Computers & Graphics (2025)

Contents lists available at ScienceDirect

Computers & Graphics

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

A general framework for adding interior control to ribbon-based multi-sided surfaces

Márton Vaitkus*, Péter Salvi, Tamás Várady

Department of Control Engineering and Information Technology, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary

ARTICLE INFO

Article history:
Received July 29, 2025

Keywords: Multi-sided Surface, Curved Domain, Medial Axis Transform, Generalized B-spline, Interior Control

ABSTRACT

Ribbon-based multi-sided parametric patches represent a special group of free-form surfaces. While ribbons with boundary and cross-derivative constraints can sufficiently define patches, modifying their interior – in addition – may be required, as well.

We introduce a new technique to add interior control structures for Generalized B-spline (GBS) patches defined over multi-connected curved domains, where the sum of the blending functions produces a weight deficiency. A local parameterization and a parametric MAT (medial axis transform) structure is computed within the domain, which leads to a family of quad structures, called templates. We define blend functions by distributing weight deficiency amongst the vertices of the template and describe algorithms to locate related control points in 3D, yielding Template GBS patches that tightly approximate the input ribbons.

As a final step, a modified representation is created, called Hybrid GBS, that exactly interpolates the given input ribbons and inherits the interior control structure of an arbitrarily chosen Template GBS patch.

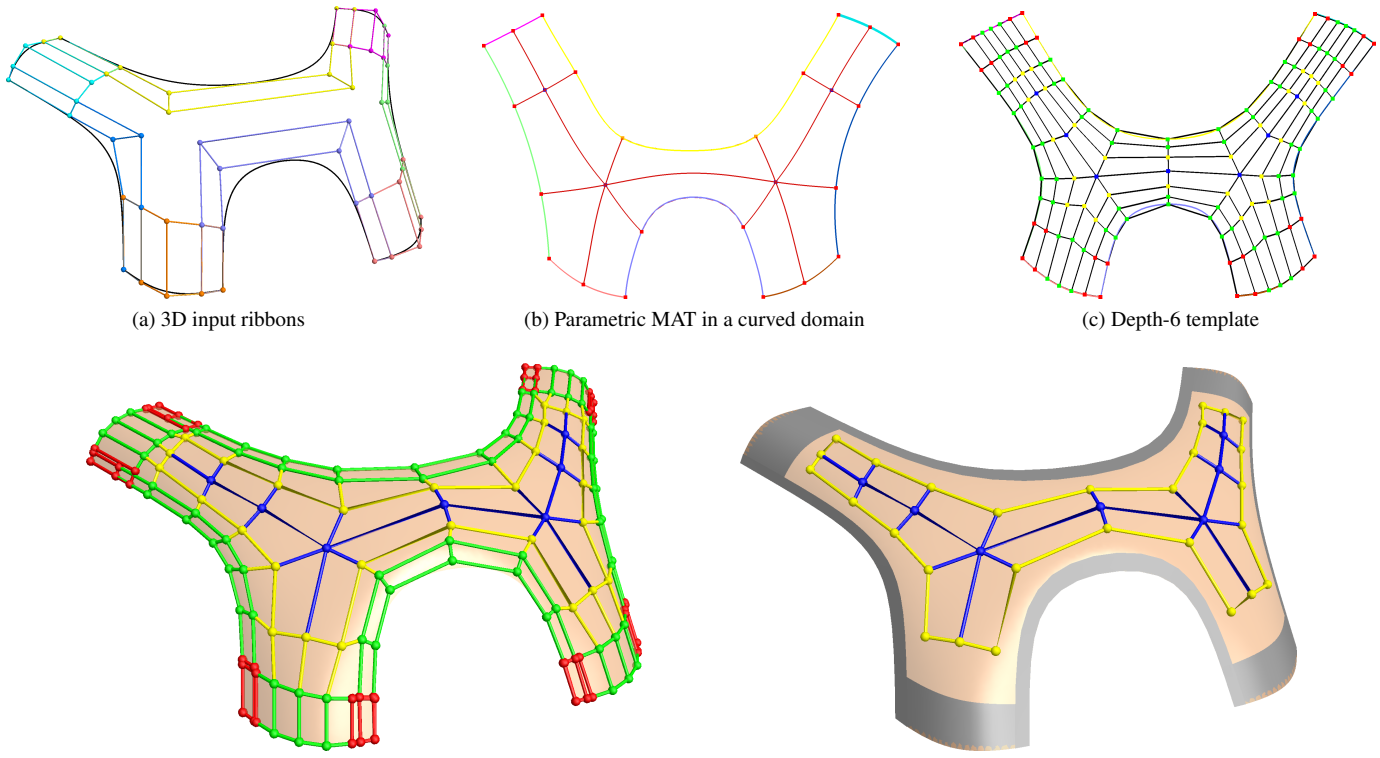
1. Introduction

Quadrilateral surfaces, including tensor product Bézier and B-spline surfaces, and Coons patches, play a dominant role in computer aided geometric design. At the same time, there are modeling tasks, such as hole filling, vertex blending or curve network based design, where constructing and stitching quadrilaterals is cumbersome and the use of n-sided patches ($n \neq 4$) is preferred. The key issue is how to produce smooth shapes that naturally connect given boundaries.

There is a rich variety of multi-sided patches – see the recent survey [1] – where many representations and related topics are discussed. In the current paper, we focus on *ribbon-based patches*, where ribbon surfaces uniquely define the boundary curves and the corresponding cross-derivatives. Similarly to quadrilateral Coons patches and their multi-sided generalizations, such as Gregory [2], Salvi et al. [3], Várady et al. [4], Vaitkus et al. [5], the interior of the patch is automatically determined by the ribbons. This can be an advantage assuming the resulting surface is satisfactory and no further modification is needed, but it can be a disadvantage, when the geometry of the interior needs to be edited in order to obtain better shapes. For example, we may want to add some internal feature or tweak the curvature distribution.

Our primary interest is to produce multi-sided patches over *curved domains*. As discussed in former papers [6, 5], the use of curved domains is indispensable for modeling complex shapes with concave boundary segments and internal hole loops. While for convex polygonal domains a variety of natural control point structures can be constructed (see for example Loop and DeRose [7], Zheng and Ball [8], Krasauskas [9], Goldman [10], Várady et al. [4]), it is a challenging problem to define an intuitive control structure for curved domains.

*Corresponding author:
e-mail: vaitkus@iit.bme.hu (Márton Vaitkus)



(a) 3D input ribbons (b) Parametric MAT in a curved domain (c) Depth-6 template

(d) Template GBS patch with interior control structure (e) Hybrid GBS patch with the original ribbons

Figure 1: Workflow to produce Template and Hybrid GBS patches.

A new kind of interior control structure was introduced in a recent paper by Vaitkus et al. [11], where *Template GB* (Generalized Bézier) patches are created combining Bézier ribbons. In this work, we introduce *Template GBS* patches that build on cubic B-spline ribbons, representing G^1 boundary conditions. The advantages of Generalized B-spline patches (see Vaitkus et al. [5]) carry over to the current setting, such as the ability to interpolate complex piecewise polynomial boundaries, and model multiply connected surfaces bounded by periodic hole loops.

There are similarities between the techniques described in this paper and those of Vaitkus et al. [11], so it is important to identify our new contributions. Both schemes are defined over a locally parameterized curved domain and a MAT (medial axis transform) structure, but the template generation is fundamentally different. Template GB patches use only the topology of the MAT diagram, while for Template GBS we need to extract geometric information from the template, as well. The former template generation method had to be replaced by a new approach, so that it can handle hole loops, as well. The template structure can be edited before constructing the surface, so a *family* of control point structures is defined. Although the concept of distributing weight deficiency is similar to template GB patches, the creation of the blend functions, using the parameterization of the domain is new. For Template GBS patches new methods are introduced to position 3D control points, together with local fairing operations. While Template GB patches exactly reproduce the Bézier boundaries, Template GBS patches can only approximate the B-spline boundaries. For this reason, we introduce a modified representation – called *Hybrid GBS* – that ensures exact interpolation and possesses an editable interior structure 'borrowed' from some Template GBS. The basic workflow is demonstrated in Figure 1.

The paper is structured as follows. After reviewing related work in Section 2, we briefly revisit Generalized B-spline patches and explain our basic concept in Section 3. Section 4 is devoted to the new template algorithm, while Section 5 deals with the blending functions. In Section 6 we describe the initial placement of 3D control points, then in Section 7 we explain the hybrid representation. The 'pros and cons' of the concept are discussed in Section 8, and several test examples are shown.

2. Previous work

In this section we briefly review weight-deficient surface formulations and other relevant works.

2.1. Weight-deficient patches

Interpolant-based multi-sided patches generally take the form

$$\mathbf{S}^*(u,v) = \sum_i \mathbf{I}_i(u,v) \cdot W_i(u,v), \tag{1}$$

where (u,v) are parameters in a domain $\Omega \subset \mathbb{R}^2$, $\mathbf{I}_i : \Omega \to \mathbb{R}^3$ are interpolant surfaces, and $W_i : \Omega \to \mathbb{R}$ are blending functions [1]. The equation for control point based patches is very similar (with $\mathbf{C}_i$ denoting the control points):

$$\mathbf{S}^*(u,v) = \sum_i \mathbf{C}_i \cdot W_i(u,v). \tag{2}$$

In both cases, the blending functions can be *weight-deficient*—they do not constitute a partition of unity ($\sum_i W_i \neq 1$), which would be needed for affine invariance. There are two possible remedies: normalization and adding extra terms, as shown below.

$$\mathbf{S}(u,v) = \frac{1}{\sum_i W_i(u,v)}\mathbf{S}^*(u,v), \qquad \text{[normalized]} \tag{3}$$

$$\mathbf{S}(u,v) = \mathbf{S}^*(u,v) + \left(1 - \sum_i W_i(u,v)\right)\mathbf{A}(u,v), \qquad \text{[augmented]} \tag{4}$$

where $\mathbf{A}(u,v)$ is an auxiliary surface. The latter solution makes it possible to augment the surface with interior control, and this is the path we have chosen in this paper.

There are several constructions of this type in the literature. Várady et al. [12] proposed a variant of Kato's interpolant-based patch [13] to add auxiliary surfaces in the interior. Martin and Reif [14] employed a similar construction to create exactly interpolating trimmed patches, using a control point based (NURBS) patch as the auxiliary. The Midpoint [15] and Generalized Bézier (GB) patches [4] are examples where the auxiliary is a single point; the main surface being interpolant- and control point based, respectively. In the case of Template GB patches [11], both the exterior and interior surfaces are defined by control points.

A somewhat different concept is the Hybrid patch suggested by Salvi [16], where $\mathbf{A}$ is defined in such a way that its blend sum matches the weight deficiency of $\mathbf{S}^*$:

$$\mathbf{S}_{\text{Hybrid}}(u,v) = \mathbf{S}^*(u,v) + \mathbf{A}(u,v). \tag{5}$$

This can also be regarded as the distribution of weight deficiency, a technique also found in the Zheng–Ball patch [8] and variations of the GB patch [17].

2.2. Other relevant works

There exist various constructions that generalize B-spline surfaces beyond regular tensor grids, but these have been generally limited to convex, polygonal domains or other special configurations. Polyhedral splines, i.e., smoothly connected piecewise-polynomials over polygonal meshes in particular have been extensively studied [18].

Recursive subdivision methods are defined for general topology control polyhedra, and have been adapted to interpolate ribbon boundary conditions [19, 20], and to approximate trimmed regions over curved domains [21]. Usually, both polyhedral splines and subdivision are derived from uniform splines, and while non-uniform extensions have been proposed [22, 23], these often impose topological constraints on the control structure.

Methods such as hierarchical splines [24] and T-splines [25] can introduce local details into regular control structures, but need to handle irregular configurations via other means, and the same holds for the non-uniform spline interpolation of Antonelli et al. [26].

Some constructions can handle arbitrary polygonal meshes, e.g. by mapping a tensor-product basis function to each mesh vertex [25, 27, 28], by solving an optimization problem [29], or by generalizing the divided difference definition of B-splines [30], but for these approaches – as well as the well-known 'meshless' methods [31] – the handling of general boundary conditions remains challenging.

The Generalized Bézier patch of Várady et al. [4] has been successfully adapted to G^2 B-spline boundary conditions by Hettinga and Kosinka [32, 33], but this approach is also restricted to uniform knots and convex polygonal domains. Our approach is an extension of the Generalized B-spline (GBS) surfaces of Vaitkus et al. [5] which supports multiply-connected curved domains and non-uniform spline ribbons. In this paper we add interior control, as summarized in the next section.

3. Basic concept

Our goal is to create surfaces that behave like GBS patches near their boundaries and have intuitive interior control. First, we will briefly review the generalized B-spline (GBS) patch (Section 3.1), then show a high-level overview on how to create approximative Template GBS patches (Section 3.2). The additional steps of creating the interpolating Hybrid GBS patch are shown in Section 3.3.

3.1. The original GBS patch

The GBS patch [5] is an n-sided surface that interpolates positional and cross-derivative boundary constraints. The surface is defined over a *curved domain* – a collection of 2D curves of the same topology as the boundary curves. (For the generation of curved domains see Appendix A.) The constraints are given in the form of *ribbons* $\mathbf{R}_i$, which are biparametric surfaces represented by a grid of control points:

$$\mathbf{R}_i(s_i,h_i) = \sum_{j=0}^{d_i}\sum_{k=0}^{e_i}\mathbf{C}_{ijk}\cdot N_j^{\Xi_i}(s_i)B_k^{e_i}(h_i), \tag{6}$$

where $\mathbf{C}_{ijk}$ is a *ribbon control point* in the kth row and jth column of ribbon i, $N_j^{\Xi_i}(s_i)$ is the jth B-spline basis function with knot vector Ξ_i, $B_k^{e_i}(h_i)$ is the kth, degree e_i Bernstein polynomial, and $(s_i,h_i) \in [0,1]^2$ is a per-side parameterization (see Fig. 2 and Appendix B). The boundary constraints are expressed as follows: the positions and cross-derivatives of the GBS patch $\mathbf{S}_{\text{GBS}}(u,v)$ at the ith boundary segment should match those of $\mathbf{R}_i(s_i,h_i)$. In this paper we deal with cubic B-splines and linear cross-derivatives ($e_i = 1$), i.e., G^1 continuity.

The patch is constructed via the same control points as the ribbons, but with different blends ($\gamma_{ijk}(u,v)$, see Appendix C):

$$\mathbf{S}^*_{\text{GBS}}(u,v) = \sum_{i=1}^{n+N_L}\sum_{j=0}^{d_i}\sum_{k=0}^{d^\perp}\mathbf{C}_{ijk}\cdot\gamma_{ijk}(u,v). \tag{7}$$

Here $d^\perp$ is the polynomial cross-degree, and N_L is the number of interior loops: boundary curves $i = 1\ldots n$ are clamped B-splines, constituting the perimeter loop, while boundaries $i =$

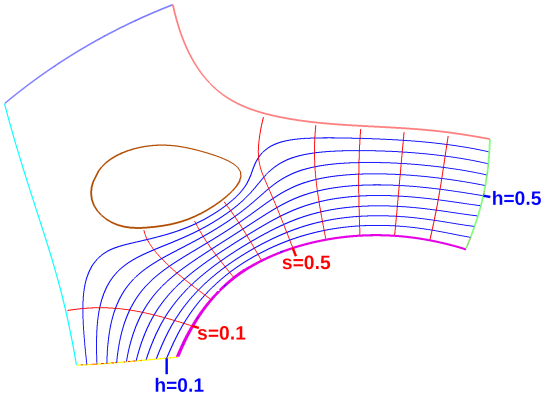


Figure 2: Constant side and distance parameter lines (s, h) for the purple side.

$n+1 \ldots n+N_L$ are periodic and define interior holes. The blend sum of this surface is thus

$$\Gamma(u,v) = \sum_{i=1}^{n+N_L} \sum_{j=0}^{d_i} \sum_{k=0}^{d^\perp} \gamma_{ijk}(u,v). \tag{8}$$

Since generally $\Gamma(u,v) \neq 1$, the patch needs to be normalized (cf. Eq. 3):

$$\mathbf{S}_{\mathrm{GBS}}(u,v) = \frac{1}{\Gamma(u,v)} \mathbf{S}^*_{\mathrm{GBS}}(u,v). \tag{9}$$

In the rest of the paper, instead of normalization, we are going to use the augmented construction of Eq. (4).

3.2. The Template GBS patch

For Template GBS patches we add *interior control points* $\mathbf{C}^{\mathrm{Int}}_\ell$ and associated blending functions $\gamma^{\mathrm{Int}}_\ell$ to a 'GBS-like' exterior defined by *exterior control points* $\hat{\mathbf{C}}^{\mathrm{Ext}}_{ijk}$ and associated blending functions $\hat{\gamma}^{\mathrm{Ext}}_{ijk}$, organized in the same side–column–row structure as the ribbon control points $\mathbf{C}_{ijk}$. (Such a control point arrangement will be shown in Figure 8b.)

$$\mathbf{S}_{\mathrm{TGBS}}(u,v) = \sum_{i=1}^{n+N_L} \sum_{j=0}^{d_i} \sum_{k=0}^{d^\perp} \hat{\mathbf{C}}^{\mathrm{Ext}}_{ijk} \cdot \hat{\gamma}^{\mathrm{Ext}}_{ijk}(u,v) + \\ + \frac{1 - \hat{\Gamma}^{\mathrm{Ext}}(u,v)}{\Gamma^{\mathrm{Int}}(u,v)} \sum_\ell \mathbf{C}^{\mathrm{Int}}_\ell \cdot \gamma^{\mathrm{Int}}_\ell(u,v). \tag{10}$$

Here $\hat{\Gamma}^{\mathrm{Ext}}$ and Γ^{Int} are the sums of the blends $\hat{\gamma}^{\mathrm{Ext}}_{ijk}$ and $\gamma^{\mathrm{Int}}_\ell$, respectively.

The basic steps for creating this patch are summarized as follows (see details in later sections):

1. The input is given as control point based *ribbons*, defining positional and cross-derivative constraints along the boundary. Based on this information, a *curved domain* is generated, with *local parameterizations* for all sides.
2. The parameterizations induce a medial axis structure, which, in turn, produces a quadrangulation of the domain (the *MAT template*).
3. The template is simplified and/or refined (as needed); the *interior* vertices of this structure (those not directly connected to the edge of the template) will correspond to control points $\mathbf{C}^{\mathrm{Int}}_\ell$, while the rest will be related to the exterior control points $\hat{\mathbf{C}}^{\mathrm{Ext}}_{ijk}$.
4. Blending functions $\hat{\gamma}^{\mathrm{Ext}}_{ijk}$ and $\gamma^{\mathrm{Int}}_\ell$ are assigned to the exterior and interior control points based on natural per-side blending functions.
5. The ribbons are approximated with knot vectors generated from the template, defining the exterior control point positions $\hat{\mathbf{C}}^{\mathrm{Ext}}_{ijk}$.
6. Interior control points $\mathbf{C}^{\mathrm{Int}}_\ell$ are placed in 3D; the Template GBS patch can now be evaluated.

3.3. The Hybrid GBS patch

Hybrid GBS patches are defined like Template GBS patches (Eq. 10), but with different exterior control points and associated blend functions, denoted without a hat ($\mathbf{C}^{\mathrm{Ext}}_{ijk}$ and $\gamma^{\mathrm{Ext}}_{ijk}$, respectively). These are determined by two additional steps:

7. Exterior control points $\mathbf{C}^{\mathrm{Ext}}_{ijk}$ of the Hybrid GBS patch are defined by scaling the input ribbons.
8. GBS blends with template-based cross-degree, $\gamma^{\mathrm{Ext}}_{ijk}$, are assigned to the exterior control points; the Hybrid GBS patch is now complete.

In Section 7.2 we will also show how to combine the internal control structure with interpolant-based surfacing schemes (instead of the GBS patch).

4. Template generation

In this section we describe the medial axis transform (MAT) diagram and the related MAT-quad structure, then continue with the template generation algorithm and demonstrate the variety of template structures.

4.1. MAT-quads

The 2D MAT is a well-known structure [34] – it is a graph within a domain whose *edges* represent the locus of points that are equal distance from two boundaries, while its *vertices* represent points with equal distances from three or more boundaries. These constitute the MAT *skeleton*. In the current context we deal with a curved domain, and (following [11]) the distance measure is not Euclidean, but *parametric*, i.e., it is computed by the per-side distance parameters h_i. The edges and vertices of the parametric MAT determine where high weight deficiency can be expected, and partition the domain according to the region of influence of each side. Similarly to the Euclidean setting [35], the parametric MAT diagram also suggests a natural quadrilateral decomposition called the *MAT-quad* structure: we drop s_i constant isolines from the given MAT vertices and connect edges to *footpoints* on the related boundaries. Due to the numerical instabilities of the MAT calculation, the default structure is automatically pruned, by snapping edges with lengths below a certain threshold.

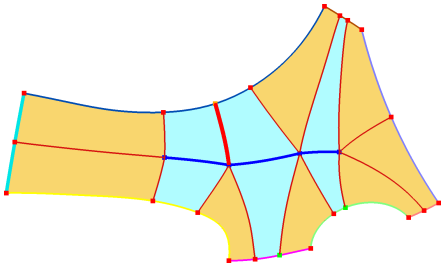


Figure 3: A MAT-quad diagram.

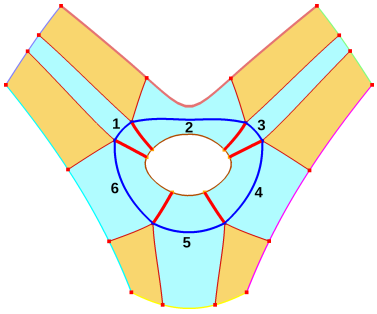


Figure 4: A MAT-quad diagram with a closed loop.

Examples of MAT-quad structures are shown in Figures 3 and 4. The edges of the MAT skeleton[1] are colored blue, the connecting edges are colored red. There are *corner-quads* (yellow) and *side-quads* (blue). The corner quads are bounded by interior *corner-edges* (thin red) and two adjacent boundary segments, meeting at a common corner. The side-quads lie between a single boundary segment and a MAT-edge. There are special *side-edges* (thick red) that occur between two adjacent side-quads. These will play an important role, in particular at closed loops, as they are always built into the template structure. Observe a single side-edge in Figure 3 and six side-edges in Figure 4.

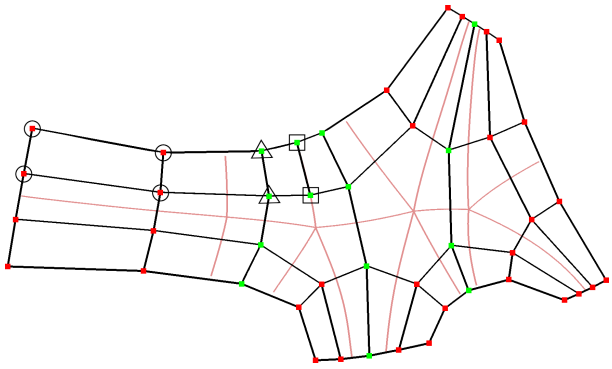
4.2. Templates

Templates are always derived by refining the initial MAT-quad structure. A basic parameter of each template is its *depth*, denoted by D, that determines the resolution of the subdivided edges and faces. (It corresponds to the polynomial degree of the transversal Bernstein polynomials $d^{\perp}$.) Depth tells us how many *vertex layers* occur as we move from the boundaries to the MAT skeleton by the formula

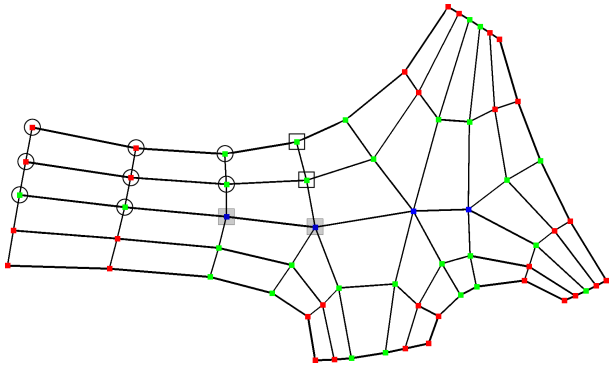
$$L = D \div 2 + 1. \tag{11}$$

Another set of parameters are internal *MAT division numbers*, which are non-negative integers. By means of these each MAT-edge can be further subdivided, as it will be discussed in the next subsection.

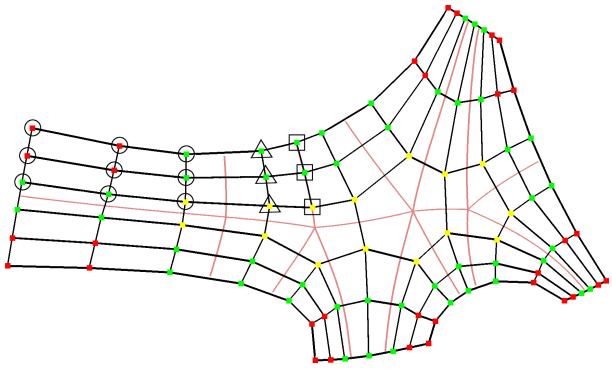
Template generation is explained by a sequence of pictures in Figure 5, where – for simplicity's sake – there are no internal divisions on the edges of the MAT skeleton. First we



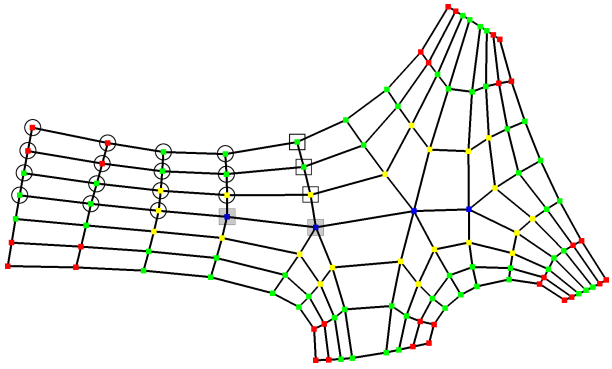
(a) Depth 3



(b) Depth 4



(c) Depth 5



(d) Depth 6

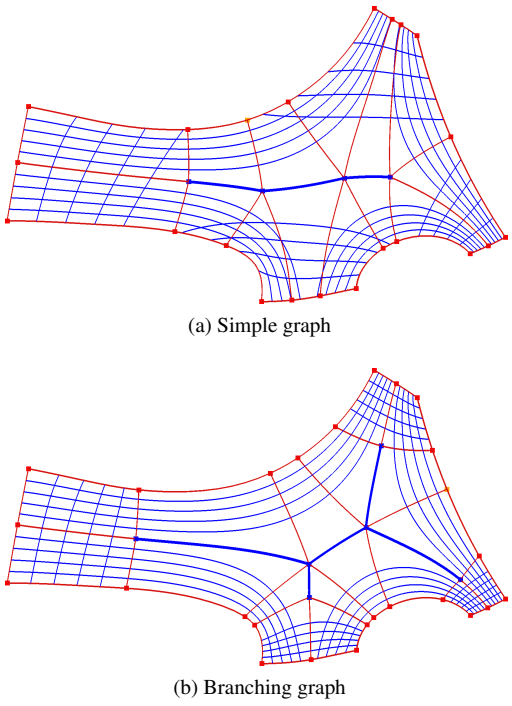
Figure 5: Family of template structures based on the MAT in Fig. 3.

[1] Note that we use the term *edge* in a topological sense, even when referring to curves or polylines.

consider *even* D-s, where the boundary segments and the connecting edges of the corner quads, and the side-edges are subdivided into L vertices. For $D = 2 \rightarrow L = 2$ and the vertices are at $\{0, 1\}$, i.e., there are no inner divisions (Fig. 3). For $D = 4 \rightarrow L = 3$ and the vertices are at $\{0, \frac{1}{2}, 1\}$. For $D = 6 \rightarrow L = 4$ at $\{0, \frac{1}{3}, \frac{2}{3}, 1\}$, and so on. The corner-quads will be subdivided into $L \times L$ vertices, i.e., for $D = 2 \rightarrow 2 \times 2$, for $D = 4 \rightarrow 3 \times 3$, $D = 6 \rightarrow 4 \times 4$ (marked with small circles). Vertices on the side-edges are marked by small squares.

For *odd* D-s, vertices are placed in the $[0, 1)$ intervals of the related edges of the corner-quads. For $D = 3$ the number of vertices equals to $L = 2$, placed at $\{0, \frac{2}{3}\}$. For $D = 5 \rightarrow L = 3$ vertices are at $\{0, \frac{2}{5}, \frac{4}{5}\}$, for $D = 7 \rightarrow L = 4$ at $\{0, \frac{2}{7}, \frac{4}{7}, \frac{6}{7}\}$ and so on. This division is performed on the side-edges, as well. Within the corner-quads $L \times L$ vertices will be computed. For templates with odd depth we have found it useful to insert further edges between the corner quads and the MAT side-edges, see vertices marked with a small triangle.

Edges are subdivided uniformly by arc length, and internal face vertices are computed using bilinear Coons interpolation of the subdivided edges. Connecting the adjacent vertices is straightforward, yielding the final template structure of small quadrilateral faces. This process can easily be generalized for higher depths, as well. Note that for odd D-s, the MAT skeleton is not part of the template structure, but n-sided polygons are formed instead. In our example five- and six-sided polygons were obtained in the interior (see Figures 5a and 5c).



(a) Simple graph

(b) Branching graph

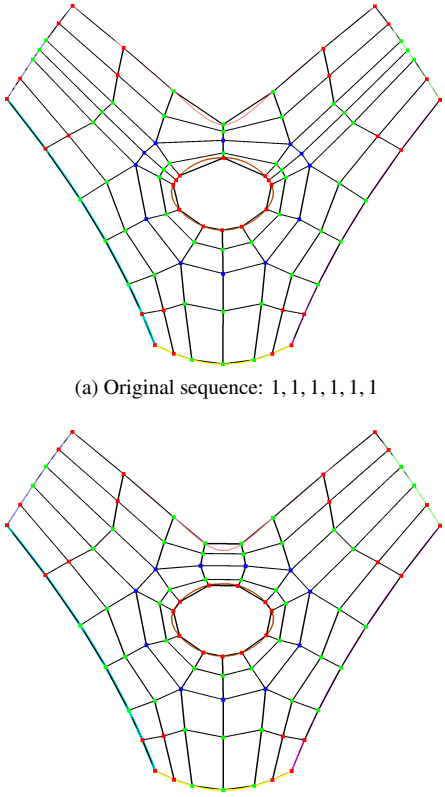
Figure 6: Different MAT-quad diagrams defined by different parameterizations.

4.3. Editing templates

The template structure can be modified in various ways, if for some reason the default configurations are not satisfactory. We discuss two additional MAT editing techniques that we found useful in practice: modifying the per-side parameterizations, and refining the interior MAT edges. In our experience the effect of these operations are fairly straightforward and predictable, and an acceptable template can be easily produced in the vast majority of cases.

4.3.1. Modifying the parameterizations

First it should be observed that the h_i parameterization determines the region of influence of the individual ribbons, thus by changing related boundary conditions one can modify the MAT diagram and the corresponding MAT-quads. By default, the boundary conditions of h_i parameters increase from 0 to 1 linearly along the entire length of the $i-1, i+1$ boundary curves, however optionally we may modify the length of these linear segments, producing different h_i parameterizations. In Figure 6 an example is shown, where the $h_i = \frac{1}{10}, \ldots, \frac{4}{10}$ isolines are displayed. Figure 6a shows a simple MAT graph using the default h parameterization. In Figure 6b, some sides have been reparameterized producing a branching MAT, and thus implying a different MAT-quad structure.



(a) Original sequence: 1, 1, 1, 1, 1, 1

(b) Modified sequence: 0, 2, 0, 1, 1, 1

Figure 7: Changing division numbers.

4.3.2. Refining the MAT

The second editing technique concerns MAT division numbers that can be assigned to each individual edge of the MAT skeleton. A division number zero means that the given edge is

not subdivided. This default setting may be overwritten (independently for each template depth) and accordingly the number of footpoints and the density of the related faces will change. An example can be seen in Figure 7, where a multi-connected model with $D = 4$ is shown. In Figure 7a each MAT-edge was subdivided once, then in Figure 7b the configuration has been modified by setting edges 1 and 3 to zero and edge 2 to two (see edge numbering in Figure 4).

These examples show that by varying the initial MAT-graph and/or refining the MAT-quad structure a family of different templates can be produced.

5. Distributing weight deficiency

In this section we describe how blending functions are assigned to the interior vertices of the template. We adapt the Bézier-based construction of [11] to handle (clamped or periodic) B-spline boundaries with non-uniform knots. We construct *exterior* and *interior blend functions* by naturally propagating *per-side basis functions* towards the interior.

5.1. Per-side blending functions

For each side i of a template with depth D, there exist L layers of vertices, each with a corresponding side-based longitudinal index $j \in [0, 1, \ldots, d_i]$, and a layer index $k \in [0, 1, \ldots, L-1]$ – see two adjacent grids of vertices in Figure 8a. We define per-side blending functions for each index tuple (i, j, k) similarly to GBS patches:

$$\sigma_{ijk}(u, v) = \mu_{ijk}(u, v) \cdot N_j^{\Theta_i}(s_i(u, v)) B_j^D(\hat{h}_i(u, v)). \tag{12}$$

where the longitudinal knot vector Θ_i, which is different from that of the input ribbon, defines a cubic B-spline with d_i control points,[2] μ_{ijk} is a rational weighting term (see Appendix C), and $\hat{h}_i$ is a modified h-parameter (described in Appendix D). We would like to ensure that these blend functions take their maximum in the vicinity of the corresponding template vertex to avoid creating blends with multiple maxima when these functions are combined – this motivates a careful choice of the longitudinal knot vectors, as well as the modification of the h-parameters.

5.1.1. Longitudinal knot vectors

We depart from the 'topological' construction of [11], by defining the knot vectors $\Theta_i = [0, 0, 0, 0, \theta_{i,0}, \ldots, \theta_{i,d_i-4}, 1, 1, 1, 1]$ based on the template geometry. Assuming a sequence of template vertices $(\mathbf{v}_{i,0}, \ldots, \mathbf{v}_{i,d_i})$ associated with side i and corresponding s-parameters $(s_{i,0}, \ldots, s_{i,d_i})$, we set the (interior) knots as $\theta_{i,j} = s_{i,j+2}$. (Accordingly, the first and last two vertices are skipped).

[2] We describe the non-periodic case. The periodic case can be handled analogously.

5.2. Exterior and interior blend functions

We present a color-coded arrangement of template vertices that helps explaining how exterior and interior blend functions are constructed from side-based blends. Due to the special MAT-based structure of our templates, the vertices are arranged into concentric layers and can be classified into exterior and interior vertices, as illustrated on Figure 8b.

The first two layers correspond to exterior control points (*CP*s) and define the boundary curves and cross-derivatives (green and red). In the corners (red) these will be overlapping, so control points (and blend functions) will be duplicated, each copy corresponding to a different ribbon and blend function. The *exterior blend functions* are simply the per-side blend functions of the template:

$$\hat{\gamma}_{ijk}^{\mathrm{Ext}}(u, v) = \sigma_{ijk}(u, v). \tag{13}$$

The remaining vertices (yellow and blue) correspond to interior CPs. Given a template vertex ℓ, let us denote the collection of index tuples with *minimal* layer index by $I(\ell) = \{(i, j, k)\}$, (i.e., those with minimal 'distance' from the boundary). *Interior blend functions* are defined by averaging the corresponding per-side functions:

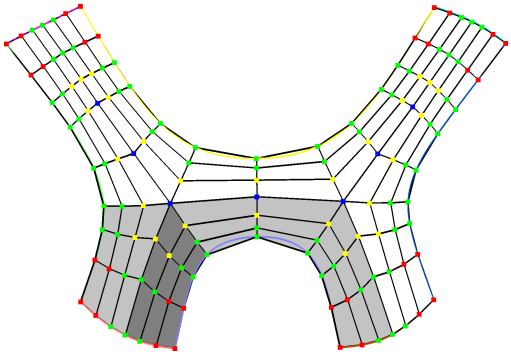
$$\gamma_{\ell}^{\mathrm{Int}}(u, v) = \frac{1}{|I(\ell)|} \sum_{(i,j,k) \in I(\ell)} \sigma_{ijk}(u, v). \tag{14}$$

Note that when per-side blends have little overlap, such averaging might result in a multi-modal function – the previously described longitudinal knot vectors and transversal reparameterizations were proven to be critical to mitigate this issue. Figure 9 shows the isolines of a blend function averaged from three distinct sides.

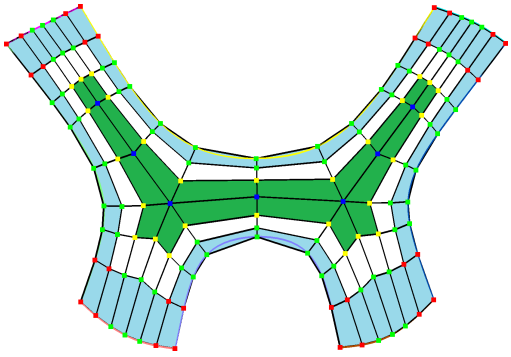
Given these blending functions we can use normalized interior blends to distribute the weight deficiency generated by the exterior blends, following Equation (10), defining the Template GBS patch.

6. Positioning the control points in 3D

In the previous chapters we defined a template topology and associated blending functions for interior control. In this section, we discuss how to position the related control points in 3D. For this purpose, it would be natural to apply variational design, i.e., to optimize a functional that measures surface quality. We mention two fundamental difficulties that arise in our context: (i) representations over curved domains are based on local parameters discretized over a mesh, making efficient gradient-based optimization difficult, (ii) functionals based on curvature or curvature variation lead to optimization problems that are highly non-linear, ill-conditioned and sensitive to initialization. With this in mind, we develop a set of heuristics that provide a reasonable initial setting for the control points with relatively little computational effort: one method based on control structure optimization (6.2), one based on approximating a high-quality guiding surface (6.3), and finally a local fairing method (6.4). All examples shown in this paper were created using these techniques.



(a) Per-side decompositions overlapping in a corner



(b) Exterior and interior vertices (blue and green regions)

Figure 8: Different decompositions of a MAT-based template.

6.1. Positioning exterior CPs by ribbon approximation

The CPs belonging to the outer two layers of the control structure ($\hat{\mathbf{C}}^{\mathrm{Ext}}_{ijk}$) define the positions and cross-derivatives along the patch boundaries. These control points are set by approximating the input ribbons with B-spline×Bernstein functions determined by the template. We apply simple least-squares fitting to samples of the curves and cross-derivative vector functions. The resulting exterior CPs will be fixed during subsequent optimizations of the interior. In Section 8 we provide some numerical data to evaluate the positional and angular accuracy of ribbon approximations.

6.2. Positioning interior CPs by optimization

To find a natural initial placement for the control points, we can employ Laplacian smoothing to the control structure, without considering the underlying surface. We define a Laplacian matrix $\mathbf{L}$, which expresses each control point as a weighted linear combination of neighboring control points:

$$\mathbf{L}_{ij} = \begin{cases} -w_{ij} & \text{for } i \neq j, \\ \sum_{k \neq i} w_{ik} & \text{for } i = j, \end{cases} \tag{15}$$

where $w_{ij} \in \mathbb{R}$ is the weight of vertex j for vertex i. The control points are found by minimizing a quadratic form of (some power of) the Laplacian:

$$\min \mathbf{p}^T \mathbf{L}^k \mathbf{p}, \tag{16}$$

or, equivalently, by solving a linear system

$$\mathbf{L}^k \mathbf{p} = 0, \tag{17}$$

with the constraint that the vertices in the outermost two layers are fixed as discussed previously. In practice, we recommend using $k = 3$ (triharmonic smoothing).

In theory any graph or polygonal Laplacian matrix could be used [36], but in our case the polygonal mesh is not a sampling of an underlying surface, so we need to consider our choice carefully. Practical experience shows that for good surface shape the control structure should be positioned similarly to the maxima of blend functions within the domain. Thus, we construct the Laplacian by first repositioning each template vertex within the domain to the maxima of the corresponding blend function. For each vertex $\mathbf{v}_i$, we consider all vertices of adjacent polygons, and compute its generalized barycentric coordinates $(\lambda_{i1}, \ldots, \lambda_{ik})$ within the planar polygon formed by this 1-ring neighborhood. Then, the Laplacian weights are simply set as $w_{ij} = \lambda_{ij}$. We use Mean Value Coordinates [37], as they are well-defined for general non-convex polygons.

6.3. Positioning interior CPs by fitting a guiding surface

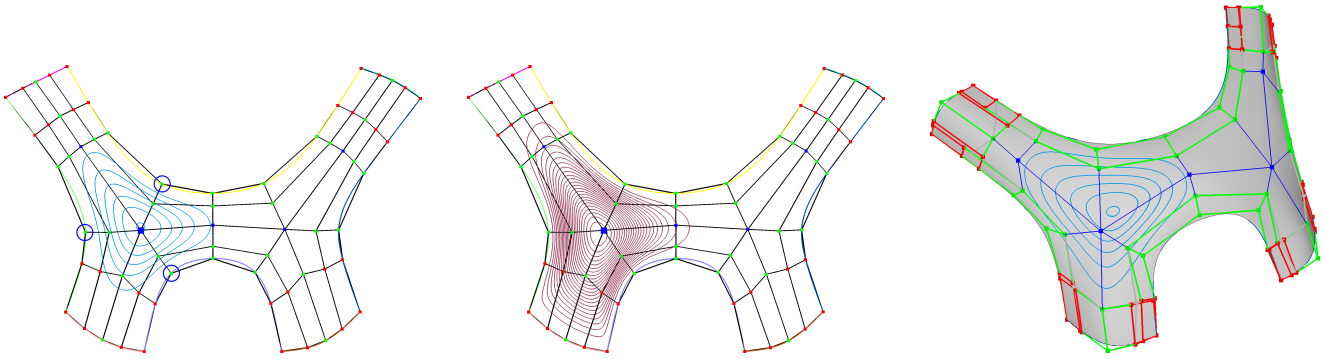
The previous method only considered the control structure, while ignoring the shape of the underlying surface. As an alternative, we propose to approximate a high-quality guiding surface. The approximation is done by simple least-squares fitting on domain mesh vertices, using only interior blending functions (with exterior CPs considered fixed). In theory any guiding surface can be used that produces a triangle mesh with good shape. One possibility is to approximate a polyharmonic surface [38] determined by the same parametric domain and local parameterization – see Vaitkus [39] for details. Another possibility is approximating a ribbon-based GBS patch [5]. We also add a regularization term that penalizes CP displacements from the positions determined by the optimization of the previous section. The examples shown in the paper were created by fitting a biharmonic surface (with fitting and regularization terms weighted equally).

6.4. Repositioning CPs by local fairing

In our experience, using either optimization or guide fitting, the manual tweaking of some control points might be recommended to even out the local curvature distribution. We have developed a method to make our manual tweaking process semi-automatic, as follows. Given a selected CP, we consider the part of the domain corresponding to the 50th percentile of the blending function – within this region, we evaluate a curvature oscillation energy (the gradient-squared or *cotangent Laplacian* of mean curvature) on the evaluated surface mesh and search for a displacement of the control point that minimizes this energy in the vicinity of the current position. In practice, a simple uniform grid search is carried out in the direction of the initial surface normal at the blend function maxima. Of course, a more global fairing could further improve surface quality – due to the aforementioned difficulties this is left for future work.

7. Hybrid representations

At this point we have already developed a representation that depends on the template, see Eq. (10). This, however, satisfies



(a) Averaged blending function (b) Normalized blend (c) Contours mapped onto the surface

Figure 9: Interior blending function.

positional and cross-derivative constraints only in an approximate sense.

In this Section, we present hybrid representations that can exactly reproduce the input ribbons. We describe a hybrid variant of the template GBS patch, as well as a scheme that is applicable to general interpolant-based surfaces.

7.1. Hybrid GBS

A key observation is that the exterior control points and blend functions in the Template GBS formula (Eq. 10) can be replaced with (rescaled versions of) input ribbon control points and GBS blend functions (with modified cross-degree), and the result will interpolate the input ribbons with G^1 continuity. This means, in essence, that we define $\mathbf{C}^{\text{Ext}}_{ijk}$ and $\gamma^{\text{Ext}}_{ijk}$ in the Hybrid GBS formula

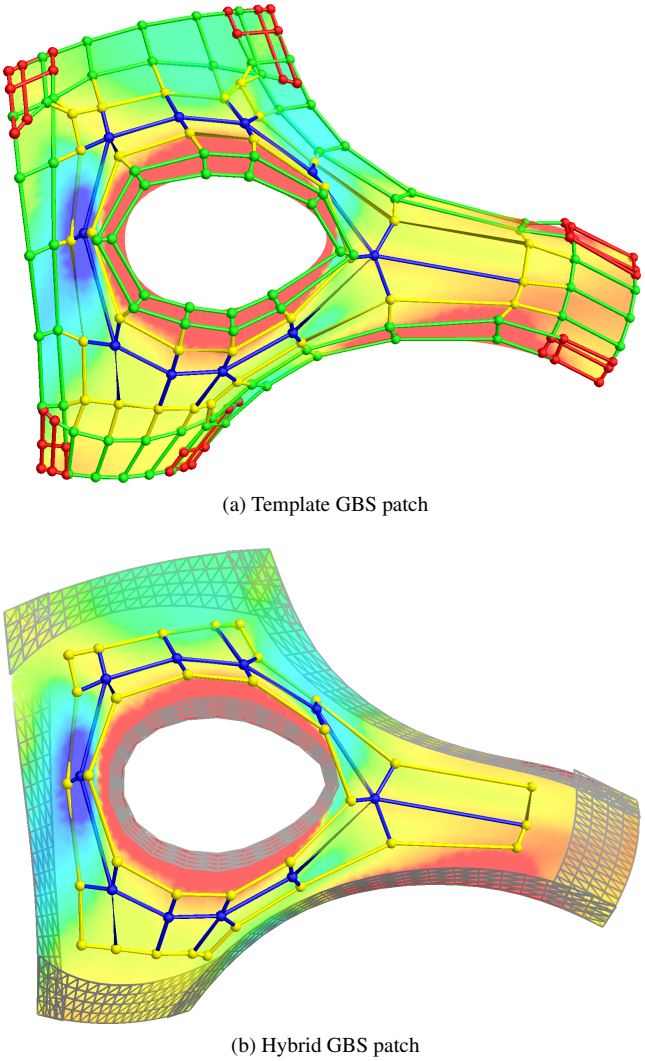
$$\mathbf{S}_{\text{HGBS}}(u,v) = \sum_{i=1}^{n+N_L}\sum_{j=0}^{d_i}\sum_{k=0}^{d^\perp}\mathbf{C}^{\text{Ext}}_{ijk}\cdot\gamma^{\text{Ext}}_{ijk}(u,v) + \\ + \frac{1-\Gamma^{\text{Ext}}(u,v)}{\Gamma^{\text{Int}}(u,v)}\sum_{\ell}\mathbf{C}^{\text{Int}}_{\ell}\cdot\gamma^{\text{Int}}_{\ell}(u,v) \qquad (18)$$

as

$$\mathbf{C}^{\text{Ext}}_{ijk} = \begin{cases}\mathbf{C}_{ijk} & \text{for } k=0,\\ \mathbf{C}_{ij(k-1)} + \frac{3}{d^\perp}(\mathbf{C}_{ijk}-\mathbf{C}_{ij(k-1)}) & \text{for } k=1,\end{cases} \qquad (19)$$

$$\gamma^{\text{Ext}}_{ijk}(u,v) = \mu_{ijk}(u,v)\cdot N^{\Xi_i}_j(s_i(u,v))B^{d^\perp}_k(h_i(u,v)), \qquad (20)$$

with Ξ_i being the input knot vectors and $d^\perp = D$. Note that the weight deficiency generated by these input-based blends is equal to the weight deficiency of the exterior template blends, because the μ_{ijk} terms are defined by inserting knots into a common cubic Bézier 'base', preserving the weight sum (see Appendix C). As a consequence, the interior blend functions of Template GBS and Hybrid GBS patches are identical and only the ribbons will differ. In our experience enforcing exact ribbon interpolation often has near-imperceptible effect on surface shape in terms of curvature or reflection line distribution. Figure 10 shows a multiply connected example.



(a) Template GBS patch

(b) Hybrid GBS patch

Figure 10: Replacing exterior layers with the original ribbons.

7.2. Interpolant-based surfaces with interior control

Our approach for introducing interior control is generally applicable, to not only control point based formulations such as

GBS patches, but also to surfaces defined by weighted interpolants as in Eq. (1). In particular, we can construct the following *interpolant-based* generalization of our scheme:

$$\mathbf{S}_{\mathrm{IB}}(u,v) = \sum_{i=1}^{n} \mathbf{I}_i(u,v) \cdot W_i(u,v) + \\ + \frac{1 - \sum_j W_j(u,v)}{\Gamma^{\mathrm{Int}}(u,v)} \sum_{\ell} \mathbf{C}_{\ell}^{\mathrm{Int}} \cdot \gamma_{\ell}^{\mathrm{Int}}(u,v), \qquad (21)$$

where $\mathbf{I}_i$ are surface interpolants and W_i are weight functions generating a weight deficiency. Some interpolant-based constructions, such as the Midpoint patches of Salvi et al. [15] are naturally weight deficient, so the above formula directly applies to them. However, other methods, including the inverse distance construction of Kato [13], or the popular corner-based patch of Charrot and Gregory [40] employ normalized blending functions, and are thus not weight deficient. In the spirit of previous works [12, 14], we can introduce an artificial weight deficiency by introducing an additional term such as $W = 1 - \prod_i h_i^2$ into each of the weight functions, which will also be equal to the weight deficiency.

8. Discussion

We believe that the parametric MAT-quad structure produces a natural partitioning for curved domains. It does work for complex surfaces, as well, as can be seen in the figures throughout the paper. It is also an attractive feature that the MAT-skeleton can be locally refined, thus the template will have more degrees of freedom where it is required. Figure 11 is an example where a D4 structure and its refined variant can be seen.

In our experience the blend functions in the interior are suitable for local editing. Our examples in Figures 12 and 13 show that lifting one or two control points yield local changes in the surface interior. The effect of relocation gets weaker as we move inwards, i.e., the yellow control points near the perimeter loop are stronger, while the blue ones are weaker.

It is a well-known property of transfinite patches, that if the ribbons are relatively narrow, then the patch interior may become somewhat flat. (See techniques to enhance the shape by ribbon- and central control points in [5].) This phenomenon can be noticed at highly curved objects, though it is less noticable for objects with hole loops, where the hole loops themselves provide sufficient geometric information for the interior. The GBS surface in Figure 14a is somewhat flat; we add an interior control structure, and the Template GBS yields a better shape (b), then it can be further improved by manual editing (c). Another example shown in Figure 15, where the isophotes of the double setback vertex blend improve after some manual tweaking. Finally, a plastic bottle surface is shown (Fig. 16), where after editing the interior control points a Hybrid GBS surface with good isophotes was obtained – note that the surface is perpendicular to the horizontal symmetry plane and thus can be smoothly joined with its other half.

In general, we have found that interior control structures with $D = 4, 5, 6$ are suitable for manual editing, but for higher depths we need to rely on the automatic procedures, such as optimization and fitting (see Section 6). The key in this project is the definition of blending functions. We have experimented with other alternatives, including uniform knots combined with various reparameterization schemes for the s-coordinates, as well as interpreting boundary vertices as Greville abscissae, and found that currently our described approach in Section 5 is the best choice in practice. At the same time, we believe other blends might also be suitable; for example, blends using mapped B-splines [28] or biharmonic weights [29].

Fig.	D	$\mathrm{Pos}_{\mathrm{avg}}$	$\mathrm{Pos}_{\mathrm{max}}$	$\mathrm{Angle}_{\mathrm{avg}}$	$\mathrm{Angle}_{\mathrm{max}}$
1d	4	0.014	0.128	0.037	0.726
	5	0.022	0.269	0.026	0.268
	6	0.005	0.072	0.011	0.147
13b	4	0.022	0.186	0.063	0.729
	5	0.024	0.179	0.06	0.638
	6	0.021	0.178	0.029	0.386
11b	4	0.014	0.074	0.168	2.683
	5	0.008	0.056	0.075	0.647
	6	0.004	0.033	0.039	0.419

Table 1: Approximation errors of the models in Figs. 1d, 13b and 11b. Positional deviations are given as percentages of the bounding box diagonal, while normal vector deviations are in degrees.

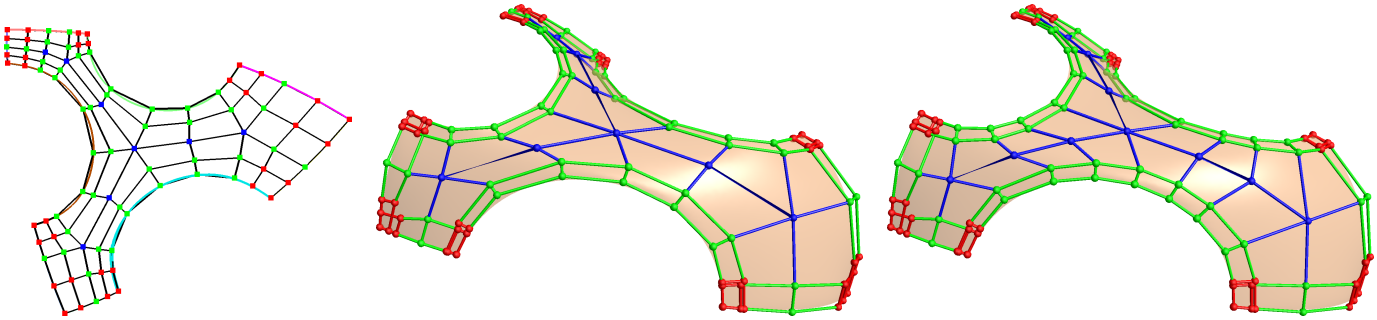
Next we show positional and angular deviations between the input ribbons and the approximate ribbons of some Template GBS test models. We have varied the depth values ($D = 4, 5, 6$) for three different example surfaces – see Table 1. As expected, higher depth yields more degrees of freedom, and thus generally better approximations. (The approximation error might increase when going from even to odd depths, due to the non-nested knot vectors.)

The use of Template GBS vs. Hybrid GBS depends on the application. The approximating Template GBS patches may represent an acceptable alternative when exact interpolation is not crucial, otherwise the use of Hybrid GBS patches is recommended.

Note that Template and Hybrid GBS patches cannot be exactly represented in NURBS format due to the use of harmonic parameterizations. Nevertheless, they can be tightly approximated using a collection of quadrilateral patches and the parametric MAT structure suggests a natural decomposition.

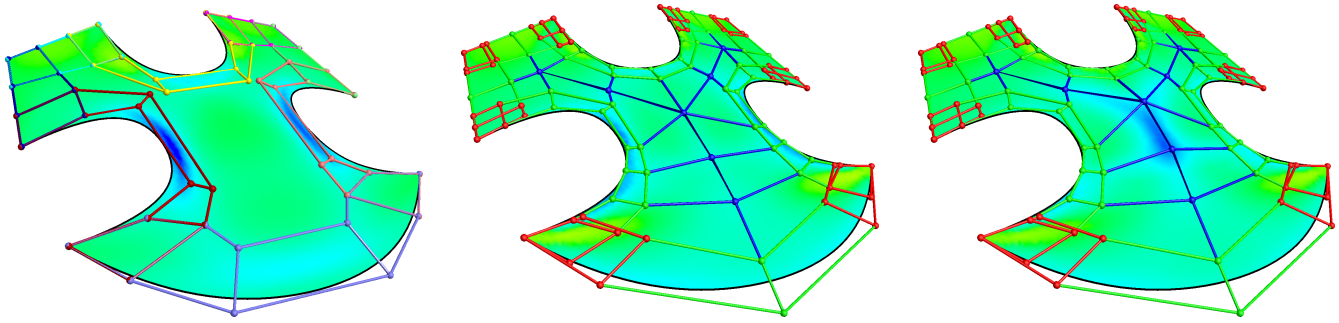
Conclusion

We have presented an approach to define interior control structures for Generalized B-spline patches; the scheme can be extended to arbitrary ribbon based surfaces, as well. The structure is built on configurable templates, constructed in the domain space by means of parametric MAT-diagrams. A set of blend functions are assigned to the template vertices, then related control points are placed in 3D either by optimization, fitting or manual editing. We have pointed out the importance of the blending functions; finding alternative solutions and further utilization of the MAT structure is subject of ongoing research.



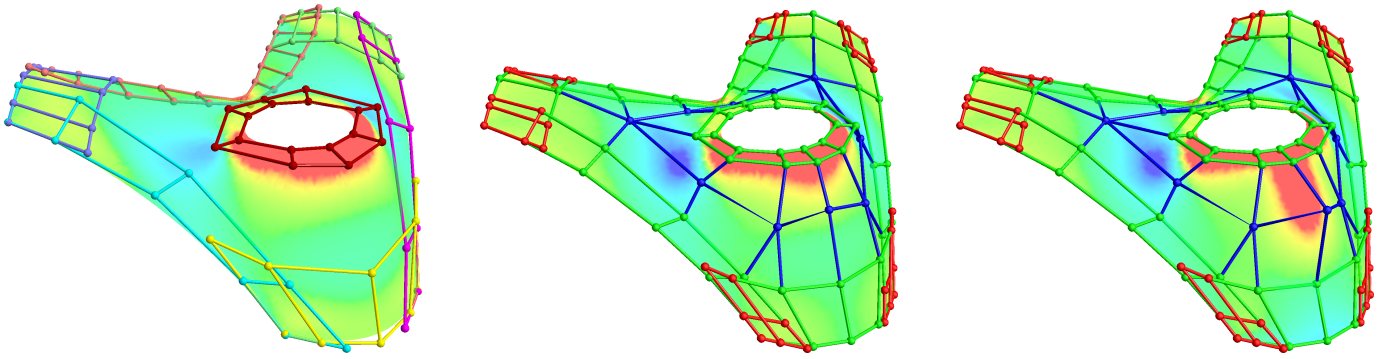
(a) Template (b) Template GBS (c) Template GBS refined

Figure 11: Interior controls in a complex configuration.



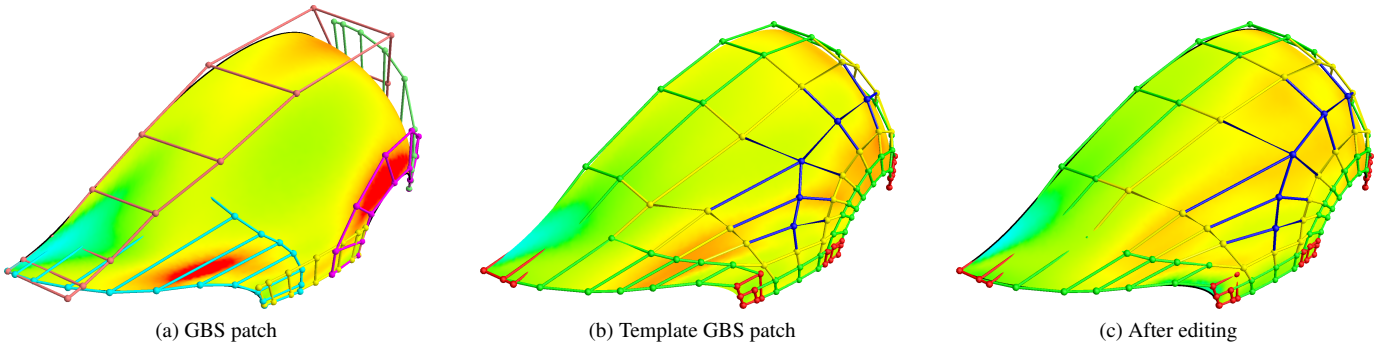
(a) GBS patch (b) Template GBS patch (c) After editing

Figure 12: Creating and editing a test model.



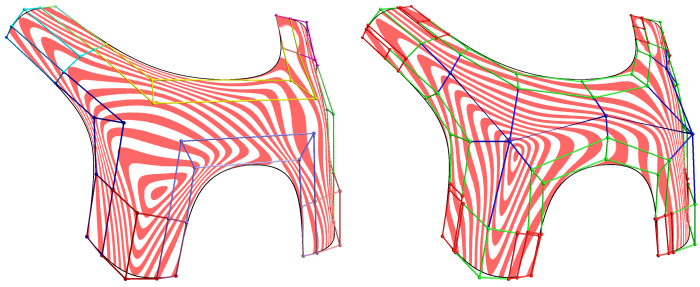
(a) GBS patch (b) Template GBS patch (c) After editing

Figure 13: Editing the ‘V-shape’ model.



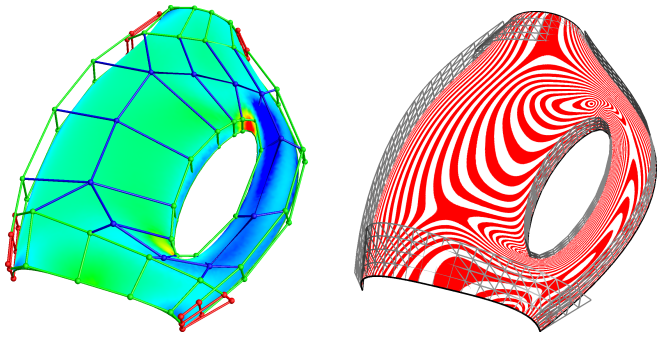
(a) GBS patch (b) Template GBS patch (c) After editing

Figure 14: Editing the ‘helmet’ model.



(a) GBS patch (b) Template GBS patch

Figure 15: Enhancing isophote lines by CP modification.



(a) Control structure (b) Hybrid GBS patch with isophotes

Figure 16: Test object with a hole loop.

Acknowledgments

This project has been supported by the Hungarian Scientific Research Fund (OTKA, No. 145970). The authors would like to acknowledge the significant programming contribution of György Karikó for extending our prototype surfacing system Sketches (ShapEx Ltd., Budapest).

References

[1] Várady, T, Salvi, P, Vaitkus, M. Genuine multi-sided parametric surface patches – a survey. Computer Aided Geometric Design 2024;110:102286. doi:10.1016/j.cagd.2024.102286.

[2] Gregory, JA. *n*-sided surface patches. In: The Mathematics of Surfaces. IMA; 1986, p. 217–232.

[3] Salvi, P, Várady, T, Rockwood, A. Ribbon-based transfinite surfaces. Computer Aided Geometric Design 2014;31(9):613–630. doi:10.1016/j.cagd.2014.06.006.

[4] Várady, T, Salvi, P, Karikó, G. A multi-sided Bézier patch with a simple control structure. Computer Graphics Forum 2016;35(2):307–317. doi:10.1111/cgf.12833.

[5] Vaitkus, M, Várady, T, Salvi, P, Sipos, Á. Multi-sided B-spline surfaces over curved, multi-connected domains. Computer Aided Geometric Design 2021;89:102019. doi:10.1016/j.cagd.2021.102019.

[6] Várady, T, Salvi, P, Vaitkus, M, Sipos, Á. Multi-sided Bézier surfaces over curved, multi-connected domains. Computer Aided Geometric Design 2020;78:101828. doi:10.1016/j.cagd.2020.101828.

[7] Loop, CT, DeRose, TD. A multisided generalization of Bézier surfaces. ACM Transactions on Graphics (TOG) 1989;8(3):204–234. doi:10.1145/77055.77059.

[8] Zheng, J, Ball, AA. Control point surfaces over non-four-sided areas. Computer Aided Geometric Design 1997;14(9):807–821. doi:10.1016/S0167-8396(97)00007-1.

[9] Krasauskas, R. Toric surface patches. Advances in Computational Mathematics 2002;17:89–113. doi:10.1023/A:1015289823859.

[10] Goldman, R. Multisided arrays of control points for multisided Bézier patches. Computer Aided Geometric Design 2004;21(3):243–261. doi:10.1016/j.cagd.2003.10.003.

[11] Vaitkus, M, Salvi, P, Várady, T. Interior control structure for Generalized Bézier patches over curved domains. Computers & Graphics 2024;121:103952. doi:10.1016/j.cag.2024.103952.

[12] Várady, T, Salvi, P, Rockwood, A. Transfinite surface interpolation with interior control. Graphical Models 2012;74(6):311–320. doi:10.1016/j.gmod.2012.03.003.

[13] Kato, K. N-sided surface generation from arbitrary boundary edges. In: Curve and Surface Design. AFA; 2000, p. 173–182.

[14] Martin, F, Reif, U. Trimmed spline surfaces with accurate boundary control. In: Geometric Challenges in Isogeometric Analysis; vol. 49 of *Springer INdAM Series*. 2022, p. 123–148. doi:10.1007/978-3-030-92313-6_6.

[15] Salvi, P, Kovács, I, Várady, T. Computationally efficient transfinite patches with fullness control. In: Proceedings of the Workshop on the Advances of Information Technology. BME; 2017, p. 96–100. arXiv:2002.11212.

[16] Salvi, P. Intuitive interior control for multi-sided patches with arbitrary boundaries. Computer-Aided Design and Applications 2024;21(1):143–154. doi:10.14733/cadaps.2024.143-154.

[17] Várady, T, Salvi, P, Kovács, I. Enhancement of a multi-sided Bézier surface representation. Computer Aided Geometric Design 2017;55:69–83. doi:10.1016/j.cagd.2017.05.002.

[18] Peters, J. Splines for meshes with irregularities. The SMAI Journal of computational mathematics 2019;5:161–183. doi:10.5802/smai-jcm.57.

[19] Levin, A. Interpolating nets of curves by smooth subdivision surfaces. In: SIGGRAPH '99. ACM; 1999, p. 57–64. doi:10.1145/311535.311541.

[20] Nasri, AH, Sabin, MA. Taxonomy of interpolation constraints on recursive subdivision surfaces. The Visual Computer 2002;18(5–6):382–403. doi:10.1007/s003710100155.

[21] Shen, J, Kosinka, J, Sabin, MA, Dodgson, NA. Conversion of trimmed NURBS surfaces to Catmull–Clark subdivision surfaces. Computer Aided Geometric Design 2014;31(7–8):486–498. doi:10.1016/j.cagd.2014.06.004.

[22] Shen, J, Kosinka, J, Sabin, M, Dodgson, N. Converting a CAD model into a non-uniform subdivision surface. Computer Aided Geometric Design 2016;48:17–35. doi:10.1016/j.cagd.2016.07.003.

[23] Feng, YF, Shen, LY, Li, X, Yuan, CM, Jiang, X. Patching non-uniform extraordinary points. IEEE Transactions on Visualization and Computer Graphics 2023;doi:10.1109/TVCG.2023.3271669.

[24] Giannelli, C, Jüttler, B, Kleiss, SK, Mantzaflaris, A, Simeon, B, Špeh, J. THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. Computer Methods in Applied Mechanics and Engineering 2016;299:337–365. doi:10.1016/j.cma.2015.11.002.

[25] Sederberg, TW, Zheng, J, Bakenov, A, Nasri, A. T-splines and T-NURCCs. ACM transactions on graphics (TOG) 2003;22(3):477–484. doi:10.1145/882262.882295.

[26] Antonelli, M, Beccari, CV, Casciola, G. High quality local interpolation by composite parametric surfaces. Computer Aided Geometric Design 2016;46:103–124. doi:10.1016/j.cagd.2016.06.005.

[27] Pla-Garcia, N, Vigo-Anglada, M, Cotrina-Navau, J. *N*-sided patches with B-spline boundaries. Computers & Graphics 2006;30(6):959–970. doi:10.1016/j.cag.2006.05.001.

[28] Yuan, X, Ma, W. Mapped B-spline basis functions for shape design and isogeometric analysis over an arbitrary parameterization. Computer Methods in Applied Mechanics and Engineering 2014;269:87–107. doi:10.1016/j.cma.2013.10.023.

[29] Jacobson, A, Baran, I, Popovic, J, Sorkine, O. Bounded biharmonic weights for real-time deformation. ACM Transactions on Graphics (TOG) 2011;30(4):78. doi:10.1145/1964921.1964973.

[30] Feng, P, Warren, J. Discrete bi-Laplacians and biharmonic B-splines. ACM Transactions on Graphics (TOG) 2012;31(4):115. doi:10.1145/2185520.2185611.

[31] Nguyen, VP, Rabczuk, T, Bordas, S, Duflot, M. Meshless methods: a review and computer implementation aspects. Mathematics and Computers in Simulation 2008;79(3):763–813. doi:10.1016/j.matcom.2008.01.003.

[32] Hettinga, GJ, Kosinka, J. A multisided C^2 B-spline patch over ex-

traordinary vertices in quadrilateral meshes. Computer-Aided Design 2020;127:102855. doi:10.1016/j.cad.2020.102855.
[33] Hettinga, GJ, Kosinka, J. Multisided B-spline patches over extraordinary regions. In: Smart Tools and Apps for Graphics. Eurographics Association; 2020,doi:10.2312/stag.20201235.
[34] Siddiqi, K, Pizer, S. Medial Representations: Mathematics, Algorithms and Applications; vol. 37 of *Computational Imaging*. Springer; 2008. doi:10.1007/978-1-4020-8658-8.
[35] Fogg, HJ, Armstrong, CG, Robinson, TT. Enhanced medial-axis-based block-structured meshing in 2-D. Computer-Aided Design 2016;72:87–101. doi:10.1016/j.cad.2015.07.001.
[36] Bunge, A, Botsch, M. A survey on discrete Laplacians for general polygonal meshes. In: Computer Graphics Forum; vol. 42. Wiley; 2023, p. 521–544. doi:10.1111/cgf.14777.
[37] Floater, MS. Mean value coordinates. Computer Aided Geometric Design 2003;20(1):19–27. doi:10.1016/S0167-8396(03)00002-5.
[38] Jacobson, A, Tosun, E, Sorkine, O, Zorin, D. Mixed finite elements for variational surface modeling. Computer Graphics Forum 2010;29(5):1565–1574. doi:10.1111/j.1467-8659.2010.01765.x.
[39] Vaitkus, M. A note on ribbon-based biharmonic surface patches. 2023. arXiv:2311.12822.
[40] Charrot, P, Gregory, JA. A pentagonal surface patch for computer aided geometric design. Computer Aided Geometric Design 1984;1(1):87–94. doi:10.1016/0167-8396(84)90006-2.
[41] Gregory, JA. Smooth interpolation without twist constraints. In: Computer Aided Geometric Design. University of Utah; 1974, p. 71–87. doi:10.1016/B978-0-12-079050-0.50009-6.

Appendix A. Curved domain generation

Here we only give a high-level description of the domain generation method – the reader should consult Vaitkus et al. [5] for more details. The domain is created from the boundary curves. First let us assume that there are no interior loops. The perimeter boundary curves are discretized, and the polylines are developed into the plane by placing the segments one by one while retaining the angle at the vertices. This will result in an open polygon, which is subsequently closed by distributing the deviation vector between the first and last points.

When the input is multiply connected, first we create a GBS patch without holes, and project the curves onto the surface to obtain interior loops in the domain.

Appendix B. Per-side parameterizations

Local parameters are computed by minimizing Dirichlet energy $\int |\nabla f|^2 \, \mathrm{d}A$ over the domain. The boundary values are set as follows:

- The *side parameter* associated with the ith side, s_i, is constant 0 on the $i-1$st side, increases according to the knot vector Ξ_i from 0 to 1 on the ith side, and it is constant 1 on the $i+1$st side. (It is unspecified on other parts of the boundary.)
- The *distance parameter* associated with the ith side, h_i, is constant 0 on the ith side, increases from 0 to 1 on the adjacent sides, and it is constant 1 on all other sides.

In the presence of periodic interior loops, side parameters need some extra care as the function values may jump at the cut line; see details in the original paper [5].

Appendix C. GBS blending function

The blending function γ_{ijk} is defined as

$$\gamma_{ijk}(u,v) = \mu_j^i(h_{i^-}, h_i, h_{i^+}) \cdot N_j^{\Xi_i}(s_i) B_k^{d^\perp}(h_i). \tag{C.1}$$

It is composed – similarly to the blends of the ribbons – of B-spline and Bernstein basis functions, and a rational weight μ_j^i that ensures the interpolation property. The notations i^- and i^+ are defined to facilitate circular indexing for the perimeter loop:

$$i^- = \begin{cases} n & \text{for } i = 1, \\ i-1 & \text{for } i \le n, \\ i & \text{otherwise.} \end{cases} \qquad i^+ = \begin{cases} i+1 & \text{for } i < n, \\ 1 & \text{for } i = n, \\ i & \text{otherwise.} \end{cases} \tag{C.2}$$

The weight function μ_j^i is constant 1 for all periodic ribbons. For the outer loop, its value can be computed by the following simple steps (corresponding to its Bézier counterpart in [6]):

1. Define quadratic rational quantities similar to Gregory's twist [41]:
$$\alpha = \frac{h_{i^-}^2}{h_{i^-}^2 + h_i^2}, \qquad \beta = \frac{h_{i^+}^2}{h_{i^+}^2 + h_i^2}. \tag{C.3}$$
2. Create a 1D cubic clamped B-spline curve of one segment with initial control points $\{C_j^{\mu_i}\} = [\alpha, \alpha, \beta, \beta]$.
3. Insert the interior knots of Ξ_i, thereby changing $\{C_j^{\mu_i}\}$. Then $\mu_j^i(h_{i^-}, h_i, h_{i^+}) = C_j^{\mu_i}$.

This process guarantees that the weight deficiency is the same regardless of the actual knot structure.

Appendix D. Transversal reparameterization

When defining per-side blending functions, we apply a reparameterization to the h-coordinate to shift the maximum of the transversal Bernstein functions towards the corresponding template vertex. More concretely, if the value of the h_i-coordinate at the vertex should be shifted to $\hat{h}_{i,D}(p) = \frac{k}{D}$ (the maxima of the Bernstein polynomial B_k^D), then we define a (vertex-dependent) quadratic reparameterization

$$\hat{h}_{i,D}(p) = c_1(v) \cdot B_1^2(h_i) + B_2^2(h_i), \tag{D.1}$$

where B_i^2 are quadratic Bernstein polynomials, and

$$c_1(v) = \max\left(0, \min\left(\frac{\hat{h}_{i,D}(p) - B_2^2(h_i)}{B_1^2(h_i(p))}, 1\right)\right). \tag{D.2}$$

Note that the reparameterization remains monotonic only when $\frac{1}{2}\hat{h}_{i,D}(p) \le h(p) \le \sqrt{\hat{h}_{i,D}(p)}$ holds.