Splines for fast-contracting polyhedral control nets

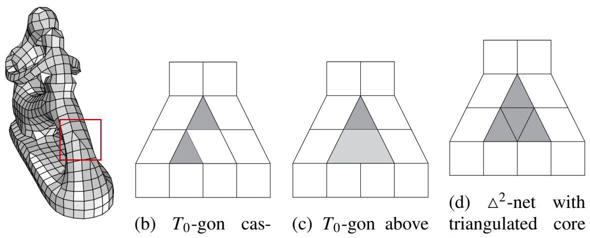
Erkan Gunpinar[a], Kęstutis Karčiauskas[b], Jörg Peters[c,*]

[a] *Istanbul Technical University, Turkiye*
[b] *Institute of Mathematics, Vilnius University, Lithuania*
[c] *Department CISE, University of Florida, USA*

Abstract

Rapid reduction in the number of quad-strips, to accommodate narrower surface passages or reduced shape fluctuation, leads to configurations that challenge existing spline surface constructions. A new spline surface construction for fast contracting polyhedral control-nets delivers good shape. A nestedly refinable construction of piecewise degree (2,4) is compared with a uniform degree (3,3) spline construction.

Keywords: control-net contraction, polyhedral-net spline, geometric continuity

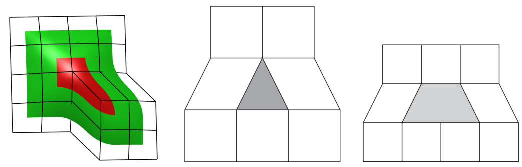


(a) (b) T_0-gon cascade (c) T_0-gon above T_1-gon (d) $\triangle^2$-net with triangulated core (gray)

Figure 1: Configurations for rapid contraction: (a) shows fast contraction 'in the wild'. (b) cascading triangles (T_0-gons), (c) T_0-gon + T_1-gon (a pentagon with 3-valent vertex), (d) new $\triangle^2$-net with triangulated gray core as generalization of (b) and (c): removing one edge yields (b), while removing both bottom edges yields (c).

1. Introduction

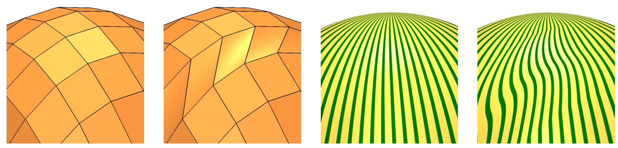
Quadrilateral control-nets for surfaces typically follow a sampled or imagined cross field to capture two orthogonal directions of shape variation. Where such regular, tensor-product nets meet, n-gons, polar or star-like configurations arise. These configurations have been the focus of numerous surface constructions (see the review in Section 1.1). By contrast, when the goal is to accommodate narrower surface passages or reduced shape fluctuation, a rapid reduction in the number of parallel quad-strips is needed. Patterns like Fig. 2 b,c achieve slow contraction. However, fast re-meshing algorithms such as [1, 2] and some hand-made quad-dominant meshes implement rapid contraction, see Fig. 5 , and pack contracting mesh configurations too tightly for the meshes to serve as control nets for the existing slow contraction spline surface algorithms: typically, irregular net configurations need to be separated by a border of quadrilaterals. Available mitigation range from ad hoc designer intervention, to an improved Doo-Sabin refinement step [3, 4], to special re-meshing rules for T_0- and T_1-locations, [5]. The drawback of these approaches is not just an increase in the number of patches, but a



(a) 5-3 neighborhood (b) τ_0-net (c) τ_1-net

Figure 2: (a) Diagonal 5-3 contraction in quad mesh design. (b,c) Single, direction-aligned contraction.

decrease in the surface quality as the natural cross field (flow) of the geometry is altered to enforce the necessary combinatorial structure. Fig. 3 a illustrates, for simplicity in a regular B-spline mesh, how a change of connectivity causes oscillation in the highlight line distribution. This change of flow is also a drawback of the frequently employed configuration with the nodes of valencies 5 and 3 in one quad, e.g. Fig. 2 a.



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

Figure 3: A re-connection (a) → (b) that ignores the flow of the principle curvature cross field can lead to surface artifacts (c) → (d) already for a regular tensor-product control net.

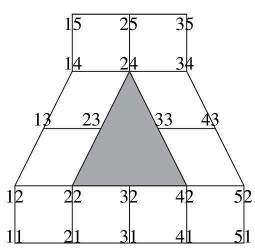


Figure 4: Rapid contraction $\triangle^2$-net with the labels of its 20 nodes $\mathbf{d}_{ij}$.

This paper offers a new, rapid contraction option to the set

*Corresponding author
Email addresses: gunpinar@itu.edu.tr (Erkan Gunpinar), kestutis.karciauskas@mif.vu.lt (Kęstutis Karčiauskas), jorg.peters@gmail.com (Jörg Peters)

of admissible irregular control nets within a bi-quadratic (bi-2) spline control net. The resulting Fast Contraction (FC) surface pieces join abutting regular tensor-product surface with an empirically good highlight line distribution [6].

The underlying fast-contracting net, typeset as $\triangle^2$-net, is shown in Fig. 4 . The internal partition of the $\triangle^2$-*core* (gray in Fig. 1) can be and is ignored, because the interior partition does not enter into the constraints for a smooth join of the FC surface with the surrounding surface. Ignoring the interior partition allows for a joint treatment of the configurations Fig. 1 b,c,d. Since the outer nodes of the $\triangle^2$-net can have any valence, $\triangle^2$-nets can share quad facets with other irregular configurations and the resulting FC surface can share points or boundary segments with those of other overlapping non-regular net configurations. Fig. 5 shows some cascade-nets arising 'in the wild' as output from quad-dominant meshing methods [1, 7, 2].

In summary, the contributions are

- a C^1 surface FC^4 for $\triangle^2$-nets consisting of three degree (2,3) patches and six degree (2,4) patches, forming together a 3×3 macro-patch. FC^4 joins by default G^1 with any surrounding bi-2 (B-)splines or polyhedral-net splines [8];
- a proof that no analogous surface construction exists that uses 3×3 pieces of degree bi-3;
- an alternative 11-piece FC^3 surface construction of degree bi-3, suitable for extending the range of polyhedral-net splines [8].

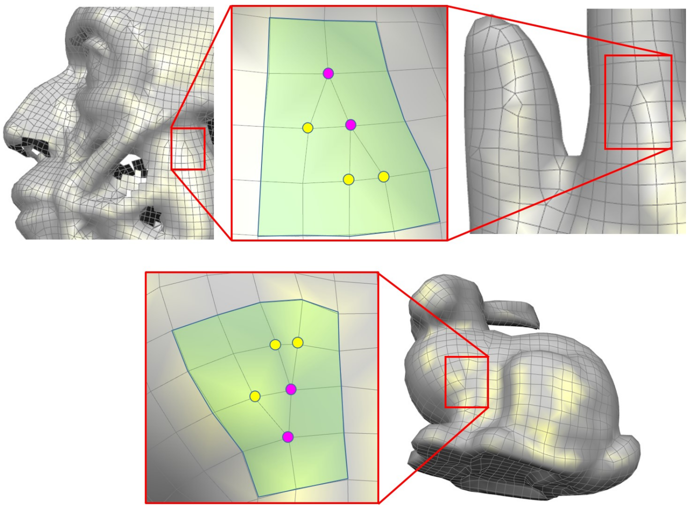


Figure 5: Cascading triangles from quad-dominant meshing methods [1, 7, 2]. Colors indicate valence 5 and valence 4 triangle vertices.

1.1. FC^4 and FC^3 surfaces in the zoo of surface constructions

An irregular configuration within an otherwise regular, grid-like, tensor-product net, can be associated with a variety of surface representations. The three major families of surface constructions are singular surface parameterizations, rational multi-sided surfaces such as [9, 10, 11], and geometrically continuous surfaces. Singular surface parameterizations include both classic subdivision surfaces [3, 12, 13] and recent algorithms built on the idea of a guide shape [14, 15]; alternative singular parameterizations use edge collapse, e.g. polar surfaces [16, 17, 18], and vertex singularity [19, 20, 21, 22]. Other surface algorithms use rational singularities [23, 24].

FC surfaces fall into the family of geometrically smooth surface constructions. Geometrically smooth surface constructions assemble a finite number of polynomial pieces to join smoothly after a change of variables. Smoothness ranges from curvature continuous surfaces of degree bi-7 [25] or degree bi-6 [26], to lower-degree tangent-continuous splines [27, 28, 29, 30]. Note that satisfying the algebraic smoothness constraints does not by default yield surfaces with acceptable highlight line distributions. Consequently, several publications focus on empirically good highlight lines. Examples of bi-5 caps are [31], and the macro-patch bi-4 caps of [32, 33]. There are even bi-3 caps with very small normal mismatch [34] that result in good highlight line distributions. Additionally, there is a hybrid family that combine a finite number of subdivision rings with a cap to serve as nestedly refined space for engineering analysis, e.g. [35].

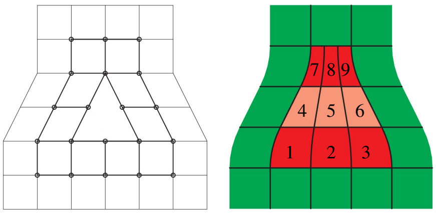
We focus on geometrically smooth constructions for irregularities in a C^1 bi-quadratic (bi-2) tensor-product surface. Bi-2 splines have minimal polynomial bi-degree for smoothing out a quadrilateral mesh. The classic generalization of bi-2 splines [3] consists of an infinite sequence of nested (contracting) bi-2 polynomial surface rings but fails to yield good shape due to artifacts generated already in the first steps. Augmented Subdivision [4] improves shape by adding a carefully chosen central guide point. Polyhedral-net splines [8] generalize tensor-product bi-quadratic (bi-2) splines by combining algorithms from [36, 37, 18] that use a *finite* number of polynomial pieces of degree at most bi-cubic (bi-3). The degree bi-2 construction [38] is degree-wise optimal, but has unsatisfactory shape.

Another type of non-regular mesh configurations are τ_0-, τ_1-nets [39, 40], see Fig. 2 b,c. Their cores, displayed in grey, are called T_0-gon (a triangle, but with particular vertex valences 4,4,5), respectively T_1-gon (a pentagon with vertex valences 3,4,4,4,4). For the treatment of τ_1-nets T-splines [41] come to mind, but T-splines are primarily useful to refine an existing quad partition, and are known to fail, due to their global parameterization requirement for the prescribed reductions in the number of quad strips, see [42, Fig 2], [43, Fig 6]. For τ-nets, smooth surfaces of bi-degree $(2,4)$ ([39]) or bi-3 ([40]) can be produced that, together with a surrounding spline, form a smooth surface of good quality.

The $\triangle^2$ algorithm is partly motivated by the output of quad-dominant meshing algorithms such as [1, 2], that avoid the complexity and higher quad-count of strict quad-meshing algorithms by introducing (fast) mesh contractions: while high resolution meshes are almost always avoid rapid contraction, the desirable low quad-count typically results in $\triangle^2$ configurations.

2. Setup

Classic tensor-product spline control nets have two distinguished directions, and so do τ_0, τ_1 and $\triangle^2$ nets. However, for the latter three, in one direction (vertical in Fig. 6) the number of mesh lines is reduced or expanded. (In the following, 'vertical' and 'horizontal' refer to the standard layout in Fig. 6 .) Although the output of the $\triangle^2$ construction are tensor-product macro-patches, the changing number of mesh lines forces a change of parameterization to achieve *geometric continuity*. Compared to τ_0 and τ_1 constructions, the $\triangle^2$ construction is more challenging due to an increased number of coefficients that do not enter formal smoothness constraints but whose careful choice is crucial for good final shape. The $\triangle^2$ macro-patch partion for the main algorithm, FC^4, is shown in Fig. 6 b: patches 1,2,3,7,8,9 are of bi-degree $(2,4)$, patches 4,5,6 are of bi-degree $(2,3)$ where the first degree, 2, refers to the degree in the horizontal direction.



(a) Extended net (b) Surface pieces

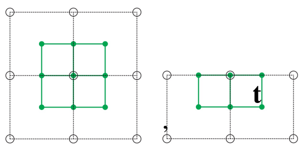
Figure 6: (a) A $\triangle^2$-net surrounded by a ring of quads.The $\triangle^2$ construction requires only the $\triangle^2$-net (inner mesh). The ring of quads is used to additionally generate (b) a ring of uniform bi-quadratic (bi-2) C^1 spline patches to allow judging the quality of transition to the regular spline complex. The inner, red 9-piece macro-patch corresponds to the FC surface.

2.1. Parameterization

The macro-patch FC surfaces consist of tensor-product pieces of polynomial bi-degree (d, d') in Bernstein-Bézier form (*BB-form*, [44]). That is, for Bernstein polynomials $B^d_k(t) := \binom{d}{k}(1-t)^{d-k}t^k$, the patch $\mathbf{p}$ of bi-degree (d, d') is defined as

$$\mathbf{p}(u, v) := \sum_{i=0}^{d}\sum_{j=0}^{d'} \mathbf{p}_{ij} B^d_i(u) B^{d'}_j(v), \quad 0 \le u, v \le 1.$$

With the convention that d denotes the polynomial degree in the parameter tracing out the horizontal direction, the bi-degrees in addition to the regular bi-2 uniform B-spline patches are $(2, 4)$, $(2, 3)$ for FC^4 and $(3, 3)$ for FC^3. Connecting the *BB-coefficients* $\mathbf{p}_{ij} \in \mathbb{R}^3$ to $\mathbf{p}_{i+1,j}$ and $\mathbf{p}_{i,j+1}$ wherever well-defined yields the *BB-net.*



(a) bi-2 (b) tensor-border

Figure 7: B-to-BB conversion and tensor-borders $\mathbf{t}$ as Hermite input data. Circles $\circ$ mark B-spline control points, solid disks $\bullet$ mark BB-coefficients of the full patch, respectively tensor-border.

2.2. Conversion from B- to BB-form and tensor-borders

Any 3×3 grid can be interpreted as the control net of a uniform bi-2 spline in uniform knot B-spline form. In Fig. 7 the B-spline control points are marked $\circ$. The *B-to-BB conversion* (e.g. by knot insertion) expresses the spline in bi-2 BB-form illustrated by the green BB-nets in Fig. 7. Conversion of a partial sub-grid yields a partial BB-net $\mathbf{t}$, called *tensor-border*, that defines position and first derivatives across an edge.

2.3. Geometric continuity and reparameterization

Two polynomial pieces $\mathbf{p}$ and $\mathbf{q}$ join G^1 along the common sector-separating curve $\mathbf{p}(u, 0) = \mathbf{q}(u, 0)$ with BB-coefficients $\mathbf{p}_{i0} = \mathbf{q}_{i0}$ if, see e.g. [45],

$$\mathbf{p}(u, v) := \mathbf{q} \circ \rho(u, v), \qquad \rho(u, v) := (u + b(u)v, a(u)v) \tag{1}$$

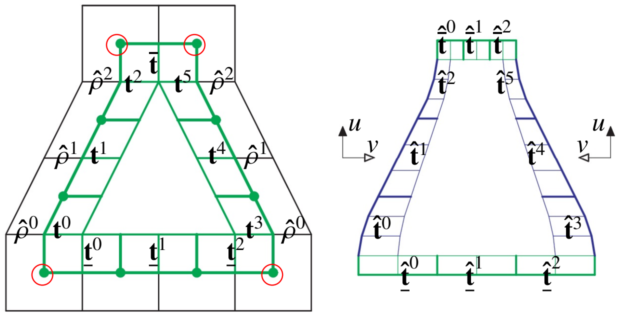
$$\partial_v \mathbf{q}(u, 0) = a(u)\partial_v \mathbf{p}(u, 0) + b(u)\partial_u \mathbf{p}(u, 0), \quad (u, v) \in [0..1]^2. \tag{2}$$

Besides the shared BB-coefficients of the common boundary, only the layers of BB-coefficients $\mathbf{p}_{i1}$ and $\mathbf{q}_{i1}$ of adjacent patches enter the G^1 continuity constraints. In the derivation, u-, v-directions can be assigned as convenient. By default, u is used to parameterize along the boundary and v in the orthogonal direction of the tensor-border, towards the interior or core.

3. The FC^4 construction

FC^4 consists of three layers: three patches of bi-degree $(2, 4)$, three of bi-degree $(2, 3)$ and again three patches of bi-degree $(2, 4)$. This choice of layout (red pieces in Fig. 6 b) and degree minimizes the number of free parameters that need to be carefully set for good surface quality as measured by uniformity of highlight line distribution [46]. Moreover, degree 4 in the contracting direction is least to obtain geometrically smooth splines of good quality that are *nestedly refinable*, see [47, 39] and Section 5.

3.1. Tensor-border frame from input Hermite data



(a) input bi-2 data (b) final frame

Figure 8: (a) $\triangle^2$-net and input bi-2 tensor-border frame obtained from the $\triangle^2$-net by B- to BB-conversion; (b) (*left*, *right*) blue reparameterization with $\hat\rho^s$. The *bottom* green tensor-border is obtained by degree-raising, the *top* green by split and degree-raising. Here and in the following figures the u- and v-arrows indicate the bi-2 tensor-border reparameterization directions.

In the following let $s = 0, 1, 2$. The bi-2 tensor-borders are initialized by partial B-to-BB conversion from control-net points $\mathbf{d}_{i,j}$ whose indices are shown in Fig. 4 :

$$\begin{array}{cll}
\textit{left} & \mathbf{d}_{i,s+j},\ i = 1, 2,\ j = 1, 2, 3 & \to \quad \mathbf{t}^s, \\
\textit{right} & \quad\text{symmetric to } \textit{left} & \to \quad \mathbf{t}^{s+3}, \\
\textit{top} & \mathbf{d}_{i,5-j},\ i = 1, 2, 3,\ j = 0, 1 & \to \quad \bar{\mathbf{t}}, \\
\textit{bottom} & \mathbf{d}_{s+i,j},\ i = 1, 2, 3,\ j = 1, 2 & \to \quad \underline{\mathbf{t}}^s.
\end{array}$$

Since they stem from C^1 splines, the resulting adjacent bi-2 tensor-borders are C^1-connected, and their 2×2 overlapping BB-coefficients agree at the four corners, marked in Fig. 8 a by $\circ$. While *left* and *right* sides of the tensor-border frame have matching 3 pieces, the *bottom* consists of 3 but the *top* of only one piece. To match the bottom structure, the *top* must be split, horizontally, into 3 pieces. This destroys consistency with the *left* and *right* tensor-border, to be re-established by reparameterizing the tensor-borders $\mathbf{t}^s$ and $\mathbf{t}^{s+3}$, with $\rho^s(u, v) := (u, a^s(u)v)$ where, to match the maximum degree of the FC^4, a^0 and a^2 can be at most quadratic and so have BB-coefficients

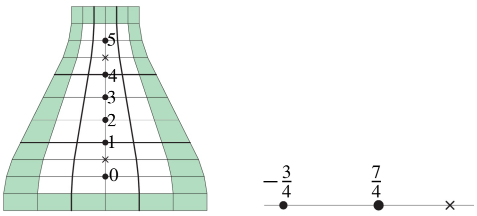
$$\begin{aligned}
&[a^0_0, a^0_1, a^0_2] := [1, 1, \tfrac{5}{6}]; \quad [a^2_0, a^2_1, a^2_2] := [\tfrac{1}{2}, \tfrac{1}{3}, \tfrac{1}{3}] \\
&\text{while } a^1(u) \text{ is linear,} \quad [a^1_0, a^1_1] := [\tfrac{5}{6}, \tfrac{1}{2}].
\end{aligned} \tag{3}$$

Since the functions a^s are C^1-connected, so are the tensor-borders $\hat{\mathbf{t}}^s := \mathbf{t}^s \circ \rho^s$, $\hat{\mathbf{t}}^{3+s} := \mathbf{t}^{3+s} \circ \rho^s$. Since the tensor-borders

$\underline{\mathbf{t}}^s$ and (the split) $\bar{\mathbf{t}}$ are not reparameterized, their degree in horizontal direction is 2. Therefore, $\hat{\mathbf{t}}^0$, $\hat{\mathbf{t}}^2$, $\hat{\mathbf{t}}^3$, $\hat{\mathbf{t}}^5$ are presented in bi-degree $(4,2)$ and $\hat{\mathbf{t}}^1$, $\hat{\mathbf{t}}^4$ in bi-degree $(3,2)$ form. This implies that tensor-borders $\underline{\mathbf{t}}^s$ and (the split) $\bar{\mathbf{t}}$ are presented as $\hat{\underline{\mathbf{t}}}^s$, $\hat{\bar{\mathbf{t}}}^s$ in bi-degree $(2,4)$ form. The resulting tensor-border frame is C^1. Appendix A lists the explicit formulas.

3.2. Setting the free parameters

Fig. 9 a shows as • the 'spine' of 6 BB-coefficients that together with the surrounding tensor-border surface frame (Fig. 8 b and, equivalently, the green net in Fig. 9 a) determine a space of C^1 macro-patches: The two BB-coefficients marked × are defined so that the central vertical curve is C^1, i.e. are set by the stencil of Fig. 9 b. With the vertical spine fixed, C^1 continuity in the horizontal direction defines the remaining BB-coefficients as averages of their two neighbors, one on the spine and the other on the tensor-border.



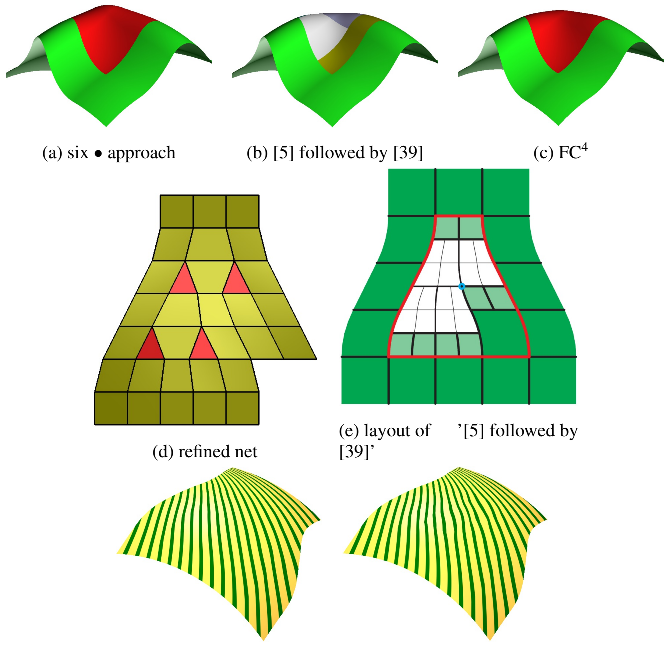
(a) BB-coefficients • (b) stencil for ×

Figure 9: Completion of the FC[4] macro-patch. (a) BB-coefficients indicate that bottom three and top three patches are of bi-degree $(2,4)$ while middle three patches are of bi-degree $(2,3)$. The 'spine' of six $\bullet^s$, $s = 0,\dots,5$ are unconstrained. (b) The stencil (rule) to join C^1 a degree 3 segment with BB-coefficients • to a degree 4 segment with BB-coefficient ×. The shared end-point of the segments is marked as the larger • and weight $7/4$ (index (1) or (4) in (a)).

While formally smooth for any choice, good shape requires a careful choice of the six coefficients marked as •. We found the best choice, among many tested, is to minimize the functional $\mathcal{F}_4 := \int_0^1\int_0^1 \sum_{i+j=4,i,j\ge 0} \frac{4!}{i!j!}(\partial_s^i\partial_t^j f(s,t))^2\, ds\, dt$ over all 9 patches of macro-patch. The resulting surfaces have good highlight line distributions for challenging convex shapes, such as Fig. 11 a,*left*, but not for the $\triangle^2$-net Fig. 11 a,*right* where the surface looks creased at the meeting of the two orthogonal feature lines. Another option is to treat the core of the $\triangle^2$-net asymmetrically as in Fig. 1 b, i.e. as two cascading T_0-gons plus one quad. Refinement of this net using the rules of [5], see Fig. 10 d, yields, asymmetrically, one additional regular bi-2 patch in Fig. 10 e, where regular bi-2 patches are colored green, and additional ones from the refinement, colored light green. Then the algorithm for τ_0-nets in [39] yields a 2×2 macro-patch with pieces of bi-degree $(2,4)$ for each τ_0-net, see Fig. 10 e. The global shape in Fig. 10 b is reasonable, but slightly dips at same location where the initial surface Fig. 10 a peaks sharply.

While many other approaches were investigated, the best choice turned out to merge the $\mathcal{F}_4$ minimization, of the spine of six •, with the refined cascade approach ('[5] followed by [39]'). The resulting surface is displayed in Fig. 10 c. Good highlight line distribution is confirmed by many other challenging inputs (see Limitations in Section 4 for an exception).

Construction summary and precalculated tables. Let $\mathbf{C}_{ini} := \mathbf{p}(\frac{1}{2},\frac{1}{2})$ be the central point of the six • construction, i.e. the center point of the central $(2,3)$ patch labeled 5 in Fig. 6 b, and let $\mathbf{C}_\ell$ be the point marked ∘ in the layout of Fig. 10 e of the '[5] followed by [39]' option. Let $\mathbf{C}_r$ be the analogous point for the left-right flipped cascade configuration. Then we set



(a) six • approach (b) [5] followed by [39] (c) FC[4]
(d) refined net (e) layout of '[5] followed by [39]'
(f) improved (g) [5] followed by [39]

Figure 10: Improving the initial macro-patch. (b) '[5] followed by [39]' is the result of refinement according to [5] followed by [39].

$$\mathbf{C} := (2\mathbf{C}_{ini} + \mathbf{C}_\ell + \mathbf{C}_r)/4 \tag{4}$$

and we proceed, as for the six • approach, to minimize $\mathcal{F}_4$, but now only over five •s since one • is symbolically set to ensure interpolation of $\mathbf{C}$.
Then for $k = 0,\dots,5$,

$$\bullet^k := \sum_{i=1}^{5}\sum_{j=1}^{2}\mu_{ij}^k\mathbf{d}_{ij} + \sum_{i=1}^{4}\mu_{i3}^k\mathbf{d}_{i3} + \sum_{i=1}^{3}\sum_{j=4}^{5}\mu_{ij}^k\mathbf{d}_{ij} \tag{5}$$

is an affine combination of the $\triangle^2$-net nodes $\mathbf{d}_{ij}$ as labeled in Fig. 4. Without loss of quality, the coefficients μ each have 5 decimals accuracy and are corrected by less than 0.00009 so they form a partition of 1. That is, the weights μ_{ij}^s listed in Appendix B are exact, not approximations of the implementation weights.

While the resulting 9 patches can be jointly encoded into a 126×20 matrix M, see below, we present the algorithm explicitly in four steps:

FC[4] Algorithm

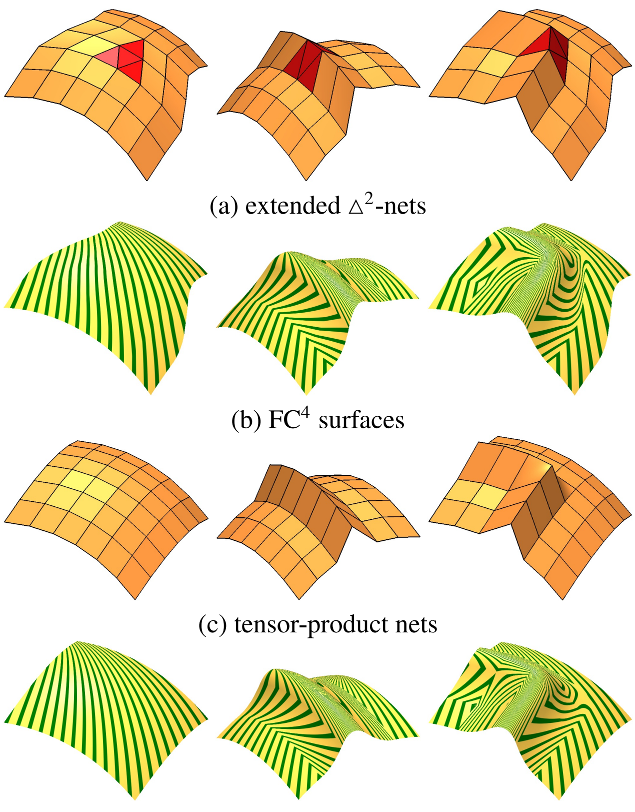
1. Compute the tensor-border frame of Section 3.1 (light green in Fig. 9 a) by B-to-BB conversion and Appendix A formulas.
2. Compute the spine (6 • in Fig. 9 a) as an affine combination of the $\triangle^2$-net nodes by Eq. (5). The weights μ_{ij}^k are listed in Appendix B.
3. Compute the two × in Fig. 9 a using stencil Fig. 9 b.
4. Set all remaining BB-coefficients as $1/2$ equal averages of neighbor BB-coefficients to enforce C^1 continuity in the horizontal direction (see Section 3.2).

Running the algorithm in a symbolic solver, in terms of symbolicly-represented points $\mathbf{d}_{ij}$ of the $\triangle^2$-net, yields the BB-coefficients $\mathbf{b}_k$ as a linear combination of the $\mathbf{d}_{ij}$. The linear combination weights are tabulated in the matrix M. For easy implementation, given M and a generic control net modifier like [8], the code consists of gathering the $\triangle^2$-net in the vector of points $\mathbf{d} \in \mathbb{R}^{20\times 3}$ and computing the vector $\mathbf{b}$ of the BB-coefficients as $\mathbf{b} = M\mathbf{d}$.

4. Analysis: Comparisons, Examples and Alternatives

By construction, the 3×3 FC^4 macro-patch is internally at least C^1 and G^1 connected to the surrounding bi-2 surface. When the $\triangle^2$-net is extended as in Fig. 6 a, a *frame* (colored green in Fig. 6 b) of regular bi-2 patches surrounds the FC^4 surface. This bi-2 frame is important to judge the quality of transition from any surrounding surface to FC^4. As is customary, we assess good shape as uniform highlight line distribution [6]. Extended $\triangle^2$-nets allow shape prediction and emphasize flaws that large compound nets (e.g. Fig. 22) might hide or cause by poor macro-scale mesh layout. For visualization, we show a triangulated $\triangle^2$-net *core* to hint at design intent where features are introduced.

As a baseline, Fig. 11 juxtaposes (a) three challenging $\triangle^2$-net configurations (c) with their regular counterparts (c). Remarkably the highlight line distributions of the tensor-product and FC^4 surfaces are alike.



(a) extended $\triangle^2$-nets

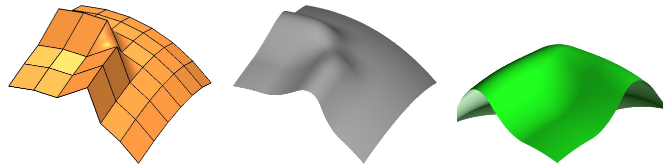
(b) FC^4 surfaces

(c) tensor-product nets

(d) tensor-product bi-2 B-spline surfaces

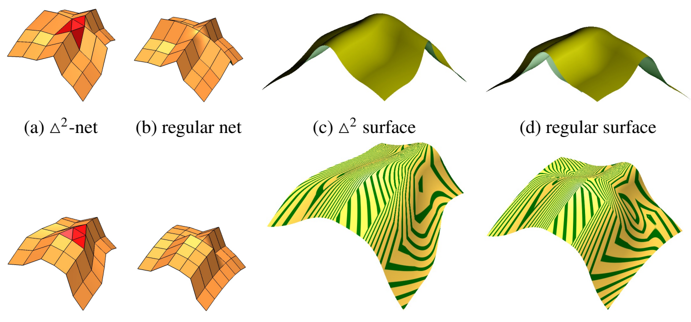
Figure 11: *Row 1:* Extended $\triangle^2$-nets, *Row 3:* Tensor-product control nets. *Row 2,4:* the corresponding surfaces with highlight lines.

The net of Fig. 12 a is a slight modification of that in Fig. 11 (c,*right*): the ridge corner has been pulled to the lower level to better mimic the ridge rounding of Fig. 11 (a,*right*). The resulting uniform bi-2 tensor-product surface Fig. 12 b disappoint: the tensor-product net Fig. 12 a tries to capture a feature 'diagonal' to the two preferred directions of the tensor-product splines, resulting in an undesirable dip. Note that the tensor-product net of Fig. 11 (c,*right*) fully aligns with the preferred directions and therefore succeeds in a sharp turn of the ridge (as does the initial construction displayed in Fig. 10 a.) By contrast, FC^4 models the diagonal direction well: Fig. 12 c reorients FC^4 of Fig. 11 (a,*right*) to show a well-preserved ridge.



(a) B-spline control net (b) bi-2 (B-)spline surface (c) FC^4

Figure 12: What is the regular counterpart of $\triangle^2$-net in Fig. 11 top,right?



(a) $\triangle^2$-net (b) regular net (c) $\triangle^2$ surface (d) regular surface

(e) $\triangle^2$-net (f) regular net (g) $\triangle^2$ surface (h) regular surface

Figure 13: Top row: crossing features. Bottom row: 'squeezing' feature.

Fig. 13 ,*top* shows that crossing features in preferred directions create bumps both for B-spline surfaces and FC^4. Nets (a) and (b) result in similar highlight lines for the more flexible FC^4 as well as the regular tensor-product spline, see Fig. 13 c,d. The reduced highlight line variation on top of the ridge advertises FC^4 for changing mesh line density.

Fig. 14 systematically reviews the effect on FC^4 of feature lines touching or straddling the $\triangle^2$-net core. Row 1 shows a partial, spine-aligned ridge (a) at the contracted and (b) at the spread-out end; both are well-shaped. Row 2 shows a partial, spine-aligned ridege combined with a horizontal ridge. This yields a slight bulge where the ridges meet – as do regular tensor-product surfaces. Row 3 compares a ridge along the contracting direction touching vs cutting through the core. Row 4 tests a full horizontal feature, see also Fig. 11 a,*middle*, with direction change. We applied plain shading to emphasize the global shape and so complement the highlight line distributions of other figures.

Limitations. FC^4 nicely follows the control net and so models the likely design intent in almost all cases of Fig. 14 , including the rounding of the right angle ridge feature. FC^4 seems less appropriate only for a single ridge running in the contraction direction through the core, see Fig. 14 k. We focus on this limitation with Fig. 15 a. Linking to the cross direction causes a dip Fig. 15 b. This same defect is well-known also for T_1-junctions and for T-splines. It is caused by the support of the spline on the ridge incorporating data from its lower neighbors.

The special case can be (partially) mended by replacing $\mathbf{C}$ in

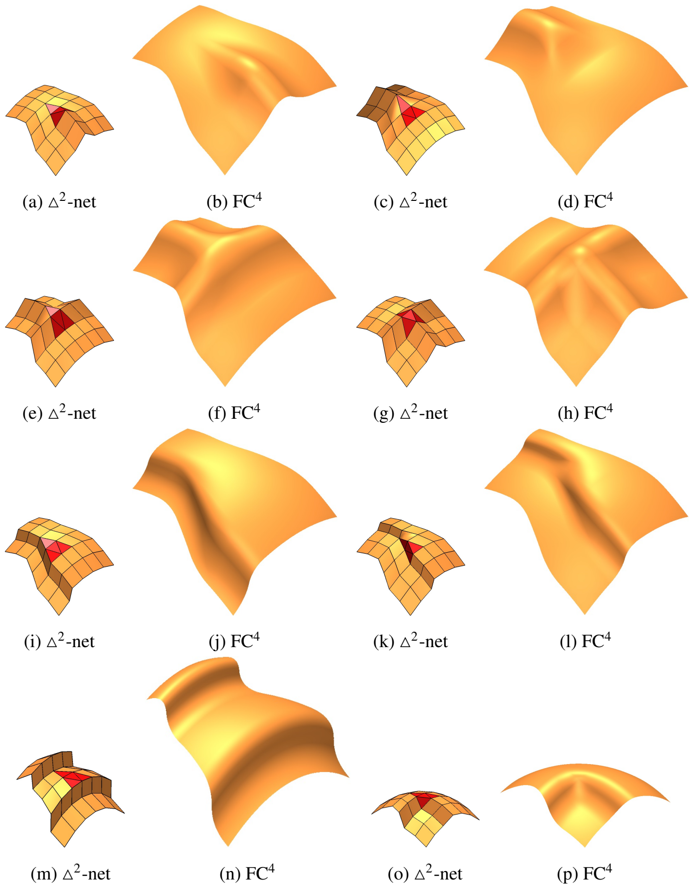


Figure 14: Gallery of feature lines touching or straddling the red core.

(4) by

$$\mathbf{C} := \alpha\frac{\mathbf{d}_{32}+\mathbf{d}_{24}}{2} + (1-\alpha)\frac{\mathbf{d}_{23}+\mathbf{d}_{33}}{2}, \quad \alpha := 0.85 \tag{6}$$

inspired by the core connectivity of Fig. 15 c,d. Here α was determined by experiment comparing highlight line distributions. Proceeding as in first approach (see Section 3.2) minimizing $\mathcal{F}_4$ yields the dip reduction shown in Fig. 15 e. A better choice may be refinement, see Fig. 20 b.

5. Nonexistence of a 3×3 C^1 bi-3 macro-patch that smoothly joins to a given input bi-2 frame

This section shows that an analogous C^1 *bi-3* surface construction that joins smoothly with a given bi-2 surrounding surface is not possible when using a 3×3 macro-patch. Such lower-bound findings characterize the complexity of the task and are therefore of scientific value. A reader purely interested in algorithms might skip this section.

5.1. Left and right reparameterizations

To obtain a 3×3 layout, and since the goal is an internally C^1 macro-patch, the top bi-2 tensor-border is uniformly split into 3 pieces, and the top and bottom borders are degree-raised to bi-3, see Fig. 16 a. To match, the left and right pieces of the tensor-border frame must be reparametrized, see Fig. 8 a: for $s = 0,1,2$

$$\tilde{\mathbf{t}}^s := \mathbf{t}^s \circ \tilde{\rho}^s, \quad \tilde{\mathbf{t}}^{3+s} := \mathbf{t}^{3+s} \circ \tilde{\rho}^{3+s}, \quad \rho^s(u,v) := (u, a^s(u)v).$$

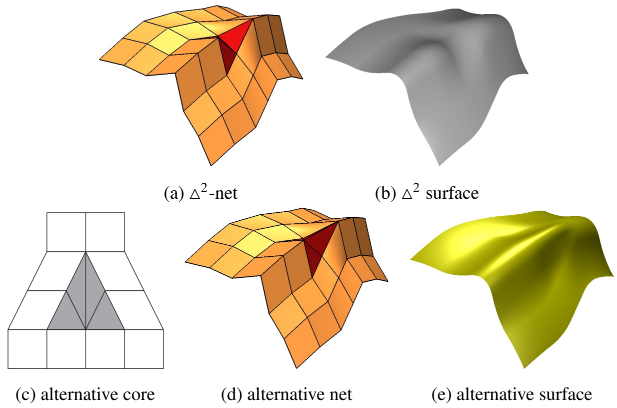


Figure 15: An alternative (rare) triangulation of a core displayed in (c) illustrates possible treatment of a poor net design.

Lemma 1 (left,right frame)**.** *The reparameterization functions $a^s(u)$, $s = 0,1,2$ must be pieces of one uniformly-split linear function $\ell(u) = 1 - u + \frac{1}{3}u$. That is, the BB-coefficients of the pieces are*

$$[a_0^0, a_1^0] = [9,7]/9; \; [a_0^1, a_1^1] = [7,5]/9; \; [a_0^2, a_1^2] = [5,3]/9. \tag{7}$$

Proof. Since the $\tilde{\mathbf{t}}^s$ are to be C^1-connected and not exceed degree 3, the $a^s(u)$ must be linear and C^1-connected, i.e. are part of one linear function ℓ. Consistency of $\tilde{\mathbf{t}}$ with top and bottom boundary curves at the coefficient marked $\circ$ in Fig. 16 a, imply the form (7). $\square$

Then the boundary BB-coefficients $\tilde{\mathbf{t}}^s_{i0}$, $i = 0,\ldots,3$, of the left piece of the frame are those of the those of the boundary quadratics degree-raised to 3. For the interior layer of the tensor-border, omitting the superscripts of the pieces $s = 0,1,2$,

$$\begin{aligned}\tilde{\mathbf{t}}_{01} &:= (1-\frac{2}{3}a_0)\mathbf{t}_{00} + \frac{2}{3}a_0\mathbf{t}_{01},\\ \tilde{\mathbf{t}}_{11} &:= (\frac{1}{3}-\frac{2}{9}a_1)\mathbf{t}_{00} + (\frac{2}{3}-\frac{4}{9}a_0)\mathbf{t}_{10} + \frac{2}{9}a_1\mathbf{t}_{01} + \frac{4}{9}a_0\mathbf{t}_{11}.\end{aligned} \tag{8}$$

The remaining BB-coefficients $\mathbf{t}_{21}$ and $\mathbf{t}_{31}$ are defined by the symmetry $\mathbf{t}_{ij} \leftrightarrow \mathbf{t}_{2-i,j}$, $i = 0,1,2$, $j = 0,1$; $a_i \leftrightarrow a_{1-i}$, $i = 0,1$. By the same logic applied along the right border, $\tilde{\rho}^{3+s} := \tilde{\rho}^s$.

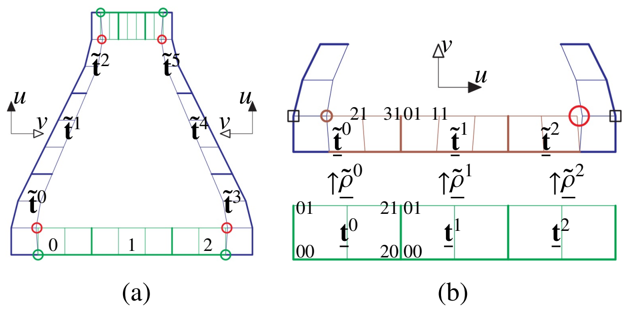


Figure 16: (a) Mismatch at the locations marked $\circ$. (b) Labels and markers $\square$, $\circ$, $\bigcirc$ for the proof of Theorem 1.

Now only a mismatch of BB-coefficients remains at the four locations marked as $\circ$ in Fig. 16 a.

5.2. Focus on the corner mismatch

To resolve the *bottom* mismatch at $\circ$, the bottom bi-2 tensor-borders $\underline{\mathbf{t}}^s$, $s = 0,1,2$, (note the under-bar) must be reparameterized:

$$\tilde{\underline{\mathbf{t}}}^s := \underline{\mathbf{t}}^s \circ \underline{\tilde{\rho}}^s, \quad \underline{\tilde{\rho}}^s := (u + b^s(u)v, v). \tag{9}$$

Lemma 2 (bottom frame)**.** *The reparameterization functions $b^s(u)$, $s = 0, 1, 2$, are of degree 2 and the leftmost, b^0 has BB-coefficients*

$$[b_0^0, b_1^0, b_2^0] = [0, -\frac{1}{9}, 0]. \tag{10}$$

Proof. For $\tilde{\underline{\mathbf{t}}}^s$ not to exceed degree 3, $b^s(u)$ can have up to degree 2. The BB-coefficients $\tilde{\underline{\mathbf{t}}}_{i0}^s$, $i = 0..3$, define the input quadratic boundary segment in degree-raised form. Omitting the superscript, we express the reparameterized BB-coefficients in terms of the bottom tensor-borders:

$$\tilde{\underline{\mathbf{t}}}_{01} := (\frac{1}{3} - \frac{2}{3}b_0)\underline{\mathbf{t}}_{00} + \frac{2}{3}b_0\underline{\mathbf{t}}_{10} + \frac{2}{3}\underline{\mathbf{t}}_{01}; \tag{11}$$

$$\tilde{\underline{\mathbf{t}}}_{11} := (\frac{1}{9} - \frac{4}{9}b_1)\underline{\mathbf{t}}_{00} + (\frac{2}{9} - \frac{2}{9}b_0 + \frac{4}{9}b_1)\underline{\mathbf{t}}_{10} + \frac{2}{9}b_0\underline{\mathbf{t}}_{20} + \frac{2}{9}\underline{\mathbf{t}}_{01} + \frac{4}{9}\underline{\mathbf{t}}_{11};$$

$$\tilde{\underline{\mathbf{t}}}_{21} := -\frac{2}{9}b_2\underline{\mathbf{t}}_{00} + (\frac{2}{9} - \frac{4}{9}b_1 + \frac{2}{9}b_2)\underline{\mathbf{t}}_{10} + (\frac{1}{9} + \frac{4}{9}b_1)\underline{\mathbf{t}}_{20} + \frac{4}{9}\underline{\mathbf{t}}_{11} + \frac{2}{9}\underline{\mathbf{t}}_{21}$$

$$\tilde{\underline{\mathbf{t}}}_{31} := -\frac{2}{3}b_2\underline{\mathbf{t}}_{10} + (\frac{1}{3} + \frac{2}{3}b_2)\underline{\mathbf{t}}_{20} + \frac{2}{3}\underline{\mathbf{t}}_{21}.$$

We now compare the four coefficients with index 00, 10, 01, 11, where the left and the bottom frame overlap. In particular, we use the formulas (8) and (11) and the fact that the input bi-2 tensor-borders are consistent, i.e $\underline{\mathbf{t}}_{ij}^0 = \underline{\mathbf{t}}_{ji}^0$, $i = 0, 1$, $j = 0, 1$. Since the outer boundaries of the reparameterized tensor-borders are degree-raised boundary quadratics, inserting the values for $a_0^0 = 1$ and $a_1^0 = 7/9$, equating the tensor-border BB-coefficients $\tilde{\mathbf{t}}_{10}^0 = \tilde{\underline{\mathbf{t}}}_{01}^0$ (marked □) in Fig. 16) results in

$$\tilde{\mathbf{t}}_{10}^0 := \frac{1}{3}\underline{\mathbf{t}}_{00}^0 + \frac{2}{3}\underline{\mathbf{t}}_{01}^0 = \frac{1}{3}\underline{\mathbf{t}}_{00}^0 + \frac{2}{3}\underline{\mathbf{t}}_{01}^0 + \frac{2b_0^0}{3}(\underline{\mathbf{t}}_{00}^0 + \underline{\mathbf{t}}_{10}^0)^{(11)} =: \tilde{\underline{\mathbf{t}}}_{01}^0.$$

This implies $b_0^0 := 0$; furthermore, matching $\tilde{\mathbf{t}}_{11}^0 = \tilde{\underline{\mathbf{t}}}_{11}^0$ (marked ∘ in Fig. 16 a) leads to

$$\tilde{\mathbf{t}}_{11}^0 := \frac{13}{81}\underline{\mathbf{t}}_{00}^0 + \frac{2}{9}\mathbf{t}^0 u_{01} + \frac{14}{81}\underline{\mathbf{t}}_{10}^0 + \frac{4}{9}\underline{\mathbf{t}}_{11}^0 = (\frac{1}{9} - \frac{4}{9}b_1^0)\underline{\mathbf{t}}_{00}^0 + \ldots =: \tilde{\underline{\mathbf{t}}}_{11}^0$$

and this implies $b_1^0 := -\frac{1}{9}$.

C^1-continuity between $\tilde{\underline{\mathbf{t}}}^0$ and $\tilde{\underline{\mathbf{t}}}^1$ implies $\tilde{\underline{\mathbf{t}}}_{11}^1 := 2\tilde{\underline{\mathbf{t}}}_{31}^0 - \tilde{\underline{\mathbf{t}}}_{21}^0$, but the expression for $\tilde{\underline{\mathbf{t}}}_{21}^0$ in (11) contains the term $-\frac{2}{9}b_2^0\underline{\mathbf{t}}_{00}^0$ that is missing in $\tilde{\underline{\mathbf{t}}}_{31}^0$ and no coefficient of $\underline{\mathbf{t}}^1$ depends on $\underline{\mathbf{t}}_{00}^0$. Hence $b_2^0 := 0$ and $\tilde{\underline{\mathbf{t}}}_{31}^0 := \frac{1}{3}\underline{\mathbf{t}}_{20} + \frac{2}{3}\underline{\mathbf{t}}_{21}$. □

We can now prove the promised (sharp) lower bound result.

Theorem 1. *There does not exist a 3×3 C^1 bi-3 macro-patch construction that guarantees a smooth join to any given bi-2 frame.*

Proof. Lemma 2 and structural left-right symmetry imply $[b_0^2, b_1^2, b_2^2] = [0, \frac{1}{9}, 0]$ (note the change of sign due to reversal of direction) Since $\underline{\mathbf{t}}_{0j}^1 = \underline{\mathbf{t}}_{2j}^0$, $j = 0, 1$, the same argument as for $b_0^0 = 0$ yields $b_0^1 := 0$. Therefore C^1 continuity of b^0 with b^1 implies $[b_0^1, b_1^1, b_2^1] = [0, \frac{1}{9}, 0]$. Retracing the arguments to C^1 continuity of b^1 with b^2 implies $b_2^1 = -\frac{1}{9}$, a contradiction. □

For illustration, calculating $\tilde{\underline{\mathbf{t}}}_{21}^2$ for $b_1^2 = -\frac{1}{9}$ to ensure C^1 continuity with the b^0 yields an inconsistency at the right location emphasized in Fig. 16 b by replacing the initial ∘ by a larger one.

6. A C^1 bi-3 macro-patch FC^3

We now leverage the findings of Section 5 and allows more pieces to successfully construct a bi-cubic FC^3 surface. The bi-3 macro-patch FC^3 consists of 11 pieces laid out in Fig. 17 b. The central patch $\mathbf{m}$ is borrowed from the FC^4 construction and degree-raised to degree bi-3. This choice turned out superior to any direct construction using functionals or heuristics. Splitting $\mathbf{m}$ uniformly into two patches in the horizontal direction yields a 4×3 layout of bi-3 patches.

To derive the bi-3 macro-patch we call two adjacent segments $\mathbf{p}^s$ and $\mathbf{p}^{s+1}$ of the bi-2 tensor-border frame *connected with ratio* $1 : \kappa$ iff

$$\mathbf{p}_{0j}^{s+1} := \mathbf{p}_{2j}^s, \quad \mathbf{p}_{1j}^{s+1} := (1 + \kappa)\mathbf{p}_{2j}^s - \kappa\mathbf{p}_{1j}^s, \; j = 0, 1.$$

As in the preceding section, let $\rho^s := (u + b^s(u)v, v)$, where $b^s(u)$ is a quadratic function with BB-coefficients b_i^s, $i = 0, 1, 2$ and define $\tilde{\mathbf{p}}^s := \mathbf{p}^s \circ \rho^s$. Then retracing the argument in Section 5.2 with the ratio $1 : \kappa$ proves the following.

Lemma 3. *Let adjacent segments $\mathbf{p}^s$ and $\mathbf{p}^{s+1}$ of the bi-2 tensor-border frame be C^1-connected with ratio $1 : \kappa$.*
Then the re-parameterized tensor-borders $\tilde{\mathbf{p}}^s$ and $\tilde{\mathbf{p}}^{s+1}$ are C^1-connected with ratio $1 : \kappa$,

$$\tilde{\mathbf{p}}_{0j}^{s+1} = \tilde{\mathbf{p}}_{3j}^s, \quad \tilde{\mathbf{p}}_{1j}^{s+1} = (1 + \kappa)\tilde{\mathbf{p}}_{3j}^s - \kappa\tilde{\mathbf{p}}_{2j}^s, \; j = 0, 1, \tag{12}$$

if and only if $b_0^{s+1} = b_2^s = 0$ and $b_1^{s+1} = -b_1^s$.

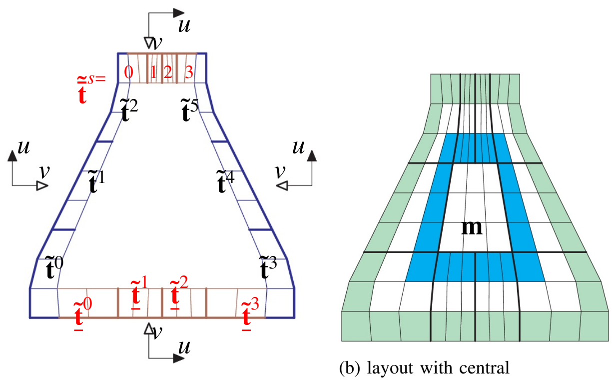
Based off this Lemma, the tensor-border frame is adjusted as follows.

6.1. Adjusting the tensor-border frame

First we consider the bottom part of frame, Fig. 17 a. For the 4×3 layout, $\underline{\tilde{\rho}}_0$ and $\underline{\tilde{\rho}}_1$ is defined as in Section 5.2, and $\underline{\tilde{\rho}}_2 := \underline{\tilde{\rho}}_0$, $\underline{\tilde{\rho}}_3 := \underline{\tilde{\rho}}_1$:

$$[b_0^s, b_1^s, b_2^s] := [0, \frac{(-1)^{s+1}}{9}, 0], \quad s = 0, \ldots, 3. \tag{13}$$

We set $\underline{\mathbf{t}}^0 := \underline{\mathbf{t}}^0$, $\underline{\mathbf{t}}^3 := \underline{\mathbf{t}}^2$, split $\underline{\mathbf{t}}^1$ uniformly in u into two pieces $\underline{\mathbf{t}}^1$, $\underline{\mathbf{t}}^2$ and define $\tilde{\underline{\mathbf{t}}}^s := \underline{\mathbf{t}}^s \circ \underline{\tilde{\rho}}^s$, $s = 0, \ldots, 3$.



(a) correct bi-3 frame

(b) layout with central bi-3 patch **m** (white)

Figure 17: (a) Correct bi-3 tensor-border frame. (b) 11 piece layout of the bi-3 macro-patch: the central patch **m** is not split.

The analogous considerations are applied to the *top* part of frame. To match a bottom layout of tensor-border frame in Fig. 17 a, the input bi-2 tensor-border $\bar{\mathbf{t}}$ (see Fig. 8 a) is split in the u-direction with ratio $(1 : \frac{1}{2} : \frac{1}{2} : 1)$ into four pieces $\bar{\mathbf{t}}^s$, $s = 0, \dots, 3$. The four bi-2 pieces $s = 0, \dots, 3$ are reparameterized with

$$\tilde{\bar{\rho}}^s := (u + b^s(u)v, v), \text{ where } [b^s_0, b^s_1, b^s_2] := [0, \frac{(-1)^s}{3}, 0].$$

That is, the explicit formulas for $\tilde{\underline{\mathbf{t}}}^s$ from $\underline{\mathbf{t}}^s$ (respectively for the top $\tilde{\bar{\mathbf{t}}}^s$ from $\bar{\mathbf{t}}^s$), for $s = 0, \dots, 3$ (superscript omitted) are that the boundary BB-coefficients are obtained by degree-raising and

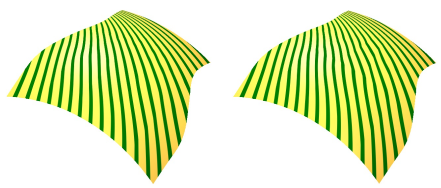
$$\begin{aligned}
\mathbf{q}_{01} :=& \frac{1}{3}\mathbf{p}_{00} + \frac{2}{3}\mathbf{p}_{01}, \ \mathbf{q}_{31} := \frac{1}{3}\mathbf{p}_{20} + \frac{2}{3}\mathbf{p}_{21}, \\
\mathbf{q}_{11} :=& (\frac{1}{9} - \frac{4}{9}\gamma)\mathbf{p}_{00} + (\frac{2}{9} + \frac{4}{9}\gamma)\mathbf{p}_{10} + \frac{2}{9}\mathbf{p}_{01} + \frac{4}{9}\mathbf{p}_{11}, \\
\mathbf{q}_{21} :=& (\frac{2}{9} - \frac{4}{9}\gamma)\mathbf{p}_{10} + (\frac{1}{9} + \frac{4}{9}\gamma)\mathbf{p}_{20} + \frac{4}{9}\mathbf{p}_{11} + \frac{2}{9}\mathbf{p}_{21},
\end{aligned} \tag{14}$$

where $\mathbf{p}$ denotes the BB-coefficients of $\underline{\mathbf{t}}^s$ ($\bar{\mathbf{t}}^s$) and $\mathbf{q}$ of $\tilde{\underline{\mathbf{t}}}^s$ ($\tilde{\bar{\mathbf{t}}}^s$) and

$$\text{bottom: } \gamma := \frac{(-1)^{s+1}}{9}, \text{ top: } \gamma := \frac{(-1)^s}{3}.$$

6.2. C^1 *completion of bi-3 macro-patch*

The central patch of FC^4 (labeled 5 in Fig. 6 b) is degree-raised to form the central bi-3 patch $\mathbf{m}$ of FC^3, see Fig. 17 b (where the tensor-border frame from Fig. 17 a is displayed as 'light green'). The C^1-extension of $\mathbf{m}$ towards the frame (displayed cyan) is uniformly split in the horizontal direction where needed, i.e. *top* and *bottom*. The resulting bi-3 macro-patch has 11 pieces. Splitting $\mathbf{m}$ yields a tensor-product 4×3 layout.



(a) 11 piece bi-3 macro-patch based on $\mathbf{m}$ (b) direct application of functional

Figure 18: Comparison of the 11-piece construction and a bi-3 construction based on functionals. The input net is Fig. 11 top,left.

7. Nested refinement of FC^4

A refinement of a spline space is *nested* if the finer space includes the coarser space. Refinement is useful, both for geometric manipulations and for engineering analysis since it exposes additional degrees of freedom while preserving the original shape or solution. The interior of FC^4 and FC^3 are C^1 splines that can be nestedly refined by knot insertion. For FC^4, the top and bottom tensor-borders are not reparameterized, only degree-raised to 4 in the vertical direction and the top is split in the horizontal direction. Hence top and bottom refinement is that of regular splines.

The G^1 transition from the input bi-2 tensor-border to FC^4 is displayed in Fig. 19. The bi-2 tensor-border *bottom-left* is reparametrized with $\rho := (u, a(u)v)$ yielding a tensor-border of bi-degree $(4, 2)$, *top-left*. For reducing the free parameters of the construction, degree 3 of the middle patch of FC^4 in the 'vertical' direction was convenient. However, for nested refinement pieces of different bi-degrees complicated the exposition. Therefore the initial patches of bi-degree $(2, 3)$ are degree-raised to $(2, 4)$ and $a(u)$ is degree-raised to 2 so that all $a(u)$ are formally quadratic.

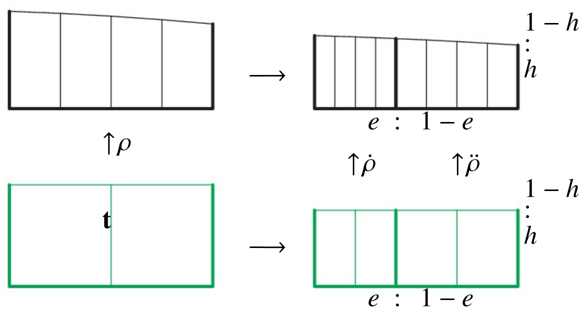


Figure 19: Diagram of nested G^1-refinement.

The input and reparameterized tensor-borders are split in some ratio $e : 1-e$ along the boundary (see $\longrightarrow$ in Fig. 19); and with ratio $h : 1-h$ into the macro-patch. By definition, *nested refinement* means that there exist reparameterizations $\dot{\rho}$ and $\ddot{\rho}$ satisfying

- (i^{nest}) the reparametrized tensor-border $\mathbf{t} \circ \rho$, split (as displayed in Fig. 19 *top-right*), equals the union of split input tensor-borders pieces (*bottom-right*) reparameterized respectively by $\dot{\rho}$, and $\ddot{\rho}$ (a commutative diagram); and
- (ii^{nest}) the C^1-continuity and bi-degree of the reparameterized tensor-borders are retained when perturbing the split input bi-2 data.

Properties (i^{nest},ii^{nest}) are satisfied if

$$\begin{aligned}
\dot{\rho}(u, v) := (u, \dot{a}(u)v), \quad \ddot{\rho}(u, v) := (u, \ddot{a}(u)v), \text{ where} \\
\dot{a}(u) := a(eu), \quad \ddot{a}(u) := a(e(1-u) + u),
\end{aligned}$$

where the BB-coefficients of $\dot{a}(u)$ and $\ddot{a}(u)$ are computed from $a(u)$ by de Casteljau's algorithm. With the BB-coefficients of the boundary obtained by degree-raising, the formulas for $\tilde{\mathbf{t}}(u, v) := \mathbf{t} \circ \rho(u, v)$, $\rho(u, v) := (u, a(u), v)$ where $a(u)$ has BB-coefficients a_0, a_1, a_2, are

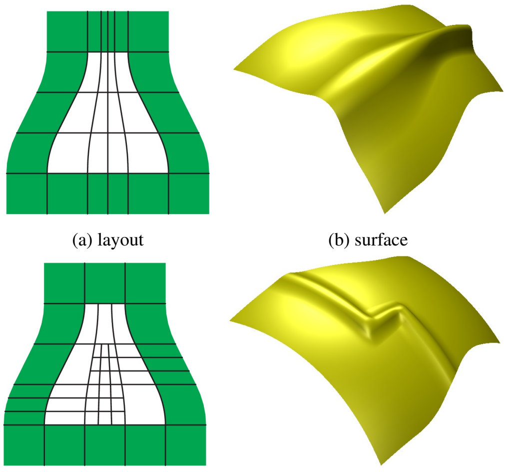
$$\begin{aligned}
\tilde{\mathbf{t}}_{01} :=& (1 - a_0)\mathbf{t}_{00} + a_0\mathbf{t}_{01}; \\
\tilde{\mathbf{t}}_{11} :=& \frac{1}{2}((1 - a_1)\mathbf{t}_{00} + (1 - a_0)\mathbf{t}_{10} + a_1\mathbf{t}_0 1 + a_0\mathbf{t}_{11}); \\
\tilde{\mathbf{t}}_{21} :=& \frac{1}{6}((1 - a_2)\mathbf{t}_{00} + (1 - a_0)\mathbf{t}_{20} + 4(1 - a_1)\mathbf{t}_{10} \\
&+ a_2\mathbf{t}_{01} + a_0\mathbf{t}_{21} + 4a_1\mathbf{t}_{11}).
\end{aligned} \tag{15}$$

The BB-coefficients $\tilde{\mathbf{t}}_{41}$, $\tilde{\mathbf{t}}_{31}$ are obtained from $\tilde{\mathbf{t}}_{01}$, $\tilde{\mathbf{t}}_{11}$ by replacing $\mathbf{t}_{ij}$ by $\mathbf{t}_{2-i,j}$ and a_i by a_{2-i}. More details on nested refinement can be found in [39, Section 3.1].

Fig. 20 ,*top* demonstrates that refinement in FC^4 adds flexibility to improve surface quality in the tricky case of Fig. 15 . Fig. 20 ,*bottom* shows how a refined layout allows introducing complex ridges.

For FC^3, G^1-refinement along left and right boundaries is the same as for FC^4, but for bi-3 tensor-border and patches, and for a linear $a(u)$, see (8). However, FC^3 is not nestedly G^1-refinable.

Lemma 4. *FC^3 is not nestedly G^1-refinable along its bottom and top borders.*



(a) layout (b) surface

(c) layout (d) surface

Figure 20: *Top* row: mimicking the surface Fig. 15 e. *Bottom* row: adding a zigzag between left and right sides.

Proof. By (i^{nest}), the reparameterized tensor-border must, after the split, be an union of the reparameterized pieces of the input tensor-border. To not exceed degree bi-3, the reparametrizations must have the form $\rho(u, v) := [u + b(u)v, v]$, with quadratic $b(u)$. By (ii^{nest}), the reparameterized pieces must be C^1-connected. Then Lemma 3 forces b(u) to be zero at the endpoints. Multiplying out, we check that no such $b(u)$ exists. □

Left and right sides of FC^3 are nestedly refinable: e.g. the features as in Fig. 20 d can be designed. The *shape* of Fig. 20 b can be modeled by non-nested refinement at the bottom and top. However, such a design is more cumbersome than introducing the details after preserving the inital surface through nested refinement.

8. Discussion, limitations and summary

Unsurprisingly, FC^4 and FC^3 generate similar surfaces, not least, because they share the central patch **m**. Options, including those used for deriving the free central-patch BB-coefficients for FC^4, resulted in poorer surfaces than the **m**-sharing FC^3 construction, as illustrated in Fig. 18 . This is likely due to the, compared to FC^4, slight distortions of the FC^3 tensor-borders that challenge derivative-based functionals.

The derivation of the FC-surfaces is intricate, but this complexity pays off in that local features diagonal to the principal parameter directions can be properly handled by FC^4 while regular B-splines result in a sequence of dips. Only for ridges splitting the core from top to bottom is the reverse true: FC^4 result in a dip. We showed two options to mend this situation: to align the core connectivity with the new cross direction as in Fig. 15 ; or, preferably, to use the G^1-refinability of FC^4. The latter increases the number of polynomial pieces but improves the surface quality.

By contrast, the implementation is simple: gather the $\triangle^2$-net in the vector of points **d** and compute the vector **b** of the BB-coefficients as $\mathbf{b} = M\mathbf{d}$. The cost of surface evaluation is very similar to evaluating a tensor-product spline by inserting knots: this ca be expressed as a matrix multiplication, followed by evaluation of the resulting Bézier form.

Physical simulation, in the sense of solving partial differential equations on the surface by Galerkin's approach, is no more difficult for geometrically smooth surfaces than for parametrically smooth surfaces [48, 49]. In particular, the expensive part of assembling the stiffness matrix, including the first fundamental form of the surface, is alike. Anyhow, geometric continuity, is already required for multi-sided smooth surfaces. For large 2D textures created in the domain, it is advantageous to have a single parameterization. But for high-end textures that are created by directly painting on the surface and pulling back the texture to domain coordinates, there is no disadvantage to geometric continuity.

Much of the technical framework of Section 3.1 and Section 6 easily generalizes to more general contractions and configurations. In particular, deriving the explicit reparameterizations of the input tensor-border frame does not pose a challenge. Rather, the challenge lies in the careful setting of free parameters (see Section 3.2) since fast contraction easily spoils the shape.

While a uniform bi-3 degree of FC^3 facilitates seamless integration into the bi-3 polyhedral-net spline code [8], as illustrated in Fig. 22 , FC^4 is preferable in applications where nested G^1-refinability ensures exact reproduction at a finer resolution, for example when using the splines both to model the surface and to solve differential equations on the surface with the same spline elements.

Acknowledgements Erkan Gunpinar acknowledges the funding from The Scientific and Technological Research Council of Turkey (TUBITAK, Project No: 123F259).

References

[1] N. Schertler, M. Tarini, W. Jakob, M. Kazhdan, S. Gumhold, D. Panozzo, Field-aligned online surface reconstruction, ACM Trans. Graph. 36 (4) (2017).

[2] W. Jakob, M. Tarini, D. Panozzo, O. Sorkine-Hornung, Instant field-aligned meshes, ACM Trans. Graph. 34 (6) (2015).

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

[4] K. Karčiauskas, J. Peters, Point-augmented biquadratic C^1 subdivision surfaces, Graphical Models 77 (2015) 18–26.

[5] K. Karčiauskas, J. Peters, Localized remeshing for polyhedral splines, Computers & Graphics 106 (2022) 58–65.

[6] K.-P. Beier, Y. Chen, Highlight-line algorithm for realtime surface-quality assessment, Comp-Aid Design 26 (4) (1994) 268–277.

[7] N. Ray, W. C. Li, B. Lévy, A. Sheffer, P. Alliez, Periodic global parameterization, ACM Trans. Graph. 25 (4) (2006) 1460–1485.

[8] J. Peters, K. Lo, K. Karčiauskas, Algorithm 1032: Bi-cubic splines for polyhedral control nets, ACM Tr on Math Software 49 (March 2023).

[9] P. Salvi, T. Várady, Multi-sided surfaces with fullness control, in: Proceedings of the Eighth Hungarian Conference on Computer Graphics and Geometry, 2016, pp. 61–69.

[10] G. J. Hettinga, J. Kosinka, A multisided C^2 B-spline patch over extraordinary vertices in quadrilateral meshes, Computer-Aided Design 127 (2020) 102855.

[11] M. Vaitkus, T. Várady, P. Salvi, Á. Sipos, Multi-sided b-spline surfaces over curved, multi-connected domains, Computer Aided Geometric Design 89 (2021) 102019.

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

[13] J. Peters, U. Reif, The simplest subdivision scheme for smoothing polyhedra, ACM Transactions on Graphics 16 (4) (1997) 420–431.

[14] K. Karčiauskas, J. Peters, Point-augmented bi-cubic subdivision surfaces, in: N. Umetani, E. Vouga, C. Wojtan (Eds.), Pacific Graphics 2022, The Eurographics Association and John Wiley & Sons Ltd., pp. 1–11.

[15] K. Karčiauskas, J. Peters, Evolving guide subdivision, Computer Graphics Forum 42 (2) (2023) 321–332.

[16] A. Myles, J. Peters, C^2 splines covering polar configurations, Computer Aided Design 43 (11) (2011) 1322–1329.

[17] D. Toshniwal, H. Speleers, R. R. Hiemstra, T. J. Hughes, Multi-degree smooth polar splines: A framework for geometric modeling and isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 316 (2017) 1005–1061.

[18] K. Karčiauskas, J. Peters, Smooth polar caps for locally quad-dominant meshes, Computer Aided Geometric Design 81 (2020) 1–12.

[19] J. Peters, Parametrizing singularly to enclose vertices by a smooth parametric surface, in: S. MacKay, E. M. Kidd (Eds.), Graphics Interface '91, Calgary, Alberta, 3–7 June 1991: proceedings, Canadian Information Processing Society, 243 College St, 5th Floor, Toronto, Ontario M5T 2Y1, Canada, 1991, pp. 1–7.

[20] U. Reif, A refineable space of smooth spline surfaces of arbitrary topological genus, Journal of Approximation Theory 90 (2) (1997) 174–199.

[21] T. Nguyen, K. Karčiauskas, J. Peters, C^1 finite elements on non-tensor-product 2d and 3d manifolds, Applied Mathematics and Computation 272 (1) (2016) 148–158.

[22] X. Wei, Y. J. Zhang, D. Toshniwal, H. Speleers, X. Li, C. Manni, J. A. Evans, T. J. Hughes, Blended b-spline construction on unstructured quadrilateral and hexahedral meshes with optimal convergence rates in isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 341 (2018) 609–639.

[23] J. A. Gregory, Smooth interpolation without twist constraints, Academic Press, 1974, pp. 71–88.

[24] C. T. Loop, S. Schaefer, T. Ni, I. Castaño, Approximating subdivision surfaces with Gregory patches for hardware tessellation, ACM Trans. Graph 28 (5) (2009) 151:1–151:9.

[25] C. T. Loop, S. Schaefer, G^2 tensor product splines over extraordinary vertices, Comput. Graph. Forum 27 (5) (2008) 1373–1382.

[26] K. Karčiauskas, J. Peters, Minimal bi-6 G^2 completion of bicubic spline surfaces, Computer Aided Geometric Design 41 (2016) 10–22.

[27] M. Kapl, G. Sangalli, T. Takacs, Dimension and basis construction for analysis-suitable G^1 two-patch parameterizations, Computer Aided Geometric Design 52-53 (2017) 75–89.

[28] A. Blidia, B. Mourrain, G. Xu, Geometrically smooth spline bases for data fitting and simulation, Computer Aided Geometric Design 78 (2020) 101814.

[29] M. Marsala, A. Mantzaflaris, B. Mourrain, G1-smooth biquintic approximation of Catmull-Clark subdivision surfaces, Computer Aided Geometric Design 99 (2022) 102158.

[30] K. Karčiauskas, J. Peters, Bi-cubic scaffold surfaces, Vol. 150, Elsevier, 2022, p. 103310.

[31] K. Karčiauskas, J. Peters, Improved shape for multi-surface blends, Graphical Models 82 (2015) 87–98.

[32] G.-P. Bonneau, S. Hahmann, Flexible G^1 interpolation of quad meshes, Graphical Models 76 (6) (2014) 669–681.

[33] K. Karčiauskas, T. Nguyen, J. Peters, Generalizing bicubic splines for modeling and IGA with irregular layout, Computer-Aided Design 70 (2016) 23–35.

[34] K. Karčiauskas, J. Peters, Can bi-cubic surfaces be class A?, Computer Graphics Forum 34 (5) (2015) 229–238.

[35] K. Karčiauskas, J. Peters, Improved caps for improved subdivision surfaces, Computer-Aided Design (2023) 103543.

[36] K. Karčiauskas, J. Peters, Smooth multi-sided blending of biquadratic splines, Computers & Graphics 46 (2015) 172–185.

[37] K. Karčiauskas, J. Peters, Low degree splines for locally quad-dominant meshes, Computer Aided Geometric Design 83 (2020) 1–12.

[38] U. Reif, Biquadratic g-spline surfaces, Computer Aided Geometric Design 12 (2) (1995) 193–205.

[39] K. Karčiauskas, J. Peters, Refinable smooth surfaces for locally quad-dominant meshes with t-gons, Computers & Graphics 82 (2019) 193–202.

[40] K. Karčiauskas, J. Peters, Low degree splines for locally quad-dominant meshes, Computer Aided Geometric Design 83 (2020) 101934.

[41] T. W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCs, in: J. Hodgins, J. C. Hart (Eds.), Proc ACM SIGGRAPH, Vol. 22(3) of ACM Tr Graph, 2003, pp. 477–484.

[42] K. Karčiauskas, D. Panozzo, J. Peters, T-junctions in spline surfaces, ACM Tr on Graphics 36 (5) (2017) 170:1–9.

[43] M. Campen, D. Zorin, Similarity maps and field-guided T-splines: a perfect couple, ACM Trans. Graph 36 (4) (2017) 91:1–91:16.

[44] G. Farin, Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide, Academic Press, 1988.

[45] T. D. DeRose, Necessary and sufficient conditions for tangent plane continuity of Bézier surfaces, Comp Aid Geom Design 7 (1) (1990) 165–179.

[46] K.-P. Beier, Y. Chen, Highlight-line algorithm for realtime surface-quality assessment, Computer-Aided Design 26 (4) (1994) 268–277, special Issue: Mathematical methods for CAD.

[47] K. Karčiauskas, J. Peters, Least degree G^1-refinable multi-sided surfaces suitable for inclusion into C^1 bi-2 splines, Computer-Aided Design 130 (2021) 1–12.

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

[49] D. Groisser, J. Peters, Matched G^k-constructions always yield C^k-continuous isogeometric elements, Computer Aided Geometric Design 34 (2015) 67–72.

Appendix A: Formulas for the tensor-border frame $\hat{\mathbf{t}}_s$

The construction is symmetric and $\underline{\hat{\mathbf{t}}}_s$, $\hat{\hat{\mathbf{t}}}_s$ are bi-2 in degree-raised $(2,4)$ form. Therefore only the first cross-derivative layer of $\hat{\mathbf{t}}_s$ needs to be specified. Omitting s, we denote the BB-coefficients of input bi-2 tensor border as $\mathbf{b}_{ij}$, $i = 0,1,2$, $j = 0,1$ and the reparameterized tensor-borders of bi-degree $(4,2)$ and $(3,2)$ as $\tilde{\mathbf{b}}_{r1}$, $r = 0,\dots,4$ and $r = 0,\dots,3$. Then $\tilde{\mathbf{b}}_{r1} := \sum_{i=0}^{2}\sum_{j=0}^{1} \nu_{ij}\mathbf{b}_{ij}$, with coefficients ν_{ij} arranged as 2×3 tables A^s_{r1} (specific for superscript s and index r) in the format $A := \left(\begin{smallmatrix}\nu_{01} & \nu_{11} & \nu_{21}\\ \nu_{00} & \nu_{10} & \nu_{20}\end{smallmatrix}\right)$. Fig. 21 , displays the labels for all reparameterized tensor-borders.

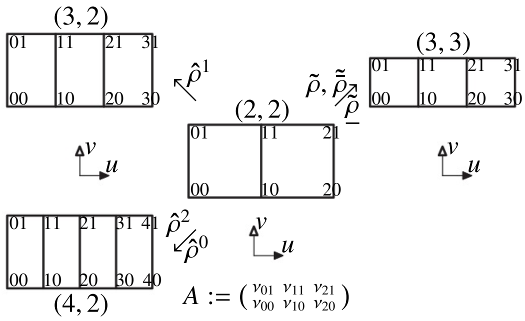


Figure 21: Reparameterized tensor-borders. center: reparameterization. left: $\hat{\rho}^0$, $\hat{\rho}^1$, $\hat{\rho}^2$ are used for the main construction. right: $\tilde{\rho}$, $\underline{\tilde{\rho}}$, $\tilde{\tilde{\rho}}$ are used for the bi-3 construction.

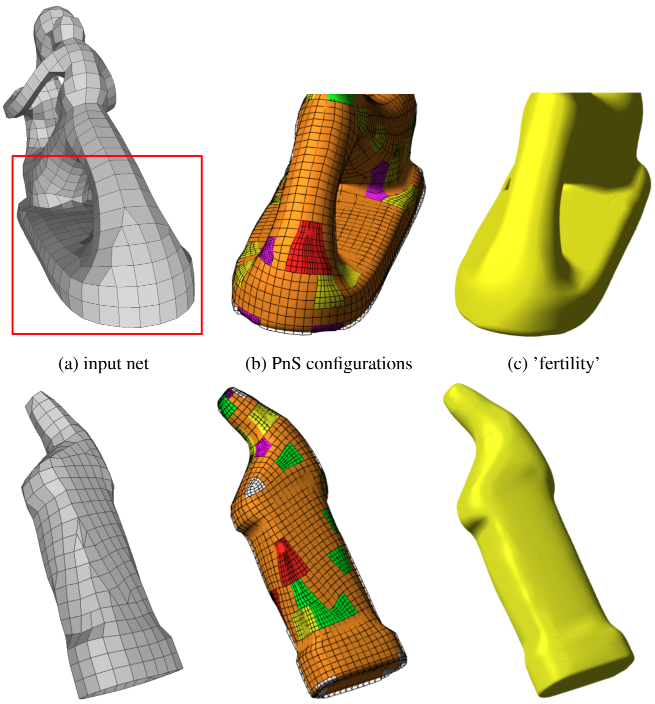
$$A^0_{01} := \begin{pmatrix}1&0&0\\0&0&0\end{pmatrix}, A^0_{11} := \tfrac{1}{2}\begin{pmatrix}1&1&0\\0&0&0\end{pmatrix}, A^0_{21} := \tfrac{1}{36}\begin{pmatrix}5&24&6\\1&0&0\end{pmatrix},$$
$$A^0_{31} := \tfrac{1}{12}\begin{pmatrix}0&5&6\\0&1&0\end{pmatrix}, A^0_{41} := \tfrac{1}{6}\begin{pmatrix}0&0&5\\0&0&1\end{pmatrix};$$
$$A^1_{01} := \tfrac{1}{6}\begin{pmatrix}5&0&0\\1&0&0\end{pmatrix}, A^1_{11} := \tfrac{1}{18}\begin{pmatrix}3&10&0\\3&2&0\end{pmatrix}, A^1_{21} := \tfrac{1}{18}\begin{pmatrix}0&6&5\\0&6&1\end{pmatrix},$$
$$A^1_{31} := \tfrac{1}{2}\begin{pmatrix}0&0&1\\0&0&1\end{pmatrix};$$
$$A^2_{01} := \tfrac{1}{2}\begin{pmatrix}1&0&0\\1&0&0\end{pmatrix}, A^2_{11} := \tfrac{1}{12}\begin{pmatrix}2&3&0\\4&3&0\end{pmatrix}, A^2_{21} := \tfrac{1}{36}\begin{pmatrix}2&8&3\\4&16&3\end{pmatrix},$$
$$A^2_{31} := \tfrac{1}{6}\begin{pmatrix}0&1&1\\0&2&2\end{pmatrix}, A^2_{41} := \tfrac{1}{3}\begin{pmatrix}0&0&1\\0&0&2\end{pmatrix}.$$

Appendix B: Weights μ of the FC^4 'spine'

$$\text{Table } M^s \text{ lists} \quad 10^5\begin{pmatrix}\mu^s_{15} & \mu^s_{25} & \mu^s_{35}\\ \mu^s_{14} & \mu^s_{24} & \mu^s_{34}\\ \mu^s_{13} & \mu^s_{23} & \mu^s_{33}\\ \mu^s_{12} & \mu^s_{22} & \mu^s_{32}\\ \mu^s_{11} & \mu^s_{21} & \mu^s_{31}\end{pmatrix}.$$

$$M^0 := \begin{pmatrix}0&0&0\\412&-825&412\\-1256&5423&5423\\987&-782&82923\\-27&-137&8662\end{pmatrix}, \quad M^1 := \begin{pmatrix}0&0&0\\1714&-3428&1714\\-3273&28273&28273\\3824&-10327&62985\\-568&2273&-3409\end{pmatrix}, \quad M^2 := \begin{pmatrix}0&0&0\\2603&-5206&2603\\-4199&45866&45866\\6924&-19420&43457\\-1246&4986&-7479\end{pmatrix},$$
$$M^3 := \begin{pmatrix}0&0&0\\2570&11526&2570\\-429&42096&42096\\3857&-16405&25095\\-1105&4421&-6631\end{pmatrix}, \quad M^4 := \begin{pmatrix}0&0&0\\5384&39230&5384\\496&24503&24503\\2232&-9572&14680\\-709&2838&-4254\end{pmatrix}, \quad M^5 := \begin{pmatrix}1008&6316&1008\\8564&66207&8564\\793&3373&3373\\303&-1369&2132\\-156&627&-941\end{pmatrix}.$$

The remaining coefficients μ^s_{ij} are obtained by symmetry; i.e. $\mu^s_{41} := \mu^s_{21}, \mu^s_{51} := \mu^s_{11}; \mu^s_{42} := \mu^s_{22}, \mu^s_{52} := \mu^s_{12}; \mu^s_{43} := \mu^s_{13}$.



(a) input net (b) PnS configurations (c) 'fertility'

(d) input net (e) PnS configurations (f) 'bottle'

Figure 22: Showcasing FC^3 within a bi-cubic polyhedral-net spline surface color-coded in (b,e) as surface pieces of type FC^3, T_0, T_1, n-sided, regular bi-2, n-valent. The BB-coefficients of the bi-3 patches are overlaid. Input nets rendered with MeshLab, output surfaces rendered with Bezierview, algorithm integrated into the Polyhedral-net Spline (PnS) code base.