

Narrowing-Cascade splines for control nets that shed mesh lines

Serhat Çam^a, Erkan Günpinar^a, Kęstutis Karčiauskas^b, Jörg Peters^{c,*}

^a*Istanbul Technical University, Türkiye*

^b*Vilnius University, Lithuania*

^c*University of Florida, USA*

Abstract

Quad-dominant meshes are popular with animation designers and can efficiently be generated from point clouds. To join primary regions, quad-dominant meshes include non-4-valent vertices and non-quad regions. To transition between regions of rich detail and simple shape, quad-dominant meshes commonly use a cascade of $n - 1$ triangles that reduce the number of parallel quad strips from $n + 1$ to 2.

For these cascades, the Narrowing-Cascade spline, short NC_n , provides a new shape-optimized G^1 spline surface. NC_n can treat cascade meshes as B-spline-like control nets. For $n > 3$, as opposed to $n = 2, 3$, cascades have interior points that both guide and complicate the construction of the output tensor-product NC spline. The NC_n spline follows the input mesh, including interior points, and delivers a high-quality curved surface of low degree.

Keywords: Control-net narrowing, polyhedral-net spline, geometric, shed mesh lines

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. 3a, 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. 3b.

Fig. 3d-f shows various Δ^3 -nets that occur in practice and Fig. 3g,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

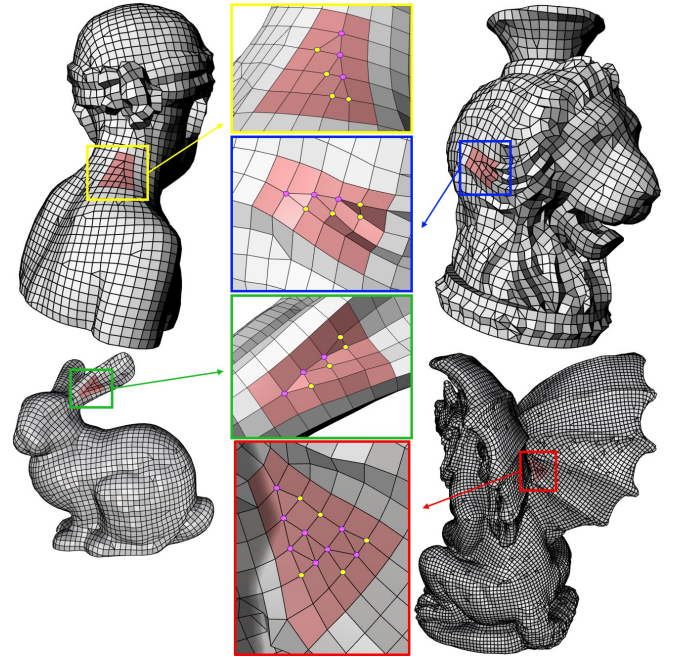


Figure 1: Δ^3 and Δ^4 configurations in quad-dominant meshes. See Fig. 2 for NC_n -surfaces. (Nodes of valence 5 are colored \bullet and nodes of valence 4 \circ .)

any net, including Δ^n , into a 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.

Instead, this paper offers a novel NC_n -spline surface for

*Corresponding author:

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

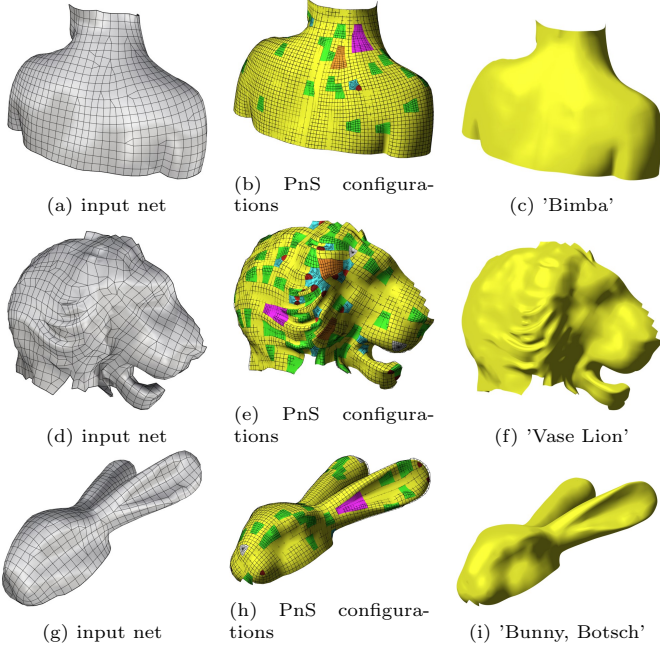


Figure 2: NC_4^n as part of a polyhedral net spline (PnS) surface, colored magenta, NC_4 , and orange NC_3 and green NC_2 for Δ^3 , Δ^2 , Δ^1 . Regular bi-2 splines are displayed in yellow and the BB-net of Bézier coefficients is superimposed. The remaining colors in (b,e,h) indicate polyhedral net splines for 3-gon (triangle with 4-valent vertices), $cop3$, $cop5$ (mesh nodes with 3, resp. 5 neighbors). The input nets are visualized using Rhinoceros3D, and the output surfaces rendered via Bezierview.

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_4^n consists of n^2 pieces: $2n$ pieces of degree (2,4) and $(n-2)n$ pieces of degree (2,3),
 - NC_3^n 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_4^n with pieces of degree 4. Section 5 presents an alternative, everywhere degree bi-3 construction NC_3^n . 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, called reparameterizations. 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.

The survey [34] lists G^k splines as one way to address the challenge of splines for meshes with irregularities. Examples include [35–38]. The survey also points to alternative singular constructions: those with singularities at corners [39–42], singular edges [43, 44] and contracting faces, a.k.a. subdivision algorithms [45]. A third approach, although not polynomial, are generalized barycentric patches such as [46] and non-4-sided transfinite constructions [47–50] that yield multi-sided patches. G^k constructions are challenging in that they have to fulfill $3k$ non-linear polynomial constraints – that can be interpreted as syzygies or Formensysteme [51] – between the derivatives of the polynomial surface patches and derivatives of their reparameterizations. (Reparameterizations locally relate the patch domains to one another.) The polynomial patch degree can be chosen sufficiently high to make these syzygies under-constrained and solvable, for example by imposing a choice of reparameterizations. However high degree is undesirable and obtaining results of minimal degree and choosing the reparameterizations to obtain good shape, as measured by highlight line distribution is a difficult challenge.

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

1 and polar vertices, and configurations that have preferred
2 directions, often induced by an orientation field. All con-
3 figurations must be isolated by a surrounding quad strip.
4 NC_n -splines have two preferred directions, one that we will
5 call *vertical*, for shedding $n - 1$ quad strips, one for each
6 of $n - 1$ *horizontal* strips. This permits tighter packing of
7 mesh line reductions.

8 3.1. Δ^{n-1} -net for NC_n splines

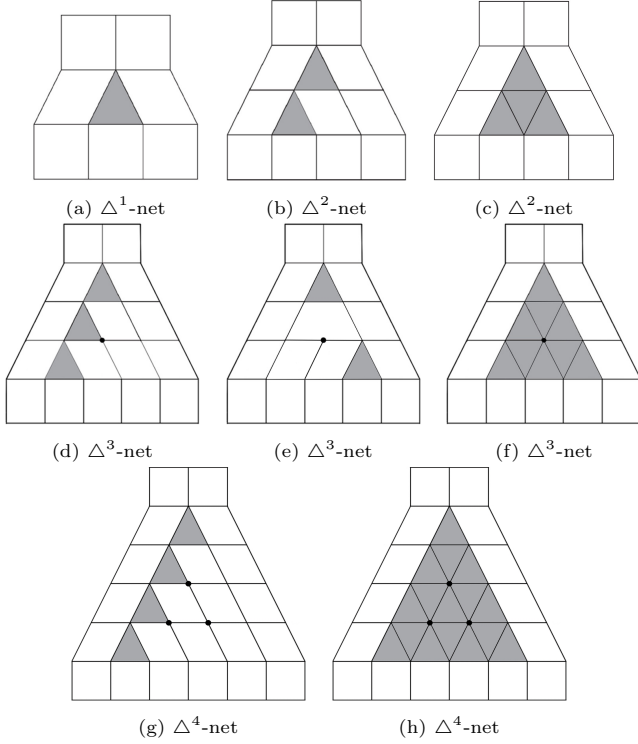


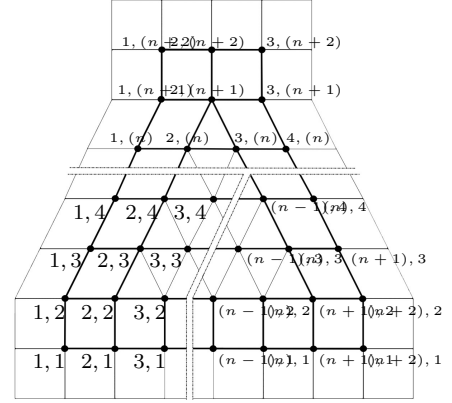
Figure 3: Examples of Δ^{n-1} -nets for NC_n -splines.

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

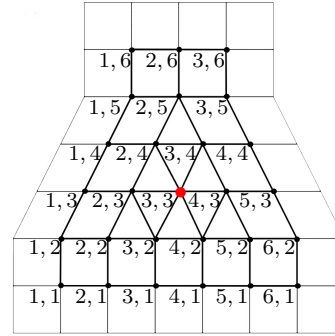
22 3.2. Surface representation

23 The NC -spline consists of tensor-product pieces of poly-
24 nomial bi-degree (d, d') in Bernstein-Bézier form (*BB-*
25 *form*, [52]). For Bernstein polynomials $B_k^d(t) := \binom{d}{k}(1 -$
26 $t)^{d-k}t^k$ a patch \mathbf{p} of bi-degree (d, d') is defined as

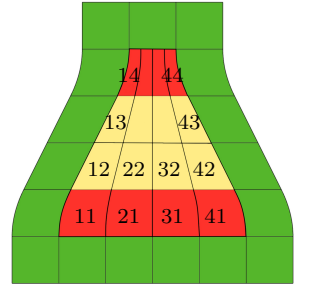
$$\mathbf{p}(u, v) := \sum_{i=0}^d \sum_{j=0}^{d'} \mathbf{p}_{ij} B_i^d(u) B_j^{d'}(v), \quad 0 \leq u, v \leq 1.$$



(a) Extended Δ^{n-1} -net



(b) Extended Δ^3 -net of NC_4^n , $n = 4$



(c) NC_4^4 surface pieces

Figure 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. 4c depicts the 4×4 patches of NC_4 for Δ^3 . The eight
red patches 11, ..., 41 and 14, ..., 44 are of degree $(2, 4)$,
the eight yellow patches 12, ..., 42, 13, ..., 43, are of de-
gree $(2, 3)$, 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).

33 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.

34 3.4. Geometric continuity and reparameterization

Two polynomial pieces \mathbf{p} and \mathbf{q} join G^1 along the com-
mon sector-separating curve $\mathbf{p}(u, 0) = \mathbf{q}(u, 0)$ with BB-

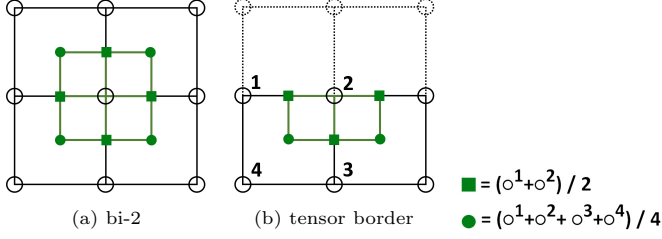


Figure 5: Tensor-borders \mathbf{t} and B-to-BB conversion from Hermite input data. Circles \circ mark control-net of the input mesh, \bullet, \blacksquare mark BB-coefficients of the bi-2 patch (a) and tensor-border (b).

coefficients $\mathbf{p}_{i0} = \mathbf{q}_{i0}$ if, see e.g. [53],

$$\begin{aligned} \mathbf{p}(u, v) &:= \mathbf{q} \circ \rho(u, v), & \rho(u, v) &:= (u + b(u)v, a(u)v) \\ \partial_v \mathbf{q}(u, 0) &= a(u) \partial_v \mathbf{p}(u, 0) + b(u) \partial_u \mathbf{p}(u, 0), & (u, v) &\in [0..1]^2. \end{aligned}$$

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, 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 $\mathbf{t}^s, \mathbf{t}^s, \bar{\mathbf{t}}, \underline{\mathbf{t}}^s$ by partial B-to-BB conversion: for $s = 1, \dots, n$

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

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: $\hat{\mathbf{t}}^s := \mathbf{t}^s \circ \rho^s$ and $\hat{\mathbf{t}}^s := \mathbf{t}^s \circ \rho^s$, 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

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

We choose the minimal horizontal degree 2 for the unreparameterized \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 $\hat{\mathbf{t}}^s, \hat{\mathbf{t}}^s$. 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 $\hat{\cdot}, \bar{\cdot}$, have first cross-derivative BB-coefficients for the corners $s \in \{0, n\}$,

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

$$\begin{aligned} \hat{\mathbf{t}}_{31}^s &\text{ from } \hat{\mathbf{t}}_{11}^s \text{ by exchange } a_i \leftrightarrow a_{i+1}, \mathbf{t}_{ij} \leftrightarrow \mathbf{t}_{i+1,j}, j = 0, 1, \\ \hat{\mathbf{t}}_{41}^s &\text{ from } \hat{\mathbf{t}}_{01}^s \text{ by exchange } a_i \leftrightarrow a_{i+2}, \mathbf{t}_{ij} \leftrightarrow \mathbf{t}_{2-i,j}, j = 0, 1, \end{aligned} \quad (2)$$

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

4.1.1. Example: the frame for NC_4 following Δ^3

Fig. 6 b,c illustrate the reparameterization 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

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

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)

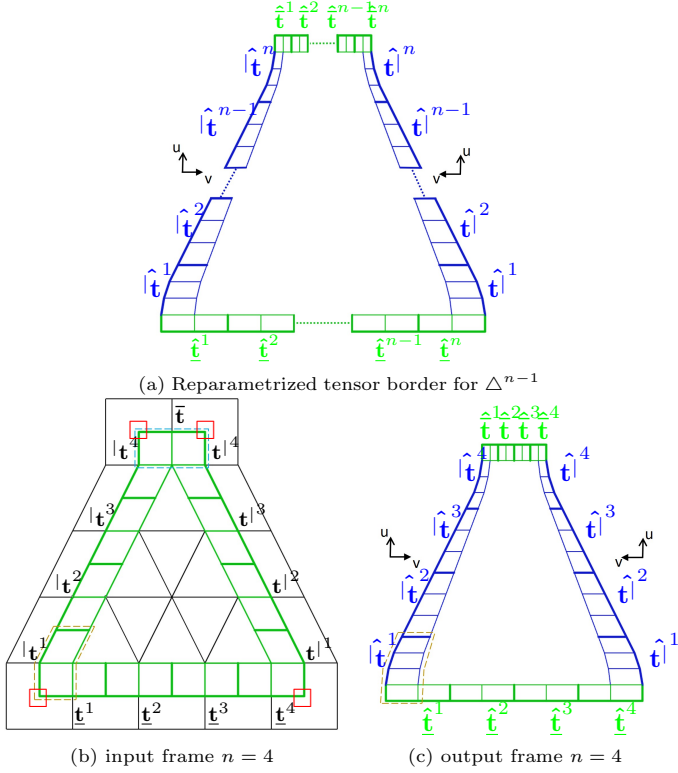


Figure 6: (a) Illustrates the general reparametrized tensor border; (b) Input **bi-2 tensor-border frame** obtained from the Δ^3 -net by B- to BB-conversion; (c) The **tensor-border** reparameterized with $\hat{\rho}^s$. The *bottom* and *top green* tensor-borders are not reparameterized but degree-raised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

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. 4c.
- arrows \uparrow and \downarrow pointing to $n-2$ vertical layers of actual degree 2 that serve as the spines of construction.

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 $[0,1] \cup [1,2] \cup \dots \cup [n-1,n] := [0,n]$.

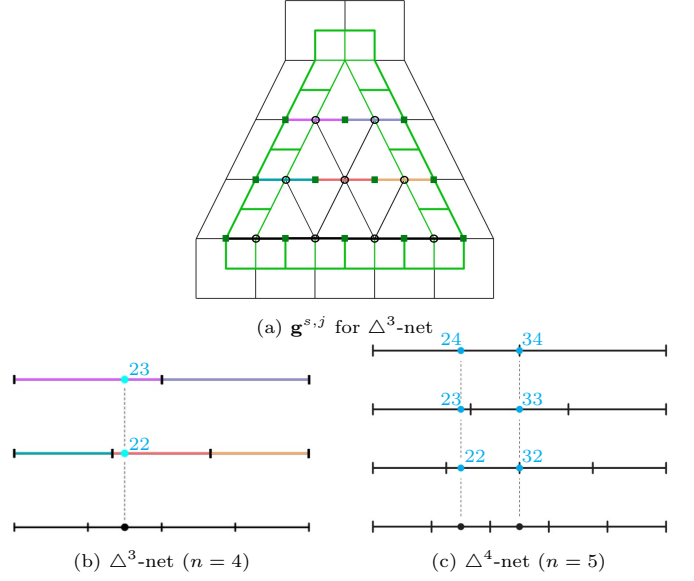


Figure 7: Relation of parameters (s, u) to \mathbf{c}_{ij} . (a) **22**: $s := 2, u := \frac{1}{8}$; **23**: $s := 1, u := \frac{3}{4}$. (b) **22**: $s := 2, u := \frac{1}{5}$; **23**: $s := 1, u := \frac{9}{10}$; **24**: $s := 1, u := \frac{3}{5}$; **32**: $s := 3, u := 0$; **33**: $s := 2, u := \frac{1}{2}$;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

- 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

$$\frac{n}{n+1-j}(s-1+u) = i - \frac{1}{2}, \quad i=2, \dots, n-1, \text{ i.e.}$$

$$r := \frac{n+1-j}{2n}(2i-1) + 1, \quad s := \lfloor r \rfloor, \quad u := r - s. \quad (5)$$

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 $B_k^d(t) := \binom{d}{k}(1-t)^{d-k}t^k$)

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

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

$$\begin{bmatrix} \mathbf{c}_{23} \\ \mathbf{c}_{33} \\ \mathbf{c}_{22} \\ \mathbf{c}_{32} \end{bmatrix} := \frac{1}{128} \begin{bmatrix} 4 & 88 & 36 & 0 & 0 & 0 & 0 \\ 0 & 36 & 88 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 49 & 78 & 1 \\ 0 & 0 & 0 & 0 & 1 & 78 & 49 \end{bmatrix} \begin{bmatrix} \mathbf{d}_{14} \\ \mathbf{d}_{24} \\ \mathbf{d}_{34} \\ \mathbf{d}_{44} \\ \mathbf{d}_{23} \\ \mathbf{d}_{33} \\ \mathbf{d}_{43} \end{bmatrix} \quad (7)$$

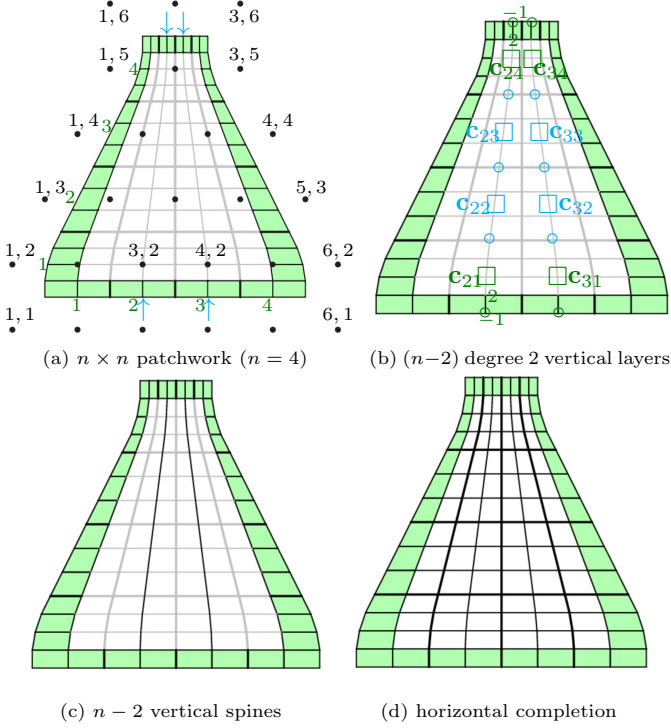


Figure 8: Construction of inner NC_4^n -spline for $n = 4$.

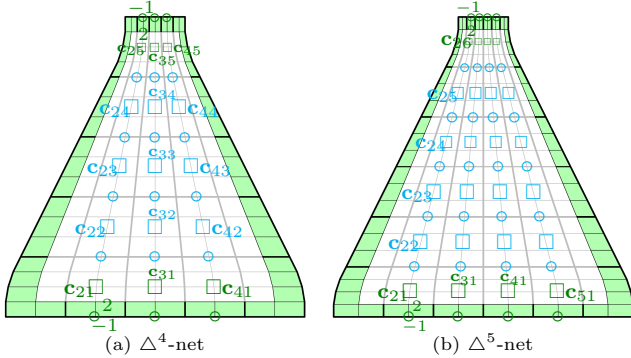


Figure 9: Degree 2 'spine' curves for Δ^4 -net ($n = 5$) and Δ^5 -net ($n = 6$).

$$\begin{bmatrix} c_{24} \\ c_{34} \\ c_{44} \\ c_{23} \\ c_{33} \\ c_{43} \\ c_{22} \\ c_{32} \\ c_{42} \end{bmatrix} := \frac{1}{200} \begin{bmatrix} 16 & 148 & 36 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 36 & 148 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 118 & 81 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 25 & 150 & 25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 81 & 118 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 64 & 132 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 132 & 64 \end{bmatrix}$$

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. 8c. The remaining BB-coefficients enforce horizontal C^1 continuity by averaging the spine coefficients (frame coefficients at the ends), see Fig. 8d.

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 c_{i1} and 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 c_{ij} by (6).
4. Compute the $(n-1)(n-2)$ points \circ as an average of their vertical neighbors c_{ij} , c_{i1} , resp. 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.

Remark. As opposed to other heuristics choices that we evaluated, the chosen c_{ij} does not reproduce the tensor-product case but results in the best highlight line distribution. For a 8×8 net, Fig. 11b 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 reparameterizing 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. 13b. Consequently, the construction consists of n^2

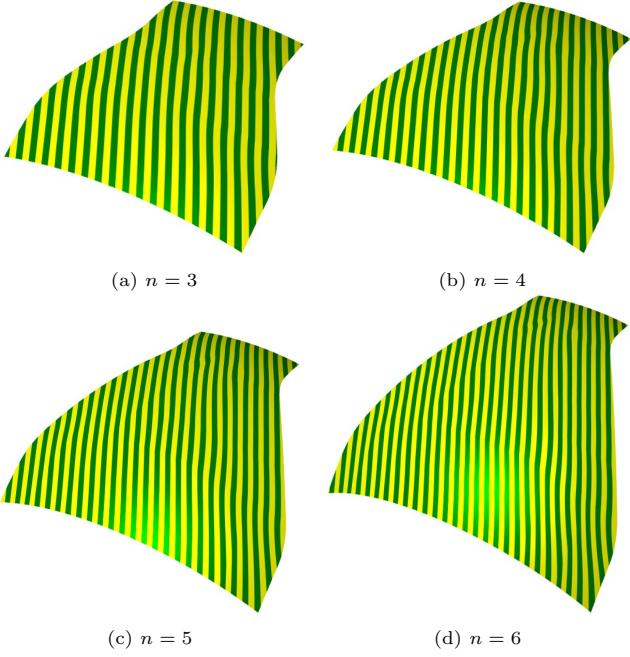


Figure 10: Highlight line distributions of NC_4^n without Appendix A improvement.

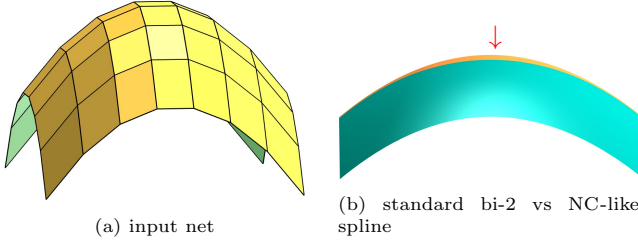


Figure 11: Mimicking, but not reproducing: the regular bi-2 spline (gold in (b), ↓) is almost hidden by the cyan NC modified one. (In (b) the outer bi-2 ring is not shown.)

1 bi-3 pieces for n even, and $n(n+1)$ bi-3 pieces for n odd.
2 Section 5.1 details the derivation of tensor-border frame,
3 while Section 5.2 explains the construction of interior control
4 points.

5.1. Adjusting the tensor-border frame

5 The bi-2 tensor border frame is initialized as described
6 in Section 4.1 and Fig. 6 a. The left, right and bottom of
7 the tensor-border frame each have n pieces, the top only
8 one. To match the bottom, the top is split (horizontally)
9 into n equal pieces if n is even and $n+1$ pieces with spacing
10 $(2, \dots, 2, 1, 1, 2, \dots, 2)/(2n)$ if n is odd. Fig. 12 shows the
11 tensor borders $\tilde{\mathbf{t}}^s$ and $\tilde{\mathbf{t}}^s$ reparameterized by $\rho^s(u, v) :=$
12 $(u, a^s(u)v)$, for $s = 1, \dots, n$. Due to the maximum degree
13 3 of NC_3^n , a^s can be at most linear. The BB-coefficients of
14 the scalar-valued polynomials a^s are
15

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

so that abutting $\tilde{\mathbf{t}}^s := \mathbf{t}^s \circ \tilde{\rho}^s$ join C^1 , as do $\tilde{\mathbf{t}}^s := \mathbf{t}^s \circ \tilde{\rho}^s$.

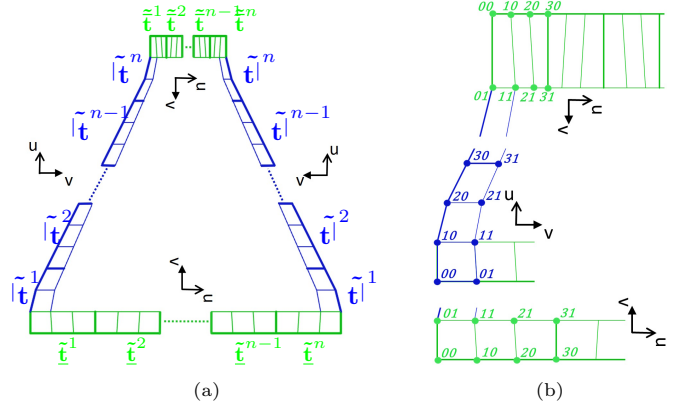


Figure 12: NC_3^n : reparameterized tensor borders.

Here $\tilde{\mathbf{t}}_{i,0}^s$ $i = 0, \dots, 3$ are the degree-raised boundary, and $\tilde{\mathbf{t}}_{01}$ and $\tilde{\mathbf{t}}_{11}$ are defined by Eq.(9) (omitting the superscripts):

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

Then $\tilde{\mathbf{t}}^s$ and $\tilde{\mathbf{t}}$ are degree-raised so that $\tilde{\mathbf{t}}^s$ and $\tilde{\mathbf{t}}$ are degree-3 in the horizontal direction. The reparameterization $\rho^s(u, v) := (u + b^s(u)v, v)$ 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 $\tilde{\mathbf{t}}^s$ and $\tilde{\mathbf{t}}$ 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 reparameterized top and bottom tensor borders, $\tilde{\mathbf{t}}^s := \mathbf{t}^s \circ \tilde{\rho}^s$ and $\tilde{\mathbf{t}}^s := \mathbf{t}^s \circ \tilde{\rho}^s$, maintain C^1 -connectivity; and $\tilde{\mathbf{t}}_{i0}^s$ and $\tilde{\mathbf{t}}_{i0}^s$ 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 reparameterizations for an even and an odd case. For $n = 4$, a Δ^3 -net the reparameterizations are

$$[a_0^s, a_1^s]_{s=1, \dots, 4} := \left[1, \frac{13}{16} \right], \left[\frac{13}{16}, \frac{10}{16} \right], \left[\frac{10}{16}, \frac{7}{16} \right], \left[\frac{7}{16}, \frac{1}{4} \right],$$

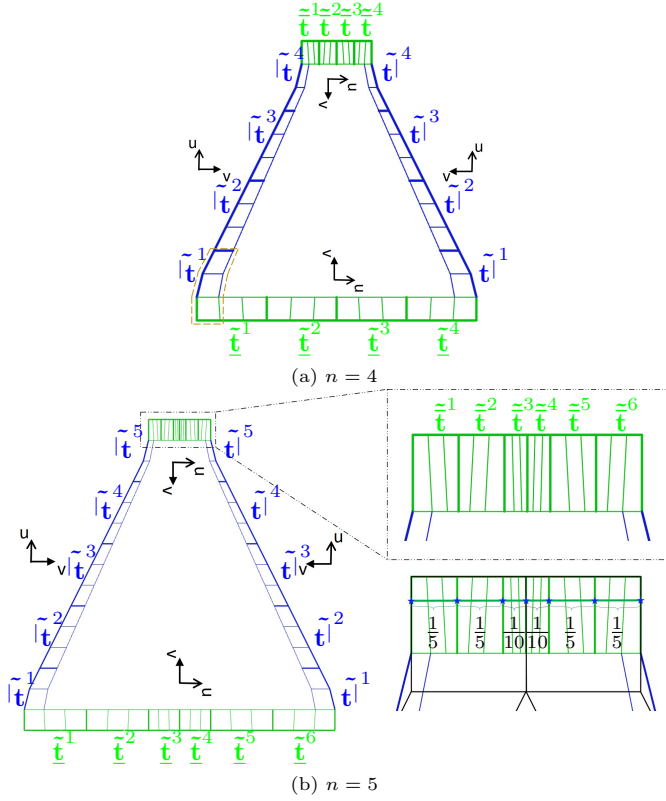


Figure 13: NC_3^4 and NC_3^5 : reparameterized tensor borders.

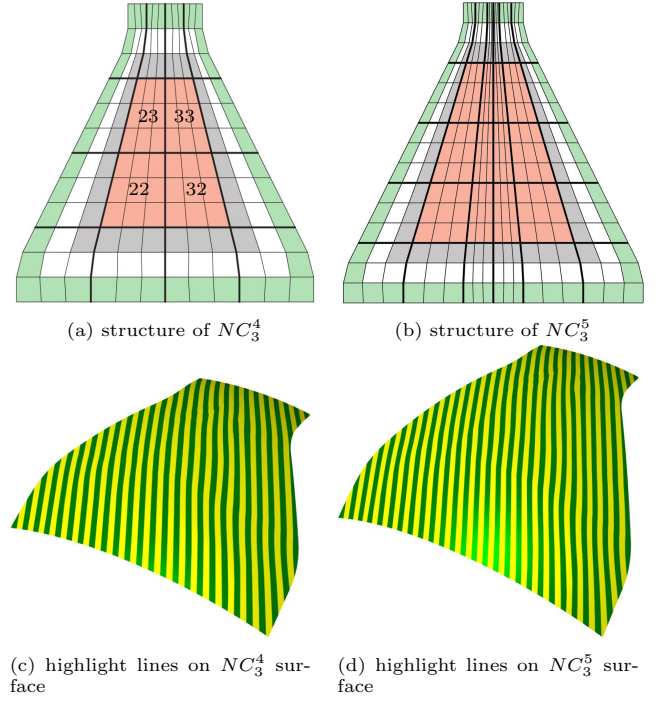


Figure 14: Bi-cubic NC_3^n surfaces consisting of $2\lfloor \frac{n+1}{2} \rfloor \times n$ pieces.

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_3^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.

14

6. Analysis: Tests and comparisons

15

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 [54] is considered high quality.

16

17

18

19

20

21

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 \mathbf{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. 15c and d confirms the preservation of preferred directions by NC_4^n surfaces.

22

23

24

25

26

27

28

29

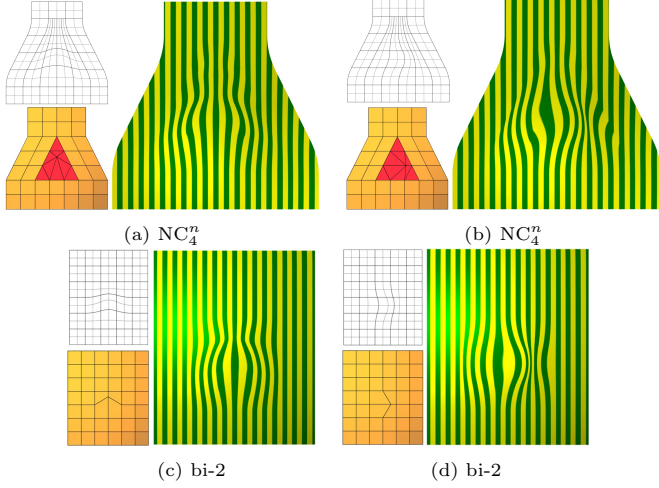


Figure 15: NC_4^n vs bi-2: irregular placement of \mathbf{d}_{33} : (a,b) Δ^3 -net with NC_4^n vs (c,d) regular bi-2 analogue.

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 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 highlight line distribution but with visible C^1 (not C^2) transitions. 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. 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 , colored yellow. (a,b) feature a central ridge of different length and NC_4 retaining the feature lines. (c,d) feature ridges following the sides of Δ^3 with (d) cutting its core. The NC_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 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.

Isotropic n -direction polar splines [18, 44, 55–57] 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.

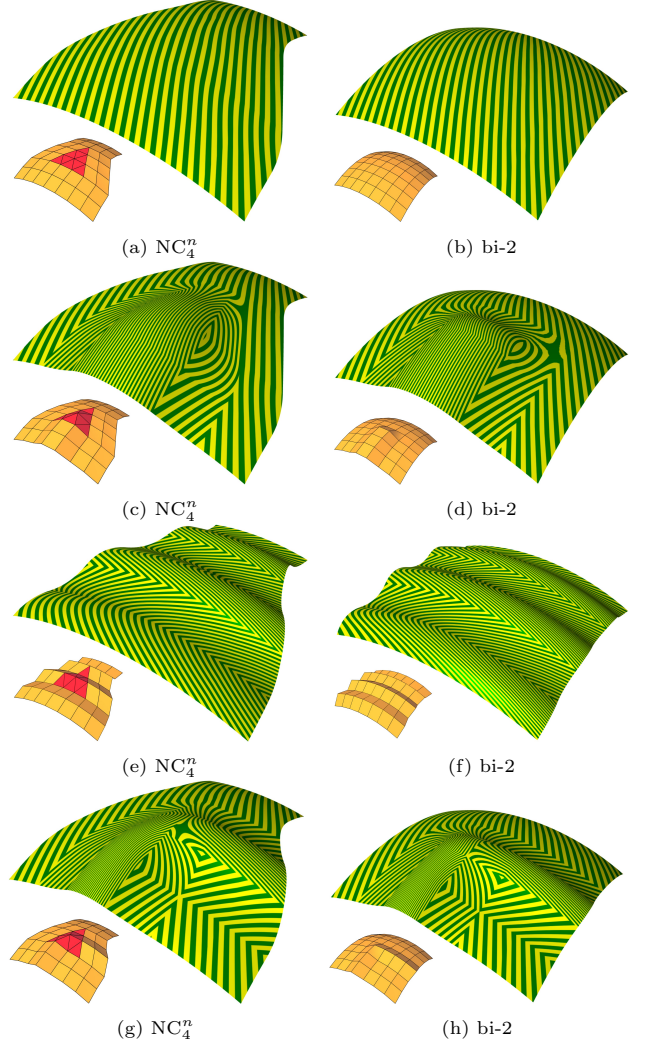


Figure 16: NC_4^n vs bi-2: *left* column: extended Δ^3 -net and NC_4 -spline; *right* column: regular bi-2 counterparts. The highlight line distributions of the NC_4^n surfaces, *left*, are very similar to tensor-product surfaces, *right*.

Locally, Δ^n nets can be re-meshed to admit a T-spline [58] 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

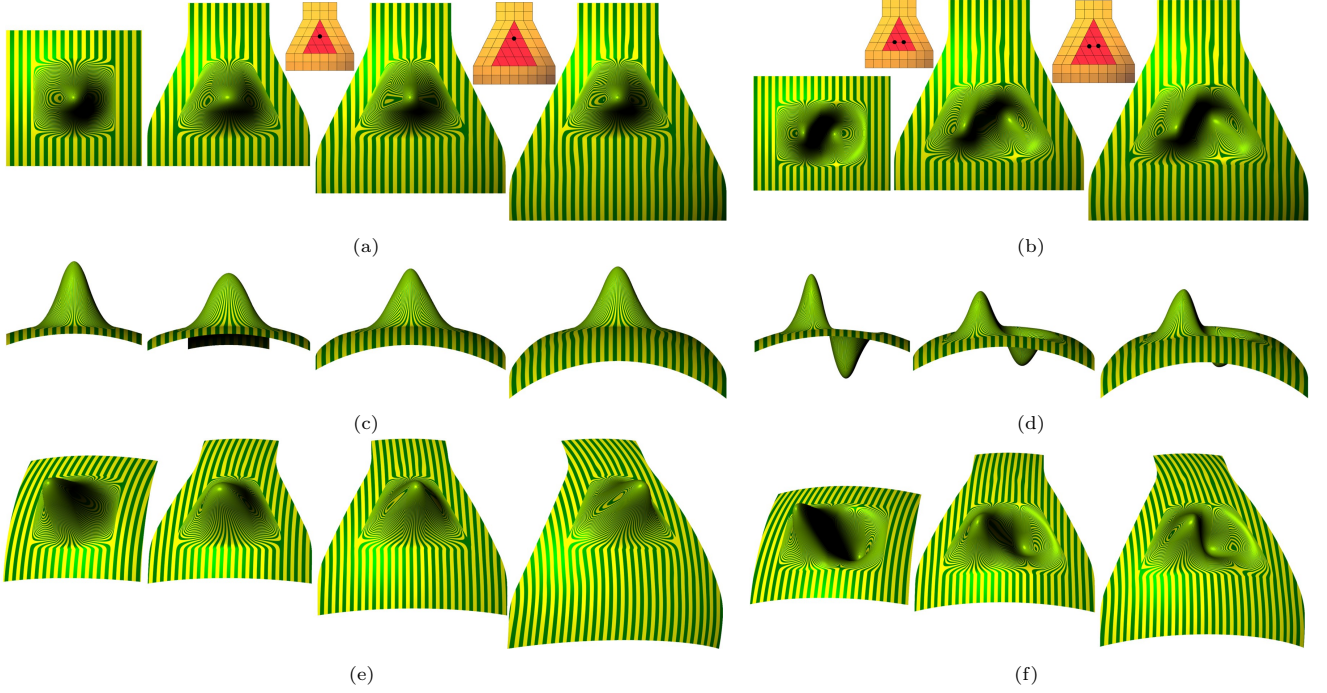


Figure 17: NC_4^n vs bi-2: *top to bottom*, 3 views juxtaposing a bi-2 spline; *left* in each subfigure, with (a,c,e) NC-splines for, *right* Δ^3 -, Δ^4 -, and Δ^5 -nets and (b,d,f) NC_5 - and NC_6 -splines. The lifted Δ^{n-1} -net point (indicated by \bullet) in the left column is \mathbf{d}_{3n} and, in the right column, \mathbf{d}_{33} and \mathbf{d}_{43} for the Δ^4 , and \mathbf{d}_{34} and \mathbf{d}_{44} for Δ^5 .

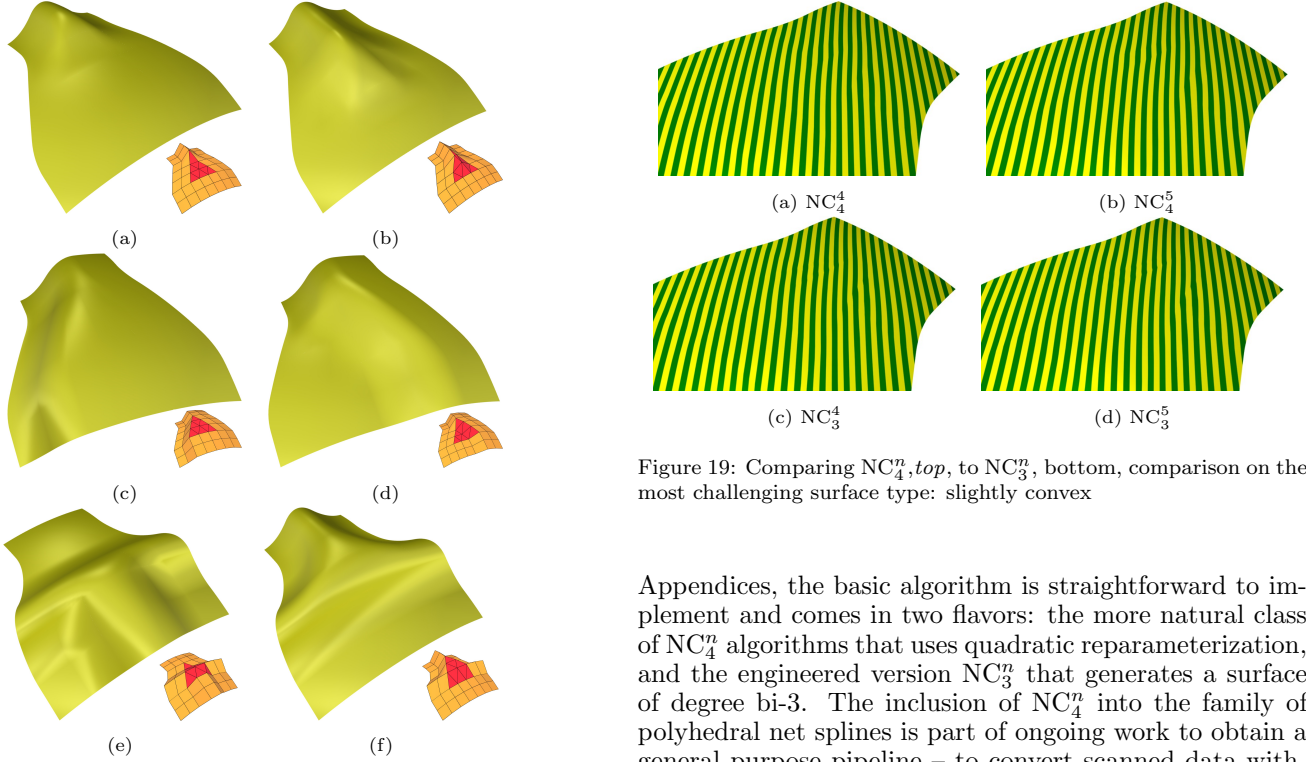


Figure 18: Extended Δ^3 -nets and corresponding NC_4^n surfaces.

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

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]. Erkan Gunpinar acknowledges the funding from The Scientific and Techno-

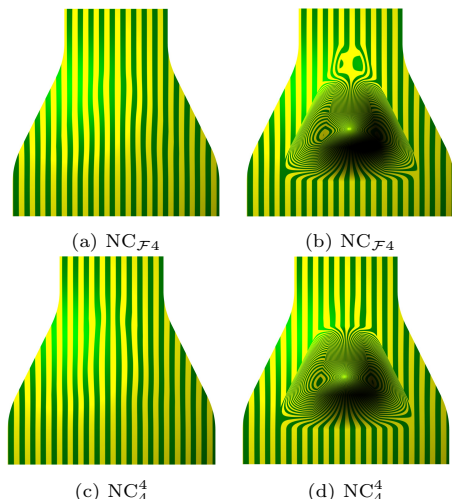


Figure 20: Comparison of (top) constructions $NC_{\mathcal{F}_4}$ based on a functional vs. NC_4^4 .

logical Research Council of Turkey (TUBITAK, Project No: 123F259). Jörg Peters acknowledges the support of the Fields Institute, Toronto, CA.

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. Hetinga, 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 approximation 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, G^2 Tensor Product Splines over Extraordinary Vertices, *Computer Graphics Forum* (2008).
- [28] K. Karčiauskas, J. Peters, Minimal bi-6 G^2 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-nurcs, *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] J. Peters, Splines for meshes with irregularities, *The SMAI journal of computational mathematics* S5 (2019) 161–183.
- [35] J. Peters, C^1 -surface splines, *SIAM Journal on Numerical Analysis* 32 (2) (1995) 645–666.
- [36] 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.
- [37] A. Collin, G. Sangalli, T. Takacs, Analysis-suitable g_1 multi-patch parametrizations for c_1 isogeometric spaces, *Computer Aided Geometric Design* 47 (2016) 93–113.
- [38] A. Blidia, B. Mourrain, G. Xu, Geometrically smooth spline bases for data fitting and simulation, *Computer Aided Geometric Design* 78 (2020) 101814.
- [39] J. Peters, Smooth interpolation of a mesh of curves, *Constructive Approximation* 7 (1991) 221–247.
- [40] U. Reif, Turbs—topologically unrestricted rational b-splines,

- Constructive Approximation 14 (1998) 57–77.
- [41] T. Nguyen, J. Peters, Refinable C^1 spline elements for irregular quad layout, *Computer Aided Geometric Design* 43 (2016) 123–130.
- [42] M. Wu, B. Mourrain, A. Galligo, B. Nkonga, Hermite type spline spaces over rectangular meshes with complex topological structures, *Communications in Computational Physics* 21 (3) (2017) 835–866.
- [43] A. Myles, J. Peters, C^2 splines covering polar configurations, *Computer-Aided Design* 43 (11) (2011).
- [44] 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.
- [45] J. Peters, U. Reif, *Subdivision Surfaces*, Vol. 3 of *Geometry and Computing*, Springer-Verlag, New York, 2008.
- [46] C. T. Loop, T. D. DeRose, A multisided generalization of Bézier surfaces, *ACM Trans. Graph.* 8 (3) (1989) 204–234.
- [47] P. Charrot, J. Gregory, A pentagonal surface patch for computer aided geometric design, *Computer Aided Geometric Design* 1 (1) (1984).
- [48] T. Várady, P. Salvi, G. Karikó, A multi-sided Bézier patch with a simple control structure, *Comput. Graph. Forum* 35 (2) (2016) 307–317.
- [49] G. J. Hettinga, J. Kosinka, A multisided $c2$ b-spline patch over extraordinary vertices in quadrilateral meshes, *Computer-Aided Design* 127 (2020) 102855.
- [50] M. Vaitkus, P. Salvi, T. Várady, Interior control structure for generalized bézier patches over curved domains, *Computers & Graphics* 121 (2024) 103952.
- [51] D. Hilbert, Über die theorie der algebraischen formen, in: *Algebra: Invariantentheorie Geometrie*, Springer, 1970, pp. 199–257.
- [52] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, Academic Press, 1988.
- [53] T. D. DeRose, Necessary and sufficient conditions for tangent plane continuity of Bézier surfaces, *Comp Aid Geom Design* 7 (1) (1990) 165–179.
- [54] K.-P. Beier, Y. Chen, Highlight-line algorithm for realtime surface-quality assessment, *Comp-Aid Design* 26 (4) (1994) 268–277.
- [55] K. Karčiauskas, A. Myles, J. Peters, A C^2 polar jet subdivision., in: *Symposium on geometry processing*, 2006, pp. 173–180.
- [56] K. Karčiauskas, J. Peters, Surfaces with polar structure, *Computing* 79 (2007) 309–315.
- [57] 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.
- [58] T. W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-nurccs, *ACM transactions on graphics (TOG)* 22 (3) (2003) 477–484.

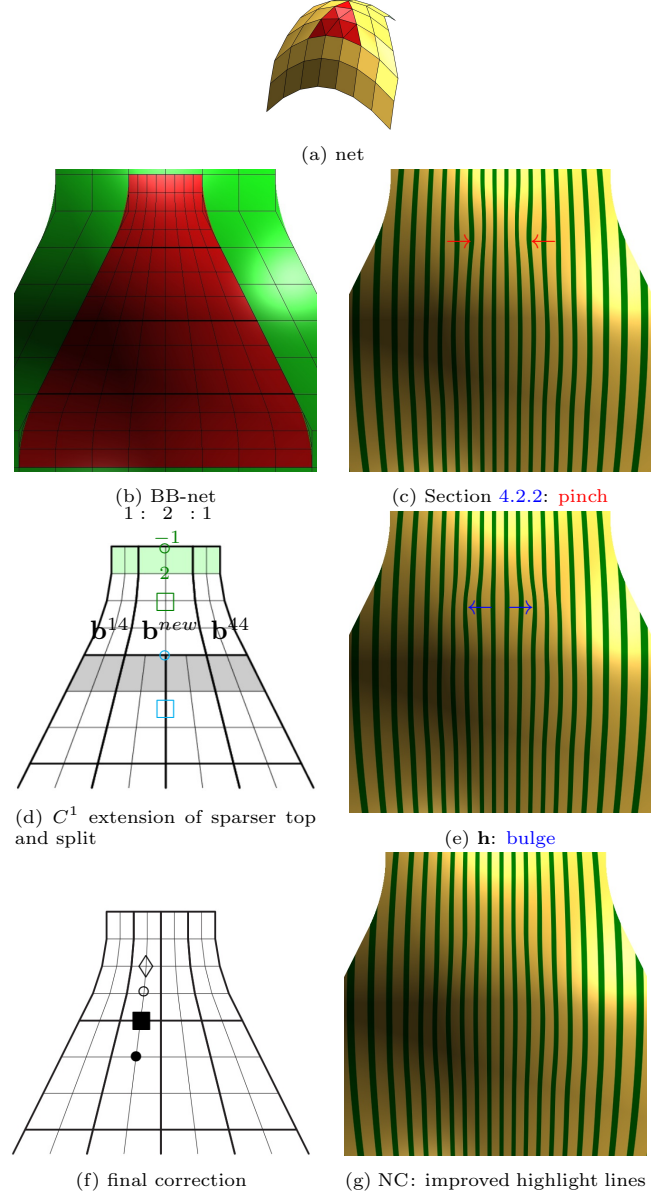


Figure 21: Improvement of top surface region of NC_4^4

3. The curve with BB-coefficients \circ , \square , \diamond 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 \leftarrow , \rightarrow) appears prompting us to average new and old formulas, see Fig. 21 f.

1. Split \mathbf{b}^{new} uniformly into two leading to

$$\circ = \mathbf{b}_{11}^{24} := \frac{1}{192} \left(6\mathbf{d}_{14} + 26\mathbf{d}_{24} + 16\mathbf{d}_{34} + 25\mathbf{d}_{15} + 110\mathbf{d}_{25} + \right.$$

- 1 $9\mathbf{d}_{35})$, (and \mathbf{b}_{11}^{34} defined symmetrically).
- 2 2. Redefine $\blacksquare = \mathbf{b}_{10}^{24} = \mathbf{b}_{13}^{23} := \frac{3}{7}\mathbf{b}_{12}^{23} + \frac{4}{7}\mathbf{b}_{11}^{24}$, $\diamond = \mathbf{b}_{12}^{24} :=$
3 $-\frac{1}{6}(\mathbf{b}_{10}^{24} + \mathbf{b}_{14}^{24}) + \frac{2}{3}(\mathbf{b}_{11}^{24} + \mathbf{b}_{13}^{24})$ and \mathbf{b}_{10}^{34} , \mathbf{b}_{13}^{33} , \mathbf{b}_{12}^{34} analo-
4 gously.
- 5 3. Averaging as in main construction joins C^1 the hori-
6 zontal layers $\mathbf{b}_{ij}^{14}, \dots, \mathbf{b}_{ij}^{44}$, $j = 0, 1, 2$ and $\mathbf{b}_{i3}^{13}, \dots, \mathbf{b}_{i3}^{43}$.

7 Appendix B: A 10-piece NC_3^4 construction

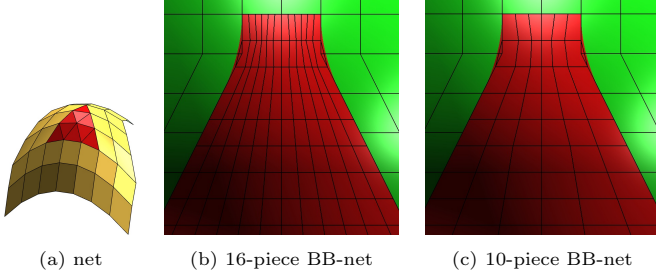


Figure 22: BB-nets of NC_3^4 : 16-piece vs 10-piece

8 The, due to degree unique, splitting of the top tensor-
9 border for the $n \times n$ layout with the interior from Sec-
10 tion 4.2 leads to subtle vertical highlight line ondulations
11 indicated by arrows \rightarrow and \leftarrow in Fig. 24 a,c. When the in-
12 terior stems from Appendix A, the fluctuations diminish,
13 see Fig. 24 b,d. Splitting the *top* bi-2 tensor-border into
14 only two pieces and reparameterizing them by $\tilde{\rho}^s$, $s = 0, 1$
15 yields a major improvement by reducing opportunities for
16 vertical parameter lines to oscillate (compare the top parts
17 Fig. 22 b vs c and Fig. 23 a vs b) – and yields fewer, namely
18 10 pieces. By joining pieces, this improvement destroys
19 the convenient tensor-product structure of the patchwork.
20 Fig. 23 (b) shows the sequential left to right, bottom to
top enumeration.

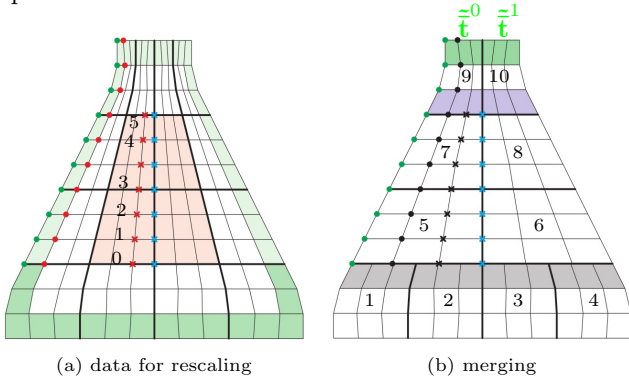


Figure 23: 10-piece bi-3 construction; only *left* part is labeled, the *right* is defined by symmetry.

- 21 1. The BB-coefficients marked \bullet or \times remain unchanged;
22 set $\bullet := 2\bullet - 1\bullet$ retaining smoothness with input bi-2
23 tensor-border.
- 24 2. For $s = 0, \dots, 5$ we set $\times_s := \beta_s \times_s + (1 - \beta_s) \times_s$, where
25 spacing
26 $[\beta_0, \dots, \beta_5] := [2, 2, 1.95, 1.9, 1, 71, 1.6]$ improves the sur-
27 face quality. Two lower BB-coefficients \times of patch 7 are
28

set to join C^1 to patch 5.

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

4. The $\tilde{\mathbf{t}}^0, \tilde{\mathbf{t}}^1$ are obtained via splitting the *top* bi-2 tensor-
border into two pieces and reparameterizing them by $\tilde{\rho}^s$,
 $s = 0, 1$; $\tilde{\mathbf{t}}^0$ is consistent with top two \bullet and \bullet .
By construction, the 10 pieces bi-3 surface is internally C^1
and G^1 connected to input bi-2 data.

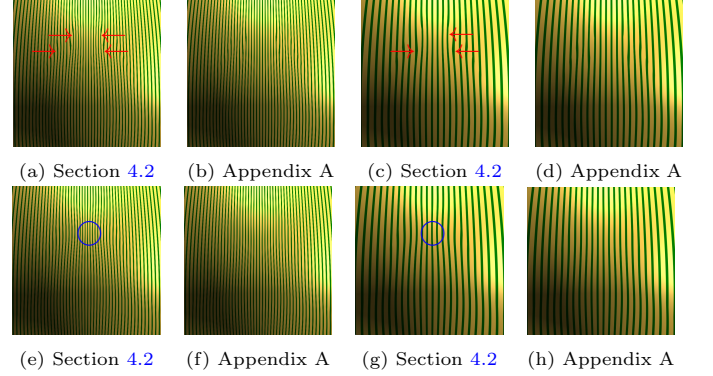


Figure 24: Effect of interior choice without (Section 4.2) and with the improvement of Appendix A. *Top* row: 16-piece construction. *Bottom* row: 10-piece construction.

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 \circ . 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.

The $n - 2$ bottom points \mathbf{c}_{i1} , $i = 2, \dots, n - 1$, (\square in
Fig. 8 and Fig. 9) are $\mathbf{c}_{i1} := \mathbf{d}_{i+1,2}$. The single piece of
the degree bi-2 top input tensor-border (see Fig. 6 b) is
uniformly split into n pieces of degree bi-2 (no degree-
raising in the vertical direction) and then setting $\mathbf{c}_{i,n-2}$ as
for the bottom. The coefficients marked \square are defined by
the tensor-border frame, combining the BB-coefficients (of
the appropriate vertical layer) with the weights 2 and -1
as illustrated in Fig. 8 and Fig. 9. The resulting degree 2
‘spine’ is completed by setting the \circ as the average of two
vertical neighboring \square ; i.e. the n sequences $\{\circ, \square, \circ\}$ are
BB-coefficients of n C^1 -connected Bézier curves. Fig. 9
displays this for \triangle^4 - and \triangle^5 -nets.