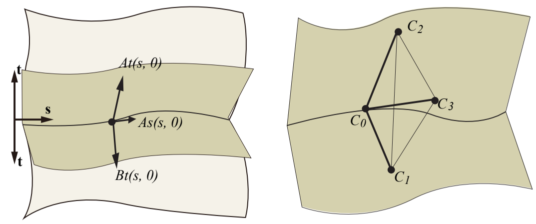


Figure 8. G^1 continuity for two Bézier patches, which is characterized by connecting functions.

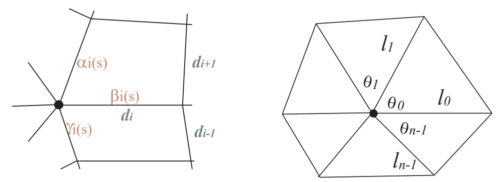


Figure 9. Define the connecting functions for spoke edges.

associated with the adjacent EP. Therefore, we need to define the constant b_i for the i-th spoke edge for each EP.

We define θ_i as the angle between $A_s(0,0)$ and $A_t(0,0)$ which is shown in Figure 9 and l_i as the length of the corresponding edge. Then we have $\sum_{i=0}^{n-1} \theta_i = 2\pi$ and

$$\frac{l_{i+1}\sin\theta_i}{l_{i-1}\sin\theta_{i-1}} = \frac{d_{i+1}}{d_{i-1}}, \quad b_i = -\frac{l_{i+1}d_{i-1}d_i \sin(\theta_{i-1}+\theta_i)}{l_i \sin\theta_{i-1}}. \tag{5}$$

The most common choice of $\theta_i = \frac{2\pi}{n}$ gives $b_i = -2d_{i-1}d_{i+1}\cos\frac{2\pi}{n}$, which is used in [32]. However, the connecting functions will lead to unsatisfactory results for non-uniform parameterization, as shown in Figures 12, 1, 13 and 15.

From the above, we observe that the connecting functions are determined by the angles. However, it is very challenging to optimize angles to improve the surface quality around the EPs. The basic idea of the present paper is that we first define a heuristic rule to define default angles and length, and then we try to solve the angles to minimize the difference between angles and length with the default ones.

Before giving the detailed equations, we first provide the following lemma with the notation $(i,j) = \sin(\theta_i)\sin(\theta_{i+2})\ldots\sin(\theta_j)$ for $i < j$.

Lemma 2. Given any $\theta_i > 0$ and $\sum_{i=0}^{n-1}\theta_i = 2\pi$, if n is odd or if n is even and $(0, n-2) = (1, n-1)$, then equation (5) always has solutions for l_i and b_i.

Proof. According to equation (5), the length l_i can be written as a function of the angle θ_i. If the valence

$n = 2k+1$ is odd, l_i can be written as

$$l_{2i} = \frac{(0,2i-2)}{(1,2i-1)} \frac{d_{2i}}{d_0} l_0; \qquad l_{2i+1} = \frac{(2i+2,2k)}{(2i+1,2k-1)} \frac{d_{2i+1}}{d_0} l_0. \tag{6}$$

If the valence $n = 2k$ is even, l_i can be written as

$$l_{2i} = \frac{(0,2i-2)}{(1,2i-1)} \frac{d_{2i}}{d_0} l_0; \qquad l_{2i+1} = \frac{(1,2i-1)}{(2,2i)} \frac{d_{2i+1}}{d_1} l_1. \tag{7}$$

With l_i and θ_i, we can compute b_i. $\square$

Referring to Figure 9, if one knot interval, for example d_i, is much smaller than the others, then the two irregular patches containing the i-th spoke edge should degenerate to two C^1 connected curves. Thus, in this case, the angle of θ_i and θ_{i-1} should be $\frac{\pi}{2}$. The default angles are defined according to this observation.

Denoting $k_i = \frac{n d_i d_{i+1}}{\sum_{i=0}^{n-1} d_i d_{i+1}}$, α_i is defined as

$$\alpha_i = \begin{cases} \frac{\pi}{2} - (2 - \frac{8}{n}) \arctan(k_i), k_i < 1 \\ \frac{(8-n)\pi}{2n} + (2 - \frac{8}{n}) \arctan(\frac{1+k_i}{2}), k_i \geq 1. \end{cases}$$

The default angles are defined as $\hat{\theta}_i = \frac{2\pi \cdot \alpha_i}{\sum_{i=0}^{n-1} \alpha_i}$. If n is even, then two of the angles should be perturbed to satisfy the other constraint. Let $\beta_1 = \hat{\theta}_{n-2} + \hat{\theta}_{n-1}$, and $\beta_2 = \frac{\sin(\hat{\theta}_0)\sin(\hat{\theta}_2)\ldots\sin(\hat{\theta}_{n-4})}{\sin(\hat{\theta}_1)\sin(\hat{\theta}_3)\ldots\sin(\hat{\theta}_{n-3})}$, then, $\hat{\theta}_{n-1} = \arctan(\frac{\beta_2 \sin(\beta_1)}{1+\beta_2\sin(\beta_1)})$ and $\hat{\theta}_{n-2} = \beta_1 - \hat{\theta}_{n-1}$.

After defining the angles, we can define the default length $\hat{l}_i$. The formula is similar to the length formula of eigen-polyhedron in [19].

$$\hat{l}_i = d_i + \sum_{j=1}^{n-1} d_j \left(\cos\left(\sum_{k=0}^{j-i-1} \hat{\theta}_{i+k} \right) \right)_+ ,$$

where $f_+ = \begin{cases} f, & f \geq 0; \\ 0, & f < 0. \end{cases}$.

The final angles and length are computed by solving the following optimization problem,

$$\begin{aligned} \min_{\theta} \quad & E = ||\hat{l}_i - l_i||_2^2 + \omega ||\hat{\theta}_i - \theta_i||_2^2, \\ s.t. \quad & \sum \theta_i = 2\pi. \\ & \theta_i > 0. \end{aligned}$$

The problem is a nonlinear least squares problem with linear constraints and we can solve the problem with a modified Levenberg-Marquardt algorithm[24] according to the geometric meaning. Lemma 2 shows that the length is uniquely determined by angles while many possible angles may exist for a given length. Thus, we solve the problem as a bivariate θ, l optimization problem rather than a univariate θ optimization problem. Each subproblem can be quickly solved because the Jacobian matrix can be explicitly computed. Suppose the angle and length of the i-th iteration are l^i and θ^i, where $l^0 = \hat{l}$ and $\theta^0 = \hat{\theta}$. In each iteration, we solve the problem with two steps. Firstly, we solve a nonlinear least squares problem using Levenberg-Marquardt algorithm to optimize θ^i by fixing l^i. In this step, we need to normalize the sum of angles to be 2π. Then, we compute l^i by fixing θ^i using the explicit expression. We repeat the alternating iterations until the step of angle is less than threshold. The parameter ω balances the differences in angles and length, which affects the final connecting functions. In our experiments, $\omega = 2$ and it works well for all our experiments.

Remark 1. Although the optimization is nonlinear, the algorithm works well in both sub-optimization problems and solutions are fast and accurate because the number of degrees of freedoms is the valence of EP which is in general very small in real applications.

Remark 2. Although we cannot guarantee that a nonlinear solver can always have a solution, in all the tests, our algorithm gives convergent results. In our implementation, if the solver does not converge in N iterations, then we will define the angles in the current iteration. And then we also have an angle configuration because we can define the length and connecting functions according to Lemma 2. In our experiments, we set $N = 200$. And in all our experiments, we have solutions within N iterations.

3.4. Solving the blending functions

Section 3.2 provides the bi-cubic representation on each face for each blending function, which is only C^0 along each spoke edge. In this section, we provide an algorithm to perturb the C^0 blending to construct a G^1-continuous blending using the connecting functions defined in Section 3.3.

With the connecting functions defined in the last section, if only one vertex is an EP, then the constraints for the G^1 continuity of the spoke edge are

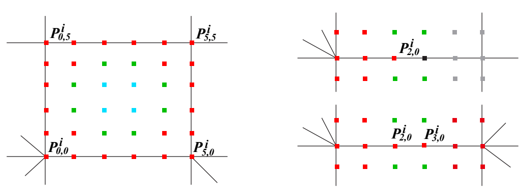
$$
\begin{cases}
((d_{i-1}+d_{i+1})d_i - b_i)P^i_{0,0} - d_i d_{i-1} P^i_{0,1} - d_i d_{i+1} P^{i-1}_{1,0} \\
\qquad\qquad\qquad\qquad\qquad + b_i P^i_{1,0} = 0, \\
b_i P^i_{0,0} + ((5d_{i-1}+5d_{i+1})d_i - 5b_i)P^i_{1,0} - 5d_i d_{i-1} P^i_{1,1} \\
\qquad\qquad\qquad - 5d_i d_{i+1} P^{i-1}_{1,1} + 4b_i P^i_{2,0} = 0, \\
b_i P^i_{0,0} - 5b_i P^i_{1,0} - ((10d_{i-1}+10d_{i+1})d_i - 10b_i)P^i_{2,0} \\
\qquad + 10d_i d_{i-1} P^i_{2,1} + 10d_i d_{i+1} P^{i-1}_{1,2} - 6b_i P^i_{3,0} = 0, \\
b_i P^i_{0,0} + 10b_i P^i_{2,0} + (10d_i(d_{i-1}+d_{i+1}) - 10b_i)P^i_{3,0} \\
\quad - 5b_i P^i_{1,0} - 10d_i d_{i-1} P^i_{3,1} - 10d_i d_{i+1} P^{i-1}_{1,3} + 4b_i P^i_{4,0} = 0, \\
\qquad P^i_{0,0} - 5P^i_{1,0} + 10P^i_{2,0} - 10P^i_{3,0} + 5P^i_{4,0} - P^i_{5,0} = 0.
\end{cases}
\tag{8}
$$

And if both vertices are EPs, then

$$
\begin{cases}
((d_{i-1}+d_{i+1})d_i - b^0_i)P^i_{0,0} - d_i d_{i-1} P^i_{0,1} - d_i d_{i+1} P^{i-1}_{1,0} \\
\qquad\qquad\qquad\qquad\qquad + b^0_i P^i_{1,0} = 0, \\
b^1_i P^i_{0,0} + ((5d_{i-1}+5d_{i+1})d_i - 4b^0_i - b^1_i)P^i_{1,0} - \\
\qquad 5d_i d_{i-1} P^i_{1,1} - 5d_i d_{i+1} P^{i-1}_{1,1} + 4b^0_i P^i_{2,0} = 0, \\
2b^1_i P^i_{1,0} + ((5d_{i-1}+5d_{i+1})d_i - 3b^0_i - 2b^1_i)P^i_{2,0} - \\
\qquad 5d_i d_{i-1} P^i_{2,1} - 5d_i d_{i+1} P^{i-1}_{1,2} + 3b^0_i P^i_{3,0} = 0, \\
3b^1_i P^i_{2,0} + ((5d_{i-1}+5d_{i+1})d_i - 2b^0_i - 3b^1_i)P^i_{3,0} - \\
\qquad 5d_i d_{i-1} P^i_{3,1} + 5d_i d_{i+1} P^{i-1}_{1,3} + 2b^0_i P^i_{4,0} = 0,
\end{cases}
\tag{9}
$$

where b^0_i and b^1_i are the constants for the two EPs defined in Section 3.3.

We first degree elevate the irregular patches to bi-quintic Bézier patches for sufficient degrees of freedom to ensure the G^1 continuity. Meanwhile, the G^1 blending functions are expected to be as close as possible to the original blending functions. Unlike that in [32], our optimization is computed locally. Denote the Bézier representation of the i-th irregular patch as $P^i_{j,k}$. As shown in Figure 10 (a), the Bézier coefficients are divided into three parts. The red ones are associated with vertices, the green ones are associated with edges and the blue ones are associated with faces.



(a) Bézier coefficients in one irregular face

(b) Two adjacent Bézier coefficients for a spoke edge

Figure 10. The bi-quintic Bézier coefficients for irregular faces, where the red and green coefficients need to be optimized. The grey coefficients are defined from Bézier extraction and the black ones are computed as a linear combination of the other coefficients.

Now, we can solve the coefficients based on the following steps according to the notations in Figure 10.

Optimize red coefficients for each EP We compute the red coefficients $P^i_{j,k},\ j+k \leq 2$ for each blending function locally for each EP. Note that for any i, $P^i_{j,0} = P^{i-1}_{0,j}$ for $j = 0, \ldots, 5$. Let V be a vector of coefficients to be optimized, where $V_0 = P^i_{0,0}$, $V_{2i+j+k} = P^i_{j,k}$. And let D be a vector of coefficients as $D_{3i} = V_{2i+1} - V_0$, $D_{3i+1} = V_{2i+2} - V_{2i+1}$, $D_{3i+2} = V_{2i+3} - V_{2i+2}$. Let $\hat{V}$ and $\hat{D}$ be the vectors of corresponding coefficients or corresponding differences from Bézier extraction after degree elevation. Then we first compute $P^i_{j,k},\ j,k \leq 1$ by solving the following least square problem with linear constraints,

$$
\begin{aligned}
\min \quad & ||V - \hat{V}||^2_2 + ||D - \hat{D}||^2_2 \\
s.t. \quad & M_v V = 0
\end{aligned}
\tag{10}
$$

where $M_v A$ are the first constraints in equation (8) or (9). We can solve all the $P^i_{j,k},\ j,k \leq 1$ under the first constraint in equation (8) or (9) using Lagrange multiplier method. And then $P^i_{2,0}$ can be computed according to the second equation in equation (8) or (9). Thus, unlike the consistent problem of $P^i_{1,1}$ in [28, 36], the constraints are always solvable because $P^i_{2,0}$ are not fixed in our constraints.

If the other vertex of the i-th spoke edge is not an EP, then grey points $P^i_{j,0}$, $P^i_{j,1}$ and $P^{i-1}_{1,j}$, $j = 4, 5$ are same as corresponding coefficients after degree elevation. And then the black point $P^i_{3,0}$ is computed according to the equation (8), i.e.,

$$
P^i_{3,0} = P^i_{2,0} + \frac{P^i_{4,0}}{2} - \frac{P^i_{1,0}}{2} + \frac{P^i_{0,0}}{10} - \frac{P^i_{5,0}}{10}.
$$

Optimize green coefficients for each spoke edge Once the vertex constraints are satisfied, the green coefficients are independent for each spoke edge. Thus we can compute the coefficients for each spoke edge by minimizing the difference between the G^1 coefficients with the C^0 coefficients of Bézier extraction after degree elevation under the remaining third or fourth constraints in equation (8) or equation (9). Similar to the red coefficients, the system can be solved through Lagrange multiplier method.

Optimize blue coefficients for each irregular face The blue coefficients can be computed directly. The basic idea is that we wish the iso-curves for each irregular patch to be cubic. We can compute the blue coefficients by the following explicit form.

$$
\begin{aligned}
P^i_{2,2} &= \frac{1}{2} f_1(P^i_{2,0}, P^i_{2,1}, P^i_{2,4}, P^i_{2,5}) + \frac{1}{2} f_1(P^i_{0,2}, P^i_{1,2}, P^i_{4,2}, P^i_{5,2}), \\
P^i_{3,2} &= \frac{1}{2} f_2(P^i_{2,0}, P^i_{2,1}, P^i_{2,4}, P^i_{2,5}) + \frac{1}{2} f_1(P^i_{0,2}, P^i_{1,2}, P^i_{4,2}, P^i_{5,2}), \\
P^i_{2,3} &= \frac{1}{2} f_1(P^i_{2,0}, P^i_{2,1}, P^i_{2,4}, P^i_{2,5}) + \frac{1}{2} f_2(P^i_{0,2}, P^i_{1,2}, P^i_{4,2}, P^i_{5,2}), \\
P^i_{3,3} &= \frac{1}{2} f_2(P^i_{2,0}, P^i_{2,1}, P^i_{2,4}, P^i_{2,5}) + \frac{1}{2} f_2(P^i_{0,2}, P^i_{1,2}, P^i_{4,2}, P^i_{5,2}),
\end{aligned}
$$

where $f_1(a,b,c,d) = -\frac{3}{10}a + b + \frac{1}{2}c - \frac{1}{5}d$ and $f_2(a,b,c,d) = -\frac{1}{5}a + \frac{1}{2}b + c - \frac{3}{10}d$.

With all the local optimization, we achieve globally G^1 blending functions. In order to define the final surface, we need to define the auxiliary control points.

3.5. Auxiliary control points computation

The last Section 3.4 provides the construction of blending functions for all the vertices of the given control grid except the EPs and we have also defined the functions for those anchors that replace the EPs. To define the final surface, we need to compute the position of those anchors, called auxiliary control points, which are totally free to choose. In this section, we first provide an explicit rule to compute these auxiliary control points as linear combinations of one neighboring control points. The basic idea is to define the auxiliary points such that the final surface has the same tangent plane as that for [19].

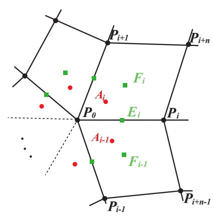


Figure 11. Notations for defining the auxiliary points.

Suppose P_0 is a valence n vertex, the neighbor control points are P_i and the knot intervals are d_i and a_i as shown in Figure 11, $i = 1, \ldots, 2n$. Then the auxiliary control points A_i are linear combination of P_i, $i = 0, \ldots, 2n$. Let the subdivision matrix for the rule in [19] be N. The details for construction of the matrix are provided in the appendix.

1. The subdivision rule in [19] requires the first neighboring control points. The only restriction for these points is that the rule should be generalized from Bézier extraction of bi-cubic NURBS. Thus we define the rule as the face, edge and vertex points F_i, E_i and V using the Bézier extraction in [32] for simplicity.

    - Face Points F_i:
    $$\begin{aligned} F_i =& \omega_i \omega_{i+1} P_{n+i} + (1-\omega_i)\omega_{i+1} P_{i+1} + \\ & (1-\omega_{i+1})\omega_i P_i + (1-\omega_i)(1-\omega_{i+1}) P_0, \end{aligned}$$
    where $\omega_i = \frac{d_{i-2}+d_{i+2}}{d_{i-2}+d_{i+2}+2d_i+2a_i}$, a_i is the knot interval of the edge which is adjacent to the i-th spoke edge.

    - Edge Points E_i:
    $$E_i = \frac{d_{i-1}}{d_{i-1}+d_{i+1}} F_i + \frac{d_{i+1}}{d_{i-1}+d_{i+1}} F_{i-1}.$$

    - Vertex Points V are computed as same as that in equation (3).

2. Compute the limit point C for the subdivision scheme based on eigen-polyhedron according to [30].
   Denote $P = [V, E_1, \ldots, E_n, F_1, \ldots, F_n]$, and the normalized left eigenvector corresponding to the eigenvalue 1 by $L^0 = [L^0_0, \ldots, L^0_{2n}]$ and then the limit point is $C = L^0 P$.

3. Compute the tangent plane for the subdivision scheme based on eigen-polyhedron according to [30].
   Define $2n \times 2n$ matrices $\widehat{N}$, where $\widehat{N}_{i,j} = N_{i+1,j+1} - N_{0,j+1}$. Let $\widehat{P} = [E_1 - V, \ldots, E_n - V, F_1 - V, \ldots, F_n - V]^T$. Note that according to [19], λ is the leading eigenvalue for the matrix $\widehat{N}$. $\widehat{N}$ can be written in the form of $\widehat{N} = U\Lambda U^{-1}$, where Λ is a diagonal matrix with the elements being the eigenvalues. Suppose i_1 and i_2 are two indices such that $\Lambda(i_1,i_1) = \Lambda(i_2,i_2) = \lambda$. Let $\widehat{\Lambda}$ be a new diagonal matrix where all the elements are zero except that $\widehat{\Lambda}(i_1,i_1) = \widehat{\Lambda}(i_2,i_2) = 1$. Then we can define a new matrix $Q = U\widehat{\Lambda}U^{-1}$. Now, we can define a set of vectors $\overline{V} = Q\widehat{P}$.

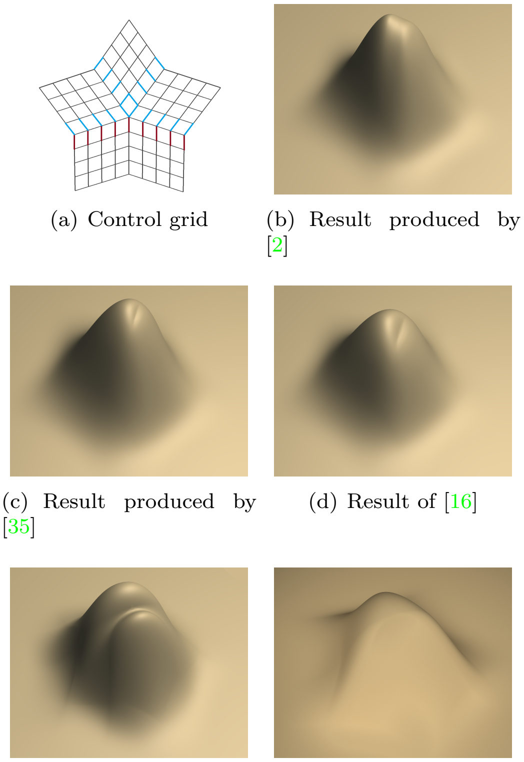
4. Define the auxiliary control points A_i
   Now $\overline{V}_{i+n}$ span the plane which is parallel to the tangle plane defined by the subdivision scheme in [19]. We then move the anchors such that the final surface has the same limit point as that in [19]. Suppose the contribution of the blending functions for A_i at the EP is c_i, and let $C' = \sum_{i=0}^{n-1} c_i \overline{V}_{i+n}$, then $A_i = C - C' + \overline{V}_{i+n}$.

3.6. Results

In this section, we show some result surfaces produced by our method and compare them with the existing nonuniform subdivision schemes [35, 2, 16] and the non-uniform patch-based method [32].

We first show the graphs of blending functions for a valence five EP with non-uniform knot intervals in Figure 12. We observe that the approaches in [35, 2, 16] produce limit surfaces with very similar quality, which is also the situation in other examples.

Creating nice-looking non-uniform subdivision schemes near EPs had been a vexing problem for a few decades, and no methods could have fixed the problem



(a) Control grid (b) Result produced by [2]

(c) Result produced by [35] (d) Result of [16]

(e) Result of [32] (f) Result of our method

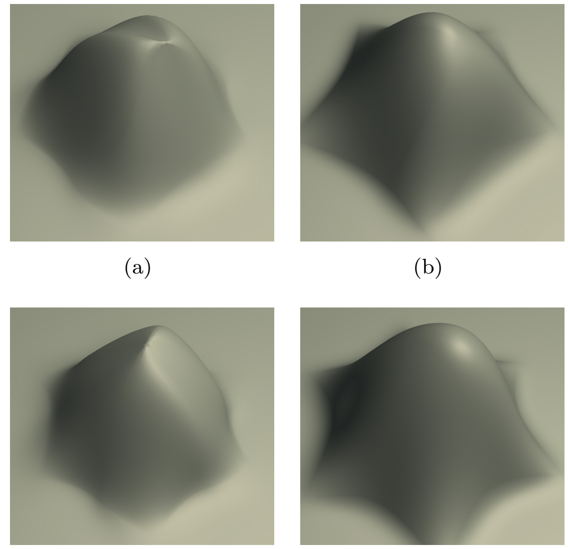
Figure 12. The surfaces generated from a control grid with non-uniform knot intervals, where the knot intervals for the red edges, blue edges are 8 and 4, and the others are one. (b), (c), (d) are produced by different subdivision schemes and (e) and (f) are produced by different patch-based methods.

until the method in [19] was discovered. In our experiments, our method produces similar quality surfaces as those in [19, 20], which are much better than the results produced by any other non-uniform subdivision schemes. Note that [32] is the only reference which considers the same problem as the present paper, so in the following, we only give detailed comparisons with method in [32].

The second experiment is some blending functions for a valence 5 EP and a valence 6 EP with 200 randomly chosen knot intervals, where the ratio of largest and smallest knot intervals for the spoke edges is between 1 to 5. For all the experiments, the surface quality of our method is obviously better than that in [32]. Figures 1 and 13 show two examples.

The improved blending functions can lead to better shape quality for the real models, two of the examples are shown in Figures 2 and14.

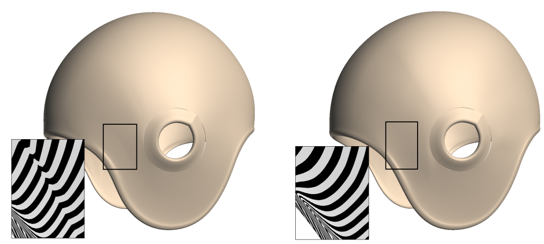
The next experiment is for the angle configuration. Optimizing the angle in terms of the knot intervals can improve the surface quality in our experiments. Figure 15 shows such an example for a valence 5 EP with knot intervals $1, 4, 1, 4, 8$, which clearly shows that angles between spoke curves can impact the surface quality.



(a) (b)

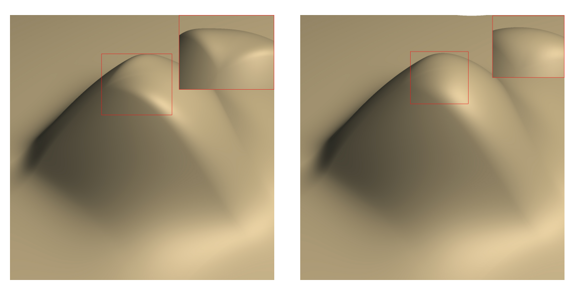
(c) (d)

Figure 13. Blending functions with random non-uniform knot intervals. The left column of pictures are the results produced by [32] while the right ones are the surface produced by our method.



(a) Surface by [32] (b) Our result

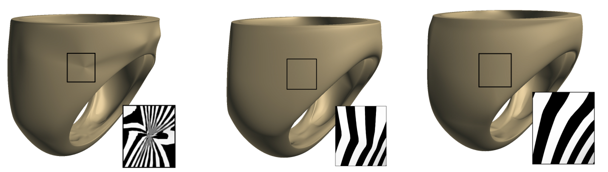
Figure 14. Real models with different approaches.



(a) $\theta_i = \frac{2\pi}{n}$ (b) Angle after optimization

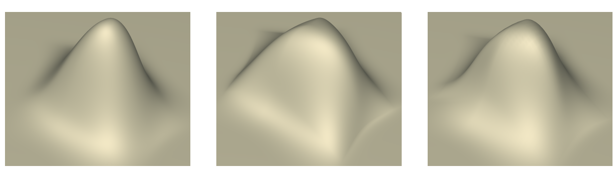
Figure 15. Blending function with uniform angles and optimized angles.

The auxiliary points can impact the surface quality, where the default ones are computed in Section 3.5. We can improve the surface quality by further optimizing these points. Figure 16 shows such an example, where (a) is the result of [32], (b) is the result of our method with default anchors and (c) is the result of optimization of anchors by minimizing the average of thin plate energy for irregular patches and the difference between the new auxiliary points with default ones. We can observe that after optimization, the surface has smoother reflection lines.



(a) Surface with [32] (b) Surface with default anchors (c) Surface with optimized anchors

Figure 16. Optimize the auxiliary control points for smoother surface.



(a) Surface with [19] (b) Surface with default anchors (c) Surface with approximating anchors

Figure 17. Approximate the limit surface in [19] using our representation.

The auxiliary points can also be computed by approximating to some target surfaces. For example, figure 17 shows another way to define the auxiliary points by approximating the limit subdivision surface using the rule in [19].

4. Application

The last section provides a set of blending function constructions for a given quadrilateral mesh with given non-uniform knot intervals, which provides many possible applications in different fields. In this section, we show several applications of the new spline representation in reverse engineering, geometric modeling and computer numerical control (CNC).

4.1. Spline surface fitting from triangle meshes

Surface fitting is the direct application for new spline representation. Because of the limitation of NURBS representation, in order to approximate a complex topological triangle mesh, most methods first need to split the triangle mesh into many pieces and then fit each piece using uniform B-splines or NURBS [7]. In the end, each NURBS is merged by adjusting the control points. The new representation allows one to approximate an arbitrary topology model directly.

In our fitting pipeline, the quadrilateral mesh is extracted from input triangle meshes as the quadrilateral mesh to define the blending functions and the knot intervals are set to be uniform without loss of generality. Then we construct the blending functions and re-parameterize the triangle mesh locally. The control points are optimized to minimize the error between triangle mesh and spline surface. There are many methods to generate quadrilateral meshes that conform to geometric features, such as [12, 10]. We are using the approach in [10] to generate the initial quadrilateral mesh. The distance between target surface and spline surface is measured by SDM [31] and a feature sensitive parametrization [18] is applied to improve the surface quality. We do not need a merging process because the blending functions are global G^1 continuous. Several fitting example spline surfaces are shown in Figure 18.

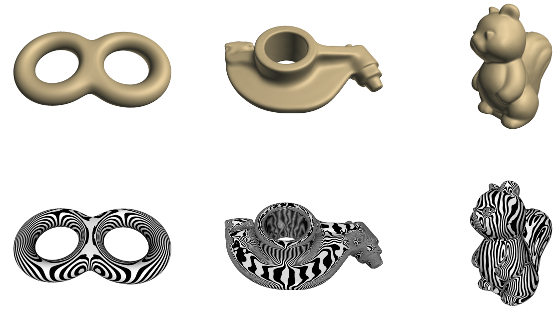


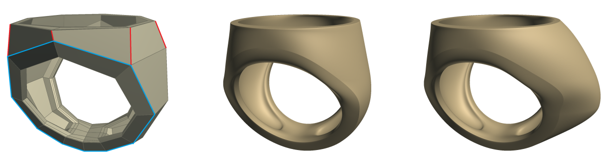
Figure 18. Our fitting results of three smooth models. The top row shows our fitting results and the bottom row is the reflection lines of fitted surfaces.

4.2. sharp features through EPs

Sharp features on spline surface are a set of curves through which the spline surface is C^0 continuous. The sharp features on NURBS surfaces play an important role in geometric modeling. As NURBS has the limitation of rectangular topology, the sharp features on NURBS surfaces can only be global iso-parameter lines. Locally iso-parameter sharp features can be created using T-splines [34, 33].

Sharp features on subdivision surfaces have been well researched. Semi-sharp features were introduced firstly in [5] and improved in [1], which provided elegant algorithmic approaches to modify the Catmull-Clark subdivision scheme. Early attempts for non-uniform sharp features include [35]. Later, it is improved in [15, 37]. All the above approaches are subdivision-based, which do not have an analog in patching and are not NURBS compatible. It is very challenging to generate complex sharp features passing through EPs in NURBS-compatible way.

The framework of our construction can be generalized to handle arbitrary sharp features in a straightforward manner. The sharp features can also pass through EPs, something that no previous patching solution has addressed. Similar to those in [37], the edges of input mesh are specified with sharp or regular tags. For sharp edges, we add more anchors that are associated with basis functions for sharp edges. For the construction, as we can see from above, the main steps for our patching algorithm are Bézier extraction, angle configuration, G^1 constraints and optimization. In the Bézier extraction step, the computation is very similar because the situation corresponds to triple knots in the NURBS case. For the angle configuration, the only difference is that the sum of angles to each side of two continuous sharp edges must both be π. And for the G^1 constraints step, we delete the G^1 constraints for the sharp edges and solve the basis functions exactly the same as before.



(a) Control grid of Ring model (b) Surface with our method (c) Surface after editing control grid

Figure 19. (a) is an input control grid with knot intervals and sharp edges, (b) is the final surface created by our method and (c) is the surface created by editing the control grid which preserves the continuous and sharp edges.

Thus, for the sharp features, we are given a control grid with knot intervals and sharp feature tags, as shown in Figure 19. In the figure, the knot intervals for the red edges are six, the others are one, the blue edges are sharp edges. And then the method defines the surface shown in Figure 19 (b), which is C^0 along the sharp edges and at least G^1 along the rest edges. Note that the continuity will be maintained when we edit the position of control grid, which is shown in Figure 19 (c). Several real models with sharp features are constructed using our representation in Figure 20.

4.3. CNC for new spline representation

Tool path planning for spline representation plays an essential role in CNC. After construction of the spline surface models, we can generate the tool path for arbitrary topological meshes which satisfies the error constraint. As the spline representation is a smoother version of the triangular mesh, the manufacture is much more precise in our experiments.

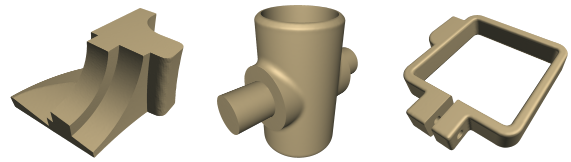
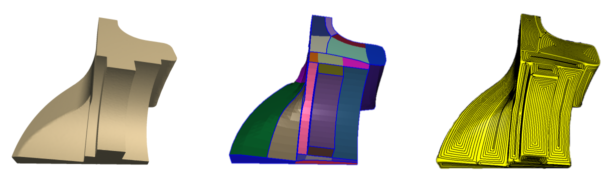


Figure 20. Real models with sharp features are represented using the present method.

We propose a two-stage method for tool path generation. Firstly a surface segmentation is used to split the control mesh into regular pieces. Secondly, we use the Fermat Spiral to construct the tool path. In segmentation part, we search for optimal one in all possible cases. In this process, each edge is assigned the weights measuring the complexity of the segmentation if it were cut, and the final segmentation becomes the minimal spanning tree problem which is solved by Kruskal algorithm [17]. After segmentation, each part becomes a regular domain, in which the Fermat Spiral can be computed as the tool path [39]. Figure 21 shows the segmentation results for input surfaces and the generated tool path.



(a) Model Surface (b) Segmentation (c) Tool Path

Figure 21. The generated CNC tool path for the models represented using the present method.

5. Conclusions and future work

The present paper provides an algorithm to define a set of blending functions for unstructured quadrilateral control grid with non-uniform knot intervals and labels for sharp edges. The main challenge comes from the issue of non-uniform parameterization for the irregular patches. The non-uniform parameterization is essential for the NURBS compilable geometric modeling kernel and local insertion operation. The other challenge is to generate arbitrary sharp features in the above geometric modeling tools. Both problems are solved in the same framework with two key ideas: auxiliary points and angle configuration.

The angle configuration is solved through a nonlinear optimization with a heuristic default values. The reason to use heuristic default values is because there is no explicit relation between the angles and the final surface quality. However, in our experiments, such heuristic values give acceptable surface quality if the ratio of knot intervals is between 1 and 5. How to formulize the angle configuration based on the final surface quality and how to solve the problem will be left as future work.

Other possible further research is for connecting functions. The current connecting functions are defined such that α and γ are constants for simplicity. How to construct the surface for the other choice of connecting functions is also an interesting future problem.

The third possible extension is for fairing energy. The present paper uses the same faring energy as that in [32]. The degrees of freedom are optimized by designing fairing energy similar to those in [25] for non-uniform knot intervals needs additional studies.

The idea in the present paper can be generalized to handle T-junctions in the control grid trivially if the T-junctions are far away from the EPs, but how to construct the blending functions if there are T-junctions on the edges of irregular faces needs more study. Of course, it is also an important and interesting problem to generalize the continuity from G^1 to G^2.

Appendix

The appendix gives the subdivision rule in [19]. Suppose given a valence-n EP V with the neighboring control points E_i and F_i, the knot intervals d_i, $i = 0, \dots, n-1$ are knot intervals for the spoke edges. The new control points are denoted as $\overline{V}$, $\overline{E}_i$ and $\overline{F}_i$. Denote $\gamma = \frac{4}{c+1+\sqrt{(c+9)(c+1)}}$ and $\lambda = \frac{1+\gamma}{4\gamma}$, where $c = \cos(\frac{2\pi}{n})$. Let $l_i = \frac{2d_i+d_i^-}{3}$, where $d_i^- = \frac{d_{i-2}d_{i+1}+d_{i+2}d_{i-1}}{d_{i-1}+d_{i+1}}$. And a set of points $\widehat{V}$, $\widehat{E}_i$ and $\widehat{F}_i$ in R^2 are defined as $\widehat{V} = (0,0)$, $\widehat{E}_i = l_i(\cos\frac{2i\pi}{n}, \sin\frac{2i\pi}{n})$ and $\widehat{F}_i = \gamma(\widehat{E}_i + \widehat{E}_{i+1})$. The subdivision rules in [19] can be defined in the following.

- Vertex Point:

$$\overline{V} = \frac{n-3}{n}V + \frac{3}{n}\frac{\sum_{i=0}^{n-1}(m_i M_i + f_i\overline{F}_i)}{\sum_{i=0}^{n-1}(m_i + f_i)}, \qquad (11)$$

where $f_i = d_{i-1}d_{i+2}$, $m_i = f_i + f_{i-1}$. Let $\widehat{V}$ be the vertex point by replacing V, E_i and F_i in the above equation with $\widehat{V}$, $\widehat{E}_i$ and $\widehat{F}_i$.

- Face points:

$$\begin{aligned}\overline{F}_i =& \alpha_{i,1}\alpha_{i,2}F_i + (1-\alpha_{i,1})\alpha_{i,2}E_{i+1} + \\ & \alpha_{i,1}(1-\alpha_{i,2})E_i + (1-\alpha_{i,1})(1-\alpha_{i,2})V,\end{aligned} \qquad (12)$$

where $\alpha_{i,1}$ and $\alpha_{i,2}$ are the unique solutions of the function

$$\begin{aligned}\overline{V} + \lambda\widehat{F}_i =& \alpha_{i,1}\alpha_{i,2}\widehat{F}_i + (1-\alpha_{i,1})\alpha_{i,2}\widehat{E}_{i+1} + \\ & \alpha_{i,1}(1-\alpha_{i,2})\widehat{E}_i + (1-\alpha_{i,1})(1-\alpha_{i,2})\widehat{V}.\end{aligned}$$

- Edge points:

$$\begin{aligned}\overline{E}_i =& (1-\beta_{i,2})(\frac{1-\beta_{i,1}}{2}P_{i,1} + \frac{\beta_{i,1}}{2}P_{i,2} + \frac{1}{2}V) + \\ & \beta_{i,2}(\frac{1-\beta_{i,1}}{2}P_{i,3} + \frac{\beta_{i,1}}{2}P_{i,4} + \frac{1}{2}E_i),\end{aligned}$$

where

$$\begin{aligned}P_{i,1} &= (1-\alpha_{i-1,1})V + \alpha_{i-1,1}E_{i-1},\\ P_{i,2} &= (1-\alpha_{i,2})V + \alpha_{i,2}E_{i+1};\\ P_{i,3} &= (1-\alpha_{i-1,1})E_i + \alpha_{i-1,1}F_{i-1},\\ P_{i,4} &= (1-\alpha_{i,2})E_i + \alpha_{i,2}F_i,\end{aligned}$$

and $\beta_{i,1}$ and $\beta_{i,2}$ are the unique solutions of the function

$$\begin{aligned}\overline{V} + \lambda\widehat{E}_i =& \beta_{i,2}(\frac{1-\beta_{i,1}}{2}\widehat{P}_{i,3} + \frac{\beta_{i,1}}{2}\widehat{P}_{i,4} + \frac{1}{2}\widehat{E}_i) \\ & +(1-\beta_{i,2})(\frac{1-\beta_{i,1}}{2}\widehat{P}_{i,1} + \frac{\beta_{i,1}}{2}\widehat{P}_{i,2} + \frac{1}{2}\widehat{V}),\end{aligned}$$

where $\widehat{P}_{i,j}$ are defined by replacing V, E_i and F_i in the above equation with $\widehat{V}$, $\widehat{E}_i$ and $\widehat{F}_i$.

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. 61872332)

References

[1] H. Biermann, A. Levin, and D. Zorin. Piecewise smooth subdivision surfaces with normal control. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00, pages 113–120. ACM Press/Addison-Wesley Publishing Co., 2000. 10

[2] T. J. Cashman, U. H. Augsdörfer, N. A. Dodgson, and M. A. Sabin. NURBS with Extraordinary Points: High-degree, Non-uniform, Rational Subdivision Schemes. ACM Transactions on Graphics, 28(3):1–9, 2009. 1, 2, 8, 9

[3] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. Computer Aided Design, 10:350–355, 1978. 1, 2

[4] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, and Y. Feng. Polynomial splines over hierarchical T-meshes. Graphical Models, 74:76–86, 2008. 3

[5] T. DeRose, M. Kass, and T. Truong. Subdivision surfaces in character animation. In Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pages 85–94, New York, NY, USA, 1998. ACM. 10

[6] T. Dokken, T. Lyche, and K. F. Pettersen. Polynomial splines over locally refined box-partitions. Computer Aided Geometric Design, 30:331–356, 2013. 3

[7] M. Eck and H. Hoppe. Automatic reconstruction of b-spline surfaces of arbitrary topological type. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 325–334, 1996. 10

[8] G. E. Farin. NURBS curves and surfaces: from projective geometry to practical use. A. K. Peters, Ltd., Natick, MA, 2002. 1, 4

[9] D. R. Forsey and R. H. Bartels. Hierarchical B-spline refinement. Comput. Graph., 22:205–212, 1988. 3

[10] 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. 10

[11] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. Computer Methods in Applied Mechanics and Engineering, 194:4135–4195, 2005. 2

[12] W. Jakob, M. Tarini, D. Panozzo, and O. Sorkine-Hornung. Instant field-aligned meshes. ACM Trans. Graph., 34(6):189–1, 2015. 10

[13] K. Karciauskas, D. Panozzo, and J. Peters. T-junctions in spline surfaces. ACM Transactions on Graphics, 36(5):1–9, 2017. 1, 2

[14] K. Karciauskas and J. Peters. Multi-sided completion of c^2 bi-3 and c^1 bi-2 splines: A unifying approach. Computer Aided Geometric Design, 86:101978, 2021. 1, 2

[15] J. Kosinka, M. A. Sabin, and N. A. Dodgson. Subdivision surfaces with creases and truncated multiple knot lines. Computer Graphics Forum, 33(1):118–128, 2014. 10

[16] D. Kovacs, J. Bisceglio, and D. Zorin. Dyadic T-mesh subdivision. ACM Transactions on Graphics (Siggraph 2015), 34(4):1–8, 2015. 1, 2, 8, 9

[17] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical society, 7(1):48–50, 1956. 11

[18] Y.-K. Lai, S.-M. Hu, and H. Pottmann. Surface fitting based on a feature sensitive parametrization. Computer-Aided Design, 38(7):800–807, 2006. 10

[19] X. Li, G. T. Finnigin, and T. W. Sederberg. G^1 non-uniform Catmull-Clark surfaces. ACM Transactions on Graphics (Siggraph 2016), 35(4):1–8, 2016. 1, 2, 6, 8, 9, 10, 12

[20] X. Li, X. Wei, and J. Y. Zhang. Hybrid non-uniform recursive subdivision with improved convergence rates. Computer Methods in Applied Mechanics and Engineering, 352:606–624, 2019. 1, 2, 9

[21] X. Li, J. Zheng, T. W. Sederberg, T. J. Hughes, and M. A. Scott. On the linear independence of T-splines blending functions. Computer Aided Geometric Design, 29:63–76, 2012. 2

[22] W. Ma. Subdivision surfaces for cad: an overview. Computer-Aided Design, 37(7):693–709, 2005. 1

[23] M. Majeed and F. Cirak. Isogeometric analysis using manifold-based smooth basis functions. Computer Methods in Applied Mechanics and Engineering, 326:547–567, 2017. 2

[24] J. J. Moré. The levenberg-marquardt algorithm: implementation and theory. In Numerical analysis, pages 105–116. Springer, 1978. 6

[25] H. P. Moreton and C. H. Séquin. Functional optimization for fair surface design. In Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '92, pages 167–176, New York, NY, USA, 1992. Association for Computing Machinery. 12

[26] K. Muller, C. Funfzig, L. Reusche, D. Hansford, G. Farin, and H. Hagen. DINUS-Double Insertion, Non-Uniform, Stationary Subdivision Surfaces. ACM Transactions on Graphics, 29:1–21, 2010. 1, 2

[27] K. Muller, L. Reusche, and D. Fellner. Extended subdivision surfaces: building a bridge between NURBS and Catmull-Clark surfaces. ACM Transactions on Graphics, 25:268–292, 2006. 1, 2

[28] J. Peters. Patching catmull-clark meshes. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00, pages 255–258, New York, NY, USA, 2000. ACM. 1, 2, 3, 7

[29] J. Peters. Splines for meshes with irregularities. The SMAI journal of computational mathematics, S5:161–183, 2019. 1, 2

[30] J. Peters and U. Reif. Subdivision Methods for Geometric Design. Morgan Kaufmann Burlington, MA, 2008. 2, 8

[31] 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. 10

[32] 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, 2, 5, 7, 8, 9, 10, 12

[33] 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, 23 (3):276–283, 2004. 3, 10

[34] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCSs. ACM Transactions on Graphics, 22 (3):477–484, 2003. 3, 10

[35] T. W. Sederberg, J. Zheng, D. Sewell, and M. Sabin. Non-uniform recursive subdivision surfaces. In Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '98, pages 387–394. ACM Press/Addison-Wesley Publishing Co., 1998. 1, 2, 8, 9, 10

[36] X. Shi, T. Wang, and P. Yu. A practical construction of g1 smooth biquintic b-spline surfaces over arbitrary topology. Computer-Aided Design, 36(5):413–424, 2004. 2, 7

[37] Y. Tian, X. Li, and F. Chen. Non-uniform subdivision surfaces with sharp features. Computer Graphics Forum, 39(6):232–242, 2020. 10, 11

[38] D. Toshniwal, H. Speleers, and T. J. Hughes. Smooth cubic spline spaces on unstructured quadrilateral meshes with particular emphasis on extraordinary points: geometric design and isogeometric analysis considerations. Computer Methods in Applied Mechanics and Engineering, 327:411–458, 2017. 2

[39] H. Zhao, H. Zhang, S. Xin, Y. Deng, C. Tu, W. Wang, D. Cohen-Or, and B. Chen. Dscarver: decompose-and-spiral-carve for subtractive manufacturing. ACM Transactions on Graphics (TOG), 37(4):1–14, 2018. 11