Low degree splines for locally quad-dominant meshes

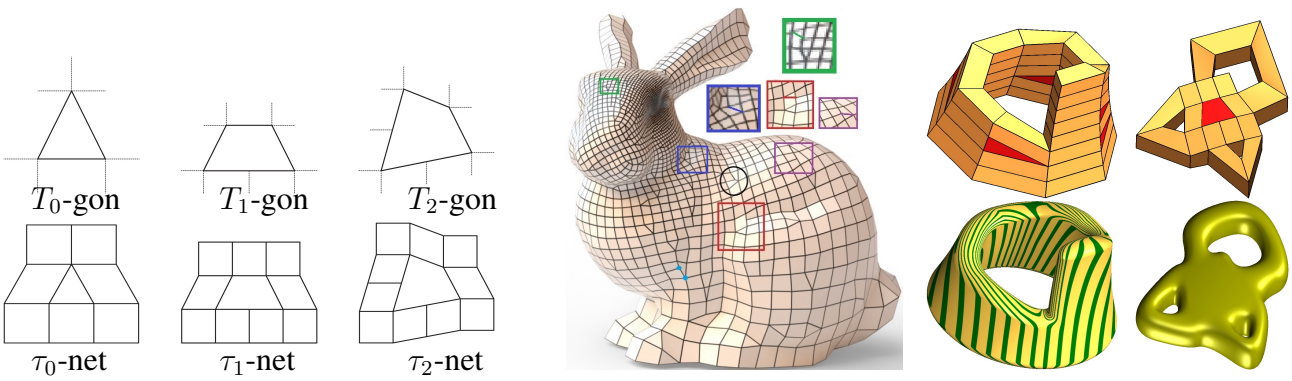
Kęstutis Karčiauskas[a], Jörg Peters[b,*]

[a] *Vilnius University*
[b] *University of Florida*

Abstract

A mesh is locally quad-dominant (lqd) if all non-4-sided facets are surrounded by quadrilaterals. Lqd meshes allow for irregular nodes where $n \neq 4$ quads meet and for multi-sided facets, called T-gons, that end quad-strips and so adjust mesh density.

This paper introduces a new class of bi-cubic (bi-3) Geometric T-joint (GT) splines whose control nets are τ-nets, i.e. T-gons surrounded by quads. These GT-splines join smoothly with each other, bi-3 G-splines and regular C^1 bi-quadratic splines to form smooth surfaces of degree at most bi-3.



(a) Main τ-net patterns (b) for lqd meshing (c) in modeling and design

Figure 1: Control τ-nets and GT-spline surfaces. (a) τ-nets are locally quad-dominant (lqd) mesh patterns that can serve as GT-spline control nets; (b) local re-connection of a quad-dominant mesh (from [12]) yields lqd meshes with τ-nets and irregular nodes with $n \neq 4$ neighbors , eg. of valence 3 and 5 within ◯. (c) GT-splines support flexible modeling: *left*: τ_0-nets spawn additional quad-strips; *right*: tight placement of a τ-net with irregular nodes.

1. Introduction

Tensor-product splines provide a principled approach to smoothing faceted meshes by interpreting the mesh as a control net [5, 11] and associating a smooth piecewise polynomial map with each mesh node. However, this interpretation requires the mesh to have the connectivity of a grid. A natural generalization is the class of *quad-dominant meshes*. Quad-dominant meshes allow irregular nodes where $n \neq 4$ quads join and multi-sided facets, called T-gons, that end quad-strips and so adjust mesh density as in Fig. 1a. Smoothing out quad-dominant meshes by interpreting its nodes as control points of smoothly-joined piecewise polynomial surface patches results in a surface description compatible with the industry's NURBS standard and can accommodate general shapes without having to resort to trimming.

*Corresponding author
Email address: jorg.peters@gmail.com (Jörg Peters)

Quad-dominant meshes including T-gons arise in polyhedral modeling and are output by re-meshing algorithms [1, 23, 12, 29] that enjoy increased locality and parallelism compared to strict quad re-meshing algorithms [26, 25, 13, 3]. Fig. 1a lists three mesh patterns consisting of a non-4-sided polygon, the *T-gon*, surrounded by quads. In contrast to the neighborhood of irregular nodes, these τ*-nets* have two preferred directions. A triangle with one $n = 5$-valent and two 4-valent nodes forms the core of any τ_0 control-net; a (nominally) 5-sided polygon with 4-valent nodes and one T-vertex is the core of any τ_1-net; and 6-sided polygon with 4-valent nodes and two T-nodes is the core of any τ_2-net. Fig. 1b illustrates how τ_0-, τ_1- and τ_2-nets enable spawning or terminating quad strips.

This paper focuses on surfaces that use τ-nets as control nets. Given the massive scale at which quad-dominant meshes can be generated, it is important to keep any corresponding spline representation as simple as possible, i.e. to minimize the number of pieces and their polynomial degree. It is then natural to interpret grid-like sub-meshes as bi-2 spline control nets and so replace a control net with a curved, smooth shape. The construction in [17], abbreviated G_{KP} hereafter, complements bi-2 C^1 splines by providing n-sided *surface caps* of bi-3 pieces covering the neighborhood of each irregular node of valence $n \neq 4$. The new *GT-splines* further extend the number of control patterns τ-nets, using polynomial pieces of degree bi-3. Together, bi-2 splines, G_{KP} and GT-splines support polyhedral shape design including change of control net density as illustrated in Fig. 1c. Notably, due to the small support of the bi-2 B-splines, T-junctions for GT-splines may be placed in close proximity to irregular nodes, see Fig. 1c, *right* and irregular nodes may be direct neighbors of each other, see the cyan points in Fig. 1b.

To ensure that the GT-splines of this paper can be applied, a designer or meshing algorithm need only guarantee *local quad dominance* (lqd): non-4-sided facets with T-junctions must be surrounded by quads. That is, for the three types of T-gons, the mesh must be their corresponding τ-net shown in Fig. 1a.

In summary, the contributions of the is paper are

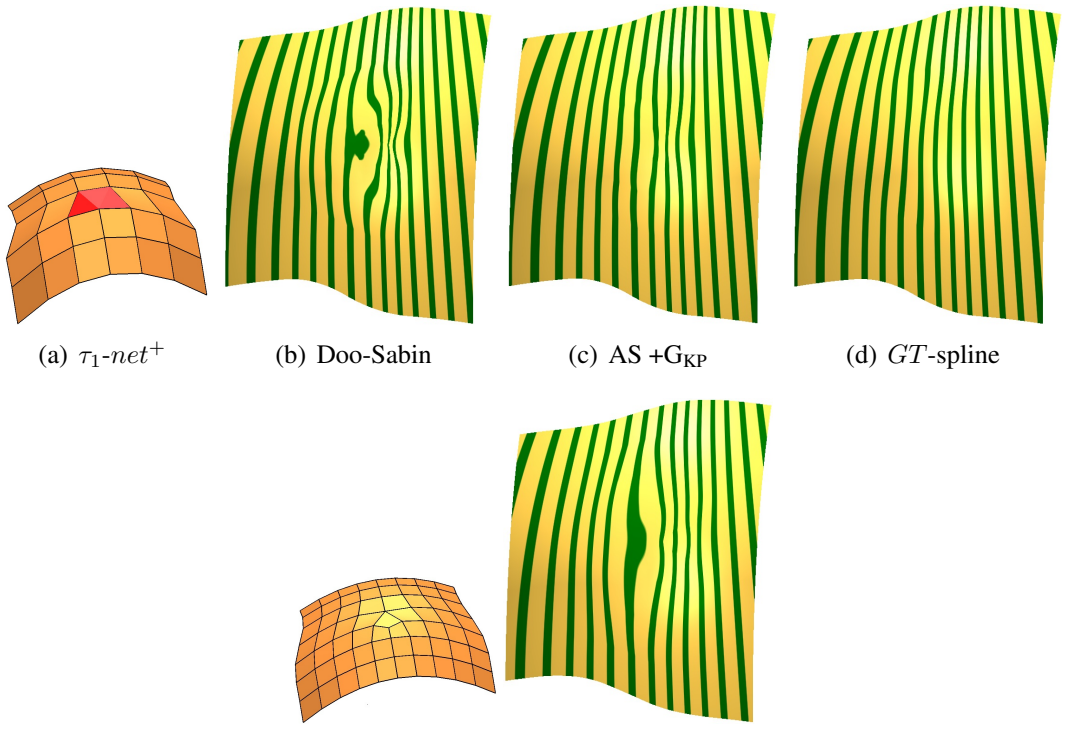
- bi-3 GT-splines generalizing bi-2 splines to τ-nets (with T-junctions) as control nets;
- a framework, for spline surfaces controlled by lqd meshes, that combines bi-2 splines on regular sub-meshes with bi-3 splines for τ-nets and bi-3 splines for non-4-valent, non-T-gon nodes.

Outline Section 1.1 further motivates GT-splines by reviewing existing constructions that generalize bi-2 splines. Section 2 defines lqd meshes. Section 3 defines GT-splines. Section 4 assesses and compares the surface quality of GT-splines.

1.1. Prior surface constructions generalizing bi-2 splines

When the goal is to smooth a quad-dominant mesh with minimal polynomial degree, Doo-Sabin subdivision [10] naturally comes to mind. However, this classic generalization of bi-2 splines not only generates an infinite sequence of polynomial pieces, but also clearly fails to yield good shape – as the highlight line distribution [2] of Fig. 2b demonstrates. The early onset of the highlight line distortion, in the first step, hints that Doo-Sabin subdivision is also not useful as a preprocessing step. Augmented Subdivision (AS) [16] yields much better but still unsatisfactory shape: compare Fig. 2c with new construction Fig. 2d. One step of Catmull-Clark (CC) subdivision [4] is often used in remeshing algorithms to turn a quad-dominant mesh into a pure quad mesh (see for example in [12]). However, already this single CC step results in shape artifacts, as Fig. 2e illustrates (In fact, [15] demonstrated that for meshes with T-junctions and a default of bi-3 splines, Catmull-Clark subdivision is not a viable option.)

Multi-sided holes in a regular bi-2 spline complex can be filled with the G_{KP} algorithm of [17], yielding surface caps with good highlight line distributions. G_{KP} caps nicely complement τ-surfaces: for $n = 3, 5$ each sector of an n-sided surface cap is a single patch of degree bi-3 and for $n > 5$ the surface cap consists of n bi-3 2×2 macro-patches. An alternative for filling of multi-sided holes in a bi-2 patch complex is the



(a) τ_1-net^+ (b) Doo-Sabin (c) AS +G_{KP} (d) GT-spline

(e) CC refined mesh + G_{KP}

Figure 2: Subdivision disappoints already on (a) the simple convex T_1-configuration: (b) highlight lines of the resulting Doo-Sabin (DS) subdivision surface; (c) of two steps of Augmented Subdivision (AS) followed by the G_{KP} algorithm; (d) new τ_1-construction. (e) Quad mesh obtained from (a) applying one Catmull-Clark (CC) step, fitting bi-2 splines where possible and G_{KP} bi-3 patches otherwise.

mathematically elegant S-patch construction, [24]. However, S-patches are higher rational degree and are less compatible with existing infrastructure. Remarkably, the construction, [28] stays within the space of bi-2 splines but typically has worse highlight line distributions than G_{KP}.

Creating T-junctions by locally inserting additional knot lines can add extra freedom to model detail for tensor-product spline surfaces. In the bi-quadratic context this approach has been investigated, for example in [7, 32, 33]. Working with these spaces assumes a careful, strictly hierarchical surface design that globally keeps track of spacing between knots to assure compatibility. If the layout of the control net is given, by a prior model or by a designer using a polyhedral modeler, such a global knot distribution has to be discovered; and it may not exist: for the mesh in Fig. 1c,*left*, the sum of *knot spacings* (differences between consecutive knot values [6]) in the vertical direction has to stay fixed when travelling around the ribbon even as the number of quad strips increases. The sums can only agree on return if as many knot spacings are zero as to make the bi-2 spline formally non-smooth.

In [15] T-configuration splines of degree bi-4 were introduced as an alternative to hierarchical splines [22, 30, 9, 14, 21] that require global management and consistency of knot spacings. [15] does not include τ_0-nets and employs higher-degree splines so that its control nets require more separation than the τ-nets of GT-splines, analogous to the larger support of C^2 bi-3 B-splines compared to C^1 bi-2 splines. Its bi-4 surface covers isolated T-junctions in a bi-3 C^2 spline complex. Just as the surrounding bi-3 splines are smoother than bi-2 splines, the bi-4 surfaces are often smoother than GT-splines. The extra cost of the bi-4 surfaces is, however, not just the polynomial degree, but, more importantly, a larger footprint and more stringent requirements on the sub-meshes: τ_0 can not be used, all outermost nodes of τ-nets must be regular and the τ-net must be surrounded by the quads. By comparison, GT-splines appreciably reduce

the burden on the designer or meshing algorithm, and the smaller footprint avoids extra mesh refinement of [19]. The focus of [20] is on refinability and therefore uses bi-4 patches rather than bi-3 patches in this paper. The construction in [18] requires higher polynomial degree to deliver class A surfaces.

2. Meshes, Control Nets and Splines

2.1. T-gons and τ-nets

A *T-junction* is where two quads on one side meet one facet on the other. For $m > 4$, a T_{m-4}-gon, short T-gon, is an m-gon surrounded by quads such that all its vertices are either T-junctions or have valence 4 (cf. Fig. 1a). A T_1-gon has one T-junction and is formally a pentagon, a T_2-gon has two T-junctions and is formally a hexagon. A T-gon and its surrounding layer of quads is called a τ-net.

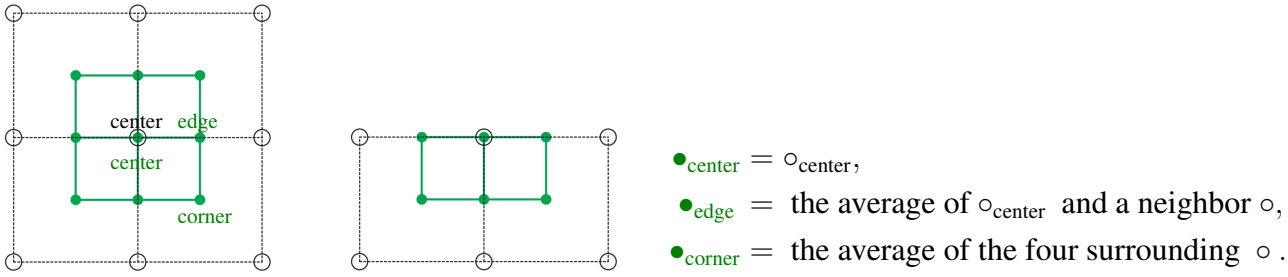
The choice of bi-2 splines for the regular mesh allows for an additional useful case: the τ_0-net. This configuration frequently arises in quad-dominant meshing algorithms, for example [12, 29] and is useful in faceted modeling. Note that the T_0-gon is not an arbitrary triangle: it has two vertices of valence 4 and one of valence 5 and hence is surrounded by 7 quads. Although τ_0-nets have no T-junction, the GT-spline for this configuration is based on the same approach as for τ_1 and τ_2 nets. More complex T-configurations can often be converted, by localized re-connection or refinement, to the three main τ-nets.

2.2. Splines and surface continuity

GT-splines convert a τ-net into a collection of tensor-product patches in Bernstein-Bézier form (BB-form; see e.g. [11]):

$$\mathbf{f}(u,v) := \sum_{i=0}^{d}\sum_{j=0}^{d} \mathbf{f}_{ij} B_i^d(u) B_j^d(v)\ , \quad (u,v) \in [0..1]^2,$$

where $B_k^d(t) := \binom{d}{k}(1-t)^{d-k}t^k$ are the Bernstein polynomials of degree d and $\mathbf{f}_{ij}$ are the BB-coefficients. This paper uses $d = 2$ or $d = 3$, i.e. the BB-patches are bi-2 (bi-quadratic) or bi-3 (bi-cubic). Connecting $\mathbf{f}_{ij}$ to $\mathbf{f}_{i+1,j}$ and $\mathbf{f}_{ij+1}$ wherever possible yields the BB-net of BB-coefficients. A useful operation on polynomials in BB-form is their *splitting* into two pieces, say a left half and a right half, by the well-known de Casteljau algorithm [11].



(a) B $\rightarrow$ BB (b) tensor-border $\mathbf{t}$ (c) B-to-BB conversion rules

Figure 3: Bi-2 B-to-BB conversion. Circles $\circ$ mark B-spline control points, Solid disks $\bullet$ mark BB-coefficients.

The regions of the mesh that are not τ-nets, but form regular 3×3 grids are interpreted as the control net of a bi-2 spline in B-spline form [6] with equally-spaced knots, i.e. in uniform B-spline form. Re-expressing the spline in bi-2 BB-form is called *B-to-BB conversion*. The straightforward rules for this basis conversion are given in Fig. 3c and illustrated in Fig. 3a. Conversion of a partial mesh yields a part of a BB-net that defines position and first derivatives across an edge using six BB-coefficients. In Fig. 3b

the edge corresponds to the bottom three •. The partial BB-net is called a *tensor-border* (of degree 2) and is denoted by the letter $\mathbf{t}$.

Since the τ_0, τ_1 and τ_2 GT-splines apply where the quad strips change, their construction is naturally based on the toolkit of geometric continuity [8]. Patches $\mathbf{f}$ and $\tilde{\mathbf{f}}$ that share an edge parameterized by $(u, 0 = v)$ are G^1-*connected* if they have matching derivatives after the change of variables $\rho(u, v) := (u + b(u)v, a(u)v)$:

$$\partial_v \tilde{\mathbf{f}}(u, 0) - a(u)\partial_v \mathbf{f}(u, 0) - b(u)\partial_u \mathbf{f}(u, 0) = 0. \tag{1}$$

3. The GT-spline Construction

Applying the GT-spline construction to τ_0, τ_1 and τ_2-nets produces surfaces, called τ-*surfaces*, with, respectively, 2×2, 4×2 and 4×4 polynomial pieces of degree bi-3. τ-surfaces are G^1 connected to surrounding bi-2 spline surfaces, other τ- or G_{KP} multi-sided surface caps.

For each type of T-gon, Table 1 lists the N distinct reparameterizations $\rho_j, j = 0, \ldots, N-1$ for the G^1 transition across the boundaries. All b_j are of the form $b_j(u) := \beta_j(1-u)u$ and the β_j are given in Table 1 below. The τ_0 construction has $N = 4$ distinct reparameterizations. The first parameterization sets $a_0(u) := \alpha(\frac{u}{2}) = (1 - \frac{u}{2}/2) = 1 - \frac{u}{4}$, $\beta_0 := 0$ and hence is $\rho_0(u, v) := (u + 0(1-u)u, (1 - \frac{u}{4})v)$. This ρ_0 reparameterizes the tensor-border $\mathbf{t}_0$ of the top row labeled τ_0 in Fig. 4b. Analogously $\rho_3(u, v) := (u + \frac{1}{2}(1-u)u, v)$ for τ_0 reparameterizes $\mathbf{t}_3$. The τ_1 construction uses $N = 6$ distinct ρ_j, e.g. $\rho_0 := (u, (1 - \frac{u}{6})v)$ and, since $a_1(u) := 1 - (\frac{1}{2} - \frac{u}{2})/3$ and $\beta_1 := 0$, hence $b_1(u) = 0$, $\rho_1(u) := (u, 1 - \frac{1+u}{6}v)$. The $N = 8$ distinct ρ_j of τ_2 are defined by $a_{j+4}(u) := a_j(u)$ for $j = 0, 1, 2, 3$ and $[a_0, a_1, a_2, a_3] := [\alpha(\frac{1}{3}u), \alpha(\frac{1}{3}(1-u) + \frac{1}{2}u), \alpha(\frac{1}{2}(1-u) + \frac{2}{3}u), \alpha(\frac{2}{3}(1-u) + u)]$.

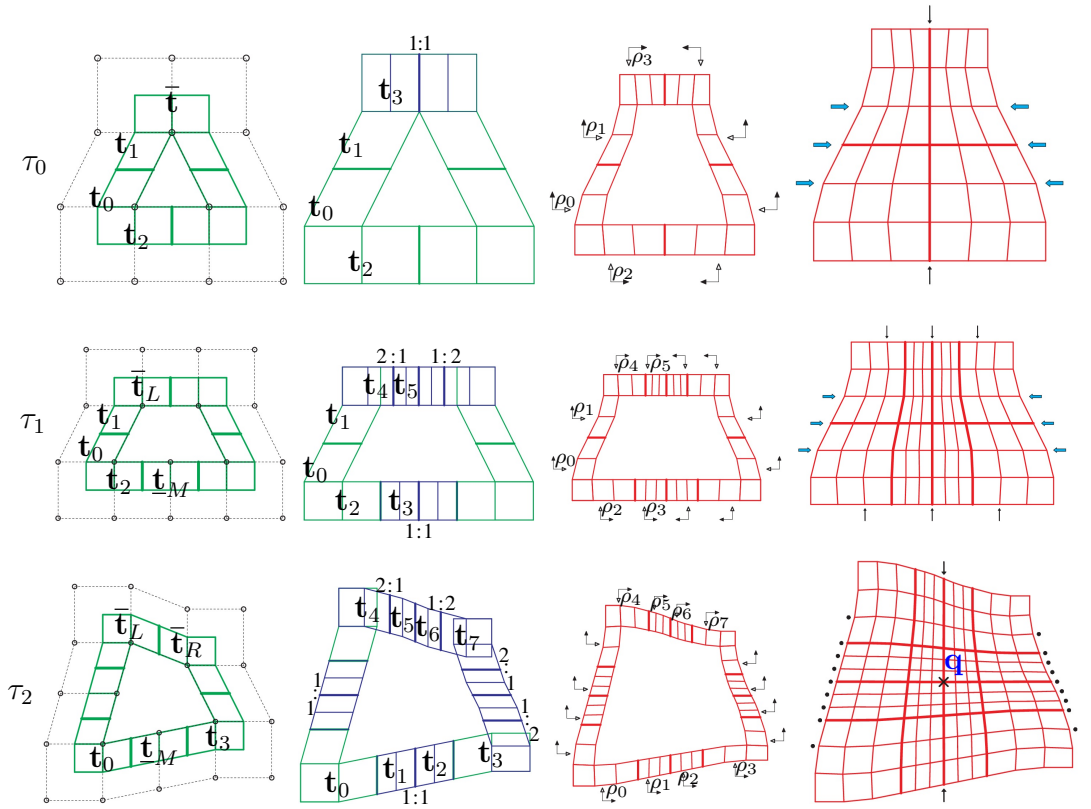
config	$\alpha(u)$	$[\ldots, a_j(u), \ldots]$	$[\ldots, \beta_j, \ldots]$
τ_0	$1 - \frac{u}{2}$	$[\alpha(\frac{u}{2}), \alpha(\frac{1}{2} + \frac{u}{2}), 1, 1]$	$[0, 0, -1, 2]/4$
τ_1	$1 - \frac{u}{3}$	$[\alpha(\frac{u}{2}), \alpha(\frac{1}{2} + \frac{u}{2}), 1, 1, 1, 1]$	$[0, 0, -2, 2, 3, -3]/12$
τ_2	$1 - \frac{u}{3}$	see text above Table 1	$[-2, 2, -2, 2, 3, -3, 3, -3]/18$

Table 1: Reparameterizations (change of variables) $\rho_j(u, v) := (u + b_j(u)v, a_j(u)v)$ as indexed in Fig. 4c.

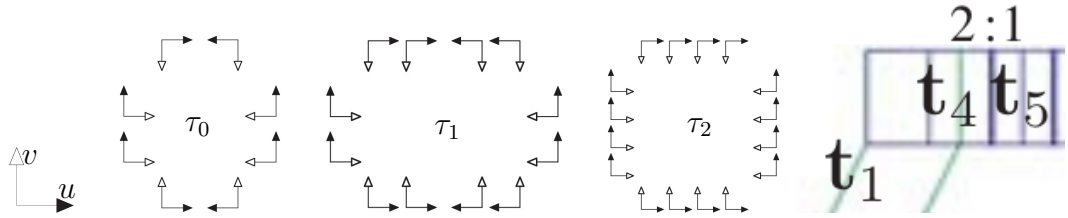
The steps of the three constructions are succinctly summarized in the following GT-Algorithm and illustrated in Fig. 4. All three constructions share a common framework where only the last step, illustrated by Fig. 4d, differs by case. The same symbol $\mathbf{t}$ is used for all three versions of tensor-borders, (a) input, (b) split and (c) reparameterized, corresponding to Steps (a-c) of the algorithm. They represent the same Hermite information in different form: splitting produces the layout required for the bi-3 τ-surface macro-patch, reparameterization makes the corner derivatives consistent.

config	1:1 split	2:1 split	1:2 split
τ_0	$\overline{\mathbf{t}} \to (\mathbf{t}_3, *)$		
τ_1	$\underline{\mathbf{t}}_M \to (\mathbf{t}_3, *)$	$\overline{\mathbf{t}}_L \to (\mathbf{t}_4, \mathbf{t}_5)$	
τ_2	$\underline{\mathbf{t}}_M \to (\mathbf{t}_1, \mathbf{t}_2)$	$\overline{\mathbf{t}}_L \to (\mathbf{t}_4, \mathbf{t}_5)$	$\overline{\mathbf{t}}_R \to (\mathbf{t}_6, \mathbf{t}_7)$

Table 2: Splitting of Step (b). See Fig. 4a and Fig. 4b for the naming and location of the tensor-borders. Unlisted tensor-borders are not split. Due to left-right symmetry, only one half of the constructions need to be listed. The entry $*$ indicates that the other half of the split belongs to the ($\mathbf{t}_3$ of the) right symmetric group.



(a) input $\mathbf{t}$s (b) split $\mathbf{t}$s (c) reparameterized $\mathbf{t}$s (d) BB-net



(e) u, v-coordinate systems for the reparameterizations (f) zoom of τ_1 (b)

Figure 4: The three GT-spline constructions. (a) shows the tensor-borders of degree 2 derived by partial B-to-BB conversion from the respective τ-net. The τ-net is shown as $\circ$ connected by dashed lines. (b) To achieve internal C^1-continuity tensor-borders with displayed ratios are split in the ratios. Table 2 lists the split pairs. Since the split tensor-borders are not consistent at the top two corners for τ_0 and τ_1 and all but the bottom-left corner for τ_2, the reparameterizations ρ_j (c) of Table 1 yield the final form of the tensor-borders of degree 3. (d) BB-net of the τ-surface with arrows and dots indicating the algorithmic steps for constructing the interior for constructing the interior. (e) (u, v)-coordinate systems of the reparameterizations ρ_j. (f) Enlargment to better see the purple degree 2 tensor-border after splitting.

GT-Algorithm

Input: a τ-net.

Output: a bi-3 surface that smoothly fills a frame of bi-2 patches with pieces of degree bi-3.

(a) Fig. 4a: generate the coefficients of the input tensor-borders (dark green net) from the τ-net nodes $\circ$ by partial B-to-BB conversion.

(b) Fig. 4b: split the tensor-borders indicated by ratios as listed in Table 2.

(c) Fig. 4c (u, v directions are shown in Fig. 4e): for each tensor-border $\mathbf{t}_j$, the first-order v-expansion of $\mathbf{t}_j \circ \rho_j(u, v)$ at $v = 0$ provides the final reparameterized tensor-border of degree 3 that agrees with

its neighbors when

$$\rho_j : [0..1]^2 \to \mathbb{R}^2, \qquad \rho_j(u,v) := (u + b_j(u)v, a_j(u)v), \qquad (2)$$
$$a_j(u) \text{ linear }, b_j(u) := \beta_j(1-u)u, \text{as listed in Table 1.}$$

(d) Fig. 4d: C^1 Completion

τ_0 construction

1. Construct the central vertical BB-net layer marked by the vertical arrows as C^1 connected cubics according to the formulas presented in Appendix B.
2. Set the remaining BB-coefficients of the horizontal layers marked by horizontal cyan arrows to form C^2-connected cubics.

τ_1 construction

1. Construct the vertical BB-net layers marked by the vertical arrows as C^1-connected quadratics, represented in degree 3 form.
2. Complete the horizontal layers of BB-coefficients marked by horizontal arrows as C^2 connected cubics.

τ_2-construction

1. Construct the vertical BB-net layer marked by the vertical arrow as C^2-connected cubic curves.
2. Complete the horizontal layers of BB-coefficients marked by *dots* as C^2-connected cubic curves.
3. To make the construction (diagonally) symmetric, repeat Steps 1 and 2 creating first the central horizontal layer and then the nine vertical layers.
4. Average the BB-coefficients from Steps 1,2 with those from Step 3.

The curves in Steps d1 and d2 are two smoothly-joined cubics represented as four pieces whose formulas are listed in Appendix A. The key to well-shaped τ_2 surfaces is the central BB-coefficient $\mathbf{q}$, see Fig. 4d, row τ_2. This central coefficient is an affine combination of the τ_2-net. The explicit formula for $\mathbf{q}$ is presented in Appendix A.

The pattern for splitting of the original tensor-borders and the scalar components (a_j, b_j) of ρ_j for each tensor-border customizes the Algorithm to one of the τ_0, τ_1, τ_2 configurations. Due to combinatorial symmetry (left/right for τ_0 and τ_1, diagonal for τ_2) of the constructions, only half the ρ_j and half of the splits need to be listed in Table 1 and Table 2.

Appendix B succinctly presents the complete set of explicit formulas for constructing τ_0-surfaces by the GT-algorithm.

The motivation for the algorithmic steps is as follows.

- The initial tensor-borders are split according to Table 2 so that they match the layout of the C^1 bi-3 τ-surface.
- Since the degree of the τ-surface is not to exceed bi-3, a linear $a_j(u)$ and a quadratic $b_j(u)$ are of maximal degree for an internal C^1 join and to achieve consistency at the corners of the reparameterized tensor-borders.
- Split tensor-borders $\mathbf{t}_j$ are reparametrized since the splitting changes (shortens) the tensor-borders $\mathbf{t}$ so that they no longer agree where they meet at the corners. (For τ_0 and τ_1 the splitting yields a mismatch only at the top corners of the bi-3 τ-surface, but the linear $a(u)$ forces all tensor-borders $\mathbf{t}_j$ to be reparameterized.)

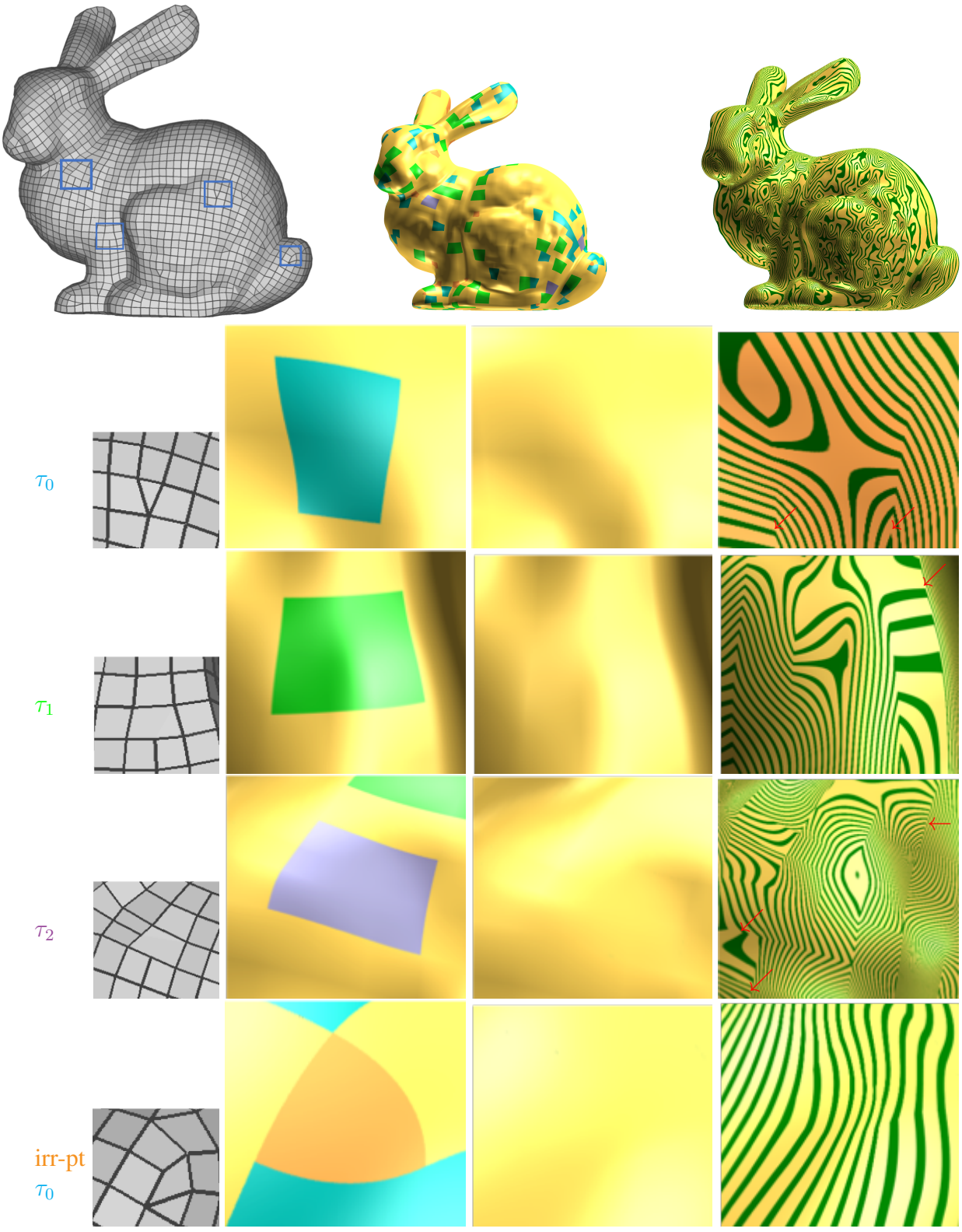


Figure 5: T-gons and irregular node (irr-pt). The four frames on the bunny outline the zoomed-in configurations. All sharp transitions in the highlight line distribution, indicated by ↙, originate in the bi-2 C^1 spline surface. They can propagate to boundaries between bi-2 and GT-spline patches but not into the GT-spline patches. The colors indicate T0-gons, T1-gons, T2-gons, eops and bi-2 regular tensor-product spline.

- The constructions in (d) were selected by testing the highlight line distribution over many inputs.

The smoothness of the surface is characterized by the following two observations. First, splitting and reparameterizing the tensor-borders of the τ-net preserves the first-order Hermite data. That is the τ-surface joins G^1 with the surrounding surface. Second, due to Step (d) and the ρ_j joining C^1 along the boundary segments, the τ-surface is internally C^1. Together this proves the following theorem.

Theorem 1. *The τ-surfaces are internally C^1 and are G^1-connected to the remainder of the surface.*

4. Assessment of spline quality

A regular surface consisting of bi-2 splines has curvature discontinuities. We can therefore not expect the GT-spline-enriched surface constructions to deliver higher-end surfaces when already the surrounding bi-2 complex has sharp transitions in the highlight lines. This is illustrated in the re-meshed bunny shown in Fig. 5: the arrows ↙ point to purely bi-2 on bi-2 transitions of the regular tensor-product spline. For example, the right crease in row τ_1 of Fig. 5 is outside the τ configuration. Such transitions can propagate to boundaries of GT-spline macro-patches. Internally the macro-patches have no such flaws and appear to the naked eye like a single patch.

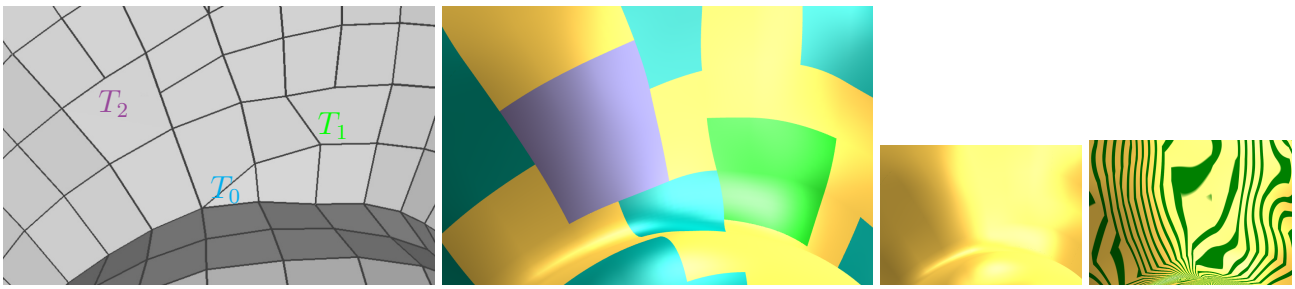
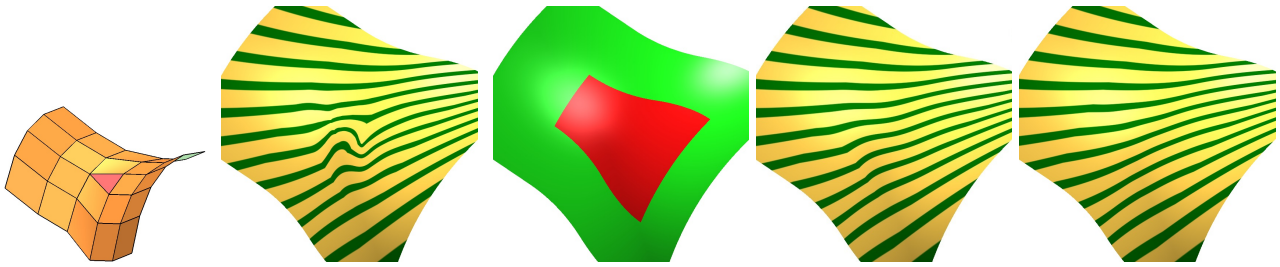


Figure 6: Zoom in above the tail of the bunny Fig. 5. (*middle*): T_0-gons, T_1-gons, T_2-gons, and bi-2 regular tensor-product spline.

Full views of finely-resolved, possibly subdivided meshes as in Fig. 1b or Fig. 5, *top*, are clearly not useful for testing the quality of the τ-constructions. The validation of spline surface quality must zoom in on quad-dominant sub-meshes as in Fig. 5, bottom four rows, or Fig. 6 – or focus on engineered τ-nets, collections of τ-nets and tight combinations of T-gons as discussed below. To inspect the transition from the regular spline and calibrate quality expectations, we surround isolated τ-nets such as in Fig. 7a by one or more layers of quads that define a surrounding surface annulus. The so-extended net is called a τ^+-net.

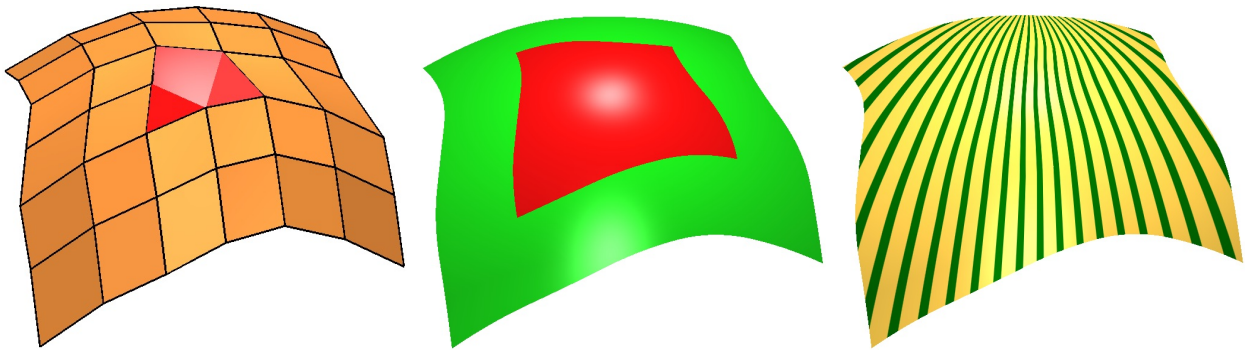
Fig. 7 demonstrates shape artifacts of Doo-Sabin subdivision surfaces, typical for hyperbolic configurations. Achieving good highlight line distributions for this input mesh is not trivial. Mimicking the τ_1-construction, by assembling the central vertical layer (see Fig. 4d) from two C^1-connected quadratic curves (represented in degree 3 form), still yields slightly oscillating highlight lines, see Fig. 7c,middle. Only the more complex formulas of the default construction in Appendix B yields acceptable shape Fig. 7c,right. (Bi-2 C^1 surfaces usually do not achieve highlight line distributions suitable for outer surface design, but can provide sufficient smoothness for interior features.)

Input τ_2-nets, such as Fig. 8a, are difficult since T-junctions from two directions concentrate at one spot. Fig. 8c demonstrates good highlight line distribution, even for convex inputs that present a challenge for subdivision and finite constructions alike. Remarkably, the highlight lines do not betray the location of the τ_2-surface.



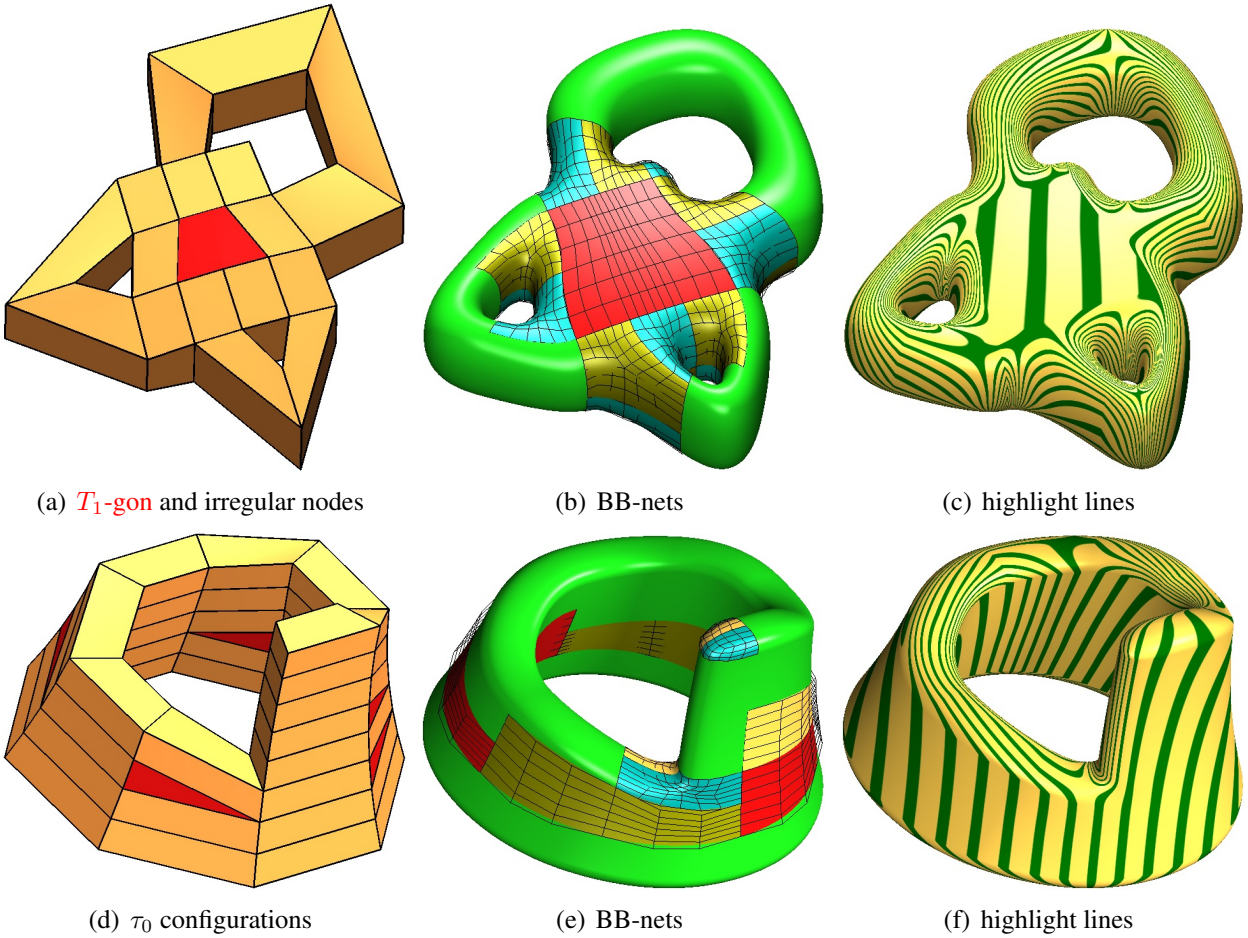
(a) τ_0^+-net (b) Doo-Sabin (c) τ_0 construction: τ_1-like and default

Figure 7: Comparison of constructions for hyperbolic τ_0 shape. In (c) the τ-surface is colored red; here and in the other examples the regular bi-2 patches are colored green.



(a) τ_2-net (b) bi-2 frame bi-3 τ-surface (c) highlight lines

Figure 8: τ_2-construction. (The τ_2-gon in (a) is rendered as two triangles and one brighter quad.)



(a) T_1-gon and irregular nodes (b) BB-nets (c) highlight lines

(d) τ_0 configurations (e) BB-nets (f) highlight lines

Figure 9: Designs with τ_0- and τ_1-nets and n-valent (irregular) points, $n = 3, 5$. (b) Surface layout with tensor-product bi-2 splines, τ-surfaces and multi-sided G_{KP} caps. All bi-3 patches are overlaid with their BB-net.

Fig. 9 *top*, illustrates efficient minimalist design with lqd meshes: the outer nodes of the τ_1-net in (a) are almost all irregular. The multi-sided regions are covered by bi-3 G_{KP} surface caps. The highlight lines are mostly smooth and only change noticeably where two irregular regions abut. Fig. 9 *bottom*, demonstrates the usefulness of τ_0-nets for rapidly reducing the number of strips without global coordination of knot sequences (that would be impossible for these examples). The only visible kinks in the highlight lines are between pieces of the tensor-product bi-2 spline.

The flexibility of the control net and simplicity of the pieces enables some appealing designs. As with tensor-product patches, some *basic rules* should be followed.

- T-gons should be aligned with creases.
- T-gons should not have many more T-junctions on one side than on the other.

Violating the first rule yields artifacts akin to the 'camel-back' oscillations in tensor-product splines with control points elevated along a diagonal. Violating the second rule corresponds to poorly graded transitions. Let r denote the ratio of the vertex count at the sparse side divided by the dense side of the T-gon, e.g. 1:2 for τ_0 since there is one node at the sparse and two at the dense end. Since $\alpha(u) := 1 - (1 - r)u$ in Table 2, the transversal derivative shrinks from 1 to r along the edge. For $[\tau_0, \tau_1, \tau_2]$, the ratios are $r = [1{:}2, 2{:}3, 2{:}3]$. We were unable to find good bi-3 constructions when the r drops below these ratios, i.e. when the contraction is faster.

All major formulas were derived and verified by symbolic computation (Maple). A C++ implementation of the formulas, leveraging a generic half-edge graph traversal data structure, computes the surfaces at the speed of B-spline to Bézier conversion. The input meshes of Fig. 5 and Fig. 6 were generated using the code of [12]. All surfaces were rendered by BezierView [31].

5. Conclusion

Using locally quad-dominant τ meshes as control nets, the new class of GT-splines adds modeling flexibility that can be used to balance mesh and spline complexity. While similar constructions based on splitting and reparameterization exist for additional τ patterns, in practice the patterns where T-joints occur on opposing sides of the T-gon can be often reduced to one of τ_0, τ_1 and τ_2 by inserting edges. When multiple T-joints occur all on one side, the parameterization changes rapidly and such poorly graded transitions typically result in poor shape.

In combination with G_{KP} and bi-2 splines, GT-splines offer an efficient and principled way for smoothing lqd-meshes: the G_{KP} algorithm fills multi-sided holes in a bi-2 spline complex and GT-splines provide a low-degree solution where quad-strips end. Meshes yielding smooth surfaces of good quality therefore have a clean characterization as lqd meshes. We hope that this insight motivates more spline-aware re-meshing algorithms.

Acknowledgements. This work was supported in part by DARPA HR00111720031 and NIH R01 EB018625. Kyle Lo implemented the code and rendered Fig. 5 and Fig. 6.

References

[1] Pierre Alliez, David Cohen-Steiner, Olivier Devillers, Bruno Lévy, and Mathieu Desbrun. 2003. Anisotropic polygonal remeshing. In *Proceedings of ACM SIGGRAPH 2003 (ACM Transactions on Graphics)*, Jessica Hodgins and John C. Hart (Eds.), Vol. 22(3). ACM Press, 485–493.

[2] Klaus-Peter Beier and Yifan Chen. 1994. Highlight-line algorithm for realtime surface-quality assessment. *Computer-Aided Design* 26, 4 (1994), 268–277.

[3] David Bommes, Marcel Campen, Hans-Christian Ebke, Pierre Alliez, and Leif Kobbelt. 2013. Integer-grid maps for reliable quad meshing. *ACM Trans. Graph* 32, 4 (2013), 98:1–98:12.

[4] E. Catmull and J. Clark. 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design* 10 (Sept. 1978), 350–355.

[5] C. de Boor. 1978. *A Practical Guide to Splines*. Springer.

[6] Carl de Boor. 1987. B-Form Basics, Gerald Farin (Ed.). SIAM, Philadelphia, 131–148.

[7] Jiansong Deng, Falai Chen, and Liangbing Jin. 2013. Dimensions of biquadratic spline spaces over T-meshes. *J. Computational Applied Mathematics* 238 (2013), 68–94.

[8] Tony D DeRose. 1990. Necessary and sufficient conditions for tangent plane continuity of Bezier surfaces. *Computer Aided Geometric Design* 7, 1 (1990), 165–179.

[9] Tor Dokken, Tom Lyche, and Kjell Fredrik Pettersen. 2013. Polynomial splines over locally refined box-partitions. *Computer Aided Geometric Design* 30, 3 (2013), 331–356.

[10] D. Doo and M. Sabin. 1978. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design* 10 (1978), 356–360.

[11] Gerald Farin. 1988. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press.

[12] Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. 2015. Instant field-aligned meshes. *ACM Trans. Graph* 34, 6 (2015), 189:1–189:15.

[13] Felix Kälberer, Matthias Nieser, and Konrad Polthier. 2007. QuadCover - Surface Parameterization using Branched Coverings. *Comput. Graph. Forum* 26, 3 (2007), 375–384.

[14] Hongmei Kang, Jinlan Xu, Falai Chen, and Jiansong Deng. 2015. A new basis for PHT-splines. *Graphical Models* 82 (2015), 149–159.

[15] Kęstutis Karčiauskas, Daniele Panozzo, and Jörg Peters. 2017. T-junctions in spline surfaces. *ACM Tr on Graphics, ACM Siggraph* 36, 5 (2017), 170:1–9.

[16] Kęstutis Karčiauskas and Jörg Peters. 2015. Point-augmented biquadratic C^1 subdivision surfaces. *Graphical Models* 77 (2015), 18–26.

[17] Kęstutis Karčiauskas and Jörg Peters. 2015. Smooth multi-sided blending of biquadratic splines. *Computers & Graphics* 46 (2015), 172–185.

[18] Kęstutis Karčiauskas and Jörg Peters. 2019. High quality refinable G-splines for locally quad-dominant meshes with T-gons. *Computer Graphics Forum* 38, 5 (Aug 2019), 151–161.

[19] Kęstutis Karčiauskas and Jörg Peters. 2019. Localized G-splines for quad & T-gon meshes. *Computer Aided Geometric Design* 71 (May 2019), 244–254.

[20] Kęstutis Karčiauskas and Jörg Peters. 2019. Refinable smooth surfaces for locally quad-dominant meshes with T-gons. *Computers& Graphics* 82 (Aug 2019), 193–202.

[21] Denis Kovacs, Justin Bisceglio, and Denis Zorin. 2015. Dyadic T-mesh Subdivision. *ACM Trans. Graph.* 34, 4, Article 143 (July 2015), 12 pages.

[22] Rainer Kraft. 1998. *Adaptive und linear unabhängige Multilevel B-Splines und ihre Anwendungen*. Ph.D. Dissertation. University of Stuttgart.

[23] Yu-Kun Lai, Leif Kobbelt, and Shi-Min Hu. 2008. An incremental approach to feature aligned quad dominant remeshing. In *Symposium on Solid and Physical Modeling*, Eric Haines and Morgan McGuire (Eds.). ACM, 137–145.

[24] Charles Loop and Tony DeRose. 1990. Generalized B-Spline Surfaces of Arbitrary Topology. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, Forest Baskett (Ed.), Vol. 24. ACM, 347–356.

[25] Ashish Myles, Nico Pietroni, and Denis Zorin. 2014. Robust Field-aligned Global Parametrization. *ACM Trans. Graph.* 33, 4, Article 135 (July 2014), 14 pages.

[26] Steven Owen, Matthew Staten, Scott Canann, and Sunil Saigal. 1999. Q-MORPH: an indirect approach to advancing front quad meshing. *Itl J Numer Methods Eng* 44 (03 1999), 1317–1340.

[27] J. Peters. 2002. Geometric Continuity. In *Handbook of Computer Aided Geometric Design*. Elsevier, 193–229.

[28] Ulrich Reif. 1995. Biquadratic G-spline surfaces. *Computer Aided Geometric Design* 12, 2 (1995), 193–205.

[29] Nico Schertler, Marco Tarini, Wenzel Jakob, Misha Kazhdan, Stefan Gumhold, and Daniele Panozzo. 2017. Field-aligned online surface reconstruction. *ACM Trans. Graph* 36, 4 (2017), 77:1–77:13.

[30] Thomas W. Sederberg, Jianmin Zheng, Almaz Bakenov, and Ahmad Nasri. 2003. T-splines and T-NURCCs. In *Proceedings of ACM SIGGRAPH 2003 (ACM Transactions on Graphics)*, Jessica Hodgins and John C. Hart (Eds.), Vol. 22(3). ACM Press, 477–484.

[31] M. Malciewicz) SurfLab (J. Peters, X. Wu. 1990–2020. BézierView: a light weight viewer for analyzing Bézier patches. (1990–2020).

[32] Chao Zeng, Fang Deng, Xin Li, and Jiansong Deng. 2015. Dimensions of biquadratic and bicubic spline spaces over hierarchical T-meshes. *J. Computational Applied Mathematics* 287 (2015), 162–178.

[33] Chao Zeng, Meng Wu, Fang Deng, and Jiansong Deng. 2016. Dimensions of spline spaces over non-rectangular T-meshes. *Adv. Comput. Math* 42, 6 (2016), 1259–1286.

Appendix A: τ_2 center point construction

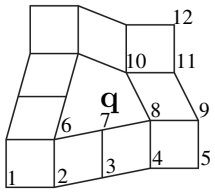


Figure 10: Labeling of the τ_2-net.

We execute Step (d) of the Algorithm for τ_2-nets with the central BB-coefficient $\mathbf{q}$ undetermined – in order to solve for it. For fixed tensor-borders and central coefficient, the four rules in Fig. 11 define all coefficients of the layers in Steps 1-4. Then $\mathbf{q}$ is the minimizer of the functional $\mathcal{F}_3 f := \int_0^1 \int_0^1 \sum_{i+j=3, i,j\geq 0} \frac{3!}{i!j!} (\partial_s^i \partial_t^j f(s,t))^2 ds dt$, summed over all 16 patches of the τ-surface. For ease of reproduction, the weights w_j of the resulting formula $\mathbf{q} := \sum_{j=1}^{20} w_j \mathbf{c}_j$ in terms of the τ_2-net vertices $\mathbf{c}_j$ labeled in Fig. 10 are

$$\begin{array}{lcccccc} w_{1\ldots6}: & 0.01 & -0.014 & -0.019 & -0.014 & 0.008 & 0.071 \\ w_{7\ldots12}: & 0.217 & 0.179 & -0.022 & 0.286 & -0.02 & 0.003 \end{array}$$

(Due to diagonal symmetry only 12 need to be listed). The weights are adjusted to have 3 digits and to sum to 1.

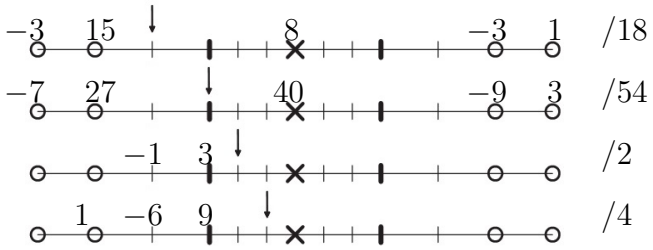
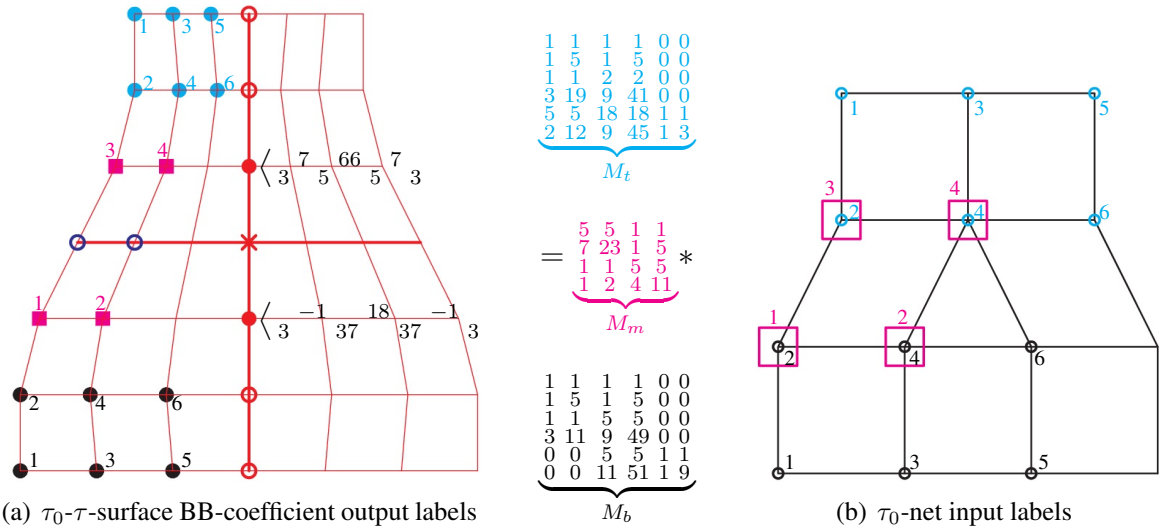


Figure 11: Four construction stencils (weights sum to 1) of C^2-connected layers that form two C^2-connected cubic curves in four pieces. The curves are defined by two coefficients (marked as the circles) at either end and the central BB-coefficient of the layer (marked as $\times$).



(a) τ_0-τ-surface BB-coefficient output labels

(b) τ_0-net input labels

Figure 12: The τ_0 construction formulas (stencils): for example, each of the four ■ in (a) is a linear combination, with weights in a row of the magenta matrix, of the four □ in (b): $\overset{2}{■} = (7\overset{1}{□} + 23\overset{2}{□} + \overset{3}{□} + 5\overset{4}{□})/36.$

Appendix B: τ_0 construction via stencils

This Appendix and Fig. 12 provide explicit formulas. (stencils) for all BB-coefficients in terms of the τ-net. Each row of the three matrices M_t,M_m and M_b in Fig. 12 lists the un-normalized weights (to be normalized to sum to 1) to generate respectively 6, 4 and 6 BB-coefficients in Fig. 12a from the corresponding color-coded sub-net of the τ_0-net in Fig. 12b. Each black • is an affine combination of the ∘ with weights from one row of the black matrix M_b: $\circ^j$ contributes $M_b(i,j)$ to $\bullet^i$. For example $\bullet^3 := (\circ^1 + \circ^2 + 5\circ^3 + 5\circ^4)/12$. Analogously, $\bullet^i$ is an affine combination of $\circ^j$ with weights in M_t, and $■^i$ is an affine combination of $□^j$ with weights listed in the columns of M_m. Symmetry determines the mirrored construction.

BB-coefficients marked ∘ in Fig. 12a are midpoints between two $■^j$. BB-coefficients marked ∘ are averages of their 'left' and 'right' neighbors. BB-coefficients marked • are weighted sums of the two middle rows of 3+4 B-spline coefficients in Fig. 12b with 7 weights (scaled by 96) correspondingly laid out to the right of ⟨ in Fig. 12a. The BB-coefficient, marked as ×, is the average of its neighbors •.