Multi-sided Bézier surfaces over curved, multi-connected domains

Tamás Várady, Péter Salvi, Márton Vaitkus, Ágoston Sipos

Budapest University of Technology and Economics

Abstract

A new multi-sided, control point based surface formulation is proposed, extending the principles of the Generalized Bézier patch published by Várady et al. (2016). The surface is constructed from side interpolants given in Bézier form. The boundary curves may have different degrees, and the cross-derivatives can ensure arbitrary $G$-continuity with adjacent patches. The representation is capable of handling boundary curves with high curvature variations and concave angles. The surfaces are $C^\infty$-continuous and may have holes in their interior.

The most distinctive feature of the patch is that it is defined over a planar domain with curved boundaries that mimic the shape of the 3D boundary curves. Local parameters, derived from harmonic barycentric coordinates, are associated with each side of the domain. The control points are multiplied by Bernstein functions and additional rational terms that provide the same degree of freedom for shape design as their quadrilateral counterparts.

The paper introduces the full construction of Curved Domain (CD) Bézier patches including domain generation, parameterization, basis functions, and methods for additional interior shape control. Specific problems related to CD surfaces are also discussed with many examples that demonstrate the peculiarity and usefulness of the scheme.

Keywords: general topology surfaces; multi-sided Bézier patches; curved domain; holes

1. Introduction

The representation of general, multi-sided surfaces is a fundamental problem in CAGD. Such surfaces are needed for designing and reconstructing complex free-form objects, and for creating smooth transitions that connect a set of given surfaces (hole filling, vertex blending, lofting). Our main interest is to produce a collection of genuine, multi-sided surfaces where boundary curves and cross-derivatives are explicitly specified. The goal is to handle concave angles and concave boundaries (see Figs. 1a and 1b), as well, while interpolating hole loops is also permitted (see Fig. 8).

Tensor-product parametric surfaces are defined over rectangular domains, and the great majority of multi-sided patches are defined over convex polygons; however, this simplicity imposes limitations on the variation of shapes that can be constructed by these representations. Mapping a 2D convex angle to a 3D concave angle, or mapping a 2D straight line to a strongly curved concave boundary in 3D often yield unacceptable surfaces. There is a general consensus in the CAGD community, that if we wish to avoid abrupt curvature changes, we need a “natural” parametric domain, that mimics the shape of the curve or surface to be created, as in the case of non-uniform B-splines. Finite element analysis also benefits from low distortion of 2D–3D mappings (Shewchuck, 2002; Engvall and Evans, 2018).

In this research project we create patches over multi-connected, planar domains with curved boundaries, that are adapted to resemble a given set of 3D boundary loops. We deal with control point based surfaces, combining ribbons (special Bézier-like patches, see Section 3). The control points of each ribbon determine a Bézier boundary curve with arbitrary degree and geometric cross-derivative constraints towards the interior. In our previous publications, we proposed the Generalized Bézier (GB) scheme that produced a unified control structure over a polygonal domain, see Várady et al. (2016, 2017); then we extended this concept to concave polygonal domains (CGB patches) in Salvi and Várady (2018). In the current paper we introduce Generalized Bézier patches over Curved Domains, called CD Bézier patches.

Take the simple example in Figure 1 (shown with mean curvature maps). As it was discussed earlier, a convex polygonal domain (GB patch) is not suitable to produce a surface with concave angles (Fig. 1a), as the surface turns...
inside out beyond the boundaries. Concave polygonal domains (CGB patch), however, produce good shapes. In practical design, concave angles are often replaced by small filleting arcs due to aesthetic reasons, manufacturability and avoiding stress concentrations. As a result, we may obtain longer boundaries with high curvature variation. If we had a polygonal domain, the domain image of such a boundary would be a straight line, and the mapping from 2D to 3D would cause folding or strong parametric distortion, thus CGB domains would fail in these cases (Fig. 1b). Curved domains (CD patches), on the other hand, yield good surfaces, as the boundaries resemble those in 3D.

Complex multi-sided patches may have long winding boundaries, and also the lengths of the adjacent sides may differ to a large extent. Polygonal domains can hardly compensate for this, and strong parametric distortions may occur. Figure 2 shows symmetric ARAP distortion maps (Shtengel et al., 2017); we can compare the effect of parameterization for polygonal and curved domains on the same input, see also Figure 3. We return to this topic in Section 7.3.

![Figure 1: Problems to resolve.](image1)

![Figure 2: Parametric distortion for a CGB and a CD patch.](image2)

![Figure 3: Close-up views of the patches in Figure 2, showing the tessellation.](image3)
The construction of CD Bézier patches makes it possible to handle multiple boundary loops, as well. We describe the related parameterization in Section 4.2 and show test cases later in Sections 7.4–7.5.

The paper is structured in the following way. After reviewing previous work (Section 2), we introduce the basic equation of CD Bézier patches (Section 3). Algorithms of curved domain generation and parameterization are described in Section 4. We discuss how to add interior control points to gain further design freedom (Section 5), then continue with the advantages of splitting boundaries (Section 6). A few interesting test objects are examined in Section 7, and research topics for the future conclude the paper.

2. Previous work

A short review on various control point based multi-sided patches can be found in Várady et al. (2017); most of these use convex domains. To our best knowledge, there are only two surface representations – besides our own – that can handle concave configurations. Schaefer (2017) generalized the S-patch of Loop and DeRose (1989) to concave domains using mean value coordinates. Kato (1991, 2000) defined a transfinite surface which is also able to represent holes. See a summary in Table 1.

S-patches represent a natural generalization of Bézier triangles, and inherit most of the nice properties of Bézier patches. Unfortunately, they have a very complex control structure, with a large number of control points, which diminishes their appeal to interactive design applications; see Salvi and Várady (2018) for a comparison with concave GB patches. Schaefer (2017) mentions that by using mean value coordinates, S-patches can be interpreted over domains with holes. The construction, however – in particular the control point placement for concave angles and hole loops – is not detailed.

Kato’s patch uses a parameterization based on the triangle inequality, which works for multi-connected, concave domains, as well. However, this mapping is largely distorted, which results in high curvature variations near the corners, see Salvi and Várady (2018).

|                     | Concave domain | Interior control | Holes |
|---------------------|----------------|------------------|-------|
| Loop and DeRose (1989); Schaefer (2017) | ✓              | ✓                | △     |
| Kato (1991, 2000)   | ✓              | ×                | ✓     |
| Várady et al. (2016, 2017) | ×              | ✓                | ×     |
| Salvi and Várady (2018) | ✓              | △                | ×     |

Table 1: Capabilities of different multi-sided surface representations. Triangles mark partial support.
There is an important paper by Sabin (1998), that – unfortunately – has not been noticed by the CAGD community, judging by its low number of citations. The paper defines a transfinite scheme by minimizing the elastic energy of curves connecting surface points to each of the boundaries, satisfying derivative constraints. The equations also provide an exact normal vector for every surface point, which is a rare feature for multi-sided patches. The problem statement is described in a general way, but parameterization is described only for rectangular domains with circular holes.

Our own previous work includes the Generalized Bézier patch (Várady et al., 2016, 2017), and its extension, the Concave GB patch (Salvi and Várady, 2018), which forms the basis of the representation described in this paper. It is important to see how the control structure changed between these two variations, because this has made it possible to fill concave holes in a natural way. All ribbons in a GB patch have the same degree in both parametric directions, see the red and green control points in Figure 4a. The red control points at the corners are shared by adjacent ribbons. The yellow control points can be autogenerated, and then used for further editing of the interior. In contrast, the ribbons of a Concave GB patch, or a CD patch, are entirely independent: each can have a different pair of degrees, and even when the degrees are the same, the control point positions can differ, see e.g. the green–red and red–blue corners in Figure 4b. Now if two ribbons span a concave angle, the adjacent control points around the corner are positioned independently, see Figure 5.

In our work, generalized barycentric coordinates (see e.g. Hormann and Sukumar, 2017), and in particular harmonic coordinates (DeRose and Meyer, 2006), play an important role. The attractive properties of harmonic functions are well-known and widely exploited in the CAGD community (Gortler et al., 2006; Weber, 2017). Harmonic functions are “as smooth as possible” under a set of constraints, and thus generalize linear interpolation to higher dimensions. Harmonic coordinates have no closed-form solutions, except for some special domains, such as circles (Lu, 2009), and are computed using discretizations. At the same time there are alternative methods for computing harmonic coordinates that do not require the discretization of the domain interior (Rustamov, 2008; Martin et al., 2008), but these methods are computationally demanding.

3. Surface equations

This surface formulation is an extension of the concave GB patch. New features include (i) support for multiple loops, and (ii) better rational weights that leave the surface unchanged after degree elevating the boundary curves.

The Curved Domain Bézier patch is the normalized sum of $n$ ribbons. The $i$-th ribbon is determined by a quadrilateral Bézier interpolant $I_i$ of $d_i$ by $e_i$ degrees, where $d_i$ is the degree of the boundary curve, and $e_i$ is the number of prescribed cross-derivatives (independent of $d_i$). In the most frequent $G^1$ and $G^2$ cases, $e_i = 1$ or 2, i.e., there are two or three rows of control points, see Figure 4b. The equation of the interpolant is given in the following form:

$$I_i(s, h) = \sum_{j=0}^{d_i} \sum_{k=0}^{e_i} C_{j,k}^{i} \cdot B_{j,k}(s, h),$$  \hspace{1cm} (1)$$

where $s, h \in [0,1]$ are parameters along the boundary, and in the cross direction, respectively. $C_{j,k}^{i}$ refers to the $j$-th control point in the $k$-th row, being multiplied by its associated bivariate Bernstein polynomial

$$B_{j,k}(s, h) = B_j^u(s)B_k^v(h).$$  \hspace{1cm} (2)$$

As it will be shown later, the $i$-th ribbon $R_i$ reproduces $I_i$ on the $i$-th side of the domain, and vanishes on all other sides in both positional and differential sense, thus satisfying the interpolation property for the whole patch.

The CD Bézier patch is defined over a curved domain in the $(u, v)$ plane. For each side, we compute a local parameterization $(s_i, h_i)$ – see details in Section 4. The side parameter $s_i = s_i(u, v)$ varies linearly on side $i$ between 0 and 1, while the distance parameter $h_i = h_i(u, v)$ vanishes on side $i$ and increases monotonically within the domain, eventually reaching 1 as it gets to the “distant” sides (i.e., edges not adjacent to side $i$). We will show parameterization examples in Figures 7 and 8.
The ribbons are very similar to the Bézier interpolants, but while a ribbon also has \( e_i + 1 \) control rows, its cross-degree \( d_i^2 \) is not \( e_i \), but an independent value, satisfying \( d_i^2 > 2e_i \). This means that the basis functions associated with the control points change:

\[
\hat{\mathcal{B}}_{i,j,k}^k(s_i, h_i) = B_{i,j}^i(s_i)\hat{B}_j^k(h_i).
\] (3)

Ribbon control points are also multiplied by additional rational weight functions \( \mu_j^j \) (see details in Section 3.1):

\[
R_i(s_i, h_i) = \sum_{j=0}^{d_i} \sum_{k=0}^{e_i} C_i^{i,j,k} \cdot \mu_j^j(h_{i-1}, h_i, h_{i+1})\hat{\mathcal{B}}_{i,j,k}^k(s_i, h_i).
\] (4)

Note, that \( \mu_j^j = \mu_j^j(h_{i-1}, h_i, h_{i+1}) \) depends on the distance parameters of the adjacent sides \( h_{i-1} \) and \( h_{i+1} \), as well. The side index \( i \) is circular, with 1 coming after \( n \).

In order to ensure the convex combination property, we need to guarantee that the weighting functions sum up to 1 in each point of the curved domain. To this end, we normalize the patch equation by dividing with the sum of the control point weights:

\[
S(u, v) = \frac{1}{B_2(u, v)} \cdot \sum_{i=1}^{n} R_i(s_i, h_i),
\] (5)

where

\[
B_2(u, v) = \sum_{i=1}^{n} \sum_{j=0}^{d_i} \sum_{k=0}^{e_i} \mu_j^j(h_{i-1}, h_i, h_{i+1})\hat{\mathcal{B}}_{i,j,k}^k(s_i, h_i).
\] (6)

### 3.1. Rational terms

We continue with the definition of the rational \( \mu_j^j \) terms. First let us take \( G^1 \) ribbons and cubic boundaries. We are going to show that for \( k = 0, 1 \) the choice of

\[
\mu_j^j = \begin{cases} 
\alpha_j = h_{i-1}^2 / (h_{i-1}^2 + h_i^2), & \text{for } j = 0, 1 \\
\beta_j = h_{i+1}^2 / (h_{i+1}^2 + h_i^2), & \text{for } j = 2, 3 
\end{cases}
\] (7)

ensures positional and cross-derivative interpolation. On the boundary \( h_i = 0 \) all the \( \mu_j^j \) terms will become 1, thus \( S(u, v) \) reproduces the original ribbon \( i \). This ribbon will not affect the other sides, since its control points will be multiplied by Bernstein functions \( B_{i,j}^i(h_i) \) and \( B_j^k(h_i) \), that vanish in both positional and cross-derivative sense on the “distant” sides. On the \( i-1 \)-th side we have \( h_{i-1} = 0 \) and \( s_j = 0 \), so the first two columns of the control points are eliminated by \( \alpha_j = 0 \), while the last two columns are multiplied by the Bernstein functions \( B_{i,j}^i(s_i) \) and \( B_{j}^k(s_i) \), that vanish due to \( s_j \). Concerning side \( i + 1 \) similar reasoning holds.

This logic nicely extends for \( G^r \) ribbons with general degree \( d_i \) boundaries, as long as

\[
\mu_j^j = \begin{cases} 
\alpha_j = h_{i-j-1}^{r+1} / (h_{i-j-1}^{r+1} + h_i^{r+1}), & \text{for } 0 \leq j < \nu \\
\beta_j = h_{i+j}^{r+1} / (h_{i+j}^{r+1} + h_i^{r+1}), & \text{for } d_i - \nu \leq j \leq d_i 
\end{cases}
\] (8)

(assuming \( d_i \geq 2v + 1 \)). As for the remaining \( j \) indices \( \nu < j < d_i - \nu \), we choose the values in such a way that degree elevation of the boundary curves will not change the surface. This can be achieved in the following way: take the \( \mu \) terms associated with the minimal degree (i.e., \( 2v + 1 \)), and create a unary Bézier curve with the control points \( \{\mu_j^j\}_{j=0}^{2v+1} \). Elevating this to degree \( d_i \) will result in the correct \( \mu \) values. For example, let us look at a quintic boundary with \( G^1 \) constraints. Since \( v = 1 \), we start from the cubic values shown in Eq. (7), and elevate the degree twice:

\[
\begin{align*}
\text{cubic:} & \quad \begin{bmatrix} \alpha & \alpha & \beta & \beta \end{bmatrix} \\
\text{quartic:} & \quad \begin{bmatrix} \alpha & \alpha & \frac{1}{2}(\alpha + \beta) & \beta & \beta \end{bmatrix} \\
\text{quintic:} & \quad \begin{bmatrix} \alpha & \alpha & \frac{7}{10} \alpha + \frac{3}{10} \beta & \beta & \beta \end{bmatrix}
\end{align*}
\] (9)

So \( \mu_2^j = (\frac{7}{10} \alpha_i + \frac{3}{10} \beta_i) \) etc. Simple algebra shows that elevating the degree of a \( 2v + 1 \)-degree ribbon would lead to these weighting terms, so the surface does not change.
3.2. Surfaces with multiple loops

For multi-loop configurations our formula needs to be modified only to a minimal extent. Assume we have \(L\) loops: a perimeter loop and several internal hole loops, with each loop being composed of \(n_l\) ribbons, \(l = 1 \ldots L\). Then we obtain

\[
S(u, v) = \frac{1}{B_2(u, v)} \sum_{l=1}^{L} \sum_{i=1}^{n_l} R_{ij}(s_{lj}, h_{lj}).
\]

(10)

We need double indices, because the \(\mu_i^j\) terms are defined by the distance parameters of the previous and next boundaries, where the index \((l, i)\) must be interpreted in a circular fashion, with \((l, 1)\) coming after \((l, n_l)\). We will return to the parameterization of multi-connected domains in Section 4.2.

We will also discuss in Section 5 how to gain further design freedom by supplementing the ribbons with additional interior control points. This practically does not change the equation of the CD Bézier patch.

4. Domain and parameterization

As we have seen in Figures 1–3, the domain definition has a crucial effect on the resulting surface. In the following we propose an algorithm for generating curved domains, and then we show a suitable parameterization.

4.1. Curved domain generation

We now discuss how to compute a curved parametric domain for our surface by mapping the 3D boundary curves to the plane. Note that the cross-derivatives of the Bézier interpolants determine the tangent plane of the surface along the boundary. Our objective is to preserve the shape of the boundary as seen from “within the surface”. In the language of differential geometry, we want to preserve the geodesic curvature of the curves, i.e., the curvature component lying in the tangent plane of the surface (Gray et al., 2006).

The above criterion is discretized in a straightforward manner as follows. The boundary curve is sampled into a polyline consisting of \(n_b\) number of points, and at each point \(p_i\) for \(i \in \{1, \ldots, n_b\}\), the tangent plane is computed from the corresponding Bézier interpolant. We project the points \(p_{i-1}, p_{i+1}\) onto the plane of \(p_i\), resulting in points \(p'_{i-1}, p'_{i+1}\) and determine the angle \(\theta_i' = \angle(p'_{i-1}, p_i, p'_{i+1})\) – see Figure 6. Then, starting from an arbitrary point, we construct a planar polyline \(q_0, \ldots, q_{n_b-1}\) so that its angles are

\[
\angle(q_{i-1}, q_i, q_{i+1}) = \theta_i', \quad \frac{(n_b - 2)\pi}{\sum_{j=1}^{n_b} \theta_j'},
\]

(11)

and its edge lengths are

\[
|q_{i+1} - q_i| = |p_{i+1} - p_i|.
\]

(12)

This polyline mimics the shape of the original loop as close as possible, but it might not close up on itself – a closed loop \(\hat{q}_1, \ldots, \hat{q}_{n_b}\) is found using the procedure suggested in Salvi and Várady (2018, Section 4.3), i.e., by setting

\[
\hat{q}_i = q_i + v \cdot \frac{\sum_{j=1}^{i} |q_{j+1} - q_j|}{\sum_{j=1}^{n_b} |q_{j+1} - q_j|},
\]

(13)

where \(v = q_1 - q_{n_b}\) is the error vector. (See also notes on discretization in Section 7.6.)

For surface patches with interior boundary loops, we make the assumption that one of the boundary loops can serve as the outer perimeter of the domain. First, the perimeter loop is flattened as before, resulting in the domain \(\Omega_F\), and the related CD Bézier patch \(S_F\) is created. For each interior boundary loop, we can create a quasi-isometric flattening using the above procedure, as well, preserving its shape, but we still need to position and orient it relative to \(\Omega_F\). To this end, we project the points of the interior loop onto \(S_F\), and determine the pre-image of these points in \(\Omega_F\). Then the flattened interior loops are displaced and oriented to fit the projected loops in a least-squares sense by computing a rigid transformation aligning the two point sets similarly to the Iterative Closest Point (ICP) algorithm.
4.2. Local parameterization

As described earlier, for each Bézier ribbon we need local \( s, h \) parameters. In other words, we need to establish a series of one-to-one mappings between the curved domain and the unit square. We compute the local coordinates from generalized barycentric coordinates \( \lambda_i \), in the following way:

\[
\begin{align*}
    s_i &= \frac{\lambda_{i+1}}{\lambda_i + \lambda_{i+1}}, \\
    h_i &= 1 - \lambda_i - \lambda_{i+1}.
\end{align*}
\]

(14)

This choice of coordinates ensures an appropriate mapping of the sides of the unit square to the sides of the domain, as \( s_i = 0 \) is mapped onto side \( i - 1 \), \( h_i = 0 \) onto side \( i \), and \( s_i = 1 \) onto side \( i + 1 \), respectively. The \( h_i = 1 \) constant parameter line is mapped to all the “distant” sides \( j \) of the domain, where \( j \notin \{i - 1, i, i + 1\} \).

Following the approach of Concave Generalized Bézier patches (Salvi and Várady, 2018), we use harmonic coordinates (DeRose and Meyer, 2006). We assume that our domain boundary is discretized into polylines, as described in the previous section. Denote the set of corners by \( \{v_0, \ldots, v_{n-1}\} \). Harmonic coordinates are defined as solutions of the Laplace equation over the domain \( \Omega \), subject to a different set of Dirichlet boundary conditions for each coordinate:

\[
\Delta \lambda_i(x) = 0 \quad \text{for} \quad x \in \Omega, \quad \lambda_i(x) = b_i(x) \quad \text{for} \quad x \in \partial \Omega,
\]

(15)

where \( b_i(x) \) equals 1 at the corner \( v_i \), decreases linearly to 0 by arc-length along the boundary segments between the corner and its neighbors, and is constant 0 everywhere else along the domain boundary. For multiply connected domains, values along disjoint boundary components are also constrained to be 0. (See implementation details in Section 7.6).

In Figures 7 and 8, we show examples of local parameterizations for a single and a multi-loop surface, respectively.

5. Editing the interior

It may occur that the patch interior of CD patches does not entirely match the designer’s intention. The control points of the ribbons ensure that the boundary constraints are satisfied, but the interior is created indirectly through an
algebraic combination of the basis functions. If the fullness of the patch is not sufficient, or further shape adjustments are required, interior control points need to be added.

In the case of the original GB patches, there is an automatic mechanism to create a uniform structure of internal control points and associated basis functions. For concave and multi-connected patches no such mechanism exists, and adding interior control points for the individual ribbons is just an option. In these cases it is generally impossible to find a single central control point, and it is hardly possible to create a unified structure of control points from the ribbons, as the degrees and the 3D locations may be different. (Recall the related control structures in Figure 4b.)

Each interior control point must be associated with a given side $i$, and multiplied by a basis function of the local parameters $s_i, h_i$ in such a way that the interpolation properties along the boundaries are preserved. Adding new control points naturally corresponds to adding new basis functions within the domain, and consequently the sum of the weighting functions, i.e., the normalization term, will also change, see Eq. (6). Our experience shows that an enriched set of control points and basis functions generally produce better shapes than just the ribbon control points which act mainly in the vicinity of the boundaries.

To create a unified, global interior control structure for CD patches is subject of future research. In our former paper (Salvi and Várady, 2018) we suggested increasing the number of rows of the ribbons. In the current CD project we propose a very simple and effective interior control, that proved to be very useful in the course of our practical testing to set the proper fullness of our patches.

Our concept can be presented in its simplest form by means of a biquartic quadrilateral Bézier patch with $G^1$ constraints. Consider the two outer rows and columns of the control grid as four side interpolants that define the boundary constraints (see Fig. 9). Then the central control point $C_{2,2}$ is ideal for modifying the interior, since its basis function $B_i^2(s)B_j^2(h) = 6(1 − s)^3s^3(1 − h)^3h^3$ will not affect the boundaries. In the multi-sided context, we have $n$ sides and local coordinates $s_i, h_i$, and we may assign an interior control point $C_{i2}$ to each side. When these are multiplied by $w_iB_i^2(s_i)B_j^2(h_i)$, the interpolation properties are preserved, and we can use them to adjust the fullness of the patch. The scalar weights $w_i$ serve to set the strength of these control points (we use a default value of $\frac{1}{3}$). The simplest placement of these interior controls comes from a generalization of the biquartic case, i.e., $C_{2,2} = C_{2,1} + (C_{2,1} − C_{2,0})$.

In the following examples we have not edited the newly supplemented interior control points, just set them to their default positions, so the changes we show are purely due to the enrichment of the control point structure. The multi-sided patch in Figure 10 has a long concave boundary. The default setting produced some unexpected saddle in the middle (Fig. 10a). Adding a new layer according to the algorithm of CGB patches produced Fig. 10b. Adding a single interior control point (shown in black) by the above algorithm yields Fig. 10c, while increasing the weight of this control point we obtain Fig. 10d.

The second example – Figure 11 – shows the surface of a helmet. The given “narrow” ribbons produced a surface portion in the interior with a relatively low curvature. When we added interior control points, the fullness of the patch significantly improved, and a more natural curvature distribution was obtained. In Figure 12a the contour lines of a biquartic basis function are shown over the curved domain of the helmet. The sum of all basis functions coming from the interior control points is depicted in Figure 12b. More test examples are presented in Section 7.

6. Split boundaries

A curved domain may have convex and concave corners, but at the same time, this scheme is capable of handling smooth corners, as well. In other words, a boundary curve can be split into segments, or adjacent boundaries can be joined smoothly, if necessary. There are two basic issues where split boundaries can be beneficial.

First, it is a well-known deficiency of Bézier curves, that control points may oscillate unexpectedly, when a curve with high curvature variations needs to be created (Figure 13). As the number of control points is raised, the degree of basis functions is elevated and the “strength” of individual control points gradually vanishes; thus it may be necessary to position them far from the curve being defined. (We can place the control points closer to the boundary, if we use B-spline curves, but then we will lose $C^0$ continuity, and the patch will inherit the continuity of the boundaries.) For CD patches, we prefer reasonably tight control structures and a “natural” control point placement of the ribbons.
(a) Original model
(b) New layer added
(c) Interior control point added
(d) Interior control point with a larger weight

Figure 10: Interior shape control.

Figure 11: Surface enhanced by adding interior control points.

(a) Basis function associated with a single interior control point
(b) Sum of basis functions associated with all interior control points

Figure 12: Interior control point basis functions.
The second issue comes from our local parameterization (Figure 14). The endpoints of the $h_i$ isolines of side $i$ are evenly distributed on the adjacent sides $i - 1$ and $i + 1$. Assume that $i - 1$ is a long boundary, then the $h_i = 0.5$ isoline will start at the halving point of side $i - 1$, far from the $i$-th side. As a consequence, the basis functions associated with side $i$ are elongated, and the control points of ribbon $i$ may create undesirable shape artifacts at places far from the $i$-th boundary. Applying a splitting operation shortens the length of the adjacent domain segment, and thus the effect of ribbon $i$ is favorably “dragged” closer to the $i$-th boundary.

Figure 13 shows a simple example of split boundaries. The top boundary curve of the surface is a degree 8 Bézier curve (see the left image). We can split this curve and approximate it by two quintic segments with a tighter control structure; see the surface on the right side. Both surfaces are smooth, but the Gaussian curvature map shows that the original surface may have some shape artifacts due to the above parameterization problem, which do not occur in the new surface. In Figure 14, the difference between the distributions of the $h_i$ isolines can be compared. The sum of the basis functions $\sum B_{jk}(s_i, h_i)$ and the weighted basis functions $\sum \mu_i'(h_{i-1}, h_i, h_{i+1})B_{jk}(s_i, h_i)$ that belong to side $i$ are also shown. These level curves vary from 1 to 0 in steps of 0.02 over the curved domain.

7. Discussion and case studies

In this section we have selected a few interesting test objects to demonstrate the capabilities of CD Bézier patches.

7.1. Test example 1

We start with a simple sheet metal part, represented by a 9-sided patch (Figure 15). The highly curved boundary on the left consists of two smoothly connected curve pieces. Here the curvature maps illustrate that the former parameterization methods produce folded or wrinkled surfaces, while a nice shape is obtained when a curved domain is used.

7.2. Test example 2

Our second example in Figure 16 is a setback-type vertex blend (Várady and Hoffmann, 1998). Four approximate rolling ball blends come together at a vertex and a smooth connection is needed to fill the hole (Fig. 16a). The radii and the setbacks are different. The bottom concave boundary has been split into two segments, so altogether we have nine boundaries. All the arcs are represented by quartic ribbons, as shown in Figure 16b. The proper fullness can be achieved by using interior control points, see curvature map (Fig. 16c), contouring (Fig. 16d) and isophotes (Fig. 16e).

7.3. Test example 3

We show a 12-sided test surface with four concave boundary curves in Figure 17, with shading, curvature map and planar slices. It was a surprise that this patch can also be produced using a CGB polygonal domain, although the mapping becomes highly distorted, as shown in Figure 18. All the angles in 3D are of 90 degrees, and the domain algorithm had to increase these angles to obtain a convex polygon.
(a) Without splitting the long boundary curve
(b) When the long boundary curve is split

Figure 14: Left: $h_i$ isolines; Middle: sum of the basis functions of side $i$; Right: sum of the weighted basis functions of side $i$.

Figure 15: Surface with a two-segment boundary: GB – CGB – CD domains.
We quantify distortion of the map between the domain and the surface as the deviation of the map from being a local isometry. The As-Rigid-As-Possible (ARAP) energy is widely used, but it is known to be biased towards shrinking and near-degenerate maps (Rabinovich et al., 2017). Instead, we use the Symmetric ARAP energy, which does not suffer from such bias (Shtengel et al., 2017). More concretely, given a Jacobian matrix $J_T$, corresponding to the local linear map between a domain triangle $T$ and its 3D image, we measure its distortion as follows:

$$E_{SARAP}(J_T) = (\sigma_1(J_T) - 1)^2 + \left( \frac{1}{\sigma_2(J_T)} - 1 \right)^2,$$

(16)

where $\sigma_1, \sigma_2$ are the maximal and minimal singular values of $J_T$.

The strong parametric distortion is drastically reduced when the curved domain patch is used. Compare the average and maximum distortion values shown in Figure 18.
7.4. Test example 4

The sketch of an iron model is shown in Figure 19a. We have picked a multi-sided patch, whose base surface is a 6-sided patch with two concave boundaries (Fig. 19b). Contours and isophotes are shown in Figures 19c and 19d. Figures 19e and 19f illustrate that it is necessary to use interior control points, as the curvature in Figure 19f is more evenly distributed. The model is also constrained by an internal hole loop positioned in the middle of the base surface. Now we have 10 boundary ribbons altogether, that define the shape depicted in Figure 19g. The domain is shown in Figure 19h, and a related curvature map in Figure 19i.

7.5. Test example 5

Here we combine a base patch and two hole loops. The base patch is 8-sided, and the hole loops are composed of four smoothly connected boundary segments. In the first case the ribbon control points are set to connect to a plane parallel to the XY plane, in the second case a steep cross-derivative direction was chosen. Figure 20 shows the input and the CD Bézier surfaces generated. The contours and the curvature maps indicate natural smooth transitions.

7.6. Implementation notes

In our project we have implemented harmonic coordinates in the following way. Since no closed-form solution exists, we discretize the domain. First we need to sample the boundaries – in our implementation uniform sampling is applied, but it may be interesting to see how a curvature-based sampling would modify the coordinates. Then the domain is tessellated into a triangle mesh using the Triangle library (Shewchuk, 1996), and we employ piecewise-linear finite elements, transforming the boundary-value problem in Eq. (15) into a sparse linear system involving the cotangent Laplace matrix (Pinkall and Polthier, 1993). Sparse direct solvers are very efficient for problem sizes typical in our setting; in our implementation we used the Eigen library (Guennebaud et al., 2010).

Conclusion and future work

In this paper we have tried to extend the limits of multi-sided parametric patches composed of Bézier ribbons. We proposed using curved domains which facilitate complex patch configurations having concave vertices, concave boundaries and internal hole loops. We have discussed related problems of domain generation, parameterization, basis functions, interior control points and split boundaries. We believe this scheme opens a new variety of genuine multi-sided surface patches. The basis functions are rational polynomials, so the patch is $C^\infty$ continuous.

At the same time, there are plenty of difficult questions for future research. Parameterization is a crucial issue; here we have derived our local coordinates from generalized barycentric coordinates that remain positive, but instead of harmonic coordinates, we would prefer alternatives that can be explicitly evaluated, even those that do not have the vertex reproduction property, as it is not exploited by our patch. Reparameterization is also a possible option.

Editing the control points requires experience, and a good default placement of the interior control points is important. For the convex GB patches there is an automatically defined, unified and refinable control point structure,
Figure 19: Iron surface with an interior hole loop.

Figure 20: Two multi-loop surfaces. Left: ribbons; Middle: contouring; Right: curvature map.
that ensures an even distribution of the basis functions, and produces nice shapes. For CD patches we search for a similar structure; for configurations with interior loops this seems to be challenging. Using B-spline interpolants instead of Bézier ribbons is a topic of ongoing research. Finally, it may be interesting to combine Sabin’s transfinite concept and our curved domains, where the points of the surface are defined by some elastic energy.

Acknowledgements

This project has been supported by the Hungarian Scientific Research Fund (OTKA, No.124727) and the National Research, Development and Innovation Fund (TUDFO/51757/2019-ITM, Thematic Excellence Program). All images were generated by the Sketches prototype system (ShapEx Ltd, Budapest); the authors acknowledge the important programming contribution of György Karikó.

References

DeRose, T., Meyer, M., 2006. Harmonic coordinates, in: Pixar Technical Memo 06-02, Pixar Animation Studios.
Engvall, L., Evans, J.A., 2018. Mesh quality metrics for isogeometric Bernstein–Bézier discretizations. arXiv preprint arXiv:1810.06975.
Gortler, S., Gotsman, C., Thurston, D., 2006. Discrete one-forms on meshes and applications to 3D mesh parameterization. Computer Aided Geometric Design.
Gray, A., Abbena, E., Salamon, S., et al., 2006. Modern Differential Geometry of Curves and Surfaces with Mathematica. 3rd ed., CRC press.
Guennebaud, G., Jacob, B., et al., 2010. Eigen. http://eigen.tuxfamily.org.
Hormann, K., Sukumar, N. (Eds.), 2017. Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics. CRC Press.
Kato, K., 1991. Generation of n-sided surface patches with holes. Computer-Aided Design 23, 676–683.
Kato, K., 2000. N-sided surface generation from arbitrary boundary edges, in: Curve and Surface Design. Vanderbilt University Press. Innovations in Applied Mathematics, pp. 173–181.
Loog, C.T., DeRose, T.D., 1989. A multised generalization of Bézier surfaces. ACM Transactions on Graphics 8, 204–234.
Lu, J., 2009. Circular element: Isogeometric elements of smooth boundary. Computer Methods in Applied Mechanics and Engineering 198, 2391–2402.
Martin, S., Kaufmann, P., Botsch, M., Wicke, M., Gross, M., 2008. Polyhedral finite elements using harmonic basis functions, in: Computer Graphics Forum, Wiley Online Library, pp. 1521–1529.
Pinkall, U., Polthier, K., 1993. Computing discrete minimal surfaces and their conjugates. Experimental Mathematics 2, 15–36.
Rabinovich, M., Poranne, R., Pandolfo, D., Sorkine-Hornung, O., 2017. Scalable locally injective mappings. ACM Transactions on Graphics (TOG) 36, 37a.
Rubtsov, R.M., 2008. Boundary element formulation of harmonic coordinates. Technical Report, Purdue University.
Sabin, M., 1998. Further transfinite surface developments, in: Mathematics of Surfaces VIII, Information Geometers. pp. 161–173.
Salvi, P., Várady, T., 2018. Multi-sided Bézier surfaces over concave polygonal domains. Computers & Graphics 74, 56–65.
Schafer, S., 2017. Multi-sided patches via barycentric coordinates, in: Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics. CRC Press, chapter 8, pp. 135–146.
Shewchuk, J.R., 2002. What is a good linear element? Interpolation, conditioning, and quality measures, in: Proceedings of the 11th International Meshing Roundtable, Sandia National Laboratories.
Shewchuk, J.R., 1996. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator, in: Lin, M.C., Manocha, D. (Eds.), Applied Computational Geometry: Towards Geometric Engineering, Springer-Verlag, volume 1148 of Lecture Notes in Computer Science, pp. 203–222. https://www.cs.cmu.edu/~quake/triangle.html.
Shenkel, A., Poranne, R., Sorkine-Hornung, O., Kovalsky, S.Z., Lipman, Y., 2017. Geometric optimization via composite majorization. ACM Transactions on Graphics (TOG) 36, 38.
Várady, T., Hoffmann, C.M., 1998. Vertex blending: problems and solutions, in: Conference on Mathematical Methods for Curves and Surfaces, Vanderbilt University, pp. 501–527.
Várady, T., Salvi, P., Karikó, G., 2016. A multi-sided Bézier patch with a simple control structure. Computer Graphics Forum 35, 307–317.
Várady, T., Salvi, P., Kovács, I., 2017. Enhancement of a multi-sided Bézier surface representation. Computer Aided Geometric Design 55, 69–83.
Weber, O., 2017. Planar shape deformation, in: Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics. CRC Press, pp. 109–133.