G^1 Non-Uniform Catmull-Clark Surfaces

Xin Li*
USTC

G. Thomas Finnigan†
Autodesk, Inc.

Thomas W. Sederberg‡
Brigham Young University

Abstract

This paper develops new refinement rules for non-uniform Catmull-Clark surfaces that produce G^1 extraordinary points whose blending functions have a single local maximum. The method consists of designing an "eigen polyhedron" in $\mathbb{R}^2$ for each extraordinary point, and formulating refinement rules for which refinement of the eigen polyhedron reduces to a scale and translation. These refinement rules, when applied to a non-uniform Catmull-Clark control mesh in $\mathbb{R}^3$, yield a G^1 extraordinary point.

Keywords: Non-uniform, Catmull-Clark surfaces, NURBS

Concepts: •**Computing methodologies** → **Parametric curve and surface models;**

1 Introduction

Several surface representations include both Catmull-Clark and NURBS surfaces as special cases [Sederberg et al. 1998; Müller et al. 2006; Sederberg et al. 2003a; Müller et al. 2010; Cashman et al. 2009; Cashman 2010; Kovacs et al. 2015]. A major aim of such surfaces is to facilitate the adoption of non-uniform subdivision surfaces by the CAD industry, where NURBS are widely used.

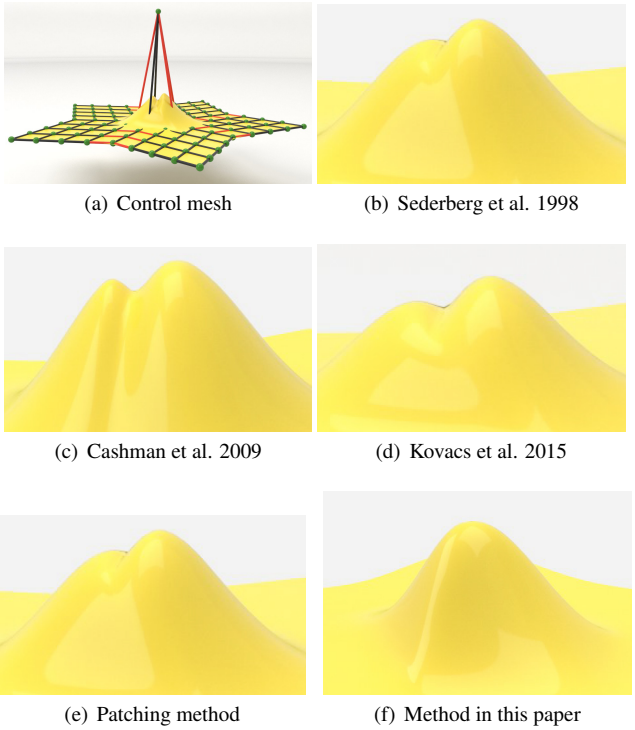
This paper solves a problem that vexes such surfaces: if knot intervals are different, the blending functions for extraordinary points can have two local maxima, as illustrated in Figure 1 (see also Figure 17 in [Kovacs et al. 2015]). Ugly blending functions unavoidably manifest themselves in real-world models, such as in Figure 2. The practical importance of this problem is elevated because such surfaces are gaining widespread commercial use, plus they play a central role in isogeometric analysis [Bazilevs et al. 2010].

The method in this paper creates G^1, wrinkle-free surfaces for any reasonable choice of knot intervals. Figure 1.f shows a blending function produced by our method.

The paper is laid out as follows. Sections 2 and 3 review refinement rules and prior art. Section 4 presents the notion of an eigen polyhedron for which refinement is simply a scale and a translation and Section 5 discusses how to design an eigen polyhedron that will specialize to NURBS in the valence four case, and to Catmull-Clark surfaces in the uniform case. Section 6 explains how to deduce refinement rules from a given eigen polyhedron, thus creating G^1 refinement. Section 7 concludes.

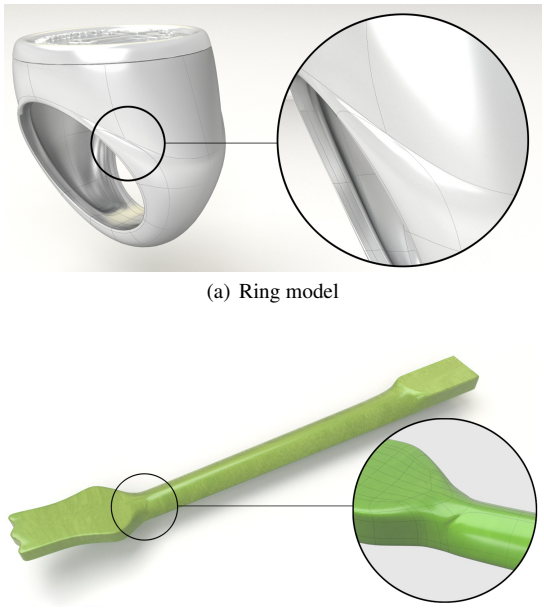
*email: lixustc@ustc.edu.cn
†email: tomfinnigan@gmail.com
‡email: tom@cs.byu.edu

SIGGRAPH '16 Technical Paper,, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925924



(a) Control mesh (b) Sederberg et al. 1998 (c) Cashman et al. 2009 (d) Kovacs et al. 2015 (e) Patching method (f) Method in this paper

Figure 1: *Valence five extraordinary point blending function. The knot interval of red edges is 10, of black edges is 1.*



(a) Ring model (b) Guitar neck model

Figure 2: *Models using a blending function such as in Figure 1b-e.*

We focus on the degree-three case, although the concepts should extend to other degrees. Our discussion assumes that all control-grid faces are four sided; if not initially, apply a single NURSS refinement [Sederberg et al. 1998]. We convey knot information by assigning a knot interval to each edge of the control grid [Sederberg et al. 1998; Sederberg et al. 2003b]. In Figure 3, d_i and e_i are knot intervals and can be any non-negative real numbers.

2 Refinement Rules

We define a CCNURBS to be a subdivision surface whose control mesh edges have knot intervals, and whose refinement equations specialize to Catmull-Clark surfaces when all knot intervals are the same and to NURBS when the valence is four and when, in Figure 3, $d_i = \tilde{d}_i = \bar{d}_i$, and $e_i = \tilde{e}_i = \bar{e}_i$, $i = 0, 1, 2, 3$.

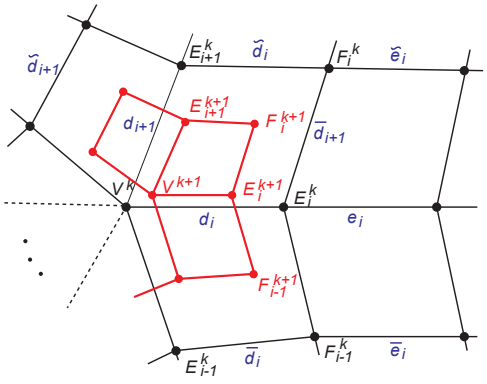


Figure 3: *Refinement Rules.*

Refinement rules for CCNURBS amount to computing face, edge, and vertex points. Figure 3 shows labels for a valence n vertex V^k with neighboring face points F^k_i and edge points E^k_i, $i = 0, \ldots, n-1$, with $V^k, F^k_i, E^k_i \in \mathbb{R}^3$. The subscripts are modulo n, the superscript k denotes refinement level, and d_i, $\tilde{d}_i$, $\bar{d}_i$, e_i, $\tilde{e}_i$, and $\bar{e}_i$ are knot intervals. We will refer to edges of a control mesh that meet at an extraordinary point as *spoke edges*, and the image of a spoke edge on the limit surface as a *spoke curve*.

Defining a $(2n+1) \times 3$ matrix

$$\mathbf{P}^k = [F^k_0 \ldots, F^k_{n-1}, E^k_0, \ldots, E^k_{n-1}, V^k]^T, \tag{1}$$

refinement can be written $\mathbf{P}^{k+1} = M^k \mathbf{P}^k$ where M^k is a $(2n+1) \times (2n+1)$ stochastic matrix whose elements M^k_{ij} are functions of knot intervals.

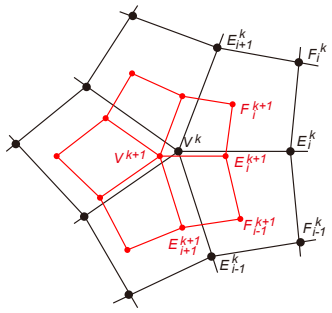


Figure 4: *Catmull-Clark subdivision rules.*

Referring to Figure 4, the Catmull-Clark refinement rules are:

$$F^{k+1}_i = \frac{V^k + E^k_i + E^k_{i+1} + F^k_i}{4}, \tag{2}$$

$$E^{k+1}_i = \frac{F^{k+1}_i + V^k + F^{k+1}_{i-1} + E^k_i}{4}, \tag{3}$$

$$V^{k+1} = \frac{\sum_{i=0}^{n-1}(F^{k+1}_i + E^k_i + V^k)}{n^2} + \frac{(n-3)V^k}{n}. \tag{4}$$

Refinement for the NURBS case amounts to conventional B-spline knot insertion: for both knot vectors, a knot is inserted midway between each pair of existing knots. Assuming for simplicity that $d_i = e_i$, $i = 0, 1, 2, 3$ (see Figure 5; this will always be the case after one refinement), the refinement rules for NURBS are

$$F^{k+1}_i = \frac{9d_i d_{i+1} V^k + (d_{i+1} + 2d_{i-1})(d_i + 2d_{i+2})F^k_i}{4(2d_i + d_{i+2})(d_{i-1} + 2d_{i+1})} + \frac{3d_{i+1}(d_i + 2d_{i+2})E^k_i + 3d_i(d_{i+1} + 2d_{i-1})E^k_{i+1}}{4(2d_i + d_{i+2})(d_{i-1} + 2d_{i+1})} \tag{5}$$

$$E^{k+1}_i = \frac{d_{i-1}F^{k+1}_i + d_{i+1}F^{k+1}_{i-1} + (d_{i-1} + d_{i+1})H^k_i}{2(d_{i+1} + d_{i-1})}, \tag{6}$$

$$V^{k+1} = \frac{V^k}{4} + \frac{\sum_{i=0}^{3}(h_i H^k_i + f_i F^{k+1}_i)}{4(d_0 + d_2)(d_1 + d_3)}, \tag{7}$$

where $f_i = d_{i-1}d_{i+2}$, $h_i = d_{i+2}(d_{i-1} + d_{i+1})$, and

$$H^k_i = \frac{3d_i V^k + (2d_{i+2} + d_i)E^k_i}{2(d_{i+2} + 2d_i)}.$$

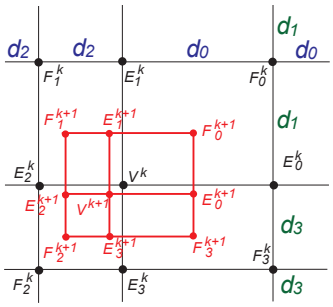


Figure 5: *NURBS Refinement.*

For the CCNURBS formulation in [Sederberg et al. 1998], any edge can be assigned any knot interval. This freedom enables the creation of local creases and darts. However, even away from extraordinary points, the surfaces are only G^1. Furthermore, the refinement matrices change at each iteration ($M^{k+1} \neq M^k$), so eigenvectors and eigenvalues are not defined.

In this paper—as well as in [Sederberg et al. 2003a], [Cashman et al. 2009], and [Kovacs et al. 2015]—we require the knot intervals on opposing edges of every face to be identical, so in Figure 3, $d_i = \tilde{d}_i = \bar{d}_i$, and $e_i = \tilde{e}_i = \bar{e}_i$, $i = 0, 1, \ldots$. This constraint causes no loss of design freedom, because local features can be added using T-junctions [Sederberg et al. 2003a]. Furthermore, two important advantages arise: the surfaces are C^2 NURBS away from extraordinary points, and the refinement matrix is stationary—$M^i = M, i = 1, \ldots, \infty$. In the remainder of this paper, we will thus simply write the refinement as

$$\mathbf{P}^{k+1} = M\mathbf{P}^k. \tag{8}$$

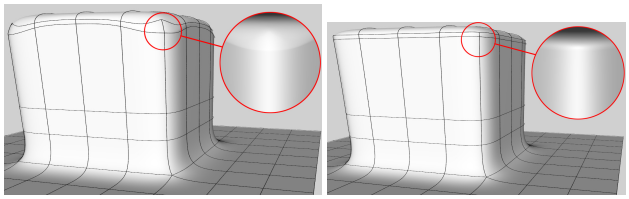
3 Prior Work

The ugly behavior illustrated in Figure 1 that often arises in CC-NURBS limit surfaces has attracted research for over a decade. [Cashman et al. 2009] proposes a strategy of minimizing the difference in knot intervals for spoke edges at an extraordinary point. It does so by performing a preprocess of repeatedly splitting the largest knot interval at an extraordinary point if it is more than twice as large as the smallest knot interval at that extraordinary point, and then performing a local refinement to make all knot intervals the same. While this provides improvement in some cases, Figure 1.c shows that the strategy does not work universally.

[Kovacs et al. 2015] modifies the CCNURBS refinement rules in [Sederberg et al. 1998], yielding improved surfaces in some cases, although Figure 1.d shows that wrinkles remain.

Another method that has been explored for dealing with this problem is to modify the "patch" method in [Peters 2000] to handle differing knot intervals. This method produces one Bézier patch (or a small number of patches) for each face of the control mesh, and is currently used commercially in the Autodesk T-Splines Plugin for Rhino, as well as in the Autodesk Inventor and Autodesk Fusion products [T-Splines 2016]. The patches are forced to be G^1 by satisfying algebraic constraints called connecting functions. While it is straightforward to modify the connecting functions to allow different knots, achieving good-looking results has proven just as elusive for patching as it has for subdivision. The reason is that no theoretically sound technique has been found for computing the position and normal for the extraordinary point and tangent vectors for spoke curves, since these are usually obtained from subdivision surfaces. While the resulting surfaces are mathematically G^1, the best implementation we know of—currently used in the Autodesk T-Splines Plugin for Rhino—often produces ugly results similar to those in Figure 1. Figure 1.e shows one such example.

Other prior art that aims to unify cubic NURBS and Catmull-Clark surfaces into a single representation includes Extended Subdivision Surfaces [Müller et al. 2006] and Dinus [Müller et al. 2010]. These schemes handle extraordinary points by reverting to Catmull-Clark rules, ignoring the actual knot intervals. While this assures smooth blending functions, the abrupt transition between the uniform extraordinary point and its nonuniform neighbors leads to undesirable results such as illustrated in Figure 6.



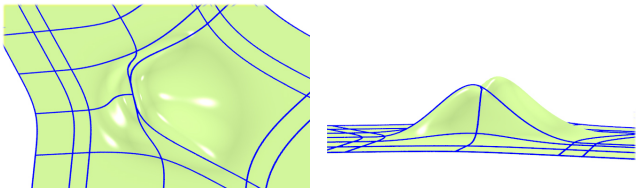
(a) Müller et al. 2006 (b) Method in this paper

Figure 6: *Example involving Müller et al. 2006.*

4 Eigen Polyhedra

In the examples in Figure 1, a single control point moved out of plane produces a limit surface that has two local maxima. We now look more closely at this curious phenomenon.

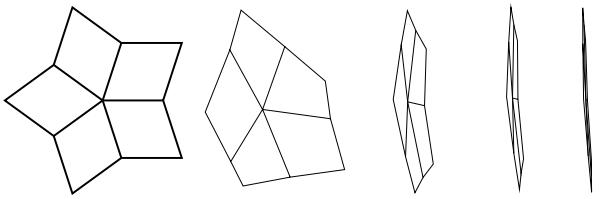
Figure 7 shows the spoke curves for the surface in Figure 1.b. Notice that the G^0 extraordinary point lies between the two local maxima, and that the angle between two pairs of adjacent spoke curves appears to be zero. The reason this happens can be understood by examining the first neighborhood control grid faces after repeated



(a) Top view (b) Perspective view

Figure 7: *Spoke curves for surface in Figure 1.b*

refinement. Figure 8 shows how those faces become more narrow with each iteration, causing the angles between some edges to approach zero upon refinement. This observation suggests that the



(a) 0 (b) 1 (c) 3 (d) 5 (e) 7

Figure 8: *First neighborhood of extraordinary point after refinements, for example in Figure 1.b. Subcaption is number of refinements. Each subfigure is enlarged so their heights are similar.*

ugly behavior in Figure 1 and Figure 7 might be eliminated if refinement rules could be devised that avoid the collapsing of faces in Figure 8. This line of thinking led to the notion of eigen polyhedra, which are meshes that lie in the x–y plane. Denote by

$$\hat{\mathbf{P}}^k = [\hat{F}_0^k \ldots, \hat{F}_{n-1}^k, \hat{E}_0^k, \ldots, \hat{E}_{n-1}^k, \hat{V}^k]^T, \tag{9}$$

a $(2n+1) \times 2$ matrix where $\hat{V}^k, \hat{F}_i^k, \hat{E}_i^k \in \mathbb{R}^2$. We will use the expression "polyhedron $\hat{\mathbf{P}}^k$" to mean the polyhedron in $\mathbb{R}^2$ whose vertices are stored in a matrix $\hat{\mathbf{P}}^k$ and whose topology is illustrated in Figure 11.a.

Definition 1. *Polyhedron* $\hat{\mathbf{P}}^0$ *is an* **eigen polyhedron** *of* M *if*

$$\hat{\mathbf{P}}^1 = M\hat{\mathbf{P}}^0 \equiv \lambda\hat{\mathbf{P}}^0 + \mathbf{I}\hat{T}^0 \tag{10}$$

where polyhedron $\hat{\mathbf{P}}^0$ *has* $\hat{V}^0 = (0,0)$, M *is a* $(2n+1)\times(2n+1)$ *matrix whose rows sum to one,* $\lambda \in \mathbb{R}^1$, $\hat{T}^0 \in \mathbb{R}^2$, *and* $\mathbf{I}$ *is a* $(2n+1) \times 1$ *vector of 1's.*

In words, $M\hat{\mathbf{P}}^0$ produces a scale of $\hat{\mathbf{P}}^0$ by a factor of λ, followed by a translation by $\hat{T}^0$. As examples, we now describe eigen polyhedra for Catmull-Clark and for NURBS refinement matrices.

Catmull-Clark eigen polyhedron. A Catmull-Clark refinement matrix can be constructed from the refinement equations (2), (3), and (4). This M has an eigen polyhedron with vertices

$$\hat{V}^0 = (0,0) \tag{11}$$

$$\hat{E}_i^0 = (\cos(\frac{2i}{n}\pi), \sin(\frac{2i}{n}\pi)), \text{ and} \tag{12}$$

$$\hat{F}_i^0 = \gamma(\hat{E}_i^0 + \hat{E}_{i+1}^0) \tag{13}$$

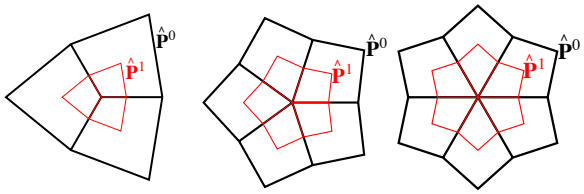
where

$$\gamma = \frac{4}{c_n + 1 + \sqrt{(c_n+9)(c_n+1)}} \tag{14}$$

and $c_n = \cos(\frac{2\pi}{n})$. In this case,

$$\lambda = \frac{1+\gamma}{4\gamma} = \frac{5 + c_n + \sqrt{(c_n+9)(c_n+1)}}{16} \tag{15}$$

and $\hat{T}^0 = (0,0)$.



(a) Valence 3 (b) Valence 5 (c) Valence 6

Figure 9: *Catmull-Clark Eigen Polyhedra.*

NURBS eigen polyhedron. The NURBS refinement matrix, constructed from (5), (6), and (7), has an eigen polyhedron with vertices

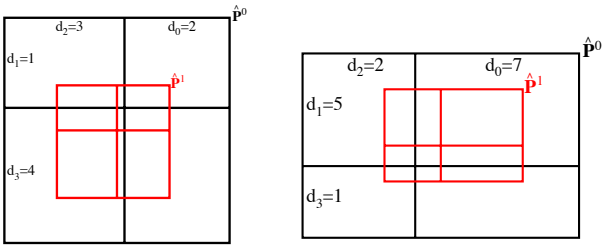
$$\hat{V}^0 = (0,0) \tag{16}$$

$$\hat{E}_i^0 = \frac{2d_i + d_{i+2}}{3}(\cos(\frac{i}{2}\pi), \sin(\frac{i}{2}\pi)), \text{ and} \tag{17}$$

$$\hat{F}_i^0 = \hat{E}_i^0 + \hat{E}_{i+1}^0. \tag{18}$$

In this case, $\lambda = \frac{1}{2}$. Since $\hat{V}^0 = (0,0)$, $\hat{T}^0 = \hat{V}^1$ is obtained by substituting (16), (17) and (18) into (7) to get

$$\hat{T}^0 = \left(\frac{d_0 - d_2}{6}, \frac{d_1 - d_3}{6}\right). \tag{19}$$



(a) Example 1 (b) Example 2

Figure 10: *Examples of NURBS Eigen Polyhedra.*

Properties of eigen polyhedra. Since the rows of M each sum to 1, $M(\mathbf{I}\hat{T}^0) = \mathbf{I}\hat{T}^0$. So, if $\hat{\mathbf{P}}^0$ and M satisfy (10),

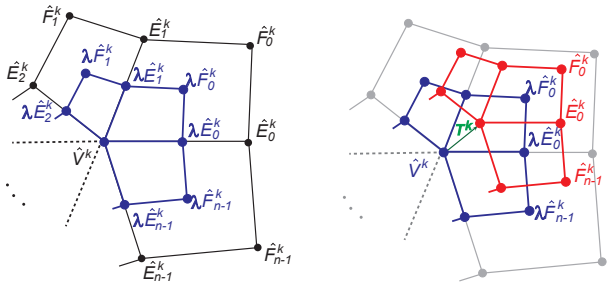
$$\begin{aligned}\hat{\mathbf{P}}^2 &= M\hat{\mathbf{P}}^1 = M(\lambda\hat{\mathbf{P}}^0 + \mathbf{I}\hat{T}^0) = \lambda M\hat{\mathbf{P}}^0 + \mathbf{I}\hat{T}^0 \\ &= \lambda^2\hat{\mathbf{P}}^0 + (1+\lambda)\mathbf{I}\hat{T}^0\end{aligned}$$

By induction,

$$\hat{\mathbf{P}}^k = \lambda^k\hat{\mathbf{P}}^0 + (1 + \lambda^1 + \ldots + \lambda^{k-1})\mathbf{I}\hat{T}^0. \tag{20}$$

From (9), the last row of $\hat{\mathbf{P}}^k$ is $\hat{V}^k$. Since $\hat{V}^0 = (0,0)$,

$$\hat{V}^k = (1+\lambda^1+\ldots+\lambda^{k-1})\hat{T}^0 = (1+\lambda^1+\ldots+\lambda^{k-1})\hat{V}^1 \tag{21}$$



(a) Scale polyhedron $\hat{\mathbf{P}}^k$ by λ with respect to V^k (b) Translate by $\hat{T}^k$

Figure 11: *Relationship between $\hat{\mathbf{P}}^k$ and $\hat{\mathbf{P}}^{k+1}$ in (23).*

and

$$\hat{\mathbf{P}}^k = \lambda^k\hat{\mathbf{P}}^0 + \hat{\mathbf{I}}V^k. \tag{22}$$

Denoting $\hat{T}^k = \hat{V}^{k+1} - \hat{V}^k$, we can obtain from (22) by straightforward algebraic manipulation,

$$\hat{\mathbf{P}}^{k+1} = M\hat{\mathbf{P}}^k \equiv [\lambda(\hat{\mathbf{P}}^k - \hat{V}^k\mathbf{I}V^k) + \hat{\mathbf{I}}V^k] + \mathbf{I}\hat{T}^k. \tag{23}$$

The geometric meaning of (23) is shown in Figure 11: If $\hat{\mathbf{P}}^0$ is an eigen polyhedron of M, then $M\hat{\mathbf{P}}^k$ is equivalent to a scale by λ about point $\hat{V}^k$ followed by a translation by $\hat{T}^k$. Note that $\hat{\mathbf{P}}^k - \hat{V}^k\mathbf{I}$, $k = 1, 2, \ldots$, are eigen polyhedra of M. It is straightforward to prove that $\hat{T}^{k+1} = \lambda\hat{T}^k = \lambda^k\hat{T}^0$.

Obviously, if polyhedron $\hat{\mathbf{P}}^0$ is an eigen polyhedron of M, polyhedron $\hat{\mathbf{P}}^0$ will not exhibit the non-uniform scaling behavior in Figure 8 when it is repeatedly refined by M.

If we subtract $\mathbf{I}\frac{\hat{V}^1}{1-\lambda}$ from both sides of (10), we obtain

$$M(\hat{\mathbf{P}}^0 - \mathbf{I}\frac{\hat{V}^1}{1-\lambda}) = \lambda(\hat{\mathbf{P}}^0 - \mathbf{I}\frac{\hat{V}^1}{1-\lambda}), \tag{24}$$

so the two columns of $\hat{\mathbf{P}}^0 - \mathbf{I}\frac{\hat{V}^1}{1-\lambda}$ are eigenvectors of M, each of whose eigenvalue is λ. This suggests that M will have an eigen polyhedron if M has two identical eigen values. Unfortunately, other than the Catmull-Clark and NURBS special cases, generic CCNURBS refinement matrices do not have two identical eigen values [Kovacs et al. 2015].

Eigen polyhedra are similar to what [Ball and Storry 1988] calls a natural configuration, and to the control grid of a characteristic map in [Reif 1995].

The main contribution of this paper is to use the notion of an eigen polyhedron to create CCNURBS refinement rules for which M has an eigen polyhedron, and thus has two identical eigenvalues.

In Section 5 we discuss how to *design* an eigen polyhedron $\hat{\mathbf{P}}^0$ *a priori*, for any set of knot intervals and any valence—without first knowing M. We then show in Section 6 how to construct a refinement matrix for which $\hat{\mathbf{P}}^0$ is an eigen polyhedron. This procedure creates CCNURBS refinement rules that produce a G^1 surface and whose blending functions have a single local maximum.

5 Creating an Eigen Polyhedron

We now explain how to design an eigen polyhedron for a valence-n extraordinary point whose edges have knot interval values

$d_0, \ldots, d_{n-1}$. We set $\hat{V}^0 = (0,0)$, and express the points $\hat{E}_i^0$ and $\hat{F}_i^0$, $i = 0, \ldots, n-1$ as equations that are functions of n and $d_0, \ldots, d_{n-1}$. We have some freedom in creating those equations, the only rigid requirements being that they must reduce to (11)–(13) when $d_0 = \ldots = d_{n-1}$ (the Catmull-Clark case) and to (16)–(18) when $n = 4$ (the NURBS case).

We first consider the angles between spoke edges in the eigen polyhedron, $\theta_i = \angle E_i^0 V^0 E_{i+1}^0$. Since all angles are the same in Catmull-Clark and NURBS eigen polyhedra, i.e.,

$$\theta_i = \frac{2\pi}{n}, \quad i = 0, \ldots, n-1, \tag{25}$$

we propose using this equal-angle formula for all eigen polyhedra.

The length of the spoke edges for NURBS eigen polyhedra are functions of knot intervals. We similarly define the length l_i of spoke edge $V^0 E_i^0$ to be

$$l_i = \frac{d_i + d_i^- + d_i^+}{3}$$

where

$$\begin{aligned} d_i^+ &= \sum_{j=i}^{i+n-1} d_i \cos(\theta_{i,j}), \quad \text{if } \cos(\theta_{i,j}) > 0 \\ d_i^- &= -\sum_{j=i}^{i+n-1} d_i \cos(\theta_{i,j}), \text{ if } \cos(\theta_{i,j}) < 0 \\ \theta_{i,j} &= \sum_{k=i}^{j-1} \theta_k, \quad i < j \end{aligned}$$

These lengths specialize to both NURBS and Catmull-Clark eigen polyhedron spoke-edge lengths.

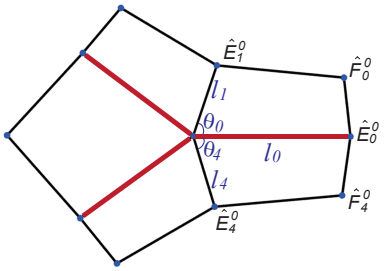


Figure 12: *Eigen polyhedron for example in Figure 1.d. Red edges have knot intervals of 10, black edges gave knot intervals of 1.*

We now define $\hat{E}_0^0 = (l_0, 0)$ to lie on the x-axis, so $\theta_{0,i}$ is the angle between spoke edge $\hat{V}^0 \hat{E}_i^0$ and the positive x-axis. Then,

$$\hat{E}_i^0 = l_i(\cos(\theta_{0,i}), \sin(\theta_{0,i})). \tag{26}$$

The points $\hat{F}_i^0$ are obtained using equation (13). The eigen polyhedron for the surface in Figure 1.d is shown in Figure 12.

6 Deriving M from an Eigen Polyhedron

Denoting a polyhedron created using the procedure in Section 5 by $\hat{\mathbf{P}}^0$, we now discuss how to create a refinement matrix M for which $\hat{\mathbf{P}}^0$ is an eigen polyhedron. We have three requirements:

1. M must satisfy (10)
2. M must specialize to Catmull-Clark refinement if the knot intervals are all equal
3. M must specialize to NURBS refinement if the valence is four and the knot intervals are as in Figure 5.

Since the rows of M express the face, edge, and vertex point computations, defining face, edge, and vertex point rules that satisfy these three requirements is equivalent to creating the desired M.

Equation (10) involves λ and $\hat{T}^0$, so we begin by finding equations for λ and $\hat{T}^0$ that specialize to the NURBS and Catmull-Clark cases. Equation (15) for computing λ meets this requirement.

Vertex point rule

Observe from (21) that $\hat{T}^0 = \hat{V}^1$. This means that our equation for $\hat{T}^0$ must not only specialize to $(0,0)$ in the Catmull-Clark case and to (19) in the NURBS case, but that it also will serve as our vertex point equation. These requirements are met by using (27) as the vertex point equation. (This is a minor modification of the vertex point rule from [Sederberg et al. 1998].)

$$\hat{V}^{k+1} = \frac{n-3}{n}\hat{V}^k + \frac{3}{n}\frac{\sum_{i=1}^n (m_i H_i^k + f_i G_i^k)}{\sum_{i=1}^n (m_i + f_i)} \tag{27}$$

where

$$\begin{aligned} H_i^k &= g_i \hat{E}_i^k + (1-g_i)V^k, \\ G_i^k &= g_i(1-g_{i+1})\hat{E}_i^k + g_{i+1}(1-g_i)\hat{E}_{i+1}^k \\ &\quad + g_1 g_2 F_i^k + (1-g_i)(1-g_{i+1})V^k, \\ g_i &= \frac{d_{i-2} + d_{i+2} + d_i}{d_{i-2} + d_{i+2} + 4d_i}, \\ f_i &= \prod_{j=1, j\neq i, i+1}^{n} d_j^+, \\ m_i &= f_i + f_{i-1}. \end{aligned}$$

Since $\hat{V}^0 = (0,0)$,

$$\hat{T}^0 = \hat{V}^1 = \frac{3}{n}\frac{\sum_{i=1}^n (m_i H_i^0 + f_i G_i^0)}{\sum_{i=1}^n (m_i + f_i)}. \tag{28}$$

Labels are illustrated in Figure 13.

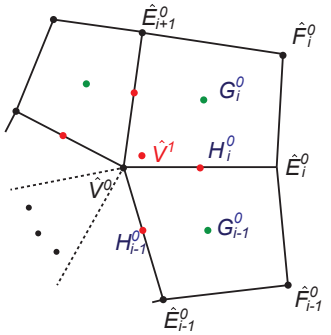


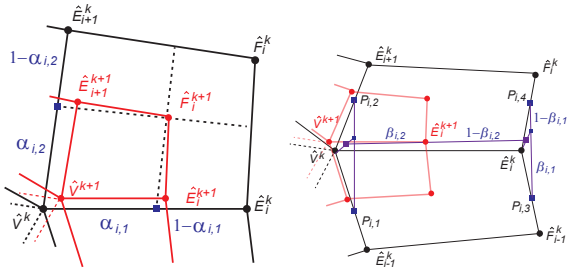
Figure 13: *Vertex point computation.*

Face point rule

From the definition of an eigen polyhedron (10), we have

$$\hat{F}_i^1 = \hat{V}^1 + \lambda \hat{F}_i^0. \tag{29}$$

At this stage of the process we know $\hat{V}^1$, λ, and $\hat{F}_i^0$, so the Cartesian coordinates of $\hat{F}_i^1$ can be computed.



(a) Face point rule (b) Edge point rule

Figure 14: *Face and edge point computation*

To devise a face point rule, we create an equation for $\hat{F}_i^1$ in terms of the four vertices of its generating face (see Figure 14.a). A reasonable way to do this is using a bi-linear equation:

$$\begin{aligned} \hat{F}_i^1 &= (1-\alpha_{i,1})(1-\alpha_{i,2})\hat{V}^0 + \alpha_{i,1}(1-\alpha_{i,2})\hat{E}_i^0 \\ &+ (1-\alpha_{i,1})\alpha_{i,2}E_{i+1}^0 + \alpha_{i,1}\alpha_{i,2}\hat{F}_i^0. \end{aligned} \tag{30}$$

The two bilinear equations in (30) can be solved via the following method from [Floater 2015]. Denote $v_1 = \hat{F}_i^1 - \hat{V}^0$, $v_2 = \hat{F}_i^1 - \hat{E}_i^0$, $v_3 = \hat{F}_i^1 - \hat{F}_i^0$, $v_4 = \hat{F}_i^1 - \hat{E}_{i+1}^0$. And let $S_i = \frac{1}{2} v_i \times v_{i+1}$, $T_i = \frac{1}{2} v_{i-1} \times v_{i+1}$, then

$$\alpha_{i,1} = \frac{2S_4}{2S_4 - T_1 + T_2 + \sqrt{D}} \tag{31}$$

$$\alpha_{i,2} = \frac{2S_1}{2S_1 - T_1 - T_2 + \sqrt{D}}, \tag{32}$$

where $D = T_1^2 + T_2^2 + 2S_1S_3 + 2S_2S_4$. Solving this for numerical values of $\alpha_{i,1}$ and $\alpha_{i,2}$, the face point rule is (30). It can be shown that (30) satisfies (10) and specializes to NURBS and Catmull-Clark.

Edge point rule

From the definition of an eigen polyhedron, we have

$$\hat{E}_i^1 = \hat{V}^1 + \lambda(\hat{E}_i^0 - \hat{V}^0) = \hat{V}^1 + \lambda \hat{E}_i^0. \tag{33}$$

Since we know $\hat{V}^1$, λ, and $\hat{E}_i^0$, the Cartesian coordinates of $\hat{E}_i^1$ can be computed.

Edge points are a linear combination of the six vertices of the two adjacent faces for the associated edge (see Figure 14.b). Inspired by the non-uniform B-spline refinement rules, we first denote

$$P_{i,1} = (1-\alpha_{i-1,1})\hat{V}^0 + \alpha_{i-1,1}\hat{E}_{i-1}^0; \tag{34}$$

$$P_{i,2} = (1-\alpha_{i,2})\hat{V}^0 + \alpha_{i,2}\hat{E}_{i+1}^0; \tag{35}$$

$$P_{i,3} = (1-\alpha_{i-1,1})\hat{E}_i^0 + \alpha_{i-1,1}\hat{F}_{i-1}^0; \tag{36}$$

$$P_{i,4} = (1-\alpha_{i,2})\hat{E}_i^0 + \alpha_{i,2}\hat{F}_i^0. \tag{37}$$

then the edge point is computed via the following equation

$$\begin{aligned} \hat{E}_i^1 =& (1-\beta_{i,2})\left(\frac{1-\beta_{i,1}}{2}P_{i,1} + \frac{\beta_{i,1}}{2}P_{i,2} + \frac{1}{2}\hat{V}^0\right) + \\ & \beta_{i,2}\left(\frac{1-\beta_{i,1}}{2}P_{i,3} + \frac{\beta_{i,1}}{2}P_{i,4} + \frac{1}{2}\hat{E}_i^0\right). \end{aligned} \tag{38}$$

It is easy to see that $\hat{E}_i^1$ is also a bi-linear combination of four points $\frac{P_{i,1}+\hat{V}^0}{2}$, $\frac{P_{i,2}+V^0}{2}$, $\frac{P_{i,3}+\hat{E}_i^0}{2}$ and $\frac{P_{i,4}+\hat{E}_i^0}{2}$ with coefficients $\beta_{i,1}$ and $\beta_{i,2}$. Thus, we can solve the coefficients using the same method as above. After solving the $\beta_{i,1}$ and $\beta_{i,2}$, the edge point rule is defined via equation (38). It can be shown that (38) satisfies (10) and specializes to NURBS and Catmull-Clark.

6.1 M in the Overall Mesh Refinement Process

While the M we have just described was created in reference to a special planar polyhedron, applying M to arbitrary control meshes in $\mathbb{R}^3$ yields excellent results. We now describe how M fits into the overall mesh refinement process. The creation of M assumes that extraordinary points are separated by more than one face. If not, perform an initial refinement using any CCNURBS formulation.

In Figure 15, the gray mesh is obtained after a single CCNURBS refinement, so the two extraordinary points are not adjacent. Refining the gray mesh yields the mesh whose vertices are either red or green. In Figure 15, the red control points are obtained by applying the appropriate M to the 1-neighborhood of each respective extraordinary point. In general, each extraordinary point has its own refinement matrix, since knot intervals will generally not be the same for each extraordinary point.

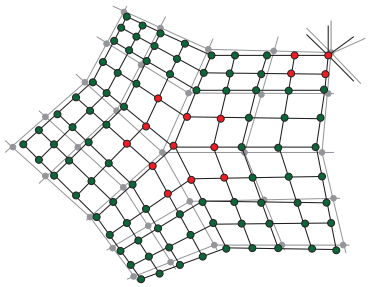
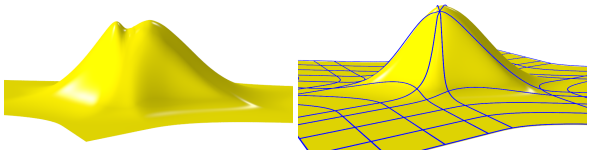


Figure 15: *Refinement in the 2-neighborhood*

The green vertices are obtained with conventional NURBS refinement rules using face, edge, and vertex point equations (5)–(7).

7 Results and Discussion

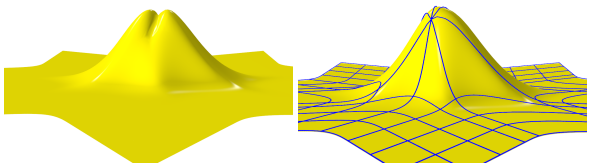
Figure 1.f and Figures 16—18 show several examples of blending functions produced by our method. Notice that the angles between spoke edges are non-zero using our method.



(a) Using Cashman et al. 2009 (b) Using Our Method

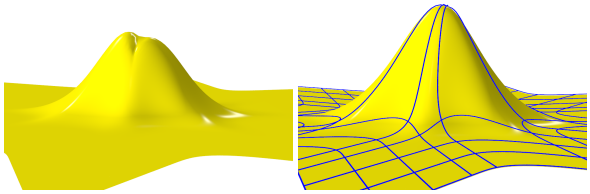
Figure 16: *Valence 6, with knot intervals 1, 10, 10, 1, 1, 10.*

For smaller knot interval ratios, wrinkles are more minor but the surface imperfection is still evident using zebra stripes, as illus-



(a) Using Sederberg et al. 1998 (b) Using Our Method

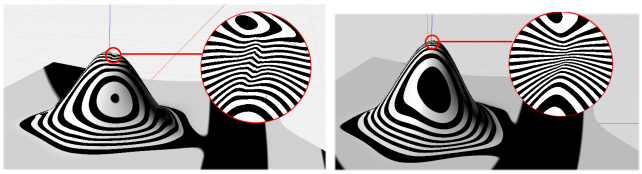
Figure 17: *Valence 7, with knot intervals 1, 8, 8, 1, 1, 1, 8.*



(a) Using Kovacs et al. 2015 (b) Using Our Method

Figure 18: *Valence 8, with knot intervals 1, 5, 5, 1, 1, 1, 5, 1.*

trated in Figure 19.a. Our method smooths out the kinked zebra stripes.



(a) Using Kovacs et al. 2015 (b) Using Our Method

Figure 19: *Zebra Stripes for valence 5 blending function with knot intervals 1,3,1,3,3.*

Improved blending functions lead to improved models. Figures 20 and 21 show the result of applying our refinement method to the model in Figure 2.a and Figure 22 shows our method applied to the guitar neck model from Figure 2.b.

We did not study valences greater than eight because high valence extraordinary points are rarely used in practice. However, the study of how well the method works for higher valences is of mathematical interest and would be worth exploring.

In the examples shown, the extraordinary point blending functions have a single local maximum and are G^1. A good problem for future study is to prove whether this is true for any choice of knot intervals and valance. We do not have an analytical proof of this, but we did test a million different extraordinary points with randomly generated knot intervals $\in [10^{-6}, 1]$ and valences of $n = 3, 5, 6, 7$ and 8 and found that, in every case, the blending function had a single local maximum and was G^1.

To verify the existence of a single local maximum, we performed five levels of refinement on each test case and confirmed that the resulting control mesh had a single vertex whose z-coordinate was larger than all neighbors. We believe that five refinements is sufficient because we have observed that the twin peak phenomenon usually manifests itself in the control grid of the second refinement.

An extraordinary point is tangent continuous if the characteristic ring is regular and injective [Peters and Reif 2008]. To verify regularity and injectivity, we need to examine the characteristic map defined by three rings of control points. To determine those $12n+1$ control points, form the $(12n+1)\times(12n+1)$ refinement matrix for those points, and compute the second and third eigenvectors. The

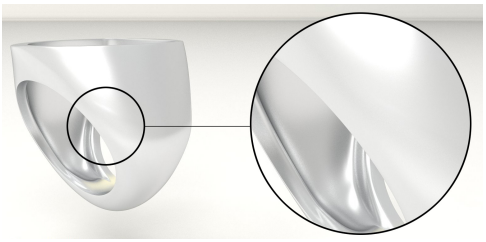
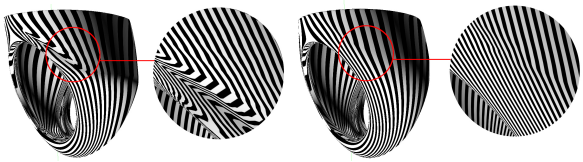


Figure 20: *Ring model from Figure 2.a using our method.*



(a) Using Kovacs et al. 2015 (b) Using Our Method

Figure 21: *Zebra Stripes for Ring Model in Figure 2.a.*

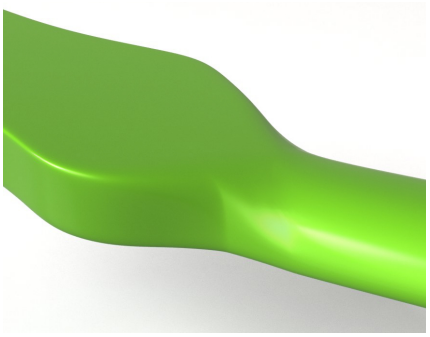
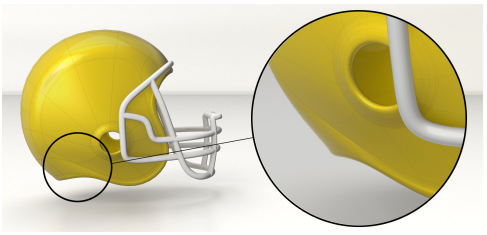
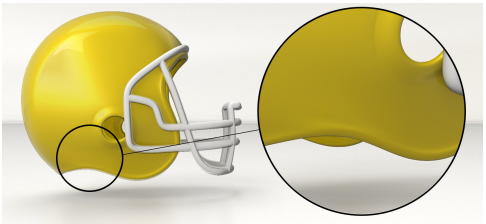


Figure 22: *Guitar neck model from Figure 2.b using our method.*



(a) Using Sederberg et al. 1998



(b) Using the method in this paper

Figure 23: *Helmet model.*

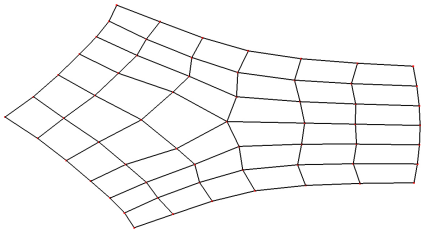


Figure 24: *Characteristic Map Control Points for the Example in Figure 12*

x-coordinates of the points are the values in the second eigenvector, and the y-coordinates in the third eigenvector. Figure 24 shows three rings of control points for the characteristic map for the eigen polyhedron in Figure 12. We verified regularity and injectivity by subdividing the control mesh of the characteristic map several times and performing numerical tests to confirm that the determinant of the Jacobian matrix does not change sign and that there are no non-local intersections.

In all million test cases, the first ring of control points for the characteristic map is a translation of the eigen polyhedron. In other words, the value of λ used in creating the eigen polyhedron always turned out to be the second and third eigenvalue of M. We have no mathematical proof that this will always be the case, and suggest that this is an interesting problem for future research. There exists some related literature on this general topic under the name "inverse eigenvalue problem."

In Section 5, equation (25) defines the θ_i for our eigen polyhedra to all be equal. We experimented with using different values of θ_i for cases where $n \neq 4$ and the d_i are not equal and found that minor changes in surface quality can occur. For example, if all knot intervals are the same except for one knot interval of zero, we observed slightly improved surface quality when angles next to the zero-knot-interval edge are 90° and the other angles are the same. While our preliminary results did not seem significant enough to report in this paper, this is worth studying further.

Another topic that invites future research is zero knot intervals. The algorithm requires a modification to handle a zero knot interval: splitting a zero knot interval produces two zero knot intervals, so one should be removed. Things are more complicated if several spoke edges have zero knot intervals because two adjacent zero knot intervals create a crease so there is not a unique tangent plane. In such cases, our method cannot be applied to the entire one-neighborhood, but could be applied piecewise to each domain bounded by creased spoke edges. There are numerous such cases to consider, and more theory to work out.

A closed-form equation for the Cartesian coordinates of the extraordinary point, as well as for tangent vectors for the spoke curves, can be developed from the eigen vectors. This would be helpful in performing exact evaluation of extraordinary limit points, and in developing an improved patching solution for non-uniform Catmull-Clark surfaces.

Acknowledgements

We are indebted to the reviewers, whose helpful suggestions greatly strengthened the paper. The models in this paper were created and rendered by Juan Santocono.

The first author was supported by the NKBRPC (2011CB302400), SRF for ROCS SE, and the Youth Innovation Promotion Association CAS.

References

BALL, A. A., AND STORRY, D. J. 1988. Conditions for tangent plane continuity over recursively generated b-spline surfaces. *ACM Transactions on Graphics (TOG) 7*, 2, 83–102.

BAZILEVS, Y., CALO, V. M., COTTRELL, J. A., EVANS, J. A., HUGHES, T. J. R., LIPTON, S., SCOTT, M. A., AND SEDERBERG, T. W. 2010. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering 199*, 5-8, 229 – 263.

CASHMAN, T. J., AUGSDÖRFER, U. H., DODGSON, N. A., AND SABIN, M. A. 2009. NURBS with Extraordinary Points: High-degree, Non-uniform, Rational Subdivision Schemes. *ACM Transactions on Graphics 28*, 3, 1–9.

CASHMAN, T. J. 2010. NURBS-compatible subdivision surfaces. Tech. rep., University of Cambridge.

FLOATER, M. S. 2015. The inverse of a rational bilinear mapping. *Computer Aided Geometric Design 33*, 46–50.

KOVACS, D., BISCEGLIO, J., AND ZORIN, D. 2015. Dyadic T-mesh subdivision. *ACM Transactions on Graphics (TOG) 34*, 4, 143.

MÜLLER, K., REUSCHE, L., AND FELLNER, D. 2006. Extended subdivision surfaces: Building a bridge between NURBS and Catmull-Clark surfaces. *ACM Transactions on Graphics (TOG) 25*, 2, 268–292.

MÜLLER, K., FÜNFZIG, C., REUSCHE, L., HANSFORD, D., FARIN, G., AND HAGEN, H. 2010. Dinus: Double insertion, nonuniform, stationary subdivision surfaces. *ACM Transactions on Graphics (TOG) 29*, 3, 1–21.

PETERS, J., AND REIF, U. 2008. *Subdivision surfaces*. Springer.

PETERS, J. 2000. Patching Catmull-Clark meshes. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 255–258.

REIF, U. 1995. A unified approach to subdivision algorithms near extraordinary vertices. *Computer Aided Geometry Design 12*, 153–174.

SEDERBERG, T. W., ZHENG, J., SEWELL, D., AND SABIN, M. 1998. Non-uniform recursive subdivision surfaces. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '98, ACM, 387–394.

SEDERBERG, T. W., ZHENG, J., BAKENOV, A., AND NASRI, A. 2003. T-splines and T-NURCCSs. *ACM Transactions on Graphics 22 (3)*, 477–484.

SEDERBERG, T. W., ZHENG, J., AND SONG, X. 2003. Knot intervals and multi-degree splines. *Computer Aided Geometric Design 20*, 7, 455–468.

T-SPLINES FOR RHINO3D PLUGIN DEVELOPERS, 2016. Implementation of extraordinary points. Private conversation.