



Narrowing-Cascade splines for control nets that shed mesh lines

submission 53

ARTICLE INFO

Article history:

Received July 9, 2025

Control-net narrowing, polyhedral-net spline, geometric continuity

ABSTRACT

To fit narrow surface passages, quad-dominant meshing algorithms use a cascade of $n - 1$ triangles that reduces the number of parallel quad strips from $n + 1$ to 2. A new shape-optimized G^1 spline surface, the Narrowing-Cascade spline NC^n , can treat such meshes as spline control nets. For $n > 3$, the narrowing configurations have interior mesh points that both guide and complicate the construction of the output tensor-product NC spline. The spline follows the input mesh while delivering a high-quality curved surface of low degree.

© 2025 Elsevier B.V. All rights reserved.

1. Introduction

In polyhedral modeling and quad-dominant meshing [1, 2], parameter lines are often merged. Merging serves, for example, to reduce variability of the surface, or accommodate narrow surface passages, see Fig. 1. When using meshes directly to control splines, such shedding of mesh lines causes extensive re-meshing. It is therefore of interest to avoid the re-meshing and develop spline surfaces that specifically support the narrowing transition. A first Narrowing-Cascade construction [3], illustrated in Fig. 3 a, provided a spline surface to transition vertically from 3 to 2 quad-strips, in 3 horizontal rows. Since the spline consists of $n = 2$ pieces in the narrowing direction, we call it NC^2 -spline. Its mesh is characterized by $n - 1 = 1$ triangles with valence 5 at the apex. More recently, [4, 5] derived NC^3 -splines whose control nets Δ^2 are characterized by a partition of a large triangle partitioned into $n - 1 = 2$ pieces per edge, as shown in Fig. 3 b.

Fig. 3 d-f shows various Δ^3 -nets that occur in practice and Fig. 3 g,h Δ^4 -nets with differing internal partitions. While parts of the earlier constructions easily generalize to NC^n -splines, for $n > 3$ interior mesh points come into play. On one hand these interior points can help guide the larger undetermined NC^n surface, on the other they complicate the aim to generate surfaces with a uniform highlight line distribution.

Several alternative approaches could be employed. The base cases could be used repeatedly. However this requires separation between the base cases, e.g. by inserting separating loops between triangles that share a mesh node. Another solution is global mesh refinement, such as one or more steps of Catmull-Clark subdivision [6] that convert any net, including Δ^n , into a

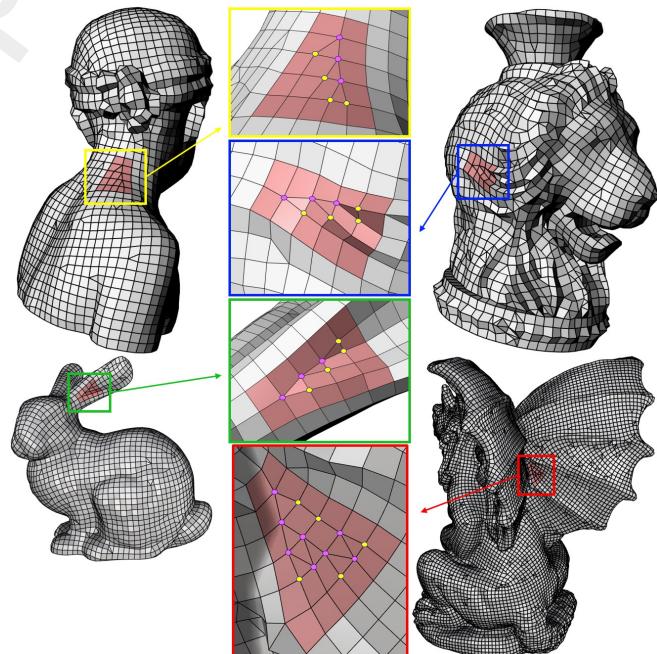


Fig. 1: Δ^3 and Δ^4 configurations in quad-dominant meshes. See Fig. 2 for NC^n -surfaces. (Nodes of valence 5 are colored ● and nodes of valence 4 ○.)

pure quad mesh. However this quadruples the number of pieces for each step and results in undesirable triangle to quads partitions. Third, special re-meshing and refinement rules can separate the triangles in a cascade [7, 8]. However, this can disturb

the highlight line distribution, decreasing the surface quality.

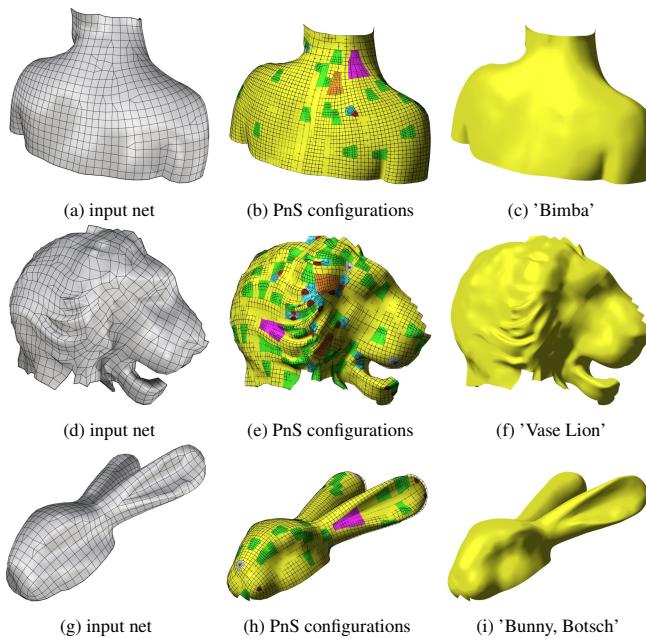


Fig. 2: NC^n_4 as part of a polyhedral net spline (PnS) surface, colored magenta, NC^4 , and orange NC^3 and green NC^2 for Δ^3 , Δ^2 , Δ^1 . Regular bi-2 splines are displayed in yellow and the BB-net of Bézier coefficients is superimposed. The remaining colors in (b,e,h) indicate polyhedral net splines for 3-gon (triangle with 4-valent vertices), cop3 , cop5 (mesh nodes with 3, resp. 5 neighbors).

Instead, this paper offers a novel NC^n -spline surface for Δ^{n-1} -nets. The NC^n -spline complements the family of polyhedral net splines [9] and so further supports designing spline surfaces by outlining them with polyhedral meshes, see Fig. 2.

NC-splines have the following properties. NC^n -splines

- join by default G^1 with any surrounding bi-2 (B-)splines or polyhedral-net splines [9];
- have a good highlight line distribution [10];
- are agnostic to internal connectivity of the Δ^{n-1} -net, see Fig. 3 c-f, but follow the placement of the internal mesh points without resorting to functionals;
- consist of a patchwork of C^1 -joined tensor-product patches controlled by the Δ^{n-1} -net, with two choices:
 - NC^n_4 consists of n^2 pieces: $2n$ pieces of degree (2,4) and $(n-2)n$ pieces of degree (2,3);
 - NC^n_3 consists of pieces of degree bi-3: n^2 pieces if n is even and $n(n+1)$ pieces if n is odd.

Overview. Section 2 reviews the pertinent literature. Section 3 introduces the technical tools. Section 4 introduces the main construction NC^n_4 with pieces of degree 4. Section 5 presents an alternative, everywhere degree bi-3 construction NC^n_3 . Section 6 illustrates, compares and, where possible, contrasts the NC-surfaces with tensor-product surfaces

2. Surface constructions generalizing bi-2 splines

There are three major families of surface constructions: singular surface parameterizations [6, 11], rational multi-sided surfaces [12–14] and geometrically smooth, G^k surfaces [8, 15–28]. Narrowing-Cascade (NC) splines belong to the family of G^1 surface constructions, where a finite number of polynomial pieces are assembled to join smoothly after a change of variables. Specifically, NC-splines fill holes in C^1 bi-quadratic (bi-2) tensor-product surfaces with G^1 continuity. Alternative approaches create an infinite series of nested bi-2 polynomial surface rings [11]). [29, 30] address Δ^1 configurations, called τ_0 -nets (as well as related τ_1 and τ_2 -nets).

An alternative spline solution for τ_1 -nets is the use of T-splines [31]. However, T-splines encounter difficulties with quad-dominant meshes due to their strict global knot compatibility requirements [31, Rule1] that often fail when the number of quad strips is reduced. This can lead to non-smooth parameterizations or require mesh modifications ([29, 32]). Enforcing these constraints frequently results in artificial extraordinary points or zero knot intervals, making them impractical for polyhedral modeling in certain situations [33].

For Δ^2 -nets, [4] generates three degree (2,3) and six degree (2,4) pieces that form a 3×3 macro-patch. An alternative bi-3 construction FC_8^3 [5] improves FC^3 by reducing the number of pieces from 11 to 8 and improves uniformity of the highlight lines. While [4, 5, 30] provide individual solutions, deriving an approach for general n is tricky due to an increasing number of interior points and the requirement of highlight lines as uniform as for the tensor-product bi-2 case.

3. Setup

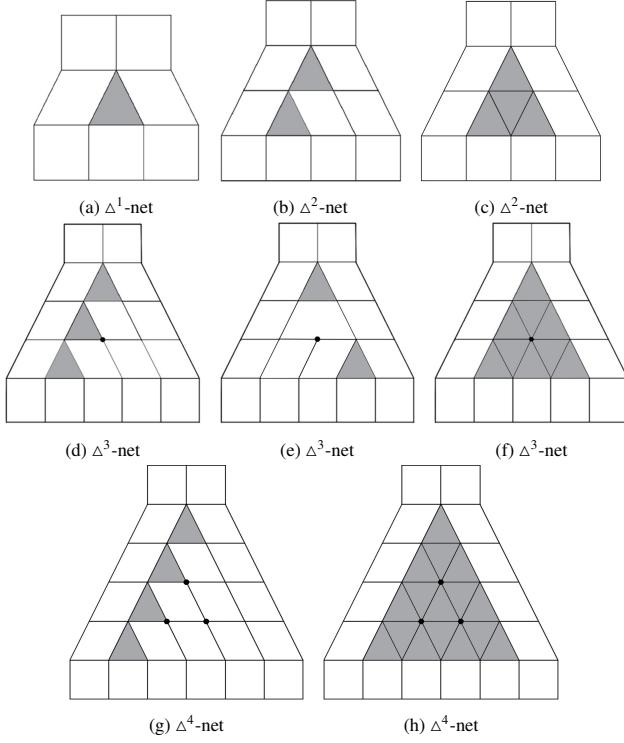
Polyhedral nets, a class of quad-dominant meshes that admit a polyhedral net spline [9], can contain configurations that join $n \neq 4$ features, such as n-gons, valence n and polar vertices, and configurations that have preferred directions, often induced by an orientation field. All configurations must be isolated by a surrounding quad strip. NC^n -splines have two preferred directions, one that we will call *vertical*, for shedding $n-1$ quad strips, one for each of $n-1$ *horizontal* strips. This permits tighter packing of mesh line reductions.

3.1. Δ^{n-1} -net for NC^n splines

Fig. 3 shows a Δ^{n-1} control net of an NC^n -spline. For $k > 2$, Δ^k has points, such as \mathbf{d}_{33} (marked • in Fig. 4 b) that does not lie in the tensor border. The inner connectivity, whose triangles are gray, is not used by the NC^n -spline construction. However interior mesh point placement serves as handles determining the shape of NC-spline surfaces. While, of course, the input Δ^{n-1} net does not have a tensor-product (regular quad-grid) structure, the output consists exclusively of a grid of $n \times n$ pieces industry-standard polynomial pieces, each in tensor-product form, see Fig. 4 c for $n = 4$. The ij -labels of the pieces should not be confused with either with BB-coefficient indices or the indices of the Δ^3 -net.

3.2. Surface representation

The NC-spline consists of tensor-product pieces of polynomial bi-degree (d, d') in Bernstein-Bézier form (*BB-form*, [34]).

Fig. 3: Examples of Δ^{n-1} -nets for NC^n -splines.

For Bernstein polynomials $B_k^d(t) := \binom{d}{k}(1-t)^{d-k}t^k$ a patch \mathbf{p} of bi-degree (d, d') is defined as

$$\mathbf{p}(u, v) := \sum_{i=0}^d \sum_{j=0}^{d'} \mathbf{p}_{ij} B_i^d(u) B_j^{d'}(v), \quad 0 \leq u, v \leq 1.$$

Fig. 4 c depicts the 4×4 patches of NC^4 for Δ^3 . The eight red patches $11, \dots, 41$ and $14, \dots, 44$ are of degree $(2, 4)$, the eight yellow patches $12, \dots, 42, 13, \dots, 43$, are of degree $(2, 3)$, where 2 is the degree in the horizontal direction. 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 (Fig. 5).

3.3. 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. 5 the control-net of the input mesh 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. 5. 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.

3.4. 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. [35],

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

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

Besides the shared BB-coefficients of the common boundary, only the layers of BB-coefficients \mathbf{p}_{il} and \mathbf{q}_{il} of adjacent

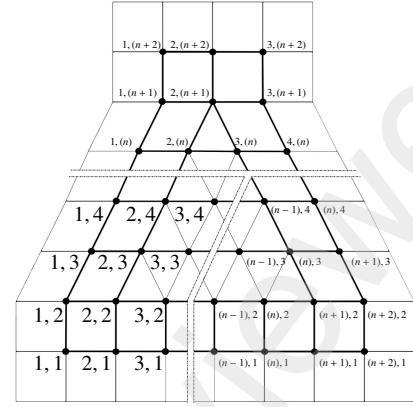
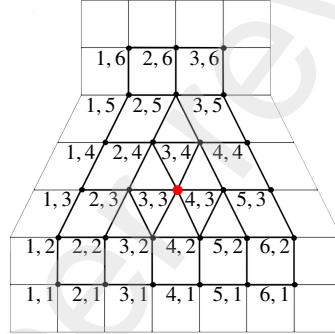
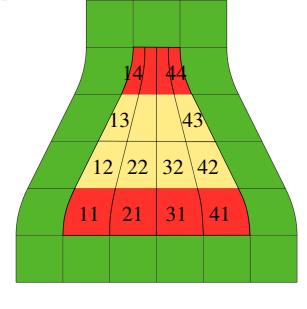
(a) Extended Δ^{n-1} -net(b) Extended Δ^3 -net of $NC_4^n, n = 4$ (c) NC_4^4 surface pieces

Fig. 4: Extended Δ^{n-1} -nets have one additional surrounding quad frame to help define a surrounding frame of regular uniform bi-quadratic (bi-2) C^1 spline patches, see (c). Here $n-1$ counts facets along edge, corresponding to n pieces along an edge. (b) Δ^3 -net with subscript labels \mathbf{d}_{ij} and middle point \mathbf{d}_{33} , marked \bullet . (c) NC_4^4 with $2n = 8$ pieces of degree $(2, 4)$, colored red, and $n(n-2) = 8$ pieces of degree $(2, 3)$, colored yellow. (The corresponding BB-nets are not displayed.) The superscript of NC_4^n indicates that the surface has n^2 polynomial pieces and the subscript the maximal of the tensor-product degree.

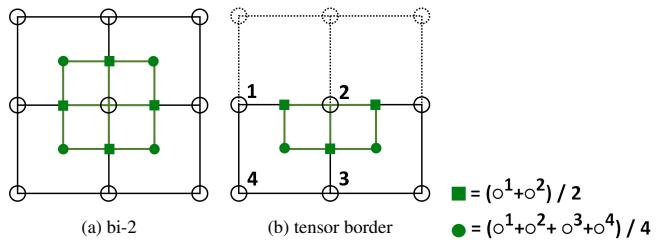


Fig. 5: Tensor-borders \mathbf{t} and B-to-BB conversion from Hermite input data. Circles \circ mark control-net of the input mesh, \bullet, \blacksquare mark BB-coefficients of the bi-2 patch (a) and tensor-border (b).

patches enter the G^1 continuity constraints. In the derivation, u -, v -directions can be assigned as convenient, but typically u is used to parameterize along the boundary and v in the orthogonal direction of the tensor-border, towards the interior core.

4. The NC_4^n construction

NC_4^n first constructs a tensor-border ‘frame’ that preserves the first-order Hermite data defined by the B-spline interpretation of the Δ^{n-1} -net and a change of variables, see Section 4.1. This frame is filled with a regular $C^1 n \times n$ macro-patch, as described in Section 4.2: The C^1 surface is constructed by first applying

B-to-BB conversion and splitting to the horizontal layers of Δ^n , and extracting vertical C^1 degree-2 control points that serve as the backbone or ‘spine’ of the remaining construction. A key challenge is to prevent unwanted oscillations. [Section 4.3](#) summarizes the algorithm.

4.1. Constructing the tensor-border frame

Given a Δ^{n-1} -net with points $\mathbf{d}_{i,j}$, see Fig. 4 a, b, we extract bi-2 tensor borders $\underline{\mathbf{t}}^s, \hat{\mathbf{t}}^s, \bar{\mathbf{t}}, \underline{\mathbf{t}}^s$ by partial B-to-BB conversion: for $s = 1, \dots, n$

$$\begin{array}{lll} \text{left} & \mathbf{d}_{i,s+j}, i = 1, 2, j = 0, 1, 2 & \rightarrow \underline{\mathbf{t}}^s, \\ \text{right} & \text{symmetric to left} & \rightarrow \hat{\mathbf{t}}^s, \\ \text{top} & \mathbf{d}_{i,(n+2)-j}, i = 1, 2, 3, j = 0, 1 & \rightarrow \bar{\mathbf{t}}, \\ \text{bottom} & \mathbf{d}_{s+i,j}, i = 0, 1, 2, j = 1, 2 & \rightarrow \underline{\mathbf{t}}^s. \end{array}$$

The left, right and bottom of the tensor-border frame consist of n pieces. However the top consists of only one piece preventing an $n \times n$ fill. To resolve this mismatch, the top is uniformly split into n pieces. This forces a reparameterization $\rho^s(u, v) := (u, a^s(u)v)$ of the left and right tensor-borders: $\hat{\mathbf{t}}^s := \underline{\mathbf{t}}^s \circ \rho^s$ and $\hat{\mathbf{t}}^s := \hat{\mathbf{t}}^s \circ \rho^s$, see Fig. 6 a. To avoid additional reparameterizations of the top or bottom, and to keep the final degrees as low as possible, the reparameterizations a^1 and a^n are chosen to be quadratic and a^2, \dots, a^{n-1} to be linear. The scalar BB-coefficients of the reparameterizations are

$$\begin{aligned} [a_0^1, a_1^1, a_2^1] &:= [1, 1, a_0^2]; \quad m = 2, \dots, n-1 (n > 2) \\ [a_0^m, a_1^m] &:= [a_0^m, a_0^{m+1}], \quad a_0^m := 1 - \frac{2m-3}{2n} \\ [a_0^n, a_1^n, a_2^n] &:= [a_0^n, \frac{1}{n}, \frac{1}{n}]. \end{aligned} \quad (1)$$

We choose the minimal horizontal degree 2 for the un-reparameterized $\underline{\mathbf{t}}^s$ and $\bar{\mathbf{t}}^s$. Due to reparameterization, the top segments $\hat{\mathbf{t}}^n, \hat{\mathbf{t}}^n$, and bottom segments $\hat{\mathbf{t}}^1, \hat{\mathbf{t}}^1$ are of degree $(4, 2)$ and degree 4 is matched by degree-raising top and bottom $\underline{\mathbf{t}}^s, \hat{\mathbf{t}}^s$. All other curves of the left and right frame (with BB-coefficients $\hat{\mathbf{t}}_{i,0}^s, \hat{\mathbf{t}}_{i,0}^s, s = 2, \dots, n-1$) are raised to degree 3.

The top and bottom of the frame, omitting superscripts and left/right indicators $|$, have first cross-derivative BB-coefficients for the corners $s \in \{0, n\}$,

$$\begin{aligned} \hat{\mathbf{t}}_{01} &= (1 - a_0) \mathbf{t}_{00} + a_0 \mathbf{t}_{01}, \\ \hat{\mathbf{t}}_{11} &= ((1 - a_1) \mathbf{t}_{00} + a_1 \mathbf{t}_{01} + (1 - a_0) \mathbf{t}_{10} + a_0 \mathbf{t}_{11}) / 2, \\ \hat{\mathbf{t}}_{21} &= ((1 - a_2) \mathbf{t}_{00} + a_2 \mathbf{t}_{01} + 4(1 - a_1) \mathbf{t}_{10} + 4a_1 \mathbf{t}_{11}) / 6 \\ &\quad + a_0 \mathbf{t}_{21} + (1 - a_0) \mathbf{t}_{20}. \end{aligned} \quad (2)$$

$\hat{\mathbf{t}}_{31}^s$ from $\hat{\mathbf{t}}_{11}^s$ by exchange $a_i \leftrightarrow a_{i+1}, \mathbf{t}_{ij} \leftrightarrow \mathbf{t}_{i+1,j}, j = 0, 1,$

$\hat{\mathbf{t}}_{41}^s$ from $\hat{\mathbf{t}}_{01}^s$ by exchange $a_i \leftrightarrow a_{i+2}, \mathbf{t}_{ij} \leftrightarrow \mathbf{t}_{2-i,j}, j = 0, 1,$

and for $s \in \{2, \dots, n-1\}$

$$\begin{aligned} \hat{\mathbf{t}}_{01} &= (1 - a_0) \mathbf{t}_{00} + a_0 \mathbf{t}_{01} \\ \hat{\mathbf{t}}_{11} &= ((1 - a_1) \mathbf{t}_{00} + a_1 \mathbf{t}_{01} + (2 - 2a_0) \mathbf{t}_{10} + 2a_0 \mathbf{t}_{11}) / 3 \\ \hat{\mathbf{t}}_{21}^s & \text{from } \hat{\mathbf{t}}_{11}^s \text{ by exchange } a_i \leftrightarrow a_{1-i}, \mathbf{t}_{ij} \leftrightarrow \mathbf{t}_{2-i,j}, j = 0, 1, \\ \hat{\mathbf{t}}_{31}^s & \text{from } \hat{\mathbf{t}}_{01}^s \text{ by exchange } a_i \leftrightarrow a_{1-i}, \mathbf{t}_{ij} \leftrightarrow \mathbf{t}_{2-i,j}, j = 0, 1. \end{aligned} \quad (3)$$

The reparameterized tensor borders $\hat{\mathbf{t}}^s$ for $(s = 1, \dots, n)$ are computed in the same manner due to left-right symmetry.

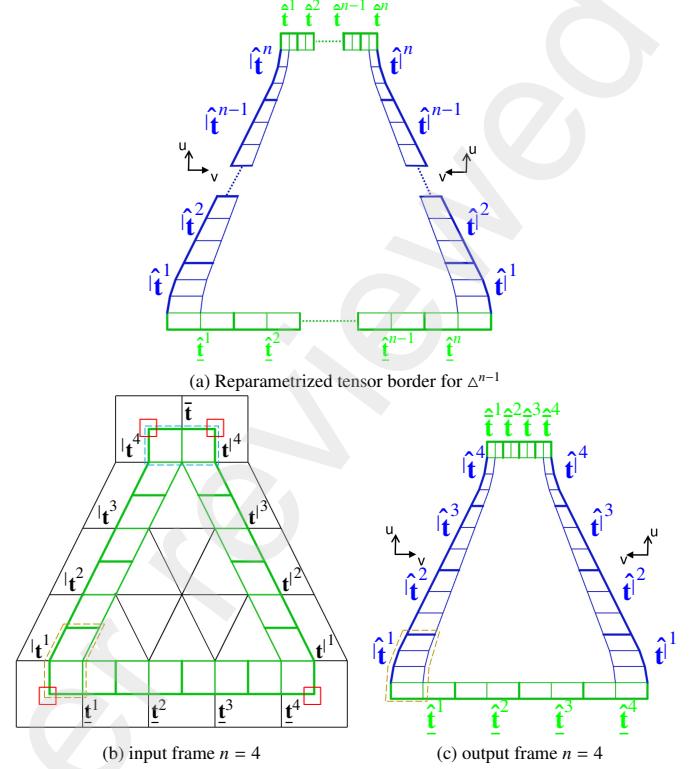


Fig. 6: (a) Illustrates the general reparametrized tensor border; (b) Input bi-2 tensor-border frame obtained from the Δ^3 -net by B- to BB-conversion; (c) The tensor-border reparameterized with $\hat{\rho}^s$. The bottom and top green tensor-borders are not reparameterized but degree-raised to 4 in the vertical direction and the top tensor-border is split.

4.1.1. Example: the frame for NC^4 following Δ^3

Fig. 6 b,c illustrate the reparametrization for NC^4 where the left, right and bottom frame consist of four matching pieces, but the top of only one. To resolve this mismatch, the top is horizontally split into $n = 4$ pieces. The reparameterization $\rho^s(u, v) := (u, a^s(u)v)$ is defined by the coefficients

$$\begin{aligned} [a_0^1, a_1^1, a_2^1] &:= [1, 1, \frac{7}{8}]; \quad [a_0^2, a_1^2] := [\frac{7}{8}, \frac{5}{8}] \\ [a_0^3, a_1^3] &:= [\frac{5}{8}, \frac{3}{8}]; \quad [a_0^4, a_1^4, a_2^4] := [\frac{3}{8}, \frac{1}{4}, \frac{1}{4}]. \end{aligned} \quad (4)$$

The tensor border reparameterized according to Eq.(2) and Eq.(3) is illustrated in Fig. 6 c.

4.2. Interior BB-coefficients

All remaining BB-coefficients are determined by the tensor-border frame and $(n-2)^2$ points \mathbf{c}_{ij} (marked \square in Fig. 8 and Fig. 9) derived from Δ^{n-1} and the tensor-border frame. Fig. 9 shows $n-2$ vertical ‘spines’ \mathbf{c}_{ij} for $n = 5, 6$. Fig. 8 displays the construction of NC_4^n for $n = 4$:

- the nodes \bullet of Δ^{n-1} (with labels introduced in Fig. 4),
- the indices $1, \dots, n$ (both in horizontal and vertical directions) of tensor-border frame pieces,
- the BB-coefficients of an $n \times n C^1$ spline patchwork, see Fig. 4 c.
- arrows \uparrow and \downarrow pointing to $n-2$ vertical layers of actual degree 2 that serve as the spines of construction.

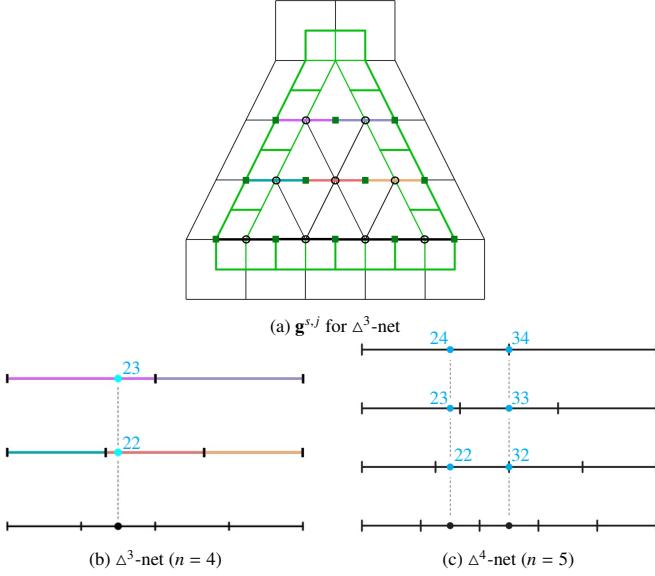


Fig. 7: Relation of parameters (s, u) to c_{ij} . (a) 22: $s := 2, u := \frac{1}{8}$; 23: $s := 1, u := \frac{3}{4}$. (b) 22: $s := 2, u := \frac{1}{5}$; 23: $s := 1, u := \frac{9}{10}$; 24: $s := 1, u := \frac{3}{5}$; 32: $s := 3, u := 0$; 33: $s := 2, u := \frac{1}{2}$; 34: $s := 2, u := 0$. The data for remaining c_{ij} can be obtained by symmetry.

4.2.1. Construction of splines of degree 2

Each horizontal sequence of points of the Δ^{n-1} -net is treated as a sequence of narrowing horizontal splines: the j -th, $j = 2, \dots, n-1$, horizontal layer of nodes $\mathbf{d}_{i,j+1}, i = 1, \dots, n+3-j$, is converted into Bézier curves $\mathbf{g}^{s,j}, s = 1, \dots, n+1-j$, of degree 2 that determine points $\mathbf{c}_{ij}, i = 2, \dots, n-1$, using heuristics inspired by regular bi-2 splines. A key moment, the location where to sample $\mathbf{g}^{s,j}$, is visualized in Fig. 7 for $n = 4$ and $n = 5$.

- The bottom layer $\mathbf{d}_{i,2}, i = 1, \dots, n+2$, defines n Bézier curve segments of uniform degree 2 C^1 spline with domain composed from n unit intervals $[0, 1] \cup [1, 2] \cup \dots \cup [n-1, n] := [0, n]$.
- The guiding curve segments $\mathbf{g}^{s,j}, s = 1, \dots, n+1-j, j = 2, \dots, n-1$, are defined over the union of $n+1-j$ unit intervals forming a composed interval $[0, n+1-j]$. This interval is scaled by $\frac{n}{n+1-j}$ to match that of the $\mathbf{d}_{i,2}$.
- The middle BB-coefficients of each degree 2 Bézier curve are associated with the midpoints of corresponding domain intervals in Fig. 7 (marked by \bullet for $\mathbf{d}_{i,2}$).

For $j = 2, \dots, n-1$, the locations are the solutions of

$$\begin{aligned} \frac{n}{n+1-j}(s-1+u) &= i - \frac{1}{2}, \quad i = 2, \dots, n-1, \text{i.e.} \\ r := \frac{n+1-j}{2n}(2i-1)+1, \quad s &:= \lfloor r \rfloor, \quad u := r-s. \end{aligned} \quad (5)$$

Here s, u determine from which guide $\mathbf{g}^{s,j}$ and at what $u \in [0, 1]$ the \mathbf{c}_{ij} is extracted as (recall that $B_k^d(t) := \binom{d}{k}(1-t)^{d-k}t^k$)

$$\begin{aligned} \mathbf{c}_{ij} &:= \frac{\mathbf{d}_{s,j+1} + \mathbf{d}_{s+1,j+1}}{2} B_0^2(u) + \mathbf{d}_{s+1,j+1} B_1^2(u) \\ &+ \frac{\mathbf{d}_{s+1,j+1} + \mathbf{d}_{s+2,j+1}}{2} B_2^2(u). \end{aligned} \quad (6)$$

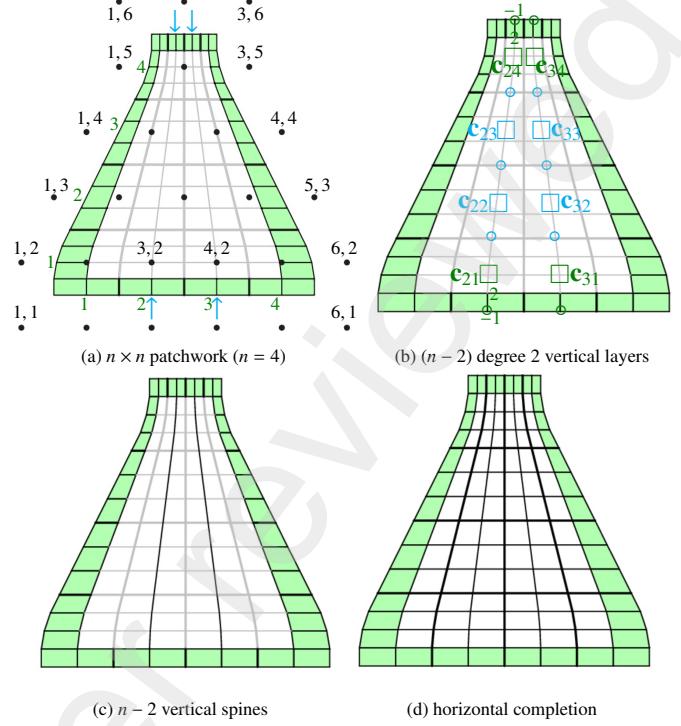


Fig. 8: Construction of inner NC_4^n -spline for $n = 4$.

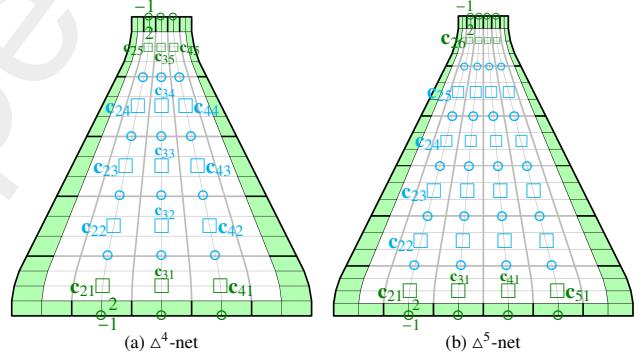


Fig. 9: Degree 2 ‘spine’ curves for Δ^4 -net ($n = 5$) and Δ^5 -net ($n = 6$).

Fig. 7 shows example (s, u) pairs and for $n = 4$ and $n = 5$,

$$\begin{bmatrix} \mathbf{c}_{23} \\ \mathbf{c}_{33} \\ \mathbf{c}_{32} \end{bmatrix} := \frac{1}{128} \begin{bmatrix} 4 & 88 & 36 & 0 & 0 & 0 \\ 0 & 36 & 88 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 49 & 78 \\ 0 & 0 & 0 & 1 & 78 & 49 \end{bmatrix} \begin{bmatrix} \mathbf{d}_{14} \\ \mathbf{d}_{24} \\ \mathbf{d}_{34} \\ \mathbf{d}_{44} \\ \mathbf{d}_{23} \\ \mathbf{d}_{33} \\ \mathbf{d}_{43} \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} \mathbf{c}_{24} \\ \mathbf{c}_{34} \\ \mathbf{c}_{23} \\ \mathbf{c}_{33} \\ \mathbf{c}_{43} \\ \mathbf{c}_{22} \\ \mathbf{c}_{32} \\ \mathbf{c}_{42} \end{bmatrix} := \frac{1}{200} \begin{bmatrix} 16 & 148 & 36 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 36 & 148 & 16 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 118 & 81 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 25 & 150 & 25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 81 & 118 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 64 & 132 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 132 & 64 \end{bmatrix} \begin{bmatrix} \mathbf{d}_{15} \\ \mathbf{d}_{25} \\ \mathbf{d}_{35} \\ \mathbf{d}_{45} \\ \mathbf{d}_{14} \\ \mathbf{d}_{24} \\ \mathbf{d}_{34} \\ \mathbf{d}_{44} \\ \mathbf{d}_{54} \\ \mathbf{d}_{23} \\ \mathbf{d}_{33} \\ \mathbf{d}_{43} \\ \mathbf{d}_{53} \end{bmatrix}.$$

The C^1 extension of the top/bottom tensor-border (with weights 2 and -1 as illustrated in Fig. 8 and Fig. 9) defines the $2(n-2)$ points marked \square . The $n-1$ \circ of each degree 2 ‘spine’ curve are set as averages of their two vertical neighbors \square (or \square and

\square ; i.e. the n sequences $\{\circ, \square, \circ\}$ are BB-coefficients of $n C^1$ -connected Bézier curves. Fig. 9 displays this for Δ^4 - and Δ^5 -nets.

4.2.2. Completion of the patchwork

The $n - 2$ bottom and $n - 2$ top spines of vertical degree 2 are degree-raised to 4, the inner $(n-2)^2$ pieces to 3, see Fig. 8 c. The remaining BB-coefficients enforce horizontal C^1 continuity by averaging the spine coefficients (frame coefficients at the ends), see Fig. 8 d.

Appendix A presents a variant of the algorithm that improves uniformity of the highlight lines near the narrow top of challenging convex input configurations. Fig. 10 shows the highlight line distribution on NC_4^n splines for $n \in \{3, 4, 5, 6\}$ without the improvement of Appendix A .

4.3. Summary

NC_4^n Algorithm

1. Compute the tensor-border frame by B-to-BB conversion and (1),(2),(3), see Fig. 6 and Section 4.1.
2. Compute the $2(n - 2)$ control points \mathbf{c}_{i1} and \mathbf{c}_{in} as affine extensions of the top/bottom tensor-border with weights 2, -1, see Fig. 9 .
3. Compute the $(n - 2)^2$ control points \mathbf{c}_{ij} by (6).
4. Compute the $(n - 1)(n - 2)$ points \circ as an average of their vertical neighbors \mathbf{c}_{ij} , \mathbf{c}_{i1} , resp. \mathbf{c}_{in} to ensure (vertical) C^1 continuity.
5. Raise the degree of the $2(n - 2)$ top and bottom pieces to 4, and the inner $(n - 2)^2$ pieces to 3.
6. Set all remaining BB-coefficients as $1/2$ averages of their immediate neighbor BB-coefficients to enforce C^1 continuity in the horizontal direction.
7. (optional) update the narrow top according to Appendix A.

16

Remark. As opposed to other heuristics choices that we evaluated, the chosen \mathbf{c}_{ij} does not reproduce the tensor-product case but results in the best highlight line distribution. For a 8×8 net, Fig. 11 b superimposes the standard bi-2 spline with a bi-2 spline whose 'horizontal' layers of B-spline nodes are re-defined using the above NC-construction, indicating only a minute change that is unlikely to surprise a designer.

5. An alternative bi-cubic NC_3^n construction

The construction NC_3^n resolves the corner mismatches of tensor-borders using degree bi-3. This requires re-parameterizing top and bottom. Since [4, Sec.5] showed that the reparameterization is unique and can only be applied if number of pieces is even, the bottom middle piece is uniformly split into two when n is odd. This forces a non-uniform split of the top as visualized for $n = 5$ in Fig. 13 b. Consequently, the construction consists of n^2 bi-3 pieces for n even, and $n(n + 1)$ bi-3 pieces for n odd.

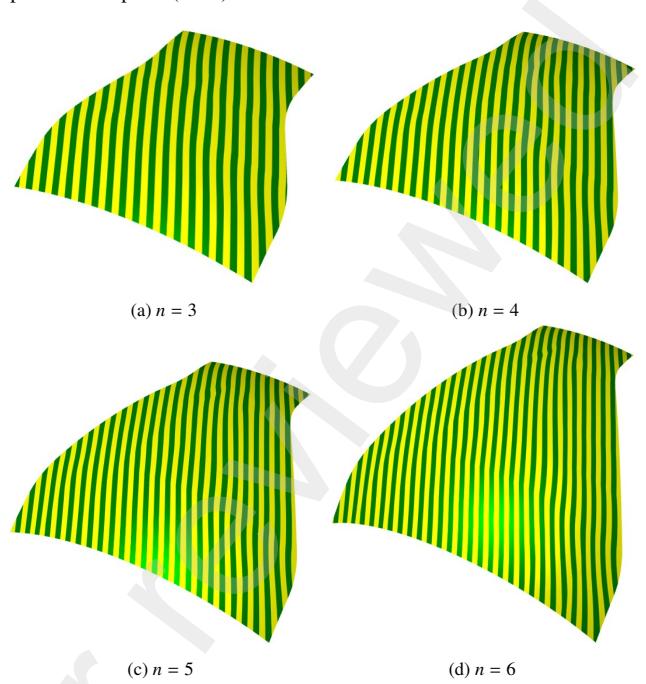


Fig. 10: Highlight line distributions of NC_4^n without Appendix A improvement.

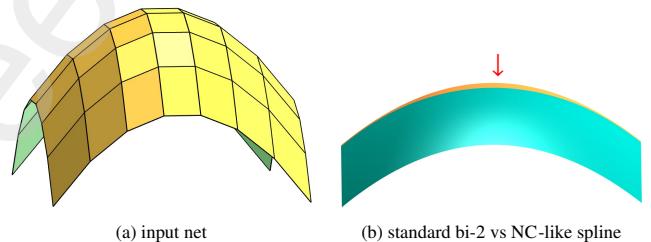


Fig. 11: Mimicking, but not reproducing: the regular bi-2 spline (gold in (b), \downarrow) is almost hidden by the cyan NC modified one. (In (b) the outer bi-2 ring is not shown.)

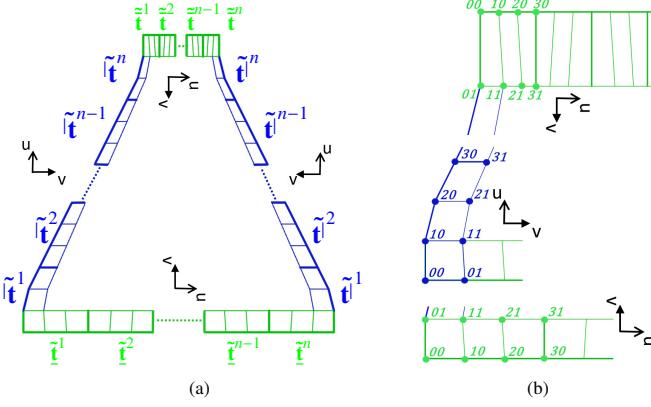
Section 5.1 details the derivation of tensor-border frame, while Section 5.2 explains the construction of interior control points.

5.1. Adjusting the tensor-border frame

The bi-2 tensor border frame is initialized as described in Section 4.1 and Fig. 6 a. The left, right and bottom of the tensor-border frame each have n pieces, the top only one. To match the bottom, the top is split (horizontally) into n equal pieces if n is even and $n + 1$ pieces with spacing $(2, \dots, 2, 1, 1, 2, \dots, 2)/(2n)$ if n is odd. Fig. 12 shows the tensor borders $|\mathbf{t}^s|$ and $|\mathbf{t}^{l^s}|$ reparameterized by $\rho^s(u, v) := (u, a^s(u)v)$, for $s = 1, \dots, n$. Due to the maximum degree 3 of NC_3^n , a^s can be at most linear. The BB-coefficients of the scalar-valued polynomials a^s are

$$\begin{aligned} [a_0^1, a_1^1] &:= [1, a_0^2], \quad m = 2, \dots, (n-1) \\ [a_0^m, a_1^m] &:= [a_0^m, a_0^{m+1}], \quad a_0^m := 1 - \frac{(n-1)(m-1)}{n^2} \\ [a_0^n, a_1^n] &:= [a_1^n, \frac{1}{n}], \end{aligned} \quad (8)$$

so that abutting $\tilde{\mathbf{t}}^s := |\mathbf{t}^s \circ \tilde{\rho}^s|$ join C^1 , as do $\tilde{\mathbf{t}}^{l^s} := |\mathbf{t}^{l^s} \circ \tilde{\rho}^s|$. Here $\tilde{\mathbf{t}}_{i,0}^s$, $i = 0, \dots, 3$ are the degree-raised boundary, and $\tilde{\mathbf{t}}_{01}$ and $\tilde{\mathbf{t}}_{11}$

Fig. 12: NC_3^n : reparameterized tensor borders.

are defined by Eq.(9) (omitting the superscripts):

$$\begin{aligned} \tilde{\mathbf{t}}_{01} &= ((3 - 2a_0)\mathbf{t}_{00} + 2a_0\mathbf{t}_{01})/3, \\ \tilde{\mathbf{t}}_{11} &= ((3 - 2a_1)\mathbf{t}_{00} + 2a_1\mathbf{t}_{01} + (6 - 4a_0)\mathbf{t}_{10} + 4a_0\mathbf{t}_{11})/9, \\ \hat{\mathbf{t}}_{21}^s &\text{ from } \hat{\mathbf{t}}_{11}^s \text{ by exchange } a_i \leftrightarrow a_{1-i}, \mathbf{t}_{ij} \leftrightarrow \mathbf{t}_{2-i,j}, j = 0, 1, \\ \hat{\mathbf{t}}_{31}^s &\text{ from } \hat{\mathbf{t}}_{01}^s \text{ by exchange } a_i \leftrightarrow a_{1-i}, \mathbf{t}_{ij} \leftrightarrow \mathbf{t}_{2-i,j}, j = 0, 1. \end{aligned} \quad (9)$$

Then $\underline{\mathbf{t}}^s$ and $\bar{\mathbf{t}}^s$ are degree-raised so that $\underline{\tilde{\mathbf{t}}}^s$ and $\bar{\tilde{\mathbf{t}}}^s$ are degree-3 in the horizontal direction. The reparametrization $\rho^s(u, v) := (u + b^s(u)v, v)$ is required for the bottom and top tensor borders to match the shared four BB-coefficients of the left and right tensor borders. The degree of b^s can be as high as 2 without $\underline{\tilde{\mathbf{t}}}^s$ and $\bar{\tilde{\mathbf{t}}}^s$ exceeding degree 3:

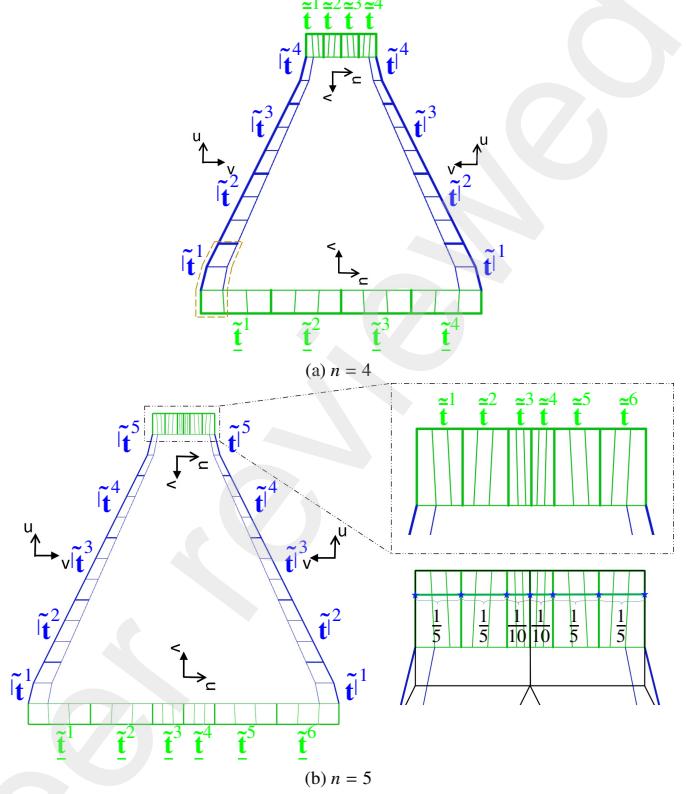
$$\begin{aligned} \text{bottom: } [b_0^s, b_1^s, b_2^s] &:= \left[0, \frac{((n-1)/2)(-1)^s}{n^2}, 0\right] \\ \text{top: } [b_0^s, b_1^s, b_2^s] &:= \left[0, \frac{((n-1)/2)(-1)^{s+1}}{n^2}, n, 0\right] \end{aligned} \quad (10)$$

The b^s ensure that the reparametrized top and bottom tensor borders, $\underline{\tilde{\mathbf{t}}}^s := \underline{\mathbf{t}}^s \circ \rho^s$ and $\bar{\tilde{\mathbf{t}}}^s := \bar{\mathbf{t}}^s \circ \rho^s$, maintain C^1 -connectivity; and $\underline{\tilde{\mathbf{t}}}_{i0}^s$ and $\bar{\tilde{\mathbf{t}}}_{i0}^s$ are obtained by degree-raising. The remaining points (omitting superscripts) are

$$\begin{aligned} \tilde{\mathbf{t}}_{01} &= (\mathbf{t}_{00} + 2\mathbf{t}_{01})/3, \\ \tilde{\mathbf{t}}_{11} &= ((1 - 4b_1)\mathbf{t}_{00} + 2\mathbf{t}_{01} + (4b_1 + 2)\mathbf{t}_{10} + 4\mathbf{t}_{11})/9, \\ \tilde{\mathbf{t}}_{21} &= ((-4b_1 + 2)\mathbf{t}_{10} + 4\mathbf{t}_{11} + (4b_1 + 1)\mathbf{t}_{20} + 2\mathbf{t}_{21})/9, \\ \tilde{\mathbf{t}}_{31} &= (\mathbf{t}_{20} + 2\mathbf{t}_{21})/3. \end{aligned} \quad (11)$$

Examples: For $n = 4, 5$ Fig. 13 illustrates the frame and reparametrizations for an even and an odd case. For $n = 4$, a Δ^3 -net the reparameterizations are

$$\begin{aligned} [\mathbf{a}_0^s, \mathbf{a}_1^s]_{s=1, \dots, 4} &:= \left[1, \frac{13}{16}\right], \left[\frac{13}{16}, \frac{10}{16}\right], \left[\frac{10}{16}, \frac{7}{16}\right], \left[\frac{7}{16}, \frac{1}{4}\right], \\ [b_0^s, b_1^s, b_2^s] &:= \left[0, \frac{3(-1)^s}{32}, 0\right]_{\text{bottom}}, \left[0, \frac{3(-1)^{s+1}}{8}, 0\right]_{\text{top}}. \end{aligned}$$

Fig. 13: NC_3^4 and NC_3^5 : reparameterized tensor borders.

For $n = 5$, a Δ^4 -net, $s = 1, \dots, 5$:

$$\begin{aligned} [\mathbf{a}_0^1, \mathbf{a}_1^1] &:= \left[1, \frac{21}{25}\right]; [\mathbf{a}_0^2, \mathbf{a}_1^2] := \left[\frac{21}{25}, \frac{17}{25}\right]; [\mathbf{a}_0^3, \mathbf{a}_1^3] := \left[\frac{17}{25}, \frac{13}{25}\right] \\ [\mathbf{a}_0^4, \mathbf{a}_1^4] &:= \left[\frac{13}{25}, \frac{9}{25}\right]; [\mathbf{a}_0^5, \mathbf{a}_1^5] := \left[\frac{9}{25}, \frac{1}{5}\right], \end{aligned}$$

and, due to uniformly splitting the middle patch (to prevent mismatch), for $s = 1, \dots, 6$

$$[b_0^s, b_1^s, b_2^s] := \left[0, \frac{2(-1)^s}{25}, 0\right]_{\text{bottom}}, \left[0, \frac{2(-1)^{s+1}}{5}, 0\right]_{\text{top}}.$$

5.2. Inner construction

For even n , the tensor-border frame (green) is constructed as in Section 5.1, and the $(n-2)^2$ inner patches (red with labels 22, 32, 23 and 33 in Fig. 14 a for $n = 4$) are taken from Section 4.2 and degree-raised to bi-3. Then the gray underlaid BB-coefficients C^1 extend the inner patches. Fig. 14 b illustrates the analogous odd case for $n = 5$ where the patches with labels 32, 33, 34 from Section 4.2 are uniformly split.

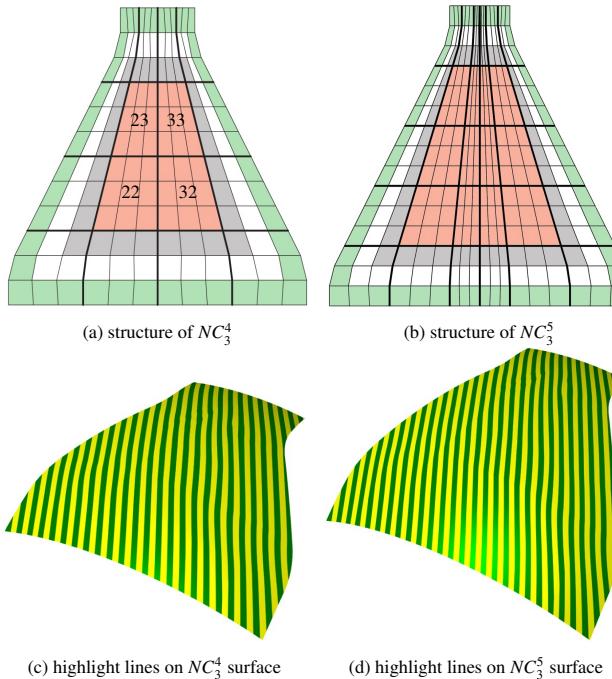


Fig. 14: Bi-cubic NC_3^n surfaces consisting of $2\lfloor \frac{n+1}{2} \rfloor \times n$ pieces.

NC_3^n Algorithm

1. Compute the tensor-border frame by B-to-BB conversion and (8),(9),(10),(11).
2. Compute the inner patches (red in Fig. 14) as in NC_4^n , and split the center pieces in the middle if n is odd.
3. Set all remaining BB-coefficients to C^1 -extend the inner patches.
4. (optional, for challenging convex configurations)
Apply the improvements of Appendix B.

1

6. Analysis: Tests and comparisons

The NC_4^n construction is analyzed with focus on the most common Δ^3 -nets, surrounded by a layer of quads to judge the quality of transition from surrounding bi-2 surfaces to NC_4^n . In the absence of specific intended features, uniform highlight line distribution [36] is considered high quality.

We first verify that NC-splines closely reproduce the highlight line distribution of the regular bi-2 counterpart, despite the non-rectangular topology. Fig. 15 juxtaposes a Δ^3 -net with the d_{33} moved (a) vertically along the u -isolines or horizontally as in (b). The similarity of the highlight line distributions to the tensor-product analogues in Fig. 15 c and d confirms the preservation of preferred directions by NC_4^n surfaces.

Fig. 16, too, juxtaposes Δ^3 -nets with their regular tensor-product counterparts. Fig. 16 a,b compare a net placed on a sphere; here minute oscillations are visible in NC_4^4 compared to the regular case. Fig. 16 c,d compares a partial, double spine-aligned ridge at the spread-out end. Both surfaces have a similar

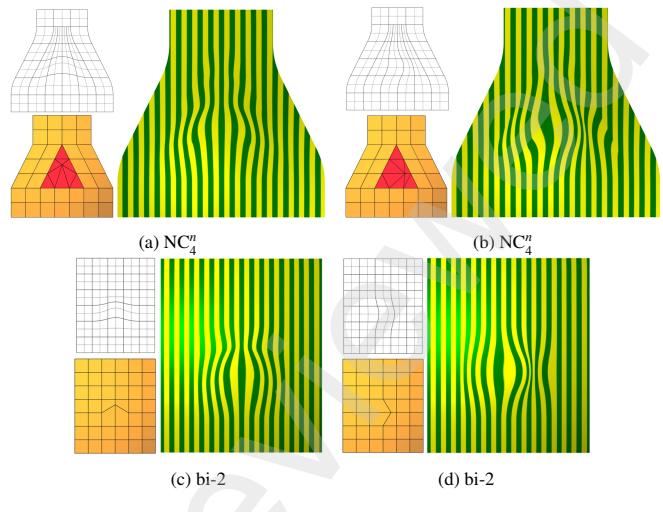


Fig. 15: NC_4^n vs bi-2: irregular placement of d_{33} : (a,b) Δ^3 -net with NC_4^n vs (c,d) regular bi-2 analogue.

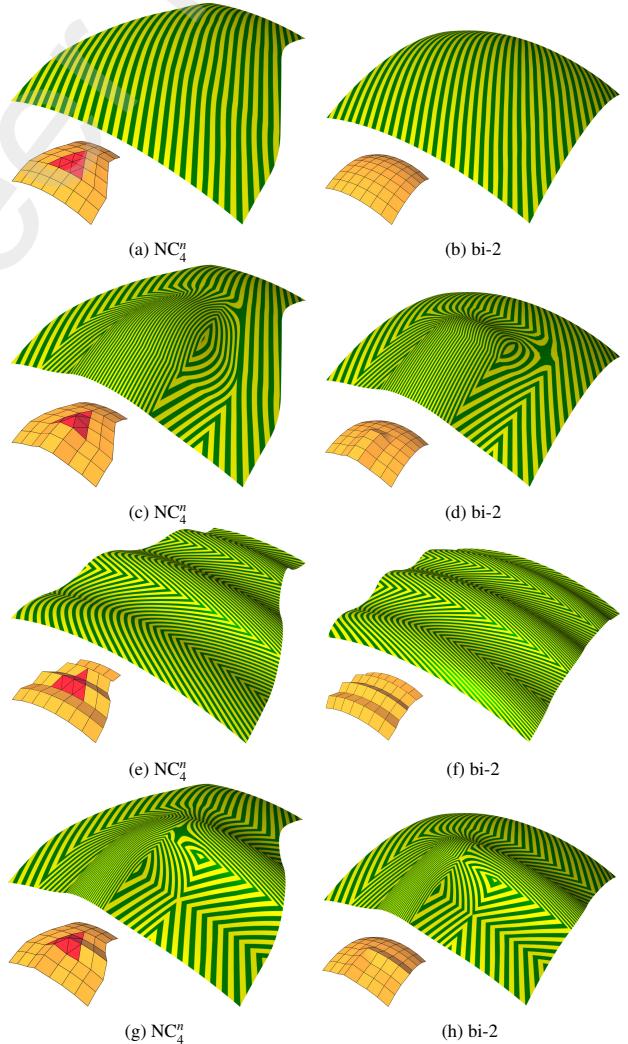


Fig. 16: NC_4^n vs bi-2: left column: extended Δ^3 -net and NC_4^n -spline; right column: regular bi-2 counterparts. The highlight line distributions of the NC_4^n surfaces, left, are very similar to tensor-product surfaces, right.

highlight line distribution but with visible C^1 (not C^2) transi-

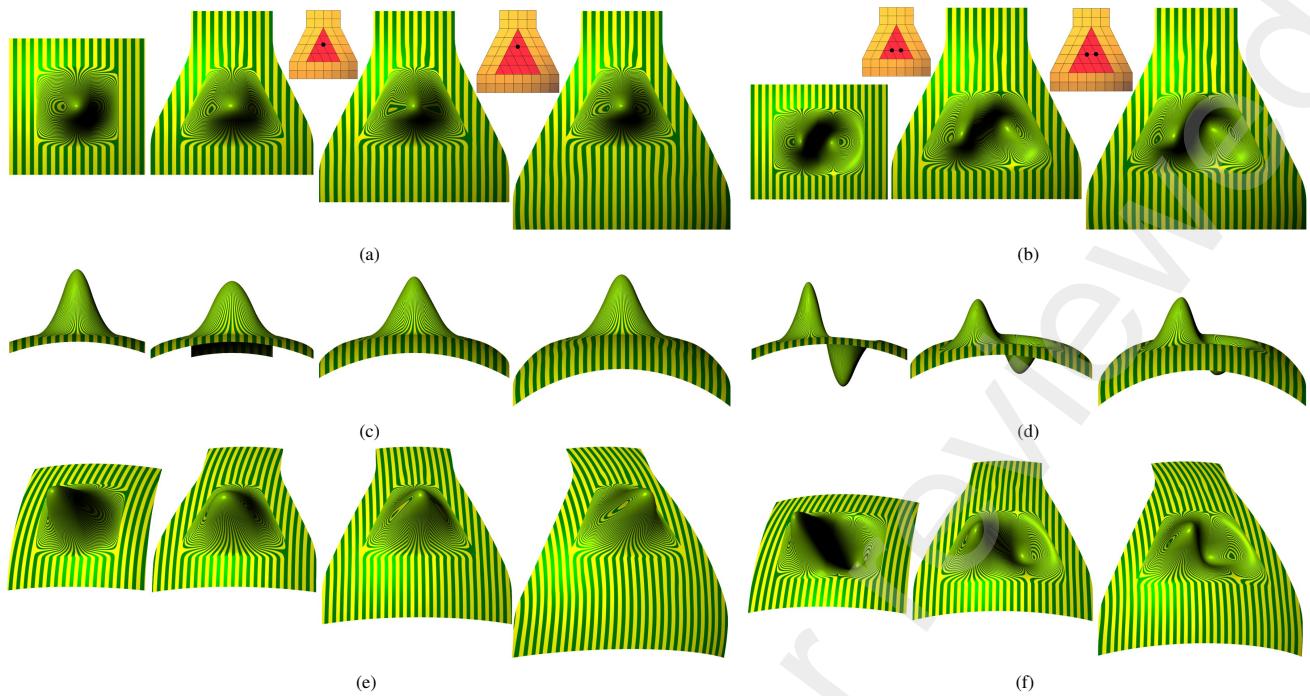


Fig. 17: NC_4^n vs bi-2: *top to bottom*, 3 views juxtaposing a bi-2 spline; *left* in each subfigure, with (a,c,e) NC-splines for, *right* Δ^3 - Δ^4 - and Δ^5 -nets and (b,d,f) NC 5 - and NC 6 -splines. The lifted Δ^{n-1} -net point (indicated by \bullet) in the left column is d_{3n} and, in the right column, d_{33} and d_{43} for the Δ^4 , and d_{34} and d_{44} for Δ^5 .

tions. Fig. 16 e,f tests horizontal ridges and Fig. 16 g,h a partial, single (left) spine-aligned ridge. Again the results are alike.

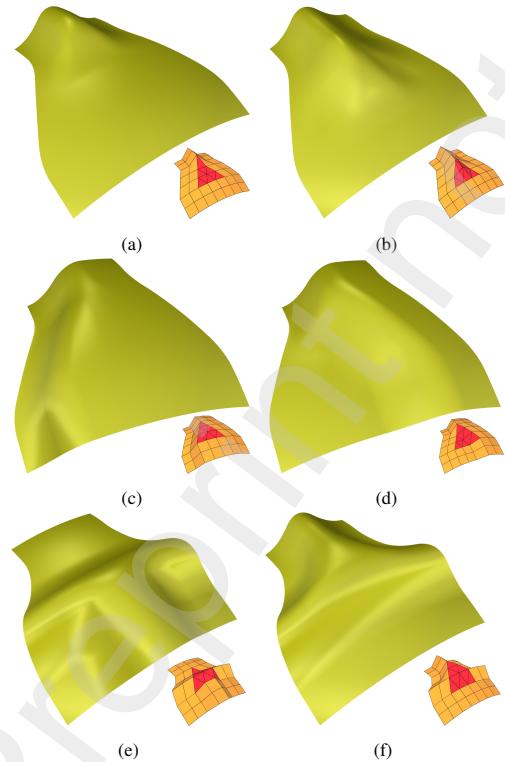


Fig. 18: Extended Δ^3 -nets and corresponding NC_4^n surfaces.

Fig. 17 compares NC_4^n , for strongly deformed Δ^3 - Δ^4 - and Δ^5 -nets with their bi-2 analogues. (a,c,e) have the top interior point lifted. As expected, the deformed area on the Δ^{n-1} -nets

narrows towards the top, but the leaves the remainder unaffected. As n increases, the spike more closely resembles its bi-2 analogue, because the horizontal curves capture the shape better. (b,d,f) visualize opposing spikes.

Fig. 18 analyzes the effect of feature lines touching or straddling the Δ^3 -net, colored orange, on NC_4^4 , colored yellow. (a,b) feature a central ridge of different length and NC_4^4 retaining the feature lines. (c,d) feature ridges following the sides of Δ^3 with (d) cutting its core. The NC_4^4 surfaces preserve these preferences. (e,f) feature partial ridges that merge with a horizontal ridge. (e) starts from the narrow end and (f) from the wide end. The slight bulge is akin to that of NC^3 , [4].

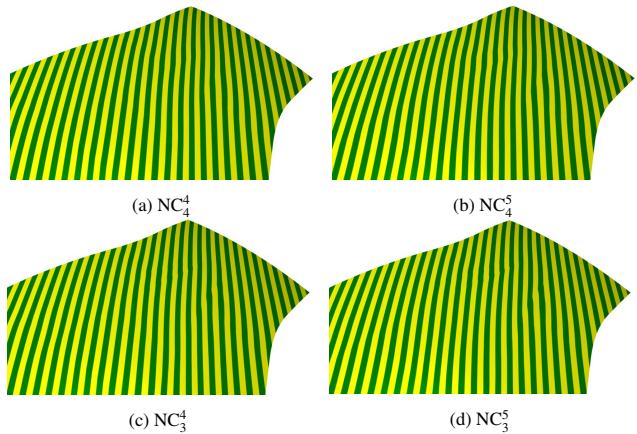


Fig. 19: Comparing NC_4^n , *top*, to NC_3^n , *bottom*, comparison on the most challenging surface type: slightly convex

Fig. 19 demonstrates that, since they share the interior construction, the highlight lines of NC_4 and NC_3 hardly differ and mainly at the transition from the top tensor-border.

While Fig. 20 a vs c demonstrates that carefully combining the functional \mathcal{F}^4 of [4] to account for interior points *can* produce a better highlight line distribution for the convex net of Fig. 16 a, in general, we were unable to find functionals that respect the interior points of the Δ^n net and yield consistently good shape, see Fig. 20 b vs d.

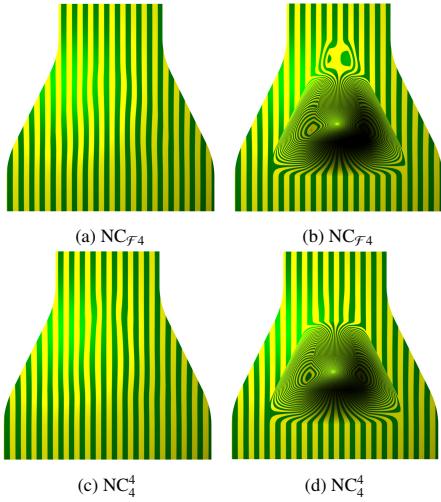


Fig. 20: Comparison of (top) constructions $\text{NC}_{\mathcal{F}4}$ based on a functional vs. NC_4^4 .

Isotropic n -direction polar splines [18, 37–40] are based on polar nets that are structurally fundamentally different from of NC_4^{n-1} and Δ^n configurations that shed mesh lines in a 2-direction grid and so need not be compared to.

Locally, Δ^n nets can be re-meshed to admit a T-spline [41] surface. But T-spline knot intervals for the narrow top part, cf. Fig. 1, must add up to the same interval sum as the wide bottom part with many more segments, making the knot spacing quite different from the geometric spacing. Moreover, no global parameterization may exist for a given input mesh, as demonstrated in [32, Fig. 2].

7. Conclusion

The addition of general rules for narrowing cascades of triangles further extends the use of sparse quad-dominant meshes as control nets. But how many special configurations should one add to polyhedral nets [9] to cover the exceptional cases in fast quad-dominant meshing algorithms? Should one not, instead, remove the exceptional cases in quad-dominant meshing? Regardless of how one answers these questions, the shedding of mesh lines is a valid category of its own, also for polyhedral net design from scratch. This justifies the presented general NC^n for narrowing cascades. And while some input meshes benefit from the improvement in the Appendices, the basic algorithm is straightforward to implement and comes in two flavors: the more natural class of NC_4^n algorithms that uses quadratic reparameterization, and the engineered version NC_3^n that generates a surface of degree bi-3. The inclusion of NC_4^n into the family of polyhedral net splines is part of ongoing work to obtain a general purpose pipeline – to convert scanned data without unnecessary trimming into CAD-compatible tensor-product surfaces.

Acknowledgements ‘Bimba’, ‘Vase Lion’, and ‘Bunny, Botsch’ in Fig. 2 were generated from sources of the same name using Instant Meshes [1].

References

- [1] W. Jakob, M. Tarini, D. Panozzo, O. Sorkine-Hornung, Instant field-aligned meshes, ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 34 (6) (2015) 189:1–189:15.
- [2] N. Schertler, M. Tarini, W. Jakob, M. Kazhdan, S. Gumhold, D. Panozzo, Field-aligned online surface reconstruction, ACM Transactions on Graphics (Proceedings of SIGGRAPH) 36 (4) (Jul. 2017).
- [3] K. Karčiauskas, J. Peters, High quality refinable g-splines for locally quad-dominant meshes with T-gons, in: Computer Graphics Forum, Vol. 38, Wiley Online Library, 2019, pp. 151–161.
- [4] E. Gunpinar, K. Karčiauskas, J. Peters, Splines for fast-contracting polyhedral control nets, Computer-Aided Design (2024) 103727.
- [5] K. Karčiauskas, K. S.-H. Lo, E. Gunpinar, J. Peters, Bicubic splines for fast-contracting control nets, Axioms 13 (6) (2024).
- [6] E. Catmull, J. Clark, Recursively generated b-spline surfaces on arbitrary topological meshes, Computer-Aided Design 10 (6) (1978) 350–355.
- [7] K. Karčiauskas, J. Peters, Localized remeshing for polyhedral splines, Computer Graphics Forum 106 (2022) 58–65.
- [8] K. Karčiauskas, J. Peters, Point-augmented bi-cubic subdivision surfaces, Computer Graphics Forum 41 (2023) 13–23.
- [9] J. Peters, K. Lo, K. Karčiauskas, Algorithm 1032: Bi-cubic splines for polyhedral control nets, ACM Tr on Math Software 49 (March 2023).
- [10] K. Beier, Y. Chen, Highlight-line algorithm for realtime surface-quality assessment, Comput. Aided Des. 26 (4) (1994) 268–277.
- [11] D. Doo, M. Sabin, Behaviour of recursive division surfaces near extraordinary points, Computer-Aided Design 10 (6) (1978) 356–360.
- [12] P. Salvi, T. Várady, Multi-sided surfaces with fullness control, in: Proceedings of the Eighth Hungarian Conference on Computer Graphics and Geometry, NJSZT, 2016, pp. 61–69.
- [13] G. Hettinga, J. Kosinka, A multisided C^2 B-spline patch over extraordinary vertices in quadrilateral meshes, Computer-Aided Design 127 (2020) 102855.
- [14] M. Vaitkus, T. Várady, P. Salvi, Ágoston Sipos, Multi-sided B-spline surfaces over curved, multi-connected domains, Computer Aided Geometric Design 89 (2021) 102019.
- [15] K. Karčiauskas, J. Peters, Evolving guide subdivision, Computer Graphics Forum 42 (2) (2023) 321–332.
- [16] D. Toshniwal, H. Speleers, R. Hiemstra, T. 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.
- [17] K. Karčiauskas, J. Peters, Smooth polar caps for locally quad-dominant meshes, Comput. Aided Geom. Des. 81 (101908) (2020) 101908.
- [18] A. Myles, J. Peters, C^2 splines covering polar configurations, Computer-Aided Design 43 (11) (2011) 1322–1329.
- [19] J. Peters, Parametrizing singularly to enclose data points by a smooth parametric surface, in: Proceedings of Graphics Interface '91, GI '91, 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, Appl. Math. Comput. 272 (P1) (2016) 148–158.
- [22] X. Wei, Y. J. Zhang, D. Toshniwal, H. Speleers, X. Li, C. Manni, J. A. Evans, T. J. R. 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 (2018).
- [23] M. Kapl, G. Sangalli, T. Takacs, Dimension and basis construction for analysis-suitable G^1 two-patch parameterizations, Computer Aided Geometric Design 52–53 (01 2017).
- [24] A. Blidia, B. Mourrain, G. Xu, Geometrically smooth spline bases for data fitting and simulation, Computer Aided Geometric Design 78 (2020) 101814.
- [25] M. Marsala, A. Mantzaflaris, B. Mourrain, G^1 -smooth biquintic approxi-

- mation of Catmull-Clark subdivision surfaces, Computer Aided Geometric Design 99 (2022) 102158.
- [26] K. Karčiauskas, J. Peters, Bi-cubic scaffold surfaces, Computer-Aided Design 150 (2022) 103310.
- [27] C. Loop, S. Schaefer, G² Tensor Product Splines over Extraordinary Vertices, Computer Graphics Forum (2008).
- [28] K. Karčiauskas, J. Peters, Minimal bi-6 G² completion of bicubic spline surfaces, Computer Aided Geometric Design 41 (11 2015).
- [29] K. Karčiauskas, J. Peters, Refinable smooth surfaces for locally quad-dominant meshes with T-gons, Comput. Graph. 82 (2019) 193–202.
- [30] K. Karčiauskas, J. Peters, Low degree splines for locally quad-dominant meshes, Comput. Aided Geom. Des. 83 (101934) (2020) 101934.
- [31] T. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-nurccs, ACM Transactions on Graphics (TOG) 22 (2003) 477–484.
- [32] K. Karčiauskas, D. Panozzo, J. Peters, T-junctions in spline surfaces, ACM Trans. Graph. 36 (5) (oct 2017).
- [33] M. Campen, D. Zorin, Similarity maps and field-guided T-splines, ACM Transactions on Graphics (TOG) 36 (2017) 1 – 16.
- [34] G. Farin, Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide, Academic Press, 1988.
- [35] T. D. DeRose, Necessary and sufficient conditions for tangent plane continuity of Bézier surfaces, Comp Aid Geom Design 7 (1) (1990) 165–179.
- [36] K.-P. Beier, Y. Chen, Highlight-line algorithm for realtime surface-quality assessment, Comp-Aid Design 26 (4) (1994) 268–277.
- [37] K. Karčiauskas, A. Myles, J. Peters, A C² polar jet subdivision., in: Symposium on geometry processing, 2006, pp. 173–180.
- [38] K. Karčiauskas, J. Peters, Surfaces with polar structure, Computing 79 (2007) 309–315.
- [39] A. Myles, K. Karčiauskas, J. Peters, Extending Catmull-Clark subdivision and PCCM with polar structures, in: 15th Pacific Conference on Computer Graphics and Applications (PG’07), IEEE, 2007, pp. 313–320.
- [40] 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.
- [41] T. W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-nurccs, ACM transactions on graphics (TOG) 22 (3) (2003) 477–484.

Appendix A: Improving the top of NC₄ⁿ

The presented construction is simple to implement and generally produces surfaces of good quality. However, well-known difficult convex input configurations, see e.g. Fig. 21 a, reveal unexpected oscillations visible as slight pinching or bulging of highlight lines. The arrows →, ← in Fig. 21 c point a pinching of highlight lines near the top. We proceed as in Fig. 21 d: to generate an intermediate surface \mathbf{h} :

1. The top tensor-border is split into three *nonuniform* pieces using the ratio 1 : 2 : 1; then it is degree-raised to 4 in the vertical direction.
2. As before, the coefficients marked □ are defined as affine combinations of inner BB-coefficients with weights (2, -1), and □ := $\frac{1}{2}(\mathbf{d}_{24} + \mathbf{d}_{34})$; □ := $\frac{1}{2}\square + \frac{1}{2}\square$.
3. The curve with BB-coefficients □, □, □ is degree-raised from 2 to 4 to yield the middle top patch \mathbf{b}^{new} with BB-coefficients

$$\mathbf{b}_{2j}^{14} = \mathbf{b}_{0j}^{new} := \frac{2}{3}\mathbf{b}_{1j}^{14} + \frac{1}{3}\mathbf{b}_{1j}^{new}, \quad \mathbf{b}_{0j}^{44} = \mathbf{b}_{2j}^{new} := \frac{2}{3}\mathbf{b}_{1j}^{44} + \frac{1}{3}\mathbf{b}_{1j}^{new}$$

for $j = 0, 1, 2$ enforcing horizontal C^1 -continuity.

4. The gray underlaid BB-coefficients enforce vertical C^1 continuity to the adjacent degree (2,3) patches by splitting and C^1 extending \mathbf{b}^{new} .

Fig. 21 e shows improved highlight lines: the undesired pinching is removed but a smaller bulge (indicated by ←, →) appears prompting us to average new and old formulas, see Fig. 21 f.

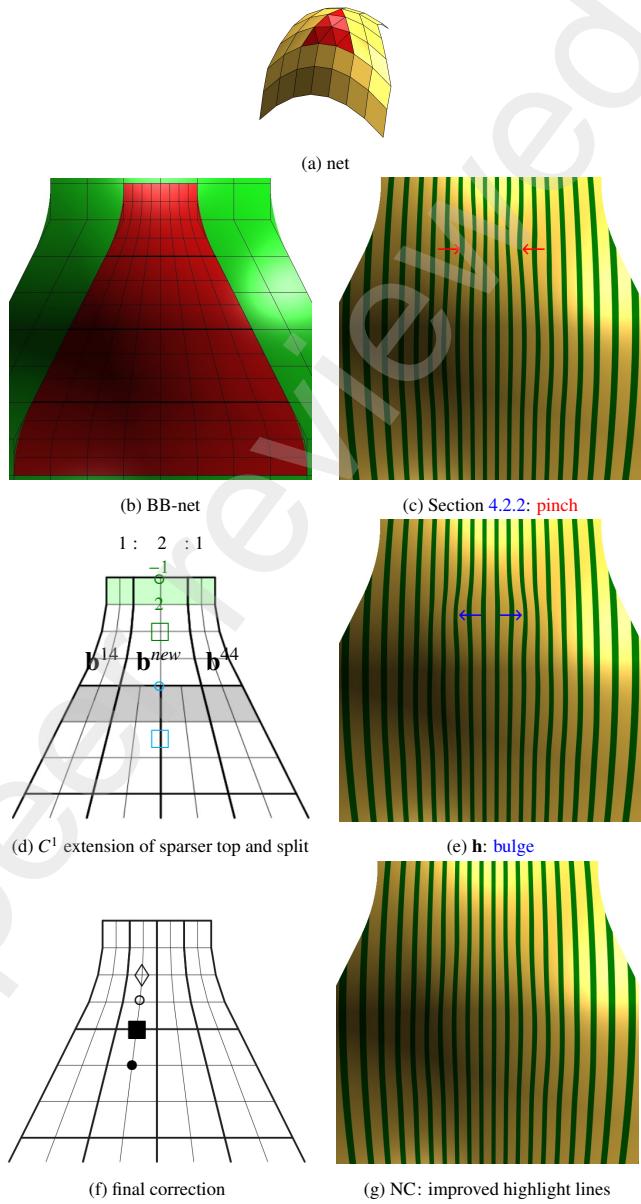


Fig. 21: Improvement of top surface region of NC₄ⁿ

1. Split \mathbf{b}^{new} uniformly into two leading to $\circ = \mathbf{b}_{11}^{24} := \frac{1}{192}(6\mathbf{d}_{14} + 26\mathbf{d}_{24} + 16\mathbf{d}_{34} + 25\mathbf{d}_{15} + 110\mathbf{d}_{25} + 9\mathbf{d}_{35})$, (and \mathbf{b}_{11}^{34} defined symmetrically).
2. Redefine ■ = $\mathbf{b}_{10}^{24} = \mathbf{b}_{13}^{23} := \frac{3}{7}\mathbf{b}_{12}^{23} + \frac{4}{7}\mathbf{b}_{11}^{24}$, ◇ = $\mathbf{b}_{12}^{24} := -\frac{1}{6}(\mathbf{b}_{10}^{24} + \mathbf{b}_{14}^{24}) + \frac{2}{3}(\mathbf{b}_{11}^{24} + \mathbf{b}_{13}^{24})$ and $\mathbf{b}_{10}^{34}, \mathbf{b}_{13}^{33}, \mathbf{b}_{12}^{34}$ analogously.
3. Averaging as in main construction joins C^1 the horizontal layers $\mathbf{b}_{ij}^{14}, \dots, \mathbf{b}_{ij}^{44}, j = 0, 1, 2$ and $\mathbf{b}_{i3}^{13}, \dots, \mathbf{b}_{i3}^{43}$.

Appendix B: A 10-piece NC₃ⁿ construction

The, due to degree unique, splitting of the top tensor-border for the $n \times n$ layout with the interior from Section 4.2 leads to subtle vertical highlight line ondulations indicated by arrows → and ← in Fig. 24 a,c. When the interior stems from Appendix A, the fluctuations diminish, see Fig. 24 b,d. Splitting the *top* bi-2 tensor-border into only two pieces and reparameterizing

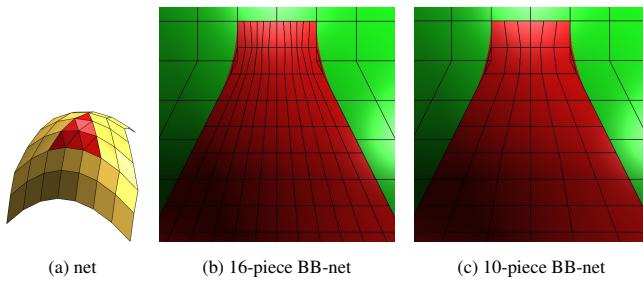


Fig. 22: BB-nets of NC_3^4 : 16-piece vs 10-piece

them by $\tilde{\rho}^s$, $s = 0, 1$ yields a major improvement by reducing opportunities for vertical parameter lines to oscillate (compare the top parts Fig. 22 b vs c and Fig. 23 a vs b) – and yields fewer, namely 10 pieces. By joining pieces, this improvement destroys the convenient tensor-product structure of the patch-work. Fig. 23(b) shows the sequential left to right, bottom to top enumeration.

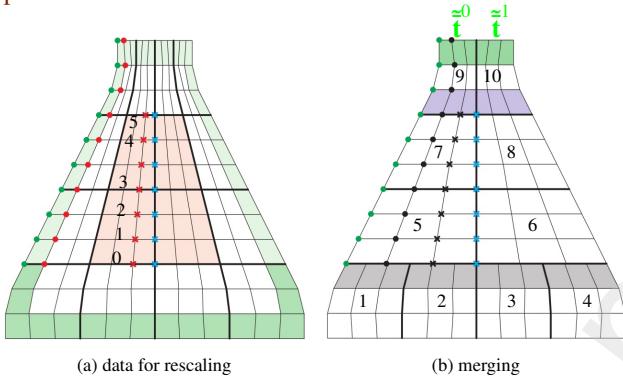


Fig. 23: 10-piece bi-3 construction; only *left* part is labeled, the *right* is defined by symmetry.

- 7 1. The BB-coefficients marked \bullet or \times remain unchanged; set
8 $\bullet := 2\bullet - 1\bullet$ retaining smoothness with input bi-2 tensor-border.

9 2. For $s = 0, \dots, 5$ we set $x_s := \beta_s x_s + (1 - \beta_s) \times_s$, where spacing
10 $[\beta_0, \dots, \beta_5] := [2, 2, 1.95, 1.9, 1, 71, 1.6]$ improves the surface
11 quality. Two lower BB-coefficients \times of patch 7 are set to join
12 C^1 to patch 5.

13 3. The gray underlaid BB-coefficients of patches 1 and 2 are
14 the C^1 extension (and split) of patch 5 and consistent with green
15 tensor-border frame; the light-blue underlaid BB-coefficients of
16 patch 9 are defined by C^1 join to patch 7.

17 4. The $\tilde{\mathbf{t}}^0, \tilde{\mathbf{t}}^1$ are obtained via splitting the top bi-2 tensor-border
18 into two pieces and reparameterizing them by $\tilde{\rho}^s$, $s = 0, 1$; $\tilde{\mathbf{t}}$ is
19 consistent with top two \bullet and \bullet .
20 By construction, the 10 pieces bi-3 surface is internally C^1 and
21 G^1 connected to input bi-2 data.
22 The 10-piece bi-3 surfaces are displayed in bottom row of
23 Fig. 24. The improvement (e) vs (a) and sparser (g) vs (c) is
24 evident and only the pinching, partly hidden by other flaws in
25 (a), (c), remains and is marked \circlearrowleft . While the improvement of (f)
26 over (b) for finely-spaced highlight lines is evident, for sparser
27 lines (h) over (d) is minute, indicating the dominance of the

The 10-piece bi-3 surfaces are displayed in bottom row of Fig. 24. The improvement (e) vs (a) and sparser (g) vs (c) is evident and only the pinching, partly hidden by other flaws in (a), (c), remains and is marked \circlearrowleft . While the improvement of (f) over (b) for finely-spaced highlight lines is evident, for sparser lines (h) over (d) is minute, indicating the dominance of the Appendix A improvement of NC surfaces.

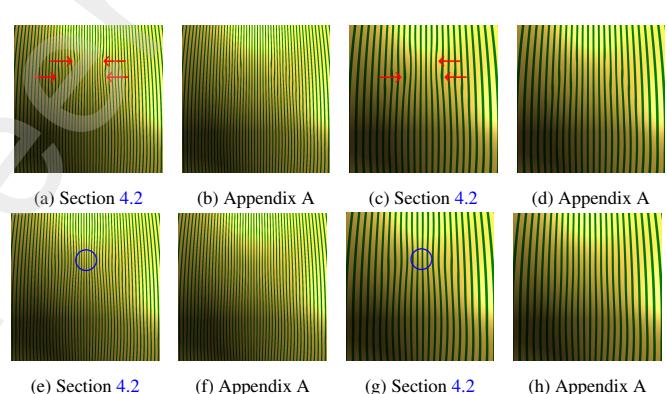


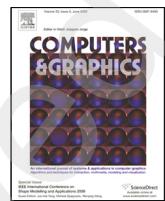
Fig. 24: Effect of interior choice without (Section 4.2) and with the improvement of Appendix A. *Top* row: 16-piece construction. *Bottom* row: 10-piece construction.



Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag



Narrowing-Cascade splines for control nets that shed mesh lines

submission 53, Serhat Cam^a, Erkan Günpinar^a, Kęstutis Karčiauskas^b, Jörg Peters^{c,*}

^a*Istanbul Technical University, Turkiye*

^b*Vilnius University, Lithuania*

^c*University of Florida, USA*

ARTICLE INFO

Article history:

Received July 9, 2025

Control-net narrowing, polyhedral-net spline, geometric continuity

ABSTRACT

To fit narrow surface passages, quad-dominant meshing algorithms use a cascade of $n - 1$ triangles that reduces the number of parallel quad strips from $n + 1$ to 2. A new shape-optimized G^1 spline surface, the Narrowing-Cascade spline NCⁿ, can treat such meshes as spline control nets. For $n > 3$, the narrowing configurations have interior mesh points that both guide and complicate the construction of the output tensor-product NC spline. The spline follows the input mesh while delivering a high-quality curved surface of low degree.

© 2025 Elsevier B.V. All rights reserved.

*Corresponding author:
e-mail: jorg.peters@gmail.com (Jörg Peters)