Computers & Graphics (2018)

Contents lists available at ScienceDirect

Computers & Graphics

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

Multi-sided Bézier surfaces over concave polygonal domains

Péter Salvi*, Tamás Várady

Budapest University of Technology and Economics, H-1111 Budapest, Műegyetem rkp. 3, Hungary

ARTICLE INFO

Article history:
Received May 9, 2018

Keywords: multi-sided surfaces, Bézier surfaces, concave domains

ABSTRACT

A new multi-sided, control point based surface representation is introduced, based on the Generalized Bézier patch [1]. While the original surface is based on convex polygonal domains and a specific, uniform arrangement of control points, the new construction permits domains with concave angles and supports a more general control point structure, where independent "half-Bézier" interpolants, or *ribbons*, are blended together. The ribbons may have arbitrary degrees along the boundary and also in the cross-derivative direction; the related control points ensure tangent- or curvature-continuous connection to adjacent quadrilateral Bézier patches and permit shape editing and optimization, when needed.

The surface comprises four components: (i) a concave domain generated from a 3D loop of boundary edges, (ii) half-Bézier interpolants, (iii) parameterizations that cover the full domain for each interpolant, and (iv) blending functions that guarantee both Bézier-like behavior along the boundaries and a smooth, C^∞-continuous composition in the interior of the patch. Editing concave Bézier patches using additional control points is also discussed. A few interesting test examples illustrate the benefits of the method.

1. Introduction

Multi-sided surfaces are often needed in geometric design, where a complex patchwork may contain non-four-sided surface elements, as well. These must smoothly connect to adjacent quadrilaterals or other multi-sided patches, satisfying positional and cross-derivative constraints along their given boundaries, and must ensure a natural blend in the interior. The constraints may be given in the form of vector functions (transfinite interpolation) or in control point based form, suitable to match adjacent patches also specified by control grids. Typical applications include curve network based design, hole filling and vertex blending.

The representation of multi-sided free-form surfaces over a concave domain is still a "terra incognita". Operations to define concave surfaces are rarely available in practice – usually trimmed tensor-product surfaces are used instead, but it is hard to directly define and modify trimming curves. Alternatively, additional subdivision curves may be inserted to split concave areas into smaller, convex ones, but this may lead to underdetermined entities, an extra burden for designers.

Our goal is to create concave multi-sided surfaces that possess features similar to those of ordinary patches. We wish to edit general topology curve networks having non-convex regions, as well, while avoiding weakly defined, artificial 3D curves that are only needed to make patch generation possible. Simple examples are shown in Figure 1. The difficulty is due to the fact that the usual techniques of convex multi-sided surfacing cannot be directly applied. Related problems include alternative ways of (i) defining the domain, (ii) the interpolants, (iii) the parameterization, and (iv) the blending functions.

In a recent publication a new representation, called Gener-

*Corresponding author. E-mail: salvi@iit.bme.hu

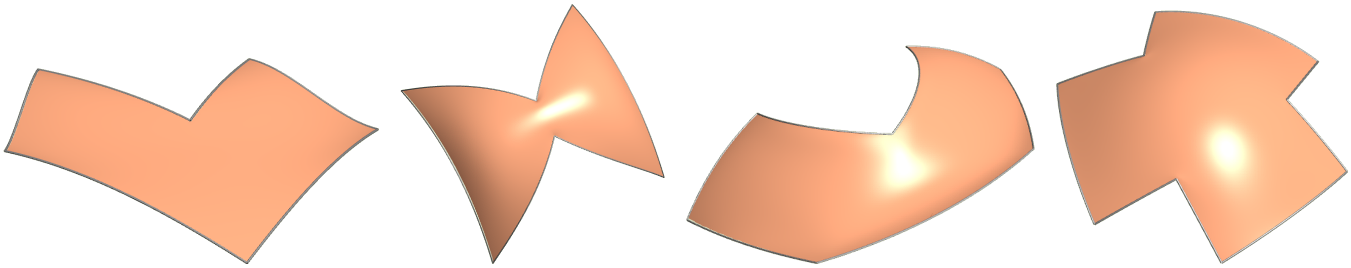


Figure 1: Simple concave patch configurations.

alized Bézier (GB) patches, was introduced [1]. This combines n side interpolants specified by rectangular grids of control points, and eventually a uniform control structure similar to a spider web is produced. GB patches satisfy our basic requirements for smooth connections and a smooth natural interior. The scheme offers additional internal control points for shape editing or optimization.

The current concave surfacing project heavily builds on this concept by having recognized that the GB patch is a specific variant of a more general surface representation, which permits the combination of "half-Bézier" interpolants, or *ribbons*, with arbitrary degrees. The essence of the concept is that the control points of adjacent ribbons do not necessarily need to be identical and in this way further design freedom can be obtained. In particular, concave corners can be represented by disjoint sets of control points, as it will be explained later.

The paper is structured in the following way. First we discuss related previous work, then in Section 3 we summarize the basic construction of GB patches and the concept of combining half-Bézier ribbons. The details of the actual construction will be discussed in Section 4, together with options to edit the interior of the surface. Test examples in Section 5 and suggestions for future work conclude the paper.

2. Previous work

There is an extensive literature on multi-sided patches; here we are going to concentrate on representations that can handle concave configurations. These can be categorized in the following way:

Multivariate Bernstein–Bézier patches. S-patches [2] represent a perfect generalization of Bézier triangles for an arbitrary number of sides. It has been pointed out recently, that these surfaces work over concave domains, as well, assuming we have a suitable parameterization with non-negative barycentric coordinates [3, 4]. S-patches – and their higher-dimensional generalizations – can be used in various applications, such as shape deformation; however, in an interactive design context, difficulties arise due to their extremely complex control point structure, even in the case of low-degree boundaries. See for example test object #1 in Section 5. Ensuring smooth connections between adjacent S-patches is also hard, although a method to set G^1 continuity has been proposed recently [5].

Transfinite interpolation methods. The pioneering work of Kato [6, 7] may be the one closest to our goals, and it also supports internal holes. It uses singular blending functions, and singular side interpolants at the concave corners. A comparison with our method is shown in test object #2 of Section 5. Sone et al. [8] concatenate adjacent sides with concave angles into a single edge; then the best correspondence is chosen by a metric based on the isolines of the resulting surface. An interesting idea with many open questions. Note that loops with more than two concave parts are not supported.

Discrete methods. The curvature-aligning technique of Pan et al. [9] and the method of Stanko et al. [10] based on biharmonic normal propagation and mean curvature fairing both fill arbitrary curve networks with a mesh. These create aesthetic, smooth surfaces, but lack the advantages of control point based schemes.

Splitting methods. A practical solution is to split concave loops into multi-sided convex patches by means of additional subdividing curves in 3D. Then the adjacent sub-surfaces need to be smoothly connected, possibly with curvature continuity, as discussed e.g. in [11]. A further restriction may be to permit only quadrilaterals, when industry-standard tensor-product patches are stitched together. While this problem was thoroughly investigated earlier [12], there is still interesting research related to composite multi-sided surfaces [13, 14]. There are special cases where the subdividing curves can be auto-generated (e.g. central splitting for convex patches), but in general the construction is shape-dependent, and updating these curves can be an extra burden on the users. Another drawback of the splitting approach is decreased continuity.

Other methods. Trimming is the prevalent solution to this problem in today's CAD systems, but it is approximative, and inherently asymmetric. Subdivision of concave faces may create unwanted protrusions, so in practice these are split into convex regions.

Our work is an extension of the Generalized Bézier patch [1]. In addition to the original paper, we have used some ideas from a follow-up article [15], as well.

3. The Generalized Bézier patch

In this section we summarize the formulation of the GB patch, as described in [1], then reinterpret it in a way that facilitates the extension to concave domains.

3.1. Basic concept

The GB patch is defined by a multi-sided control grid; it is a generalization of quadrilateral Bézier patches both in terms of control structure and behavior along the boundaries. A degree

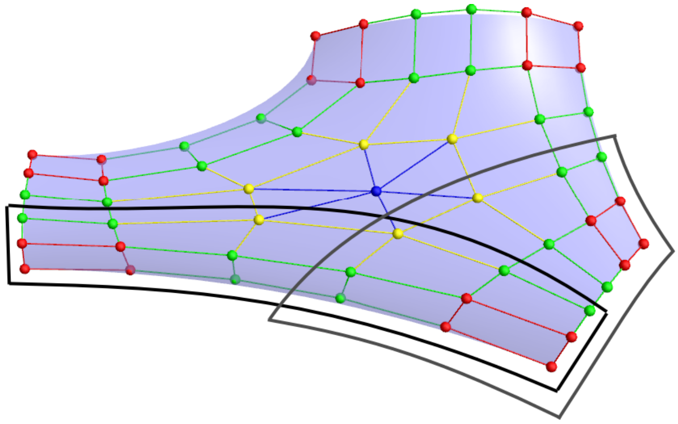


Figure 2: Control structure of the GB patch. Black frames are shown around the associated layers of two sides.

d patch interpolates n boundary curves given in Bézier form; for each side there exists a half-Bézier ribbon of $(d+1)\times\lceil d/2\rceil$ control points. (We will also use the notation $l = \lceil d/2\rceil$ for the number of *layers* or control rows.) For example, Figure 2 shows the control points of a six-sided quintic patch with three layers. It can be seen, that the control point structure associated with a given side is identical to that of a quadrilateral Bézier patch (see black frame).

For a G^1-compatible patch, the cross-derivatives along the boundaries are determined by the first two rows of control points. Coloring shows a classification: at the corners there are four *corner control points* (red), that are associated with two sides. Between these, there may be *ribbon control points* (green), associated exclusively with one side. There are *interior control points* (yellow) in the middle, that can be placed automatically by a special degree elevation algorithm. Finally, there is a single *central control point* (blue), that is responsible for the middle of the patch.

The surface is defined over a regular polygonal domain in the (u, v) plane. Each side of the domain has its own local parameterization (s_i, h_i), with mappings computed from the Wachspress barycentric coordinates $\{\lambda_1, ..., \lambda_n\}$ (see e.g. [16]):

$$s_i = \lambda_i / (\lambda_{i-1} + \lambda_i), \qquad h_i = 1 - \lambda_{i-1} - \lambda_i. \tag{1}$$

The *side parameter* s_i varies linearly on side i between 0 and 1, while the *distance parameter* h_i 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).

The GB patch is a composition of Bézier ribbons, given in the following form:

$$R_i(s_i, h_i) = \sum_{j=0}^{d}\sum_{k=0}^{l-1} C^i_{j,k}\cdot\mu^i_{j,k}B^i_{j,k}(s_i, h_i). \tag{2}$$

The indexing scheme is side-based, i.e., $C^i_{j,k}$ refers to the j-th control point in the k-th row of the i-th side. For an ordinary Bézier ribbon the control points $C^i_{j,k}$ would be multiplied by bivariate Bernstein functions $B^i_{j,k}(s_i, h_i) = B^d_j(s_i)\cdot B^d_k(h_i)$. For the GB patch, however, they need to be multiplied by additional rational weight functions $\mu^i_{j,k}$, as well. As it is explained in Section 4.2, $\mu^i_{j,k}$ depends on a pair of distance parameters: h_i and h_{i-1} or h_{i+1}. This term guarantees that R_i reproduces the ordinary Bézier ribbons 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.

The patch equation is simply given as

$$S_{GB}(u, v) = \sum_{i=1}^{n} R_i(s_i, h_i) + C_0\cdot B_0(u, v), \tag{3}$$

where C_0 is the central control point, and B_0 the *weight deficiency*, defined as

$$B_0(u, v) = 1 - \sum_{i=1}^{n}\sum_{j=0}^{d}\sum_{k=0}^{l-1}\mu^i_{j,k}B^i_{j,k}(s_i, h_i), \tag{4}$$

ensuring that the sum of basis functions equals to 1, and thus the convex combination property is also satisfied.

As shown earlier, many of the control points occur twice in consecutive Bézier ribbons. Generally speaking, $C^i_{j,k} = C^{i-1}_{d-k,j}$ for $j < \frac{d}{2}$ and $C^i_{j,k} = C^{i+1}_{k,d-j}$ for $j > \frac{d}{2}$. In our quintic patch this gives (e.g.) $C^i_{5,0} = C^{i+1}_{0,0}$, $C^i_{5,1} = C^{i+1}_{1,0}$, $C^i_{4,0} = C^{i+1}_{0,1}$ and $C^i_{4,1} = C^{i+1}_{1,1}$.

In the original paper [1] it is explained how a degree d GB patch can be composed from 2-layer (G^1) or 3-layer (G^2) interpolants of various degrees. The algorithm performs a degree reduction/elevation operation to match the highest degree $d = \max(d_1, \ldots, d_n)$, and increases the number of layers accordingly to $l = \lceil d/2\rceil$. Several new interior control points are computed in the process and incorporated into the final structure. Thus a GB patch not only interpolates positional and cross-derivative constraints from the Bézier interpolants, but also offers additional degrees of freedom for shape design and optimization. It smoothly connects to adjacent quadrilateral and multi-sided Bézier surfaces, i.e., if the corresponding Bézier ribbons satisfy G^1 or G^2 continuity, the full GB patch will ensure this in the same way.

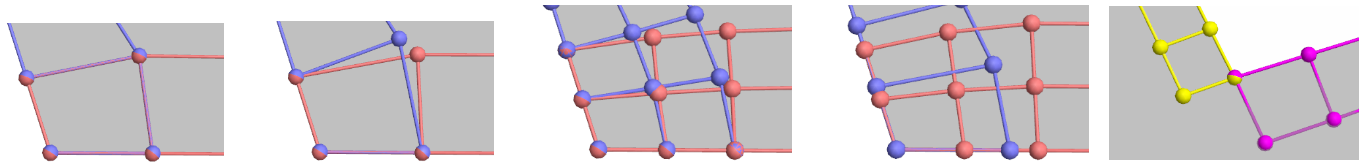
3.2. GB patches reinterpreted

The original GB patch is based on two assumptions:

1. The half-Bézier ribbons have a uniform structure, having the same degree d, and the same number of layers $l = \lceil d/2\rceil$.
2. The corresponding control points of adjacent ribbons are identical at the common corners.

We have found that these restrictions can be lifted, and the GB patch can be defined in a more general way, permitting (i) arbitrary degrees d_i and layers l_i, and (ii) independent grids of control points. The control points may coincide fully or partially with those of the neighboring ribbons, or not at all. In any case, the weighting functions $\mu^i_{j,k}$ guarantee that the half-Bézier ribbon behaves like an ordinary Bézier patch on its base edge, and vanishes on all other sides of the domain.

We show a few simple examples, and indicate how many control points with different positions remain, see Figure 3. (a) The



(a) $d_i = d_{i+1} = 3$, $l_i = l_{i+1} = 2$ (b) same as in (a) (c) $d_i = d_{i+1} = 5$, $l_i = l_{i+1} = 3$ (d) $d_i = 4$, $d_{i+1} = 5$, $l_i = 2$, $l_{i+1} = 3$ (e) same as in (a)

Figure 3: Control point coincidences at a corner.

control points fully coincide ($4 + 4 \to 4$). (b) There are two different twist control points ($4 + 4 \to 5$), i.e., the mixed partial derivatives of the adjacent ribbons are not compatible. Our scheme will produce rationally weighted twist vectors, in the same way as it was suggested by Gregory [17]. (c) Partial coincidence of 5 control points ($9+9 \to 13$). (d) Degrees and layers differ, and only the corner control point is shared ($9 + 4 \to 12$). (e) Same degrees and layers as in (a), but all the control points except the corner are fully separated ($4 + 4 \to 7$). This last structure facilitates the definition of patches with concave corners, see the next section.

4. Concave GB patch

As we start dealing with concave patches, various difficulties arise, since the methods of convex patches cannot be applied any longer. Conventional side interpolants typically sweep from the tangent vector of the previous boundary curve to the tangent of the next curve, but in the concave case these create unwanted twists: the interpolants must point towards the interior of the patch at all times, see Figure 4. This means that at a concave corner the interpolant must preferably match an inverted tangent vector. A 3D example is given in Figure 5, showing that the patch would turn inside out if we used the convex approach.

There are also other issues, as almost all components of the original formulation – blending functions, domain, parameterization – need revision. In the following we will examine these one by one. Since the control structure is no longer uniform, we need alternative means for interior control. Increasing the number of layers is fairly straightforward, but we will also explore some other possibilities at the end of the section.

4.1. Ribbons

A half-Bézier ribbon in the concave GB patch has $(d_i + 1) \times l_i$ control points, where d_i and l_i are independent values. This means that the degrees of this interpolant are d_i in the edgewise direction, and $2l_i - 1$ in the cross-boundary direction. Adding a new layer to a ribbon consequently adds two to its cross-boundary degree, so all control points need to be repositioned. The new positions $\hat{C}^i_{j,k}$ can be computed by applying the standard Bézier degree elevation algorithm twice, except for the newly inserted control row, which appears at a linear extension of the ribbon's last segment. Assuming that we are changing l_i to $l_i + 1$, these are defined as

$$\hat{C}^i_{j,l_i} = C^i_{j,l_{i-1}} + \left(C^i_{j,l_{i-1}} - C^i_{j,l_{i-2}}\right)\frac{1}{2l_i + 1}. \quad (5)$$

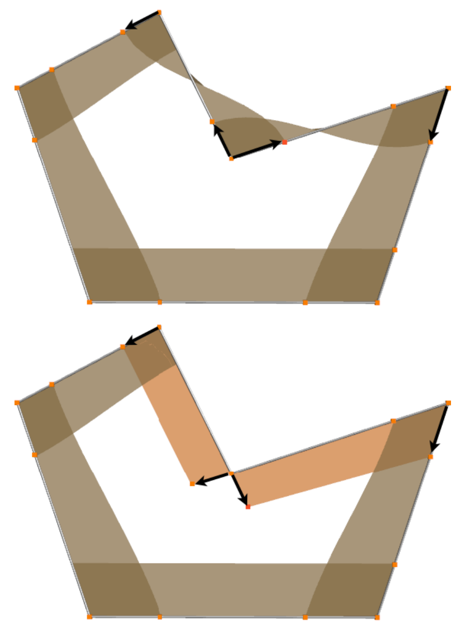


Figure 4: Default (top) and enhanced ribbons (bottom) at a concave corner.

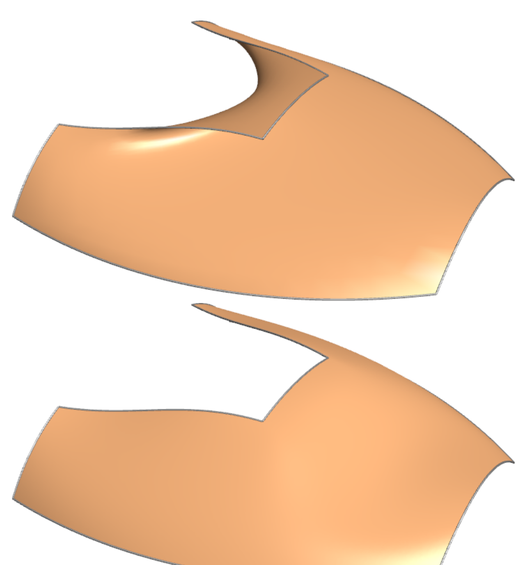
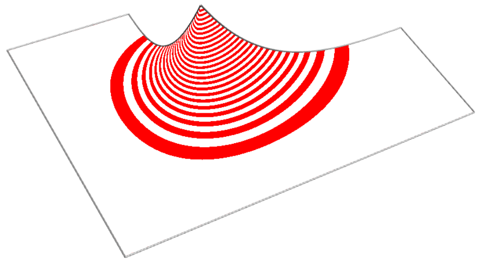
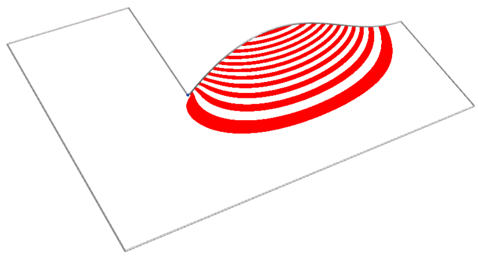


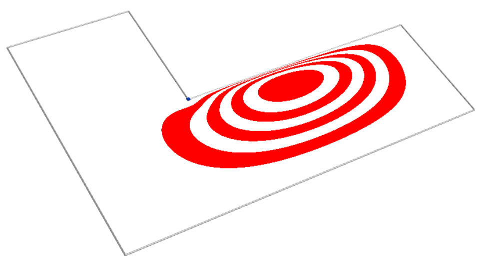
Figure 5: Patch with default (top) and enhanced ribbons (bottom).



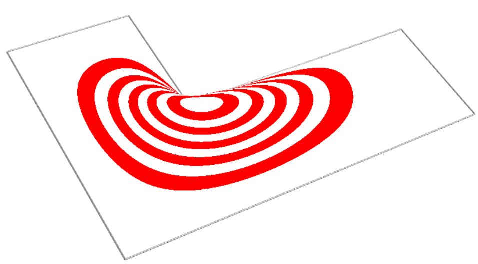
(a) $\mu^1_{0,0}B^3_0(s_1)B^3_0(h_1)$



(b) $\mu^1_{1,0}B^3_1(s_1)B^3_0(h_1)$



(c) $\mu^1_{1,1}B^3_1(s_1)B^3_1(h_1)$



(d) $\mu^1_{0,1}B^3_0(s_1)B^3_1(h_1)$

Figure 6: Blending functions near the concave corner of an L-shaped domain.

Similarly to the convex GB patches, this sort of degree elevation retains the boundaries and the cross derivatives, but it may slightly change the interior of the original surface.

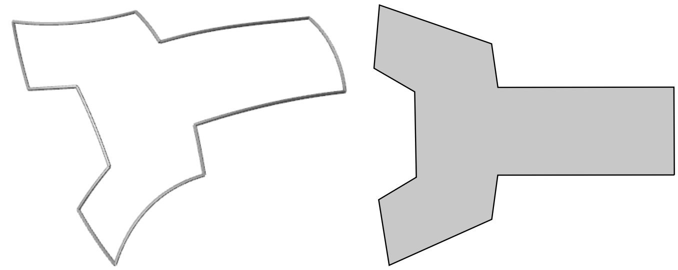
4.2. Blending functions

The bivariate Bernstein polynomial associated with a control point $C^i_{j,k}$ is

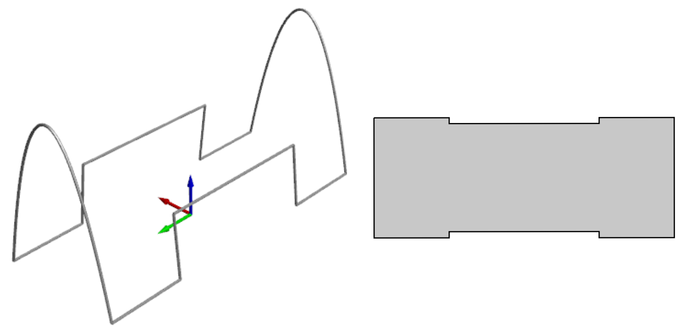
$$B^i_{j,k}(s_i, h_i) = B^{d_i}_j(s_i)B^{2l_i-1}_k(h_i), \tag{6}$$

which is then multiplied by the scalar function $\mu^i_{j,k}$, as it was mentioned earlier in Section 3. We define it as follows:

$$\mu^i_{j,k} = \begin{cases} \alpha_i = h^2_{i-1}/\left(h^2_{i-1}+h^2_i\right), & \text{when } 2j < d, \\ 1, & \text{when } 2j = d, \\ \beta_i = h^2_{i+1}/\left(h^2_{i+1}+h^2_i\right), & \text{when } 2j > d. \end{cases} \tag{7}$$



(a) Example where projection works



(b) Example where projection fails

Figure 7: Domains generated by projection.

This is somewhat simpler than the one in [15], and more suitable for handling non-coinciding control points. A ribbon is given as

$$R_i(s_i, h_i) = \sum_{j=0}^{d_i}\sum_{k=0}^{l_i-1} C^i_{j,k}\cdot\mu^i_{j,k}B^i_{j,k}(s_i, h_i). \tag{8}$$

Concave GB patches do not have such a central control point that would remove the weight deficiency (see also Section 4.5), so we propose to normalize the weights with the sum

$$B_{\mathrm{sum}}(u, v) = \sum_{i=1}^{n}\sum_{j=0}^{d_i}\sum_{k=0}^{l_i-1}\mu^i_{j,k}B^i_{j,k}(s_i, h_i) \tag{9}$$

in order to maintain convex combination. Figure 6 shows some cubic blending functions over an L-shaped domain.

Finally, we arrive at the patch equation

$$S(u, v) = \frac{1}{B_{\mathrm{sum}}(u, v)}\cdot\sum_{i=1}^{n} R_i(s_i, h_i). \tag{10}$$

4.3. Domain generation

A simple and natural method for generating a concave domain is to project and connect all vertices on a best fit plane [7], see Figure 7a. While this works for configurations that are relatively flat, there are many models with highly curved boundaries where this method cannot be applied, see Figure 7b. In the following we present an algorithm that is an extension of a heuristic procedure used for generating convex domains [18].

Let p_i denote the vertices of the domain ($i = 1\ldots n$), e_i the length of the edge $\overline{p_{i-1}p_i}$, and ϕ_i the angle at p_i. Similarly, in 3D, let $C_i(u)$ denote the Bézier curve defined by the control points $C^i_{j,0}$, E_i the arc length of C_i, and Φ_i the angle between the tangents $-C'_i(1)$ and $C_{i+1}(0)$, see Figure 8. We wish

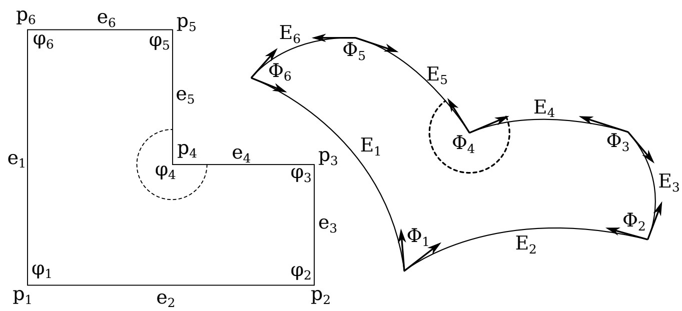


Figure 8: Notations in the domain and in 3D.

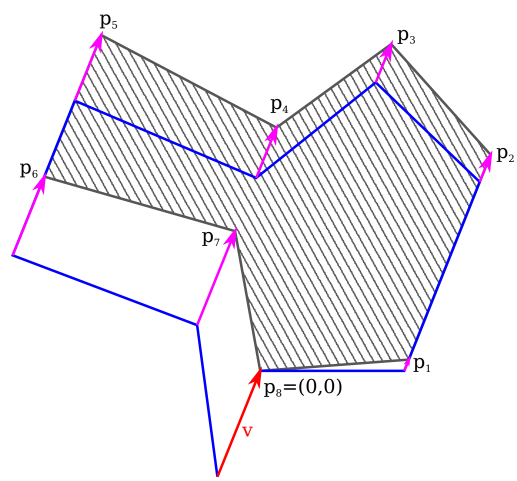


Figure 9: Domain generation.

to generate a domain such that the sizes of the edge lengths and angles are distorted minimally, i.e., $\sum_i \left(e_i - c_{\text{length}} E_i\right)^2$ and $\sum_i \left(\phi_i - c_{\text{angle}} \Phi_i\right)^2$ are minimal, where c_{length} and c_{angle} are constants.

The algorithm works as follows. We set $e_i = E_i$, and normalize the angles such that they sum to $(n-2)\pi$:

$$\phi_i = \Phi_i \cdot \frac{(n-2)\pi}{\sum_j \Phi_j}. \tag{11}$$

Then we can draw a domain by starting from the origin, drawing the edge e_1 in the direction of the x axis, then turning left at a $\pi - \phi_1$ angle, drawing e_2 etc. At the end we will not necessarily get back to the origin, resulting in an open polygon $\{\hat{p}_i\}$. The deviation is $v = (0,0) - \hat{p}_n$, which is then distributed between all points:

$$p_i = \hat{p}_i + v \cdot \frac{\sum_{j=1}^{i} e_j}{\sum_{j=1}^{n} e_j}. \tag{12}$$

Figure 9 shows the stages of the algorithm, and Figure 10 shows the domain for the same input curves as in Figure 7b.

This algorithm is also applicable in the concave case, but it does not guarantee that the resulting domain is free of self-intersections, and sometimes can produce polygons with very narrow “bottlenecks”, see the leftmost image in Figure 11. We validate the domain by requiring that the minimum distance between any two segments e_i and e_j should be larger than a parameter $\delta_{\min}$. (This is set to 10% of the domain's bounding box

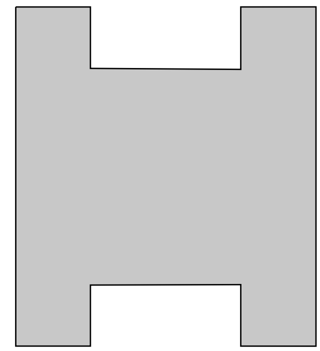


Figure 10: A good domain for the curve loop in Figure 7b.

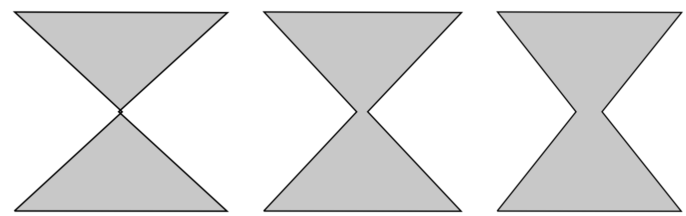


Figure 11: Two steps of widening the domain.

axis in all our examples.) If a domain is invalid, we enlarge the $\phi_i^0 = \phi_i$ angles and regenerate the polygon, iterating until it becomes valid. The enlargement is done by scaling the convex angles by a constant factor σ_{angle} (1.1 in our examples), and distributing the surplus evenly amongst the concave angles:

$$\phi_i^{k+1} = \begin{cases} \sigma_{\text{angle}} \cdot \phi_i^k, & \text{when } \phi_i \leq \pi, \\ \phi_i^k - \phi_{\text{extra}}^k / n_{\text{concave}}, & \text{when } \phi_i > \pi, \end{cases} \tag{13}$$

where n_{concave} is the number of concave angles, and

$$\phi_{\text{extra}}^k = (\sigma_{\text{angle}} - 1) \cdot \sum_{\phi_j^k \leq \pi} \phi_j^k. \tag{14}$$

An example is shown in Figure 11.

It is theoretically possible that the above algorithm fails, when some of the self-intersections cannot be eliminated. In that case, no patch is created and subdivision needs to be performed. However, our experiments show that the proposed method is fairly reliable, as we have not come across failing examples.

4.4. Parameterization

The parameterization presented in Section 3 was based on Wachspress coordinates, which cannot be used within a concave polygon. Mean value coordinates [16] are the most common alternative for the concave case, but these are nonnegative only in the kernel of the domain. We have chosen to use harmonic coordinates [19] instead, which are much more computation-intensive, but they are nonnegative inside the whole polygon. The local coordinates s_i and h_i are computed from these barycentric coordinates as before – examples are shown in Figure 12.

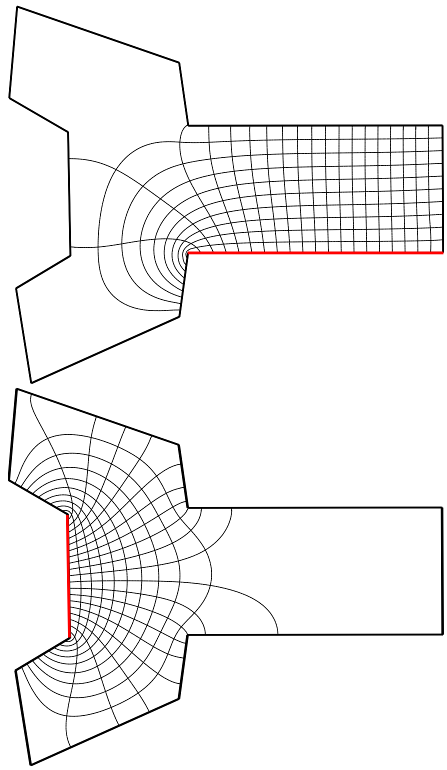


Figure 12: Local parameterization in a concave domain. The base sides are shown in red.

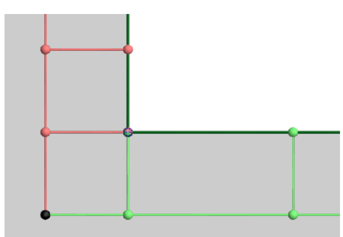


Figure 13: Concave corner control point construction.

4.5. *Additional control*

The concave patch as defined above may become flat in surface areas relatively distant from the boundaries. This effect gets stronger when n is large, and it may harm shape quality in the vicinity of concave corners. We can add extra control points and associated basis functions to change the fullness of the surface – two such constructions are presented below.

Concave corner control points

We would like to affect the shape around the concave corners, so it is a natural choice to formulate a blending function as the extension of the second control rows of the adjacent sides (colored black in Figure 13). Multiplying the cross-directional parts of the blending functions yields

$$\beta_i B_1^3(h_i) \cdot \alpha_{i+1} B_1^3(h_{i+1}). \tag{15}$$

This is a suitable term, as it vanishes on all sides of the domain. It takes a maximum value of $\frac{4}{81}$ at $h_i = h_{i+1} = \frac{1}{3}$. (For simplicity's sake here we discuss concave corner control points only for

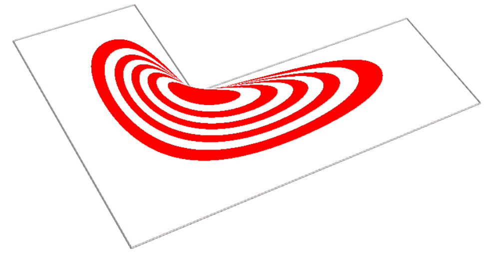


Figure 14: Blending function of a concave corner control point.

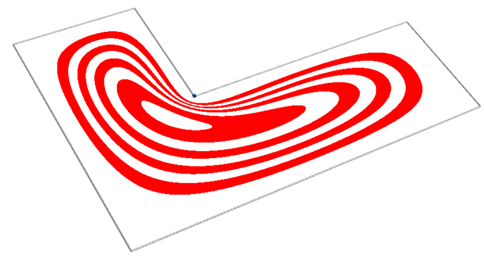


Figure 15: Blending function of the central control point.

cubic blends.) The above expression can be scaled arbitrarily; we propose to match the weight of the adjacent control points $C^i_{d,1}$ and $C^{i+1}_{0,1}$ at their maximal position, which is approximately $\frac{2}{9}$, thus leading to a multiplier of $\frac{9}{2}$. This seems to be a reasonable choice in our test examples. See Figure 14 for the blending function, and also the related test object #6 in Section 5.

Central control point

We can define a central blending function using the distance parameters as

$$\prod_i h_i^2. \tag{16}$$

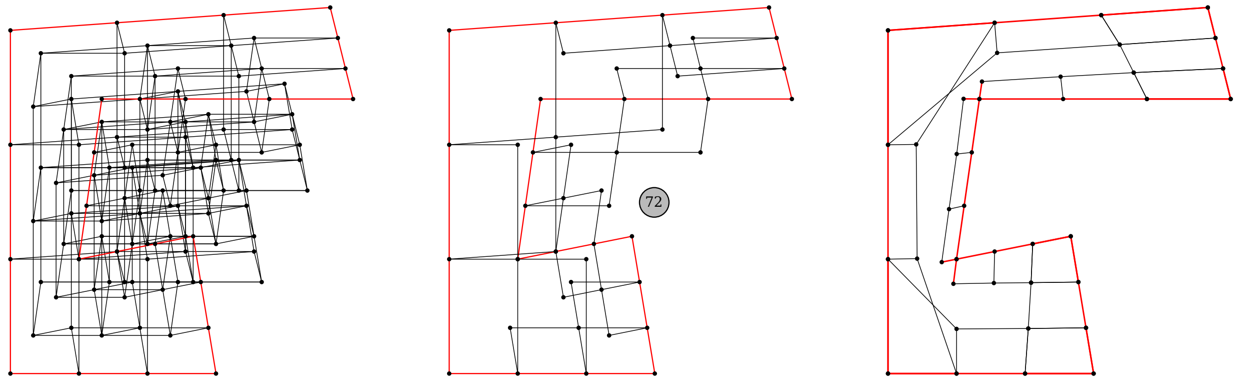
This expression also admits arbitrary scaling. We can compute a suitable multiplier by using the maximum value $\left(\frac{n-2}{n}\right)^{2n}$ we would obtain in a regular polygonal domain, where all $\lambda_i = \frac{1}{n}$ at the center. Similarly to the concave corner control points, we can then set the multiplier as $\frac{2}{9} \cdot \left(\frac{n-2}{n}\right)^{-2n}$.

See Figure 15 for the blending function, and also the related test object #4 in Section 5.

Note that using a single control point for editing the interior is not always meaningful in an arbitrary concave domain, as in some cases two or more would be needed (e.g. for symmetry reasons). The placement and weight of multiple interior control points is currently part of ongoing research.

5. Test results and discussion

In this section we present a few test examples to demonstrate the basic properties of concave GB patches. In all of the images curvature maps and isophote lines are computed by standard approximation methods [20] based on a densely sampled mesh.



(a) S-patch – full control network (b) S-patch – control points snapped (c) Concave GB patch control network

Figure 16: Control network of an 8-sided planar patch with cubic boundaries.

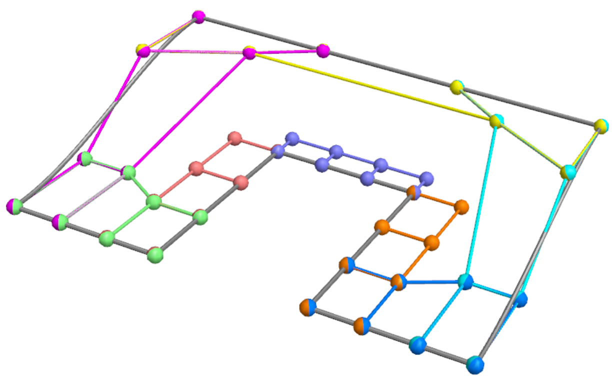
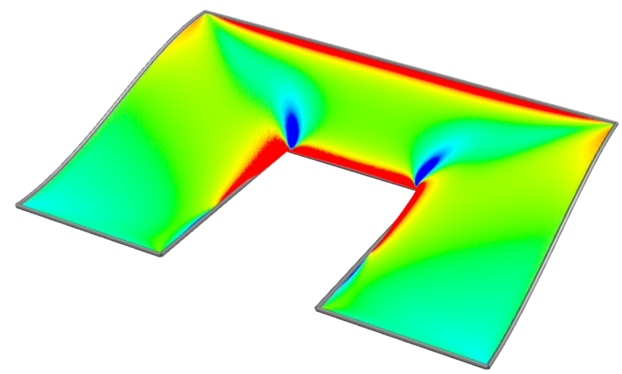
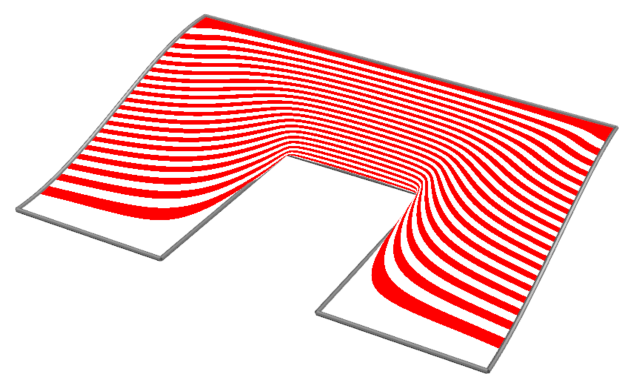


Figure 17: Curve network of test object #2.



(a) Mean curvature map



(b) Slicing

Figure 18: Patch generated by the algorithm in [7].

Test object #1

Here we compare the control network of concave GB patches to that of S-patches. As an example, we take a planar U-shaped object with eight sides.

Setting the degree to cubic, an S-patch would require 120 control points to represent this surface. The default placement of these is shown in 16a. Note that many points lie outside the concave region. It is a frequently applied procedure to merge the interior control points, i.e., those that are not adjacent to the boundaries. In our case, this would mean snapping 72 control points together and leaving the configuration shown in Figure 16b. We believe that even the remaining arrangement is not intuitive; in general it is very hard to find sensible locations for both the independent and the merged control points.

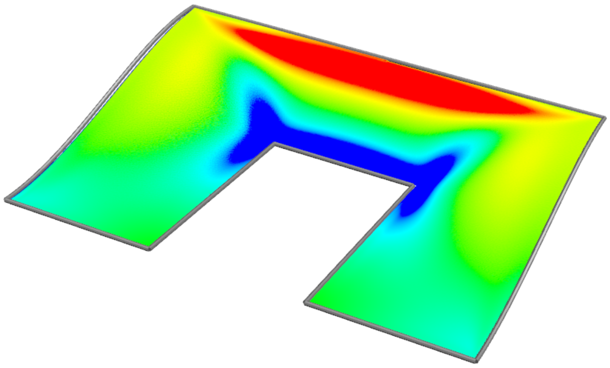
In contrast, the control points of a concave Generalized Bézier patch form an easy-to-grasp configuration. In our example in Figure 16c, there are only 16 extra control points associated with the sides (four pairs of twist control points have been merged, and two pairs remained independent).

Test object #2

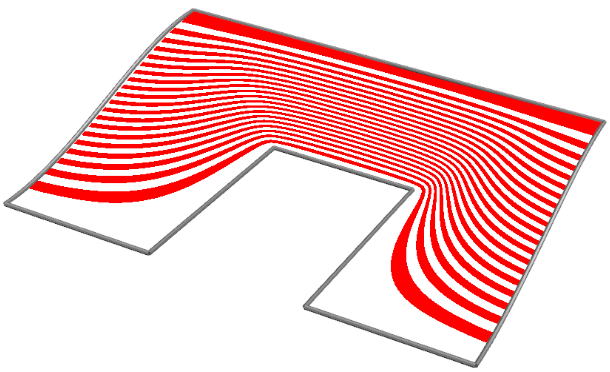
This is a simple test example with another 8-sided, U-shaped polygon. The far side is lifted and the remaining boundaries are kept in the XY-plane; the related cross-derivatives are set to ensure smooth tangential connections. The control points are shown in Figure 17. Here we wish to compare our scheme to that in [7].

Kato proposed a projective domain with a local parameterization based on edge lengths. The blends used in the paper interpolate one side and vanish on all other sides, thus they are singular at the corners. These functions decrease rapidly in the vicinity of the boundaries, which may cause sudden curvature changes.

Figures 18a and 18b indicate this shape deficiency of Kato's surface. The interior of the patch is nicely blended, but due to the above reason the influence of the ribbons is weak, see



(a) Mean curvature map



(b) Slicing

Figure 19: Concave GB patch.

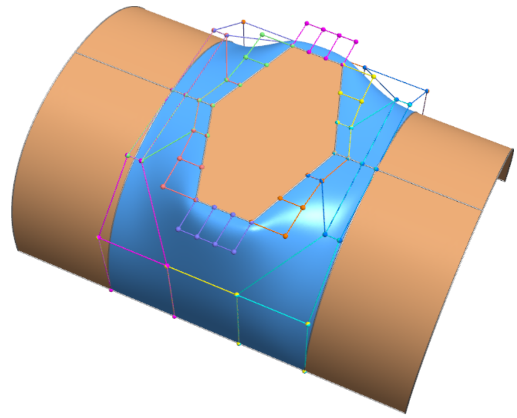
related curvature map and slicing images. Figures 19a and 19b show the concave GB surface with the same control structure. As it can be observed, the transition from the 3D boundaries towards the interior shows a more natural, even curvature and contour distribution; this also yields much better control of the cross-derivatives.

Test object #3

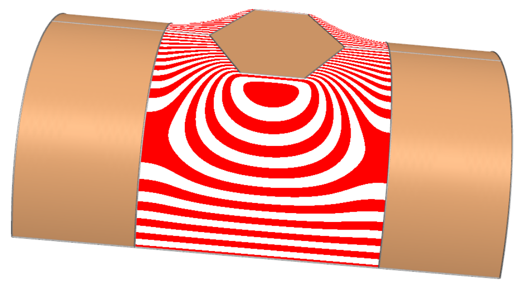
This example shows a patch layout with two 8-sided concave patches. On the left and right there are two extruded surfaces, and there is a planar six-sided face on the top. We wish to create naturally pleasing concave surfaces that smoothly connect the extrusions with the perimeter loop of the top face. The control structure and the shaded multi-sided patches are shown in Figure 20a; slicing, curvature map and isophotes are displayed in Figures 20b, 20c and 20d, respectively.

Test object #4

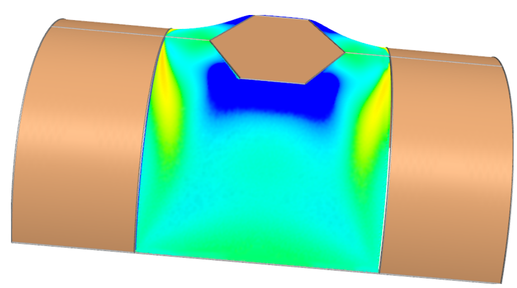
This test surface is a setback-type vertex blend that connects three edge blends. The rail-curves of the adjacent edge blends intersect at 90 degrees, and a concave patch with nine boundary curves is obtained. The patch is symmetric and in this case an obvious center control point can be defined. The sequence in Figure 21 shows three surfaces with different central control point settings. The first surface is somewhat flat; the curvature of the second one is nicely distributed; the third one has a somewhat artificial bulge in the middle.



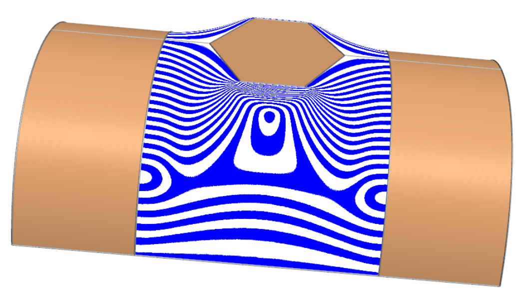
(a) Shaded patches with control points



(b) Slicing



(c) Mean curvature map

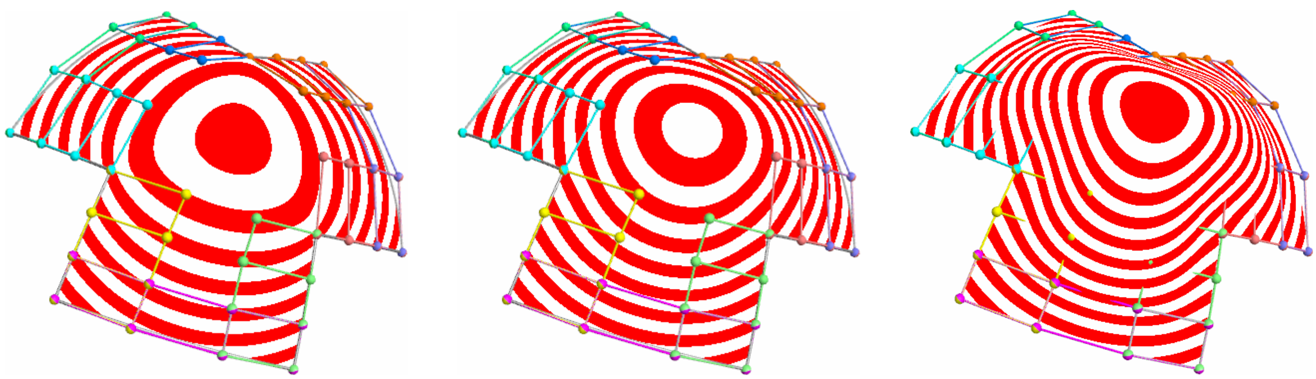


(d) Isophote lines

Figure 20: Test object #3.

Test object #5

This example shows a free-form curve network interpolated by two 5-sided convex patches, one 4-sided convex patch, and one 8-sided concave patch. The boundaries are quartic Bézier curves. The patches are smoothly connected across the shared boundaries. Editing the curve network automatically modifies the related patches. Figures 22a and 22b show the control structures and a mean curvature map, respectively, while in Figure 22c the contour lines visually indicate G^1 continuity.



(a) “Underfull” patch (b) Full patch (c) “Overfull” patch

Figure 21: Test object #4.

Test object #6

In this example we demonstrate how to edit concave GB patches. We have a simple automotive part with eight boundary curves. The reference model is defined by cubic curves and cubic cross-derivatives ($d_i = 3$, $l_i = 2$ for all i); curvature map and slicing is shown in Figure 23a.

First we add two concave corner control points (colored black) and modify the shape by means of these – see Figure 23b. Then we go back to our reference model and increase the degrees; starting from the left corner in clockwise direction we obtain curves with degrees 3,5,3,5,3,4,5,4. Note that for the two quintic boundaries the number of layers has also been increased to 3. We have edited the control points to modify the right boundary curve and the interior shape, as shown in Figure 23c.

Conclusion and future work

We have described a new patch that extends the concept of Generalized Bézier patches [1] to concave polygonal domains. Boundaries and cross-derivatives are given in control point structures compatible with quadrilateral patches. New technical solutions were introduced for half-Bézier ribbons, parameterizations and blending functions within concave domains. Editing the interior of these patches is solved using special control points.

Generating concave surfaces is a difficult problem, and our current project raised interesting questions. Compatibility issues of higher degree derivatives and the optimal placement of extra control points at the concave corners are problems of interest. We are also thinking about alternative parameterization methods with non-negative side and distance parameters to replace the current harmonic coordinates; power coordinates [21] may open an avenue to obtain direct parametric evaluations. The relative strength of the control points decreases for patches with many edges; a new parameterization method or alternative internal control structures could help to overcome this problem.

Acknowledgements

This work was partially supported by the Hungarian Scientific Research Fund (OTKA, No. 124727). All images were generated by the Sketches prototype system (ShapEx Ltd, Budapest); the authors appreciate the exceptional programming contribution of György Karikó. The authors would also like to thank István Kovács and Márton Vaitkus for the inspiring discussions.

References

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

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

[3] Smith, J, Schaefer, S. Selective degree elevation for multi-sided Bézier patches. Computer Graphics Forum 2015;34(2):609–615.

[4] Schaefer, S. Multi-sided patches via barycentric coordinates. In: Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics; chap. 8. CRC Press; 2017, p. 135–146.

[5] Hettinga, GJ, Kosinka, J. Multisided generalisations of Gregory patches. Computer Aided Geometric Design 2018; (in press).

[6] Kato, K. Generation of n-sided surface patches with holes. Computer-Aided Design 1991;23(10):676–683.

[7] Kato, K. N-sided surface generation from arbitrary boundary edges. In: Curve and Surface Design. Innovations in Applied Mathematics; Vanderbilt University Press; 2000, p. 173–181.

[8] Sone, J, Konno, K, Chiyokura, H. Surface interpolation of non-four-sided and concave area by NURBS boundary Gregory patch. In: Curve and Surface Design. Innovations in Applied Mathematics; Vanderbilt University Press; 2000, p. 389–398.

[9] Pan, H, Liu, Y, Sheffer, A, Vining, N, Li, CJ, Wang, W. Flow aligned surfacing of curve networks. ACM Transactions on Graphics 2015;34(4):127.

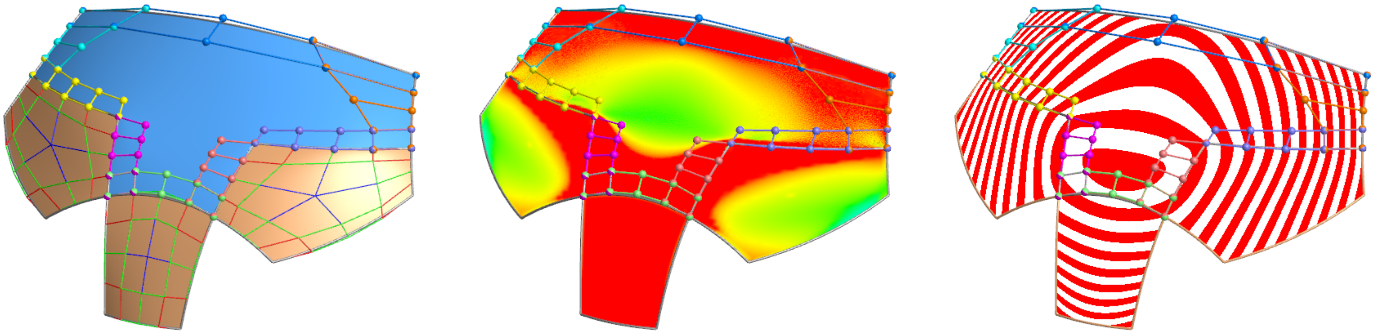
[10] Stanko, T, Hahmann, S, Bonneau, GP, Saguin-Sprynski, N. Surfacing curve networks with normal control. Computers & Graphics 2016;60:1–8.

[11] Salvi, P, Várady, T, Rockwood, A. Transfinite surface interpolation over specific curvenet configurations. In: The Mathematics of Surfaces XIV. IMA; 2013, p. 309–326.

[12] Peters, J. Smooth free-form surfaces over irregular meshes generalizing quadratic splines. Computer Aided Geometric Design 1993;10(3):347–361.

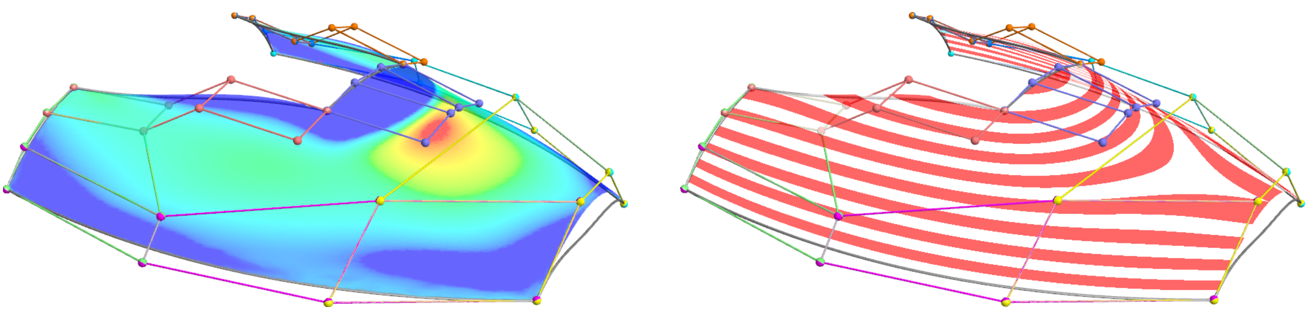
[13] Loop, C, Schaefer, S. G^2 tensor product splines over extraordinary vertices. In: Computer Graphics Forum; vol. 27. 2008, p. 1373–1382.

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

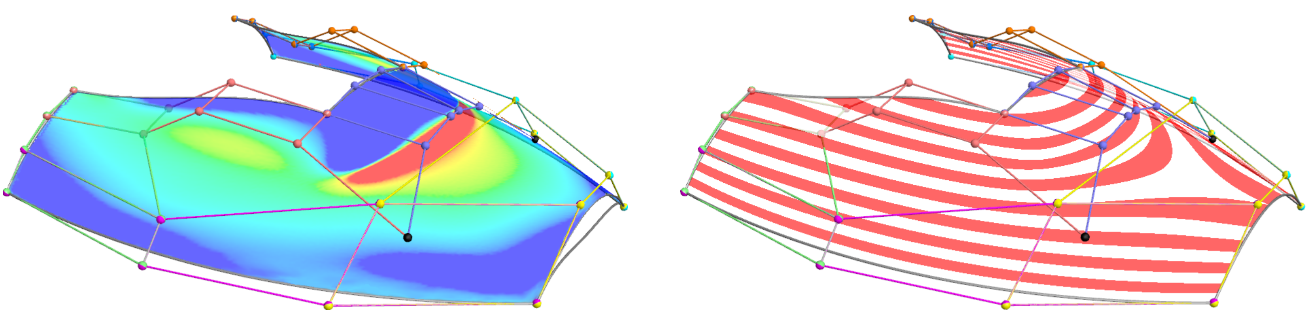


(a) Control structures (b) Mean curvature map (c) Contours

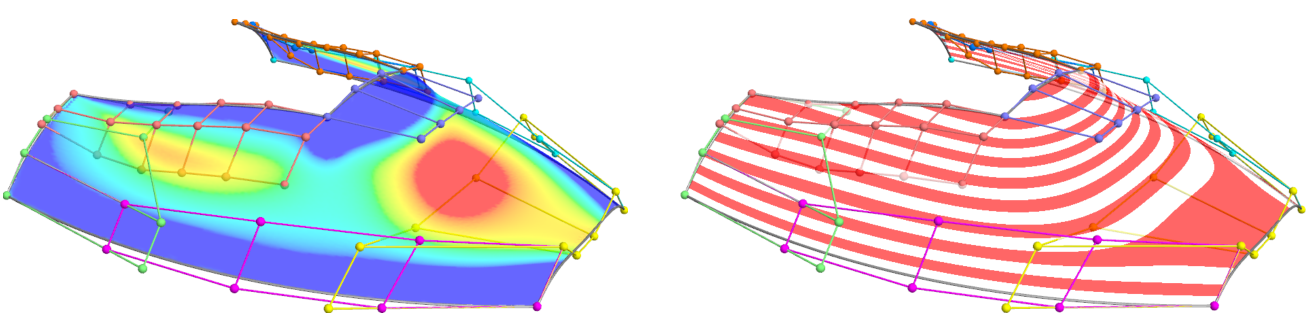
Figure 22: Test object #5.



(a) Reference patch (left: mean curvature, right: slicing)



(b) Concave corner control points added (left: mean curvature, right: slicing)



(c) Degree elevated patch modified (left: mean curvature, right: slicing)

Figure 23: Test object #6.

[15] Várady, T, Salvi, P, Kovács, I. Enhancement of a multi-sided Bézier surface representation. Computer Aided Geometric Design 2017;55:69–83.

[16] Hormann, K, Floater, MS. Mean value coordinates for arbitrary planar polygons. Transactions on Graphics 2006;25(4):1424–1441.

[17] Gregory, JA. Smooth interpolation without twist constraints. In: Computer Aided Geometric Design. Academic Press, Inc.; 1974, p. 71–88.

[18] Várady, T, Rockwood, A, Salvi, P. Transfinite surface interpolation over irregular *n*-sided domains. Computer Aided Design 2011;43(11):1330–1340.

[19] DeRose, T, Meyer, M. Harmonic coordinates. In: Pixar Technical Memo 06-02, Pixar Animation Studios. 2006,.

[20] Rusinkiewicz, S. Estimating curvatures and their derivatives on triangle meshes. In: 3D Data Processing, Visualization and Transmission. IEEE; 2004, p. 486–493.

[21] Budninskiy, M, Liu, B, Tong, Y, Desbrun, M. Power coordinates: a geometric construction of barycentric coordinates on convex polytopes. ACM Transactions on Graphics 2016;35(6):241.