

Dyadic T-mesh Subdivision

Denis Kovacs^{1,2*}

Justin Bisceglia^{1,3†}

Denis Zorin^{1‡}

¹New York University ²FiftyThree ³Blue Sky Studios

Abstract

Meshes with T-joints (T-meshes) and related high-order surfaces have many advantages in situations where flexible local refinement is needed. At the same time, designing subdivision rules and bases for T-meshes is much more difficult, and fewer options are available. For common geometric modeling tasks it is desirable to retain the simplicity and flexibility of commonly used subdivision surfaces, and extend them to handle T-meshes.

We propose a subdivision scheme extending Catmull-Clark and NURSS to a special class of quad T-meshes, dyadic T-meshes, which have no more than one T-joint per edge. Our scheme is based on a factorization with the same structure as Catmull-Clark subdivision. On regular T-meshes it is a refinement scheme for a subset of standard T-splines. While we use more variations of subdivision masks compared to Catmull-Clark and NURSS, the minimal size of the stencil is maintained, and all variations in formulas are due to simple changes in coefficients.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—[Geometric algorithms, languages, and systems];

Keywords: Subdivision surfaces, T-meshes, T-splines

1 Introduction

Subdivision surfaces are widely used to create free-form 3D shapes for computer animation and are a popular tool for conceptual design. The most obvious reason for this is that subdivision surfaces support flexible and easy-to-modify control meshes of arbitrary topology and the predominant quad mesh structure is suitable for many modeling needs.

Examination of common organic and mechanical models reveals that subdivision surface modeling often involves two tasks:

- Varying control mesh resolution between more and less feature-rich areas.
- Modifying connectivity to align features with control mesh edges.

In practice, these challenges are resolved using extraordinary vertices, most commonly in pairs of valence 3 and 5, redirecting the edge flow (Figure 1).

For commonly used schemes, the quality at the extraordinary vertices in general cannot match the surface quality of the regular parts of the surface, where it reduces to polynomial patches (Figure 5).

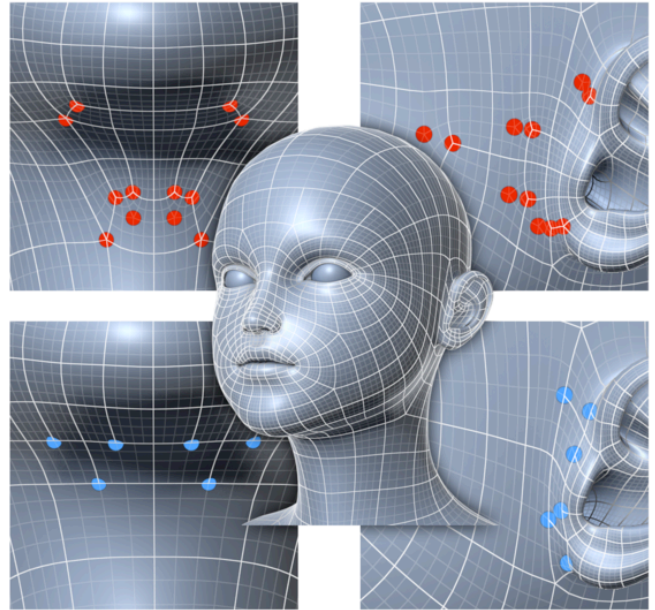


Figure 1: Top: T-joints (blue) allow for local refinement without extraordinary vertices (red). Bottom: successive subdivision turns each source T-face into mostly regular quads and a row of T-faces along the original T-edge.

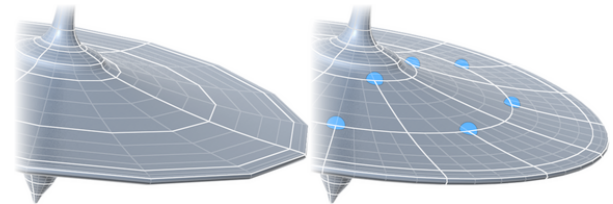


Figure 2: T-joints can be used to keep tessellated faces roughly equal sized and thus prevent under- or over-tessellation.

*e-mail: kovacs@cs.nyu.edu

†e-mail: justinb@blueskystudios.com

‡e-mail: dzorin@cs.nyu.edu

T-joints significantly increase the freedom of choosing control mesh connectivity, without creating lower quality surfaces. In a restricted form produced by adaptive hierarchical refinement, these were available in subdivision surface context for a long time, both in literature [Zorin et al. 1997] and in commercial tools (e.g., Autodesk Maya). A more flexible representation based on T-splines is the primary free-form representation in Autodesk Fusion 360.

A significant body of literature has appeared on construction of smooth surfaces with T-joints, with T-splines being the most common construction. Yet, to the best of our knowledge, no subdivision schemes were proposed that operate directly on meshes with T-joints (T-meshes). The closest approach is a relatively complex T-NURCCs scheme [Sederberg et al. 2003], which has been devel-

oped to allow local refinement of non-uniform Catmull-clark surfaces (without T-joints) by successively introducing special T-joint configurations.

Part of the difficulty in integrating T-joints with subdivision is that T-meshes have a far richer space of local connectivity configurations compared to meshes with no T-joints. Handling these configurations is a challenge, even for subdivision rules involving few points. Furthermore, only very specific classes of T-meshes lead to nested spline spaces, a prerequisite for subdivision.

In this paper, we propose a subdivision scheme extending Catmull-Clark subdivision to *dyadic* T-meshes, with at most one T-joint per quad edge. The complexity of the algorithm is only moderately higher than the complexity of adaptive subdivision (e.g., [Zorin et al. 1997]; at the same time, it considerably extends the variety of possible local and global mesh connectivities (Figures 3, 4)).

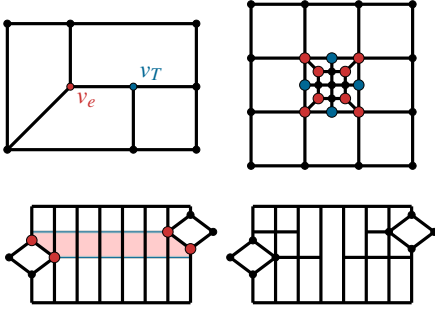


Figure 3: Examples of useful connectivities our scheme can handle: Top left: a T-joint labeled v_T next to an extraordinary vertex v_e . Top right: local refinement around extraordinary vertices due to extrusion. Bottom: extraordinary vertices due to feature vertices, resulting in long and thin quad strips without T-joints.

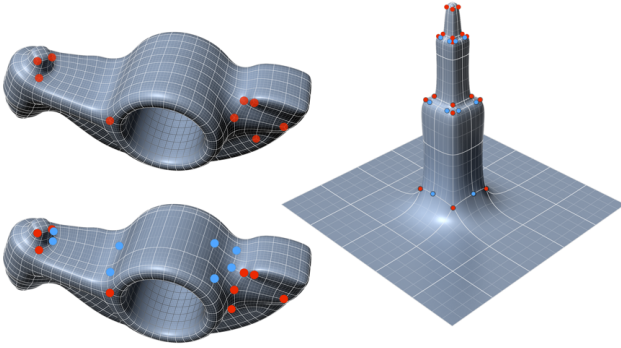


Figure 4: Realizations of the example diagrams of Figure 3. Left: a remeshed model of the rockerarm has long quad strips between extraordinary vertices. Inserting T-joints allows for simplification of the quad strip. Right: local refinement on three steps of mesh extrusion.

Our main insight is that a subdivision scheme handling all these configurations for the class of meshes that we consider can be obtained in *factorized* form, based on the original Catmull-Clark and NURCC ideas, and only a small number of cases need to be handled.

For regular T-grids, our scheme reproduces analysis-suitable T-Splines exactly. The correctness of the weight formulas was proved using a symbolic verification algorithm for all possible local T-subgrid topologies: since DAS T-meshes define nested analysis-suitable T-spline spaces, symbolic knot insertion formulas define the (unfactored) ground truth formulas for all weights involved, which then can be test these formulas symbolically against the rules defined in this paper for equality.

We demonstrate that our scheme yields surfaces of good quality for a variety of local configurations including extraordinary vertices and T-joints (in fact, extraordinary vertices themselves can be T-joints) as shown in Figure 18, while treating the whole surface in a uniform way and imposing no restrictions on proximity of T-joints and extraordinary vertices.

We conjecture that our surfaces are G^1 at extraordinary points based on the numerical evidence, although a complete rigorous analysis is beyond the scope of this paper.

We verified the practicality of our algorithm by implementing a plug-in prototype for Autodesk’s Maya, together with a set of Maya Python scripts to aid the artist in the creation and manipulation of T-meshes, which we will make publicly available. We also include the MATLAB code for computing the subdivision masks as an electronic supplement.

2 Related work

The literature on subdivision surfaces is broad, but only a few works touch on the question of T-joints. Similarly, much work has been done on T-splines and related T-mesh constructions, but with relatively little focus on subdivision schemes.

We refer the reader to a recent survey [Cashman 2012] for a general overview of recent work on subdivision; here we focus on the most closely related work on subdivision schemes allowing nonuniform knot spacing on the one hand, and schemes for constructing smooth surfaces on T-meshes on the other hand.

Nonuniform subdivision schemes. Our work is based on one of the earliest schemes for extending nonuniform splines to arbitrary meshes [Sederberg et al. 1998] (NURSS), and its restricted version described in [Sederberg et al. 2003] (NURCC). In this work, the original factorized form of surface subdivision [Catmull and Clark 1978] is extended to arbitrary knot intervals. We show how to apply this factorization in the context of T-meshes. Alternative approaches to constructing subdivision surfaces with nonuniform knots are proposed in [Müller et al. 2006], [Müller et al. 2010], and [Cashman et al. 2009a]. [Müller et al. 2006] presents a scheme resulting in stationary subdivision matrices near extraordinary vertices (which yields explicit limit point formulas) while handling arbitrary knot intervals on opposite sides of faces. The factorized form plays an important role in extending higher-order uniform B-Splines to arbitrary control meshes ([Zorin and Schröder 2001; Stam 2001]). The algorithm of [Cashman et al. 2009a], based on the factorized form of [Cashman et al. 2009b], describes a method for extending NURBS of arbitrary degree to arbitrary meshes with nonuniform knot intervals. In our work, our principal goal is *not* to handle arbitrary knots in full generality; rather, we focus on a restricted version of knot assignments, with matching knot intervals on opposite sides of faces and dyadic relations between knots, making the minimal extension to conventional subdivision that enables T-mesh refinement.

T-joints on arbitrary meshes. There are several directions of work that introduce T-joints into arbitrary triangle and quad meshes.

Adaptive refinement of basis functions is the most straightforward approach to adding fine-scale degrees of freedom to a surface. Taking a patch-based point of view, this can be considered a restricted form of T-mesh constructions. Variations of this approach were proposed in [Eck et al. 1995; Zorin et al. 1997; Lounsbery et al. 1997; Khodakovsky et al. 2000; Bertram et al. 2004] and many other works, leading to wavelet and multiscale surface representations.

[Sederberg et al. 2003] introduces *T-splines*, capable of handling a broad range of local T-configurations.

The same work also introduces *T-NURCCs*, a scheme designed to allow local refinement of meshes with extraordinary vertices, in places where more detail is needed, either to resolve the mesh shape

or to introduce more local control. It assumes a NURCC input control mesh without T-joints, and outputs a mesh with T-joints which appear as a result of the local refinement. Local refinement is split into two parts: on “suitably large” regular subgrids T-spline knot insertion is performed, while local refinement keeps the mesh sufficiently regular in two rings around the extraordinary vertex. This has two important consequences. Firstly, extraordinary vertices and T-joints have to be sufficiently far apart (separated by at least another regular vertex). This immediately excludes most examples shown in this paper (Figures 1, 4, 11, 20, 21, 22). Secondly, for DAS T-meshes that have enough separation between extraordinary vertices and T-joints, T-Spline knot insertion could be used for computing one full subdivision step (resulting in a surface identical to our scheme), but at a higher computational cost and implementation complexity, as it cannot take advantage of the factorization derived in our work; for more general T-mesh topologies, it is unclear to us if it is even possible to define a meaningful subdivision, as T-spline spaces are not guaranteed to be nested, and control point insertion can become a non-local iterative process ([Sederberg et al. 2004]).

Overall, in comparison to T-NURCCs, we opt for a different approach: rather than adding subdivision to T-splines, by creating a buffer zone around extraordinary vertices, we extend Catmull-Clark rules to a scheme that handles a number of important T-mesh configurations, without fundamentally changing the structure of the subdivision algorithm.

A C^1 polynomial basis construction for T-meshes (PHT splines) was proposed in [Deng et al. 2008], and extended to meshes with extraordinary vertices in [Li et al. 2010] (GPT splines). Due to more local basis function support, this scheme offers greater flexibility and a purely polynomial basis for T-meshes with few restrictions in the regular case, and admits simple analysis [Deng et al. 2006] (the situation with T-splines is far more complex, e.g., [Li et al. 2012; Buffa et al. 2010; Mourrain 2010]). Extraordinary faces (faces which have at least one extraordinary vertex) cannot share a T-vertex. As these bases require multiple degrees of freedom per vertex, further adaptation is needed in the context of geometric modeling. Our focus is on designing a scheme that can be easily used in the same context as Catmull-Clark subdivision is currently used.

An important recent application of T-meshes and T-splines is *isogeometric analysis* (e.g., [Cottrell et al. 2009]), i.e. methods that use the same high-order basis for geometric modeling and simulation. [Bazilevs et al. 2010] demonstrates that T-splines have substantial advantages for isogeometric analysis.

Analysis-suitable T-splines are introduced in [Li et al. 2012]; restrictions on the T-mesh structure are imposed to ensure that the resulting T-spline spaces are linearly independent. Local refinement for T-splines for analysis purposes is studied in [Scott et al. 2012; Dörfler et al. 2010]. More recent improved techniques for T-spline refinement are discussed in [Morgenstern and Peterseim 2014]. [Wang et al. 2011] proposes a method for conversion of an arbitrary quad mesh to a control mesh for an analysis-suitable T-spline, which is C^2 away from extraordinary vertices but only C^0 at some of the edges at extraordinary vertices. In FEM simulation applications, a lower order of smoothness is acceptable, as long as the approximation order is maintained; in modeling applications however it is essential to keep the surface quality high.

Recently, hierarchical splines were combined with T-splines, to have a more local but restricted scheme for refinement [Evans et al. 2014]. This work points out that single-level refinement as used in T-spline schemes is preferable for design, and hierarchical structures have advantages for simulation.

Finally, [Myles et al. 2010] describes a procedure for automatic conversion of an arbitrary mesh to a coarse T-mesh of quad patches, which often naturally have T-joints adjacent to extraordinary vertices, and require additional refinement to isolate them, as well as “nearly-aligned” extraordinary vertices shown in Figure 3, another motivation for considering constructions with no such restriction.

A more recent method for constructing T-meshes from meshes is described in [Liu et al. 2015].

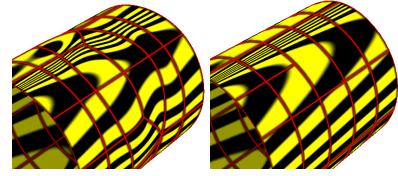


Figure 5: Local refinement on cylinder surface approximated using redirected edge flow (left) vs. T-joints (right).

3 Subdivision on Dyadic T-meshes

3.1 Notation and Definitions

We start with definitions needed to describe a T-mesh structure. A *T-joint* is a vertex in the interior of an edge. *T-meshes* consist strictly of quad faces attached along edges and may contain T-joints. Combinatorially, a T-mesh is a standard polygonal mesh where all faces are viewed as quads. Each face is defined by at least four vertices of which four vertices determine the corners of the quad and the remaining vertices are designated as T-joints. An important feature of a T-mesh is that each vertex is allowed to be a T-joint in any number of faces that share it. A *regular* T-joint vertex has valence 3 and is a T-joint with respect to a single face. We refer to the single incident edge that does not belong to that face as the *stem*.

The sequence of edges connecting two sequential corner vertices in a face defines a *T-edge*. A face is called *regular* if it has no T-joints; otherwise, it is called a *T-face*. Examples of T-joints and T-faces are shown in Figure 6. Unlike the case of conforming meshes, subdivision on T-meshes requires an implementation of knot intervals associated with mesh edges, analogous to NURBS. For regular meshes, this implementation reduces to the case of bi-cubic NURBS. We think of each quad as a rectangle in the parametric plane with knot intervals as the length of edges. Just as the lengths of opposite sides of a rectangle are equal, the knot intervals on opposing edges should be equal. The sum of edges on a T-edge should also be equal to an opposing edge. From this property, we can associate two knot intervals with the two directions of a face.

For a regular T-joint, we define a *T-joint extension* as the span of 3 knot intervals: one along the stem of the T-joint and the other two connecting the T-joint to the opposite edge of its T-face and the following face (Figure 7). The extensions of an irregular T-joint are shown in Figure 11.

In a *dyadic* T-mesh, each T-face has at most one T-joint on it, and T-edges have equal knot intervals on each sides of the T-joint. In a *dyadic analysis-suitable* (DAS) T-mesh, no two perpendicular extensions intersect (Figure 7). This is a simple sufficient requirement for the nestedness property of T-spaces needed to define subdivision [Li et al. 2012]. To summarize:

1. each face of a DAS T-mesh has at most one T-joint, splitting its T-edge into equal knot intervals;
2. the sums of knot intervals assigned to opposite sides of each face are equal;
3. perpendicular T-joint extensions never intersect.

DAS T-mesh subdivision is a primal quadrilateral 4-split [Andersson and Stewart 2010] where each iteration splits one face into four new faces. When a T-face is refined, two of the new faces are regular and the other two are T-faces (see Figure 8(Right)). Note that after refinement, T-joints become regular vertices and new T-joints appear along T-edges. The subdivision rules we define have the usual form $V' = \sum_i w_i V_i$, where the subdivision result, V' , is a linear combination of weights w_i and vertices V_i . Both V' and V_i may denote vertex positions that are either existing, newly inserted, or

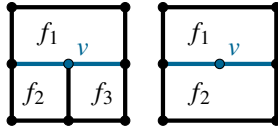


Figure 6: Left: v is a T-joint with respect to face f_1 , but a corner for f_2 and f_3 . Right: v is a T-joint with respect to both f_1 and f_2 . T-edges are shown in blue.

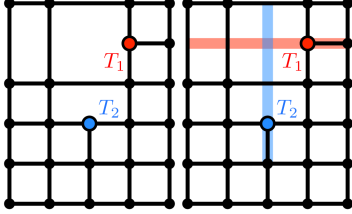


Figure 7: Left: T-joints T_1 and T_2 . Right: T-joint extensions in red and blue. Intersecting extensions imply insufficient separation between T-faces, a violation of analysis-suitable conditions.

intermediate. The weights are independent of V_i , but may depend on mesh connectivity and knot intervals. The *stencil* for a rule is a set of knot intervals and vertices used in the rule, and a *mask* is the set of weights, assigned to the vertices of the stencil.

The notion of extending a knot interval of a face across either an edge or a vertex mitigates complexity in the stencils and masks. Such extensions are denoted by dashed intervals in the figures. Hence, knot intervals are measured using edge lengths or distances across faces in the direction of a T-joint extension, as shown in Figure 8 (left and center).

The stencils in Figure 10 describe the rules for generating a refined mesh from weights and masks; we will discuss the rules in detail below.

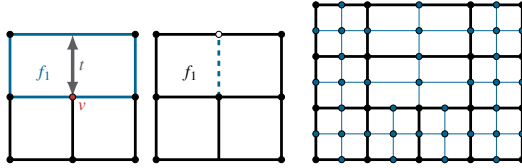


Figure 8: Left: t is the knot interval across T-face f_1 in the direction of the T-joint extension of v . Center: a dashed blue line denotes either the edge length (if it exists), or t . Right: one global refinement step of a T-mesh. The original mesh is drawn in black. Blue vertices and edges are added in refinement.

3.2 Factorized scheme on DAS T-meshes

The rules are factorized in a way similar to the common factorization of Catmull-Clark rules. Refinement of knot intervals is simple: when an edge with knot interval d is split, the two new edges of the refinement receive knot intervals $d/2$.

We define subdivision rules for face points, edge, and vertex points. Edge and vertex rules are factorized in two parts: first, intermediate points (midpoints and half-face points) are computed, which are then combined in a final rule.

Unlike the Catmull-Clark scheme, but similar to NURSS [Sederberg et al. 1998], we need more general formulas for weights, depending on the knot intervals. All rules refer to knot intervals and points shown in Figure 10. Wherever necessary, we show separate stencils for weight computations.

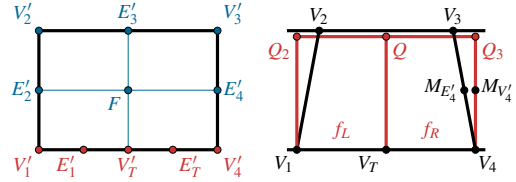


Figure 9: Left: face, edge, and vertex points generated by our scheme for a T-face. Rules for components in red use vertex points from half-faces; rules for components in blue use T-face vertices. Right: two half-faces superimposed on a T-face, with their collective vertices.

Auxiliary half-face rules. To avoid combinatorial complexity in edge and vertex rules, we create two half-faces for each T-face f_T (without discarding the original T-face), by adding points on the edge opposite to the T-joint. This splitting operation is the standard Boehm's insertion rule for one-dimensional cubic NURBS, cf. [Sederberg et al. 2003]. The analysis-suitable T-mesh topology guarantees that such local knot insertion is always possible.

Three new points Q_2, Q, Q_3 are assigned to the edge opposite the T-edge. Using the knot labeling of Figure 10(b) and the vertex labeling of Figure 9, the explicit formula for Q is:

$$Q = \frac{(s_3 + 2s_4)V_2 + (s_3 + 2s_2)V_3}{2(s_2 + s_3 + s_4)} \quad (1)$$

For Q_2 , depending on whether the face adjacent on the left is a T-face, with a T-joint on the edge continuing the T-edge of f_T or not, we have two formulas:

$$Q_2 = \frac{s_3V_L + (2s_1 + 2s_2 + s_3)V_2}{2(s_1 + s_2 + s_3)} \quad \text{with no adjacent T-joint}$$

$$Q_2 = \frac{s_3Q_L + (s_2 + s_3)V_2 + s_2Q}{2(s_2 + s_3)} \quad \text{with adjacent T-joint}$$

where Q_L is the vertex computed on $[V_L, V_2]$ according to (1). The formulas for Q_3 are obtained by reflection.

We organized the new and old points into sets that define three faces. Using Figure 9 (right), we let ζ denote the vertices of one of these faces depending on which face is used in a rule. That is,

$$\zeta = \begin{cases} \{V_T, V_1, Q_2, Q\} & \text{for } f_L \\ \{V_T, Q, Q_3, V_4\} & \text{for } f_R \\ \{V_T, V_1, V_2, V_3, V_4\} & \text{for } f_T \end{cases}$$

where f_L is the left half-face, f_R is the right half-face, and f_T is the T-face.

Face rule. The face rule is given by

$$F = Tw_TV_T + \bar{T}w_1V_1 + w_2V_2 + w_3V_3 + \bar{T}w_4V_4. \quad (2)$$

Here, $T \in \{0, 1\}$ is an indicator function and $\bar{T} = 1 - T$. The condition on all indicator functions is existence of T-joints. The stencils of Figure 10 describe where we place these conditions. The weight for each V_i is defined by a cross of knot intervals through V_i :

$$w = \frac{1}{4} \cdot \frac{s_2 + 2s_3}{s_1 + s_2 + s_3} \cdot \frac{t_2 + 2t_3}{t_1 + t_2 + t_3}. \quad (3)$$

For a T-face, the T-joint weight is

$$w_T = \frac{1}{2} \left(\frac{t_2 + 2t_3}{t_1 + t_2 + t_3} \right). \quad (4)$$

Midpoint rule. Midpoints are needed for the edge and vertex rules. As explained below, either faces or half-faces are used in an edge

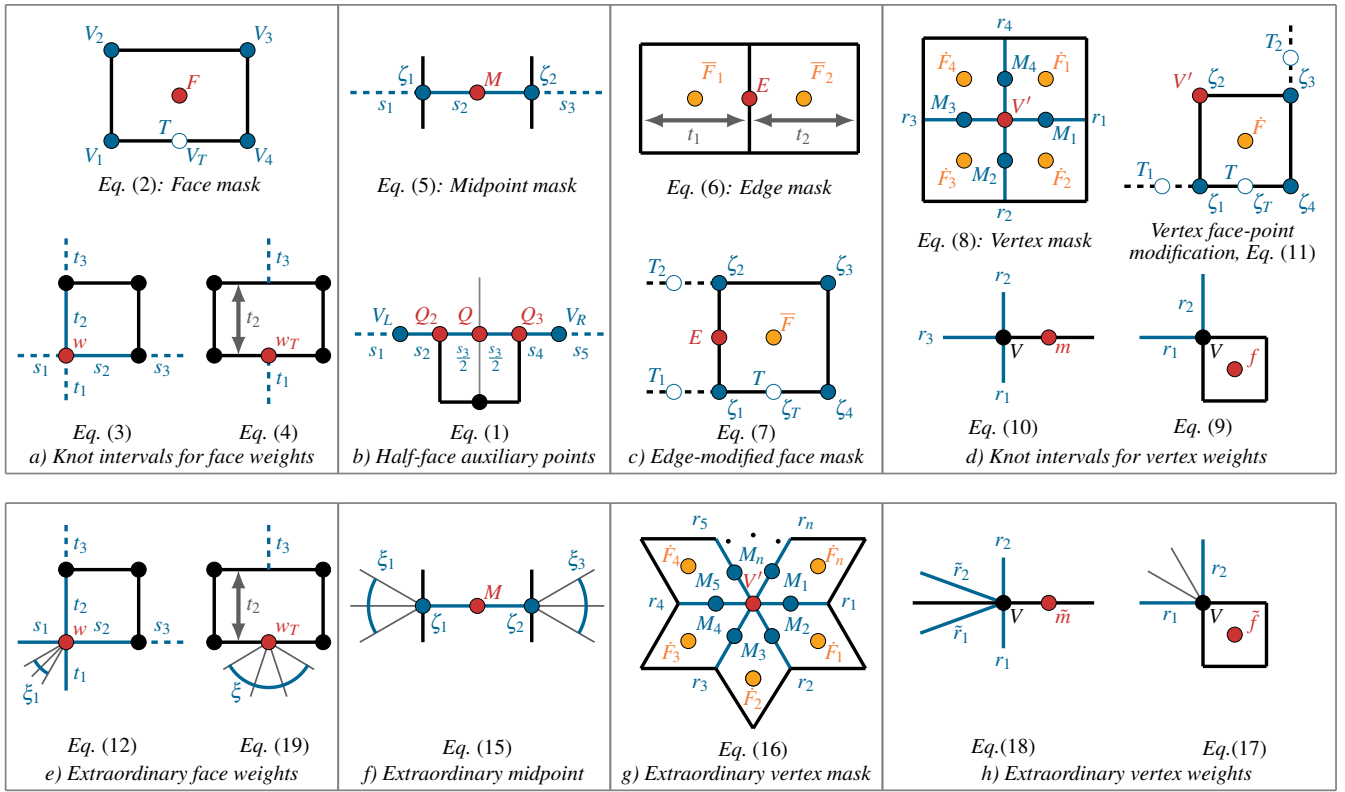


Figure 10: Top: stencils for face, midpoint, auxiliary, edge and vertex rules for regular DAS T-mesh subdivision, illustrating Equations 1-11. Red dots denote the vertex for which the rule is defined. Orange dots are showing modified face points used in edge and vertex rules. Blue and white vertices have nonzero weights in masks. A white vertex (always a T-joint) may or may not be present, and marks the location of the binary value $T \in \{0, 1\}$ associated with the edge. Lower-case letters denote knot intervals on edges as well as weights next to vertices. Bottom: stencils for DAS T-mesh subdivision rules at extraordinary vertices.

or vertex rule; the midpoints are computed using either original or half-face auxiliary points, depending on which faces are used:

$$M = \frac{1}{2} \left(\frac{s_2 + 2s_3}{s_1 + s_2 + s_3} \zeta_1 + \frac{s_2 + 2s_1}{s_1 + s_2 + s_3} \zeta_2 \right). \quad (5)$$

In the vertex rule, a T-joint will use a midpoint from the edge shared by two half-faces of a T-face. It happens that the formula for this midpoint is equivalent to that of the face point of the T-face.

The midpoints used for edge rules and vertex rules for the same edge are the same. Note however that midpoints for edges on a half-face are different from T-face edge midpoints: for instance, in Figure 9 (right) the edge midpoint $M_{E'_4}$ for the edge point E'_4 is different from one of the edge midpoints $M_{V'_4}$ used to compute the vertex point V'_4 .

Edge rule. The edge rule is essentially identical to the NURSS rule, with two important changes. First, when an edge endpoint is a T-joint in a face, we use the half-face adjacent to the edge in the computation. Second, face points are modified. These *modified face points*, $\{\bar{F}\}$, are computed using Eq. 7. Note that for each of the edge points on a face, we compute a different modified face point, and the midpoint is computed using the same face or half-face points. Figure 9 (right) shows an example where ζ is assigned half-face vertices instead of T-face vertices. When a half-face has neighboring half-faces, its face point calculation should use the knot intervals from neighboring half-faces instead of the knot intervals of the neighboring T-faces.

The edge rule is given by

$$E = \frac{1}{2} \left(\frac{t_2}{t_1 + t_2} \bar{F}_1 + \frac{t_1}{t_1 + t_2} \bar{F}_2 + M \right), \quad (6)$$

where \bar{F} is the edge-modified face point given by the rule

$$\bar{F} = T\bar{T}_1 w_T \zeta_T + (TT_1 w_T + \bar{T}(w_1 + T_1 w_4)) \zeta_1 + (w_2 + T_2 w_3) \zeta_2 + \bar{T} w_3 \zeta_3 + \bar{T} T_1 w_4 \zeta_4. \quad (7)$$

Vertex rule. Similar to the edge rule, we collect all faces incident at a vertex V , and replace each T-face, which has a T-joint on an edge incident at V with a half-face. Again, modified face points are required. Figure 9 (right) shows whether ζ is assigned vertices from f_T , f_L , or f_R .

The rule for the vertex is identical to the NURSS rule:

$$V' = \frac{1}{4} V + \frac{3 \sum_{i=1}^4 f_i \bar{F}_i + m_i M_i}{4 \sum_{i=1}^4 m_i + f_i} \quad (8)$$

with weights

$$f = r_1 \cdot r_2, \quad (9)$$

$$m = r_3 \cdot (r_1 + r_2), \quad (10)$$

The vertex modified face points are given by

$$\bar{F} = T\bar{T}_1 w_T \zeta_T + (\bar{T}(w_1 + T_1 w_4) + TT_1 w_T) \zeta_1 + w_2 \zeta_2 + (w_3 + \bar{T} T_2 w_4) \zeta_3 + \bar{T} T_1 \bar{T}_2 w_4 \zeta_4. \quad (11)$$

As in previous cases, modification of the remaining face vertices is described by cyclically rotating the stencil.

In the absence of T-faces, these rules degenerate to the case of cubic NURSS [Sederberg et al. 1998]. When there are no T-faces and all knot values are identical, the factorization is equivalent to Catmull-Clark subdivision.

3.3 Extraordinary refinement

We now extend our subdivision rules to support extraordinary vertices in such a way that an extraordinary vertex can share an edge with a T-joint or can itself act as a T-joint for some of the incident faces.

To be consistent with the goals of generalizing Catmull-Clark, we have two criteria:

1. The rules should generalize the extraordinary vertex rule for Catmull-Clark subdivision as well as Eq. 8.
2. There should be G^1 -continuity at an extraordinary vertex.

Although the face, midpoint and edge masks use the same vertices as the corresponding Catmull-Clark rules, a few changes are necessary when computing the weights on an arbitrary mesh. The weights for the rules require additional knot intervals from adjacent faces and edges.

Face rule. For faces with extraordinary vertices, we compute the weights in the following manner. We label a set of additional edges ξ_i , as shown in Figure 10(e). The new weight equation is

$$w = \frac{1}{4} \cdot \frac{s_2 + 2\tilde{s}_3}{\tilde{s}_1 + s_2 + \tilde{s}_3} \cdot \frac{t_2 + 2\tilde{t}_3}{\tilde{t}_1 + t_2 + \tilde{t}_3}. \quad (12)$$

where s_i, t_i are replaced by \tilde{s}_i, \tilde{t}_i as follows:

$$\begin{aligned} \tilde{s}_i &= \max\{s_i \cup \xi_i\} \\ \tilde{t}_i &= \max\{t_i \cup \xi_i\} \end{aligned} \quad (13)$$

These weights require normalization to preserve affine invariance. Thus, a face mask for a face touching an extraordinary vertex is:

$$F = \frac{Tw_T V_T + \bar{T}w_1 V_1 + w_2 V_2 + w_3 V_3 + \bar{T}w_4 V_4}{(Tw_T + \bar{T}w_1 + w_2 + w_3 + \bar{T}w_4)}. \quad (14)$$

Midpoint rule. Similarly, we replace the extension knot intervals used in the midpoint rule with maxima as shown in Figure 10(f): $\tilde{s}_1 = \max(\xi_1)$, $\tilde{s}_3 = \max(\xi_3)$, and

$$M = \frac{1}{2} \left(\frac{s_2 + 2\tilde{s}_3}{\tilde{s}_1 + s_2 + \tilde{s}_3} \zeta_1 + \frac{s_2 + 2\tilde{s}_1}{\tilde{s}_1 + s_2 + \tilde{s}_3} \zeta_2 \right). \quad (15)$$

Edge and vertex rules. The edge rule stays exactly the same. The vertex rule is generalized in the standard way, by summing all face points and midpoints around the vertex of valence n :

$$V' = \frac{n-3}{n} V + \frac{3 \sum_{i=1}^n \tilde{m}_i M_i + \tilde{f}_i \tilde{F}_i}{n \sum_{i=1}^n \tilde{m}_i + \tilde{f}_i} \quad (16)$$

The new stencils for weights \tilde{f} , \tilde{m} are part of the vertex rules for Figure 10(h). Their equations are

$$\tilde{f} = r_1 \cdot r_2. \quad (17)$$

$$\tilde{m} = \frac{1}{2} (r_1 + r_2) (\tilde{r}_1 + \tilde{r}_2), \quad (18)$$

Notice the equation for \tilde{f} stays the same, but the stencil is different. Extraordinary vertices may serve as T-joints. Such configurations turn into T-faces with regular T-joints after one subdivision step

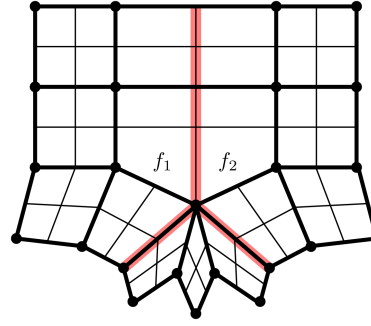


Figure 11: Refinement of a neighborhood around a vertex that is both extraordinary and a T-joint. T-joint extensions in red.

(Figure 11). To support such T-meshes, the maximum rules in Eq. 13 are applied for half-faces at the extraordinary T-joint as well. Additionally, t_1 in Eq. 4 is replaced by

$$\tilde{t}_1 = \max\{\xi\} \quad (19)$$

with ξ as shown in Figure 10(e).

3.4 Boundaries

Our factorized scheme supports boundaries similar to Catmull-Clark. Conceptually we mirror the mesh for each border face along the border edge, thereby defining knot intervals, control points, and T-joint tags past the border. In practice this is equivalent to evaluating non-uniform B-Spline boundary curves along the border. Note that for the auxiliary control points Q_i knot insertion on “both sides” of the border results in no change for the border vertex.

4 Overview of the scheme derivation

In this section, we outline how the proposed scheme was obtained. Recall that Catmull-Clark and a number of other subdivision schemes are derived from refinement relations for B-splines, i.e., expressing a basis function as a linear combination of basis functions corresponding to a finer grid (refinement relation).

On a mesh without T-joints, the topology of the refined mesh in the support of a basis function is defined uniquely, so all subdivision rules are obtained from a single refinement relation. The situation with T-meshes is completely different: a single T-spline basis function can potentially have infinitely many different T-mesh connectivities in its support. As a result, infinitely many subdivision rules may be required. We make two key observations.

1. The number of such combinations is *finite* for DAS meshes. (the number is still quite large).
2. The effects of local T-mesh connectivity on refinement relations, and, as consequence, on subdivision rules, amounts to moving weights from one vertex to another, based on the presence or absence of T-joints on nearby edges (*blocking*).

Equipped with these two observations we were able to find the set of rules that we have described, and verify correctness of these rules by explicit check on all possible local T-mesh configurations; thanks to the factorized form, the rules can be easily extended to arbitrary quad meshes, following the ideas of NURSS [Sederberg et al. 1998]. However, we found that applying the NURSS approach directly results in a loss of G^1 continuity at extraordinary vertices. We propose a modification instead, which avoids this problem.

4.1 Regular case.

T-splines on analysis-suitable meshes. To explain our approach more precisely, we review the basic T-spline definition [Sederberg et al. 2003]. Our scheme yields T-splines in the limit on regular T-grids.

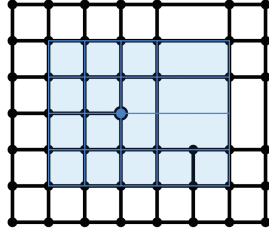


Figure 12: Support of a T-spline basis function.

We associate a spline space \mathcal{T} with a regular T-mesh by assigning a T-spline basis function to each vertex. It is constructed as a standard tensor-product non-uniform B-spline basis function, with the knot sequence determined by extending the edges at each vertex in four parametric directions by two knot intervals (we count a knot interval if the extension passes a vertex of the T-mesh or intersects an edge.) We call two sets of four knot intervals each the *knot vectors* of a T-spline basis function. The parametric location on the horizontal and vertical lines through the vertex constitutes the *basis cross* (Figure 12). A mesh admits a *standard* T-spline basis, if the basis functions constructed as above do sum up to one (not the case for all T-meshes), but any analysis-suitable mesh admits standard T-splines.

Nested analysis-suitable T-spline spaces. The subdivision rules in the case of B-splines are a consequence of the *nesting property* of the spline spaces defined for the grid M and once-subdivided grid M^1 : each basis function B_i on M is a linear combination of the basis functions B_j^1 on the grid M^1 . In the case of T-meshes, the nestedness property for quadrisecution does not always hold. However, for DAS T-meshes, it does:

Proposition 1. *The T-spline space \mathcal{T} associated with a regular dyadic analysis-suitable T-mesh T is contained in the space \mathcal{T}^1 associated with the once-subdivided mesh T^1 .*

Appendix A provides a proof.

This property means that subdivision rules abstractly can be defined in the standard way: given a surface f defined as a linear combination of T-spline basis functions on the coarse mesh $f = \sum_i P_i B_i$, we can replace each coarse basis function with $\sum_j w_{ij} B_j^1$, a linear combination of basis functions on the refined mesh, and by rearrangement of terms obtain expressions for control points of the same surface defined in terms of B_j^1 :

$$f = \sum_i P_i \sum_j w_{ij} B_j^1 = \sum_j \left(\sum_i w_{ij} P_i \right) B_j^1 = \sum_j P_j^1 B_j^1$$

where P_j^1 are the control points on the fine mesh.

To turn this into a practical scheme, however, we need to obtain the weight w_{ij} of the fine-scale basis function B_j^1 in the decomposition of every coarse-scale basis function B_i ; the number of different decompositions is infinite for general T-meshes, but for DAS meshes, the following holds:

Proposition 2. *For a regular DAS T-mesh, the control mesh of a single patch of a T-spline surface corresponding to a face can only have a finite number of possible connectivities.*

This proposition is proved in the electronic supplement.

Computing subdivision coefficients for regular DAS T-meshes. Proposition 1 asserts that the spaces \mathcal{T} and \mathcal{T}^1 are nested, and Proposition 2 suggests that a finite number of subdivision rules can be defined. Neither provide a way to compute coefficients w_{ij} for the subdivision rules.

For NURBS, one way to compute the coefficients w_{ij} is by performing knot insertion: Starting with a single control value 1 assigned to the vertex of the coarse-scale B-spline basis function B_i

and zeros assigned to all other vertices, we can insert knot lines of a fine-scale basis function B_j^1 , updating the control values accordingly. Once all knot lines are inserted, the control value of v_j , the vertex corresponding to the basis function B_j^1 , will be w_{ij} .

For nested analysis-suitable T-meshes, in particular for DAS T-meshes, the subdivision masks can be computed in the same way ([Scott et al. 2012]. Section 2.4.2).

These observations give us an initial possible approach to defining a subdivision scheme for DAS T-meshes: enumerate all possible control meshes of a single patch by Proposition 2, and then for each pair of a coarse and fine basis functions B_i and B_j^1 overlapping the patch, compute the coefficient w_{ij} using knot insertion to add all knots of B_j^1 .

However while the number of topologically distinct control meshes is finite, it is quite large: a scheme attempting to detect the local connectivity and compute the weights based on this would be quite complex. Even more importantly, it would be entirely unclear how to extend these rules to arbitrary T-meshes, as each local configuration in this case would potentially contain multiple extraordinary vertices.

Deriving a compact scheme for DAS meshes. For regular meshes or on T-meshes where the local structure is regular, knot insertion, as described above, produces the coefficients for non-uniform spline subdivision.

The coarse mesh in the support of a fine-scale basis function B_j^1 is shown in Fig. 13a, with a knot grid $[s_1, s_3, s_5] \times [t_1, t_3, t_5]$. Finer knot lines s_2 and s_4 need to be inserted into the support of any coarse-scale basis function B_i to determine the coefficient of control point P_i in the rule for computing P_j^1 .

We observe that for the coefficients to be identical to the regular case (because the knot insertion process would be exactly the same) two conditions need to be satisfied: (1) The mesh already contains horizontal and vertical coarse knot lines $[s_1, s_3, s_5]$ and $[t_1, t_3, t_5]$; (2) the original T-mesh does not contain the new knot lines $[s_2, s_4], [t_2, t_4]$.

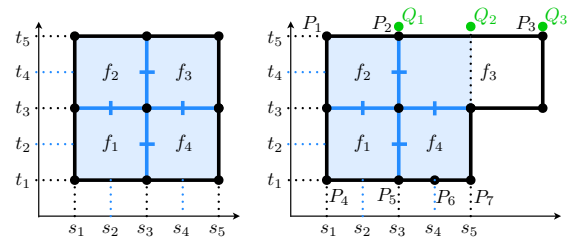


Figure 13: Knot insertion for a vertex basis function (support and knot intervals in blue) on a) a regular mesh, b) a T-mesh.

In general, both assumptions can be violated for a dyadic analysis-suitable T-mesh. In the T-mesh of Fig. 13b, the knot vectors of B_i , $i = 1, 2, 3$, associated with vertices v_i do not contain the knot line s_5 . On the other hand, the knot vectors of B_i , for $i = 4 \dots 7$ already contain s_4 . As a result, knot insertion will yield a non-standard coefficient w_{ij} for control points P_i , $i = 1 \dots 7$.

We use two different mechanisms to adjust the rules for situations of missing or extra knot lines.

- Accounting for missing knots at the coarse level is relatively easy: we simply perform temporary knot insertion, resulting in new control points Q_i , $i = 1, 2, 3$. Temporary points Q_i are used instead of corresponding control points P_i for the affected control point in the edge and vertex rules. This approach is similar to the way adaptive subdivision is performed in [Zorin et al. 1997].

- To deal with fine-scale basis function knots already present in the mesh (s_4 w.r.t. support of B_i , $i=4 \dots 7$) we modify the regular face and T-face masks (in this case for f_1 and f_4). These modifications require no recomputation of weights – rather some weights are zeroed out, or shifted to a different location; in the presented rule formulas, this movement is enabled by indicator functions T .

The intuition behind the second mechanism is based on the idea of *blocking* (Figure 14): For a coefficient w_{ij} to be nonzero, the fine-scale basis function B_j^1 needs to be a part of the decomposition of B_i , which is only possible if $\text{supp } B_j^1 \subset \text{supp } B_i$. The presence of a T-joint T on an external stencil edge of P 's edge point ensures that the basis function of the control point Q , does not contain the support of the basis function at P . As a consequence, although the stencil in the absence of T had Q in it, it cannot be present in the edge stencil. The effects of blocking have to be propagated to the face points used to compute an edge or vertex point for which a vertex in the standard stencil was blocked. This means that *modified* face points have to be computed, potentially *per corner* of a face. All blocking-related modifications can be summarized as follows:

A T-joint which is not a part of a face stencil, but is adjacent to a vertex v of a stencil, blocks the vertex of the stencil on the other side of v , and shifts its weight to v .

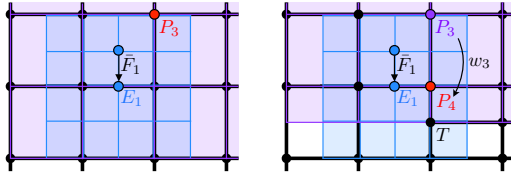


Figure 14: Left: P_3 's basis support (purple) contains E_1 's basis support (blue). Right: T blocks P_3 's basis support (purple) with respect to edge point E_1 's basis support - for \bar{F}_1 we shift w_3 to P_4 .

We emphasize that our rules were obtained by generalizing from a number of special cases and applying the blocking heuristics, not by a direct derivation from the knot-insertion rules. For this reason, our rules require a proof of validity for *all* possible stencils which is briefly discussed in Section 5 and more completely in supplementary material.

4.2 Extraordinary vertices

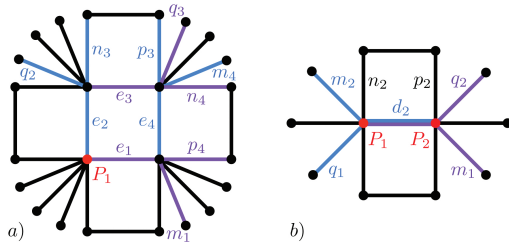


Figure 15: a) knot intervals used in face point weight w_1 's two factors (blue, purple) for original NURSS, b) knot intervals used in midpoint's two control point weights (purple, blue) for P_1 and P_2 .

We briefly discuss the NURSS scheme for extraordinary refinement of meshes with only quad faces and identical knot intervals for opposite quad edges, as we do not consider other types of meshes. Like the case of regular refinement, in the absence of T-joints our face and vertex rules for extraordinary refinement simplify to NURSS. Weights, however, are different for the face and midpoint masks. Using the knot interval labels in Fig. 15a), the NURSS face

weight is

$$w_1 = (e_1 + m_1 + p_4 + e_3 + q_3 + n_4) \cdot (e_2 + q_2 + n_3 + e_4 + m_4 + p_3)$$

while $w_2 \dots w_4$ are cyclically symmetric. Using Fig. 15b), the edge midpoints are computed as

$$M = \frac{(d_2 + q_1 + m_2)P_2 + (d_2 + q_2 + m_1)P_1}{(d_2 + q_1 + m_2) + (d_2 + q_2 + m_1)}.$$

Note that compared to our weights no maximum is computed.

Loss of tangent-plane continuity

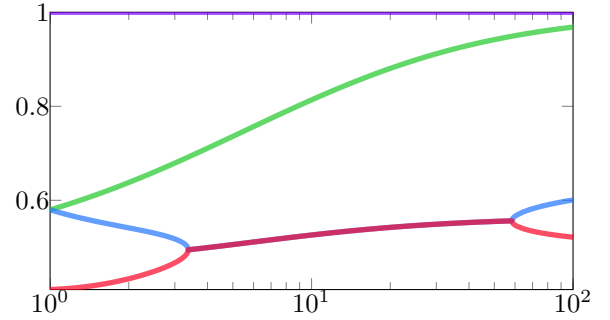


Figure 16: Modulus of four dominant eigenvectors of NURSS subdivision matrix for valence $n = 6$ and unit knot intervals except one varying.

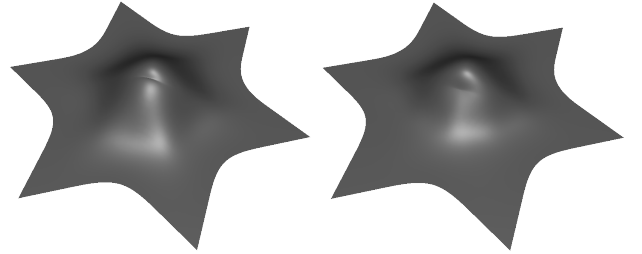


Figure 17: Extraordinary vertex of valence 6. a) NURSS, b) our max-modification.

In [Sederberg et al. 1998], the authors describe a pinching artifact at extraordinary vertices for general knot interval assignments due to different magnitudes of first and second eigenvalues of the subdivision matrix, but conjecture that the construction above nevertheless achieves tangent plane continuity. Unfortunately, we observe that in general, tangent plane continuity is not achieved. For analysis, we choose valence $n = 6$ with unit knot intervals except at one edge and vary the remaining interval. Figure 16 shows the modulus of the first four eigenvalues of the subdivision matrix. For knot intervals in the range 4-50, we can observe that the first subdominant eigenvalue stays real, while the second and third eigenvalues become complex. This results in a surface that does not have a unique tangent plane at the extraordinary vertex. Figure 17a) clearly shows that there are two separate tangent planes at the extraordinary vertex.

In comparison, this problem does not exist in the rules we propose. We refer the reader to the electronic supplement to this paper for the details on G^1 -continuity analysis of this scheme.

5 Evaluation

Verifying the limit surface in the regular case. To verify that our factorized subdivision rules yield the exact analysis-suitable T-spline surface in the limit, it is sufficient to compare one full global refinement step using standard knot insertion on analysis-suitable

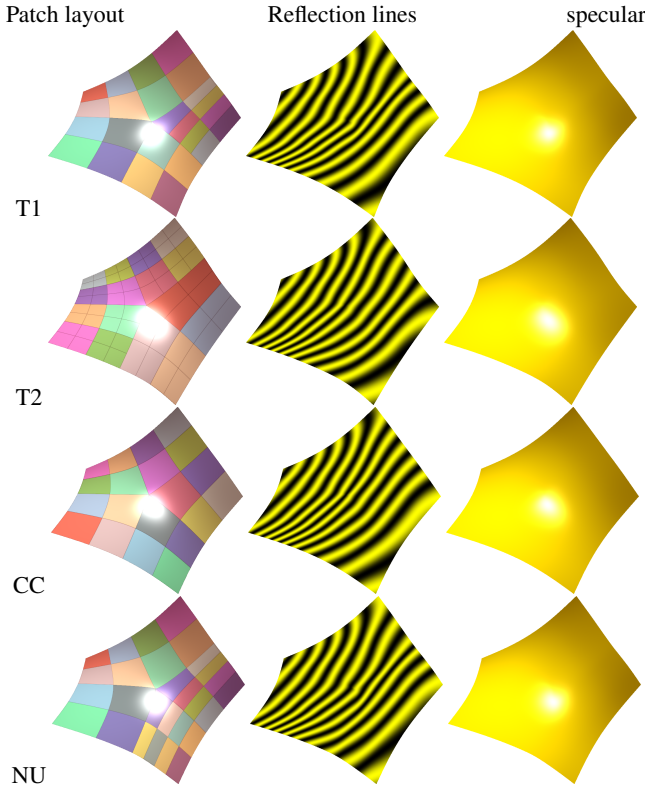


Figure 18: Valence 5 configurations: *T1*, *T2* are different T-joint configurations, *CC* is the standard Catmull-Clark surface, *NU* is NURSS.

T-spline basis functions with our subdivision formulas. As the number of local analysis-suitable T-mesh topologies on a regular grid is finite (see Appendix B), we automated their enumeration as well as the symbolic definition of both refinement and subdivision formulas, and performed all equality tests symbolically using Mathematica.

Surface quality. In Figure 18, we show a valence 5 extraordinary vertex with several possible local T-joint configurations, and compare to Catmull-Clark and NURSS with unequal knot intervals. We observe that the behavior of surfaces with T-joints is similar to that of Catmull-Clark, unless the knot spacing is unequal in two directions. Then it is slightly worse and comparable to NURSS, inheriting its pinching artifact, a result of two unequal subdominant eigenvalues in the subdivision matrix. This can be seen in the lower quality of the reflection lines.

In Figure 19, we look at the effect of increasing the vertex valence. We observe that the quality is consistent with the quality of Catmull-Clark. We note that the quality for standard Catmull-Clark decreases quickly with valence but a variety of techniques were developed to improve quality, some of which are applicable in our setting (as the scheme is stationary), although with greater difficulty.

Figure 2 shows how T-joints can be used to avoid under- or overtesellation without the introduction of extraordinary vertices, in the case where the control mesh faces vary greatly in size.

In Figure 20, we compare how similar mesh layouts are done with T-joints and conforming meshes with extraordinary vertices. Typically pairs of vertices of valence 3 and 5 need to be used to achieve the same layout.

Figure 21 shows several extraordinary vertices of valences 5 and 6 neighboring T-joints, transitioning to a coarser mesh from the fingers to the hand.

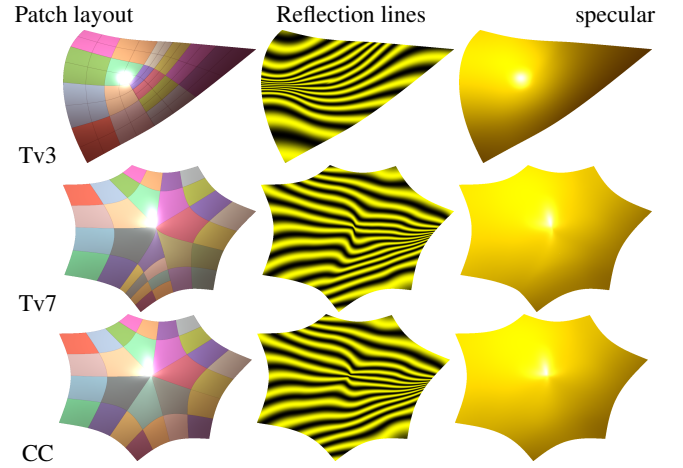


Figure 19: Varying the valence: vertices of valence 3 and 7, with T-joints on incident edges, *CC* shows Catmull-Clark for valence 7 for reference.

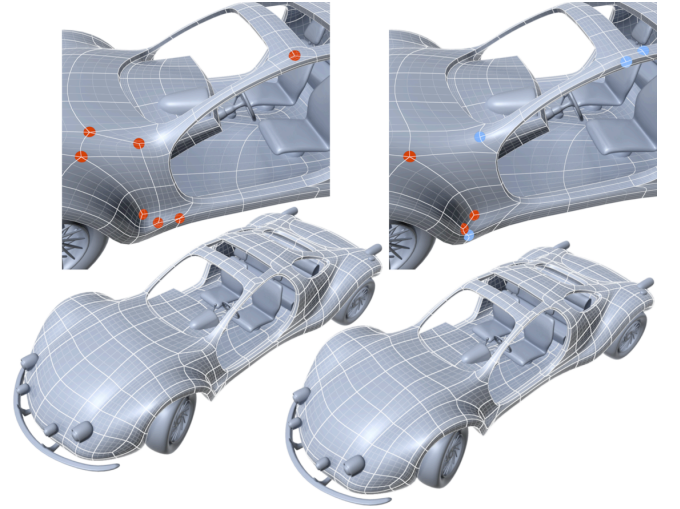


Figure 20: T-joints can be used to reduce the number of extraordinary vertices.

Figure 22 highlights all T-joints used in Figure 1 to avoid the extraordinary vertices found in the original mesh from [Bitmapworld 2006] to coarsen the mesh near the chin/neck and ear/cheek areas. It also shows an extraordinary T-joint.

6 Conclusions

We have demonstrated that for a restricted class of T-meshes it is possible to design a set of subdivision rules with a similar support and computation cost to the Catmull-Clark subdivision, and the complexity of stencils is only moderately higher compared to NURSS. The quality of surfaces is similar to Catmull-Clark near extraordinary vertices, although it degrades if knot intervals near extraordinary vertices vary greatly. Both a version of NURCC and analysis-suitable T-splines can be reproduced by our scheme.

Limitations. The most significant limitation of the proposed approach is that the T-mesh is required to be analysis-suitable. Requiring separation between T-joints limits the flexibility of T-joint insertion. On the other hand, this class of T-splines is best understood, and has a number of attractive properties not available for general T-splines.

Just as in the case with T-NURCCs, one can combine our scheme with general T-spline patching, provided that the separation be-

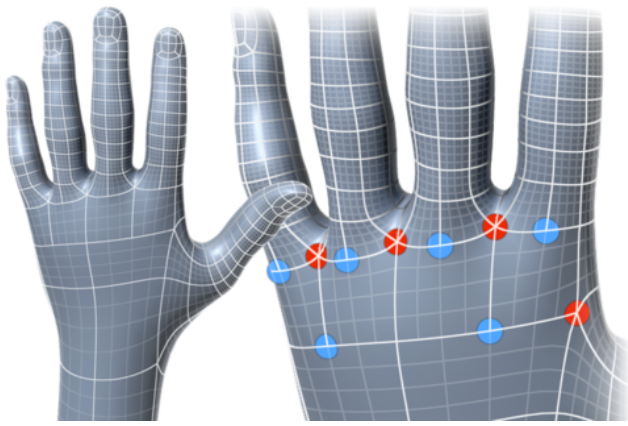


Figure 21: Several extraordinary vertices neighboring T-joints.

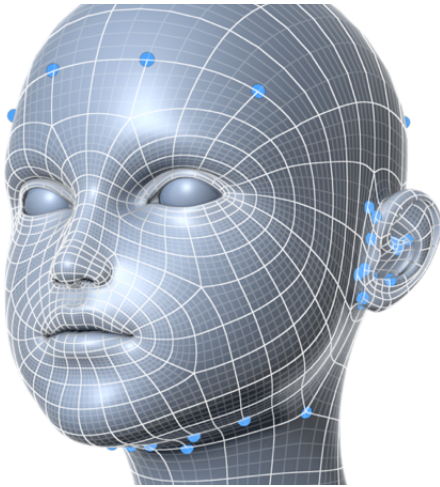


Figure 22: An extraordinary T-joint on the lower right.

tween *non-standard* regions on the regular part of the surface and extraordinary vertices is high; note that no such requirement needs to be imposed on the standard regions.

Our analysis of C^1 -continuity shows that while our modification of NURSS increases the range of valences for which the scheme is C^1 , the analysis is performed only under assumptions on independent knot intervals and for bounded valence. Even more significantly, for some of the higher-valence configurations C^1 conditions may still fail. Although practical implications of this are not high, this, along with degradation of surface quality suggests that more work is needed on improving NURSS rules.

A recent approach developed in [Prautzsch and Chen 2011] suggests that analysis of factorized schemes may be done without explicit analysis of the characteristic map; this opens up the possibility of analysis with restrictions on independent knot intervals.

Future work. In the future we want to extend the regular boundary rules of Section 3.4 to extraordinary vertices and define a full set of crease masks. We will also describe an extension of our scheme to support certain semi-standard T-spline configurations that will allow T-meshes such as a regular mesh with one quad split into four.

7 Acknowledgements

We are grateful to Bay Raitt, Jason Mitchell and Joe Demers for the discussions around T-meshes and mesh refinement that were the catalyst for this work. We would also like to thank BitmapWorld,

Bay Raitt and Greg Petchovsky for the models seen in the figures, and Michael K. Reed for proof-reading.

References

- ANDERSSON, L.-E., AND STEWART, N. F. 2010. *Introduction to the mathematics of subdivision surfaces*. SIAM.
- BAZILEVS, Y., CALO, V., COTTRELL, J., EVANS, J., HUGHES, T., LIPTON, S., SCOTT, M., AND SEDERBERG, T. 2010. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering* 199, 5-8, 229–263.
- BERTRAM, M., DUCHAINEAU, M., HAMANN, B., AND JOY, K. 2004. Generalized B-spline subdivision-surface wavelets for geometry compression. *Visualization and Computer Graphics, IEEE Transactions on* 10, 3, 326–338.
- BITMAPWORLD, 2006. Little girl head 3d model (<http://www.turbosquid.com/3d-models/polygonal-head-little-girl-3d-model/302581>).
- BUFFA, A., CHO, D., AND SANGALLI, G. 2010. Linear independence of the T-spline blending functions associated with some particular T-meshes. *Computer Methods in Applied Mechanics and Engineering* 199, 23-24, 1437–1445.
- CASHMAN, T., AUGSDÖRFER, U., DODGSON, N., AND SABIN, M. 2009. NURBS with extraordinary points: high-degree, non-uniform, rational subdivision schemes. *ACM Transactions on Graphics (TOG)* 28, 3, 46.
- CASHMAN, T., DODGSON, N., AND SABIN, M. 2009. A symmetric, non-uniform, refine and smooth subdivision algorithm for general degree B-splines. *Computer Aided Geometric Design* 26, 1, 94–104.
- CASHMAN, T. 2012. Beyond Catmull-Clark? a survey of advances in subdivision surface methods. In *Computer Graphics Forum*, vol. 31, 42–61.
- CATMULL, E., AND CLARK, J. 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design* 10, 6, 350–355.
- COTTRELL, J., HUGHES, T., AND BAZILEVS, Y. 2009. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons Inc.
- DENG, J., CHEN, F., AND FENG, Y. 2006. Dimensions of spline spaces over T-meshes. *Journal of Computational and Applied Mathematics* 194, 2, 267–283.
- DENG, J., CHEN, F., LI, X., HU, C., TONG, W., YANG, Z., AND FENG, Y. 2008. Polynomial splines over hierarchical T-meshes. *Graphical Models* 70, 4, 76–86.
- DÖRFEL, M., JÜTTLE, B., AND SIMEON, B. 2010. Adaptive isogeometric analysis by local h-refinement with T-splines. *Computer methods in applied mechanics and engineering* 199, 5-8, 264–275.
- ECK, M., DE ROSE, T., DUCHAMP, T., HOPPE, H., LOUNSBERY, M., AND STUETZLE, W. 1995. Multiresolution analysis of arbitrary meshes. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ACM, 173–182.
- EVANS, E., SCOTT, M., LI, X., AND THOMAS, D. 2014. Hierarchical t-splines: Analysis-suitability, bézier extraction, and application as an adaptive basis for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*.
- KHODAKOVSKY, A., SCHRÖDER, P., AND SWELDENS, W. 2000. Progressive geometry compression. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 271–278.
- LI, M., AND SCOTT, M. 2011. On the nesting behavior of t-splines. Tech. Rep. 11-13, ICES.

- LI, X., DENG, J., AND CHEN, F. 2010. Polynomial splines over general T-meshes. *The Visual Computer* 26, 4, 277–286.
- LI, X., ZHENG, J., SEDERBERG, T., HUGHES, T., AND SCOTT, M. 2012. On linear independence of T-spline blending functions. *Computer Aided Geometric Design* 29, 1, 63–76.
- LIU, L., ZHANG, Y., LIU, Y., AND WANG, W. 2015. Feature-preserving t-mesh construction using skeleton-based polycubes. *Computer-Aided Design* 58, 162–172.
- LOUNSBERY, M., DEROSE, T., AND WARREN, J. 1997. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics (TOG)* 16, 1, 34–73.
- MORGENSTERN, P., AND PETERSEIM, D. 2014. Analysis-suitable adaptive t-mesh refinement with linear complexity. *Arxiv preprint arXiv:1407.6175*.
- MOURRAIN, B. 2010. On the dimension of spline spaces on planar t-subdivisions. *Arxiv preprint arXiv:1011.1752*.
- 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, 25.
- MYLES, A., PIETRONI, N., KOVACS, D., AND ZORIN, D. 2010. Feature-aligned T-meshes. *ACM Transactions on Graphics (TOG)* 29, 4, 117.
- PRAUTZSCH, H., AND CHEN, Q. 2011. Analyzing midpoint subdivision. *Computer Aided Geometric Design*.
- SCOTT, M., LI, X., SEDERBERG, T., AND HUGHES, T. 2012. Local refinement of analysis-suitable t-splines. *Computer Methods in Applied Mechanics and Engineering* 213–216, 206 – 222.
- SEDERBERG, T., 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, 387–394.
- SEDERBERG, T., ZHENG, J., BAKENOV, A., AND NASRI, A. 2003. T-splines and T-NURCCs. In *ACM Transactions on Graphics (TOG)*, vol. 22, ACM, 477–484.
- SEDERBERG, T., CARDON, D., FINNIGAN, G., NORTH, N., ZHENG, J., AND LYCHE, T. 2004. T-spline simplification and local refinement. In *ACM Transactions on Graphics (TOG)*, vol. 23, ACM, 276–283.
- STAM, J. 2001. On subdivision schemes generalizing uniform B-spline surfaces of arbitrary degree. *Computer Aided Geometric Design* 18, 5, 383–396.
- WANG, W., ZHANG, Y., SCOTT, M., AND HUGHES, T. 2011. Converting an unstructured quadrilateral mesh to a standard T-spline surface. *Computational Mechanics*, 1–22.
- ZORIN, D., AND SCHRÖDER, P. 2001. A unified framework for primal/dual quadrilateral subdivision schemes. *Computer Aided Geometric Design* 18, 5, 429–454.
- ZORIN, D., SCHRÖDER, P., AND SWELDENS, W. 1997. Interactive multiresolution mesh editing. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 259–268.

A Subdivision and Nested Spaces

Proof of Proposition 1.

We show that *dyadic analysis-suitable T-splines* form nested spaces under face quadrisecting.

The extended T-mesh T_{ext} is defined as the T-mesh with all extensions included. We use *Corollary 8.10* from [Li and Scott 2011]:

Corollary 1. *Given two analysis-suitable T-meshes, T^1 and T^2 , if $T_{\text{ext}}^1 \subseteq T_{\text{ext}}^2$, then $\mathcal{T}^1 \subseteq \mathcal{T}^2$*

We prove that the extended T-mesh and the extensions of its subdivisions are nested. Let the T-mesh be T^0 and subdivided T-mesh be T^1 . Consider a face of T^0 containing a first-bay T-joint face extension (Figure 23a, face f_1 , middle extension). In this case, T^1 contains two edges covering the extension in f_1 . Suppose a face of T^0 contains a second-bay face extension (e.g., side extensions in f_1 in Figure 23a or central extension in f_0). By enumerating possible ways to connect pairs of faces for which the same extension is first- and second-bay, one can observe that there are only two valid configurations for such pairs of faces in a DAS T-mesh, identical exactly to these examples: (f_1, f_0) or (f_2, f_1) . In both cases, one can verify directly that the extended mesh of T^1 covers the extensions in f_0 and f_1 .

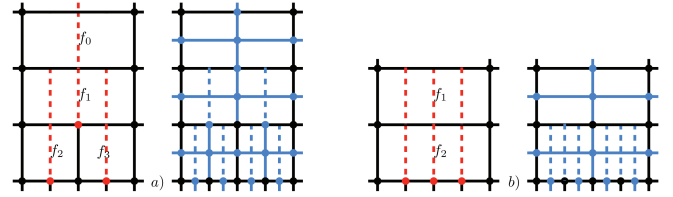


Figure 23: a) Extensions in the original mesh (red dashed lines) are also part of the extended subdivided mesh (blue) for DAS T-meshes. b) This is not the case for a non-nested space with three T-joints per edge in the original T-mesh (see extensions in f_1).

Therefore, the DAS T-spline spaces are nested. As an example of spaces that are not nested, imagine an analysis-suitable T-mesh that allows three T-joints per edge (Figure 23b). The extended original T-mesh is not contained in the extended quadrisected T-mesh.

B Stencil Enumeration

For the original T-mesh T^0 we define the T-spline space \mathcal{T}^0 , its basis functions B_i^0 and control points P_i^0 associated with vertex v_i and support $S_i^0 = \text{supp}(B_i^0)$. Analogously we define for the refined T-mesh T^1 , \mathcal{T}^1 , B_j^1 and P_j^1 associated with vertex v_j , $S_j^1 = \text{supp}(B_j^1)$.

We enumerate a set of neighborhoods in T^0 of a vertex v_j from T^1 , consisting of vertices in the stencil of v_j (*stencil candidates*.) In the electronic supplement we will show that the control points P_i^0 in these neighborhoods are in fact the only ones needed to compute P_j^1 . We distinguish 5 cases with different neighborhood topology:

1. P_j^1 is a face control point;
2. P_j^1 is an edge control point and both edge vertices are regular;
3. P_j^1 is an edge control point and one edge vertex is a T-joint with respect to one or both faces bordering the edge;
4. P_j^1 is a vertex control point, v_j is a regular vertex;
5. P_j^1 is a vertex control point, v_j is a T-joint in T^0 .

These 5 cases require different constructions of neighborhoods.

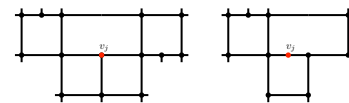


Figure 24: Examples of a) T-vertex and b) T-edge stencils

Cases 1, 2 and 4 are similar to Catmull-Clark - we define the stencils as the vertices of all faces bordering v_j . For an edge control point these are the faces bordering the edge, while for a face control point

it is the surrounding face. For cases 3 and 5, the 1-neighborhood is not big enough: there are stencil control points outside the 1-neighborhood. For 5, we add the faces neighboring the T-face to the left and right of the T-joint (Figure 24a). Similarly for case 3, we only add the face on the same side of the T-joint as the edge on which v_j^1 is (Figure 24b).

For enumeration of all connectivities we can obtain as 1-neighborhoods (cases 1,2,4) or 1-neighborhood with additional edges, the specific choice of knot intervals is irrelevant. For consistency, we fix the basis cross of B_j^1 to unit intervals. Then any knot interval incident at v_j^1 (associated with an edge or a face extent) and covered by the knot grid of B_j^1 is also fixed. This leads to fixed knot intervals marked blue in Figure 25. It is easy to see that all T-joints in a stencil topology have to be oriented either all horizontally or all vertically, except for case 2, where we have to consider both orientations. For cases 3 and 5 the orientation is fixed by the central T-joint, while for cases 1 and 4 we can choose one orientation (the other is symmetric). W.L.O.G., we assume vertical orientation. Each face in the stencil candidate can have at most one T-joint, which can be located on one of the two horizontal edges (in case 2, it can also be on one of the vertical edges) giving 3 possible states for each face (4 for case 2). In cases 1,3,4,5 we iterate for every face through the cases of a T-joint along the bottom and the top edge, observing all edge length constraints. For case 2, we consider T-joints on every edge (removing combinations that would result in horizontal and vertical T-joints). This yields a total of 171 stencil candidates. In the electronic supplement, we explain how we verify that this enumeration is exhaustive.

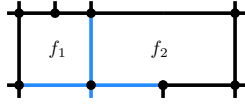


Figure 25: Edge constraints on opposite sides resulting from normalized basis cross.