




Intuitive Interior Control for Multi-Sided Patches with Arbitrary Boundaries

Péter Salvi 

Budapest University of Technology and Economics, salvi@iit.bme.hu

Abstract. In most CAD systems, multi-sided regions are typically represented as trimmed surfaces, or are split into several quadrilateral sub-patches. Both of these approaches have their drawbacks; an alternative is to use non-standard patch formulations, such as transfinite interpolation surfaces or generalized Bézier patches, but these also have their limitations.

We propose a new hybrid scheme that combines the advantages of ribbon- and control-point-based surfaces: it can handle positional and cross-derivative boundary constraints of any degree, while providing fine-grained control of the interior, as well.

Keywords: multi-sided surfaces, transfinite interpolation, control-point-based surfaces, generalized Bézier patch

DOI: <https://doi.org/10.14733/cadaps.2024.143-154>

1 INTRODUCTION

The geometric modeling of many objects, ranging from simple household items to complex aesthetic designs, involves non-quadrilateral free-form surfaces. In CAD systems these are generally represented by one of the following methods (see also Fig. 1):

- *Trimming* takes a larger four-sided patch and trims it at the specified boundaries. The connection to adjacent surfaces will generally be inaccurate (even positionally), and inherent symmetries of the multi-sided surface may not be reproduced.
- *Splitting* divides the n -sided area into subpatches. The actual choice of subdivision affects surface quality, and maintaining G^1 or G^2 continuity along the subdividing curves as the patch is modified can be difficult.

In contrast, there are non-standard representations allowing an arbitrary number of sides without the drawbacks mentioned above. These are often called *transfinite interpolation* surfaces, as they reproduce the boundaries exactly, and can also ensure smooth connections to adjacent patches with G^1 or higher continuity.

Some of these can be regarded as multi-sided generalizations of the Coons patch, in the sense that the surface depends only on the positional and cross-derivative constraints at the boundaries. This is usually achieved by blending together ribbon surfaces interpolating some of the boundaries, so we will call these

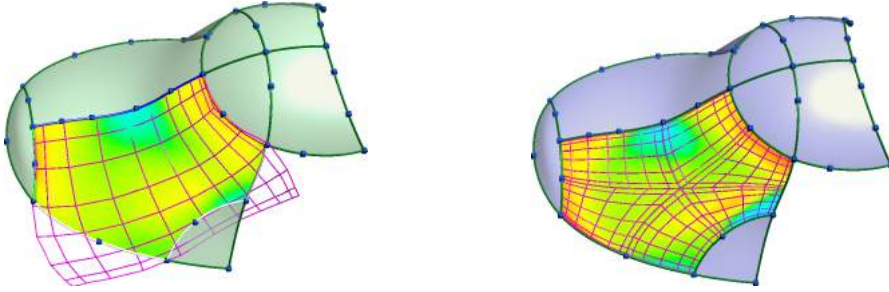


Figure 1: Representing a 6-sided surface with trimming (left) and splitting (right).

ribbon-based surfaces. Their strong point is also their weakness: since the geometry depends only on the boundaries, there is little control over the surface interior.

There are also *control-point-based* representations. These have fine-grained interior control, but are generally limited to (rational) polynomial boundaries.

The aim of this paper is to propose a genuinely multi-sided surface representation that (i) can handle any kind of boundary curves, (ii) allows connection to adjacent patches with G^1 or higher continuity, and (iii) has good control over the interior. The preliminaries and related works are described in Section 2; our main contribution is then outlined in Section 3, followed by a discussion illustrated with examples in Section 4.

2 PREVIOUS WORK

In Section 2.1 we introduce the notations, and review some of the more influential ribbon- and control-point-based formulations. A brief analysis of these and a few other relevant methods found in the literature is presented in Section 2.2.

2.1 Preliminaries

Almost all multi-sided surface formulations are defined over a 2D domain (exceptions include [15]). Its shape can be varied, but often it is a regular n -sided polygon. Points inside the domain are then mapped to some sort of local coordinates depending on the actual patch equation. Here we show a set of coordinates that can be described by generalized barycentric coordinates [2] and fits all schemes reviewed in this section.

Given a point in the domain and its generalized barycentric coordinates $\{\lambda_i\}$, we define

$$s_i(\lambda_1, \dots, \lambda_n) = \frac{\lambda_i}{\lambda_{i-1} + \lambda_i}, \quad d_i(\lambda_1, \dots, \lambda_n) = 1 - (\lambda_{i-1} + \lambda_i) \quad (1)$$

with cyclic indexing. The side parameter s_i runs from 0 to 1 as we follow the i^{th} edge of the domain (connecting the $(i-1)^{\text{st}}$ and i^{th} vertices), giving 0 and 1 on the $(i-1)^{\text{st}}$ and $(i+1)^{\text{st}}$ edges, respectively. The distance parameter d_i vanishes at the i^{th} edge and increases monotonically, reaching 1 at the non-adjacent sides. Figure 2 shows constant parameter lines of $\frac{k}{10}$ ($k = 0, 1, 2 \dots 10$). Note that $s_i, d_i \in [0, 1]$ and for a point on the i^{th} side $d_{i-1} = s_i = d_{i+1}$. (We chose Wachspress coordinates here, as these are the simplest and they work well on convex polygons.)

2.1.1 Ribbon-Based Surfaces

A ribbon \mathbf{R}_i is a quadrilateral surface interpolating the i^{th} boundary curve, and also satisfying the associated cross-derivative constraints. We can blend different ribbons together using a variation [3] of Shepard's inverse

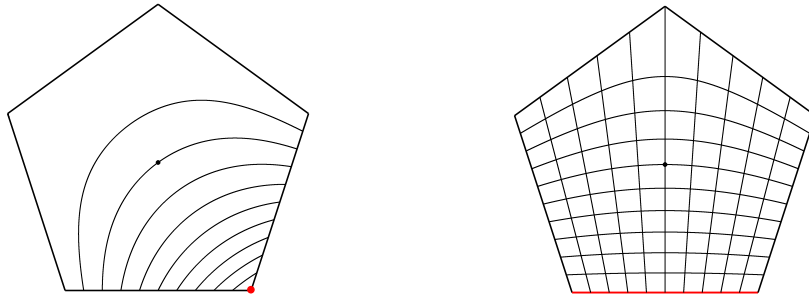


Figure 2: Parameterization of a 5-sided domain, showing contours of Wachspress coordinates λ_i associated with the bottom-right corner (left) and the (s_i, d_i) system associated with the bottom side (right).

distance weights (L_i):

$$\mathbf{S}(u, v) = \sum_{i=1}^n \mathbf{R}_i(s_i, d_i) L_i(d_1, \dots, d_n) = \sum_{i=1}^n \mathbf{R}_i(s_i, d_i) \cdot \frac{1/d_i^2}{\sum_{j=1}^n 1/d_j^2}. \tag{2}$$

Here (u, v) is a point in the 2D domain, s_i is a shorthand for $s_i(\lambda_1(u, v), \dots, \lambda_n(u, v))$ and similarly for d_i . This equation cannot be evaluated at the boundaries in this form, but this can be solved by multiplying both the numerator and denominator of L_i by $\prod_{k=1}^n d_k^2$. It is easy to see that the following properties hold for a point on the i^{th} boundary:

$$L_i = 1, \quad L_j = 0 \quad (j \neq i), \quad L'_k = 0 \quad (\forall k), \tag{3}$$

where the derivative is taken in an arbitrary parametric direction. A singularity remains, as the above equations specify both 0 and 1 for a corner point, but the limit of the sum exists. Increasing the exponents in L_i allows for connecting to adjacent surfaces with G^2 or higher continuity. Figure 3 shows a schematic depiction of this approach.

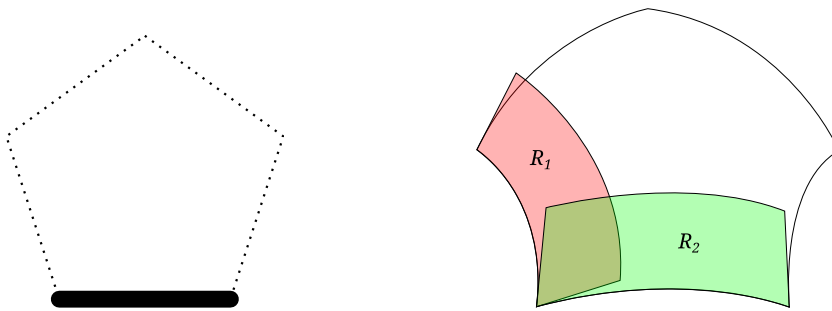


Figure 3: Inverse distance weighted patch. Left: blending scheme (1 at one side and 0 on all others); Right: two linear ribbon surfaces.

The Charrot–Gregory patch [1] uses corner interpolants $\mathbf{R}_{i-1,i}$ instead of ribbons. These interpolate two consecutive boundaries, and can be regarded as partial Coons patches created from two ribbons and a corner correction surface [10]. The blending function is very similar to the previous one, see also Fig. 4.

$$\mathbf{S}(u, v) = \sum_{i=1}^n \mathbf{R}_{i-1,i}(s_{i-1}, s_i) L_{i-1,i}(d_1, \dots, d_n) = \sum_{i=1}^n \mathbf{R}_{i-1,i}(s_{i-1}, s_i) \cdot \frac{1/(d_{i-1}^2 d_i^2)}{\sum_{j=1}^n 1/(d_{j-1}^2 d_j^2)}. \tag{4}$$

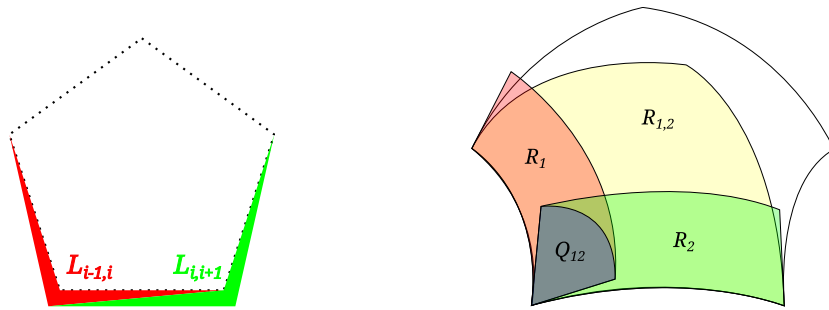


Figure 4: Charrot–Gregory patch. Left: blending scheme (1 at one corner, gradually vanishing on the adjacent boundaries); Right: creating a corner interpolant from ribbon surfaces ($\mathbf{R}_{i-1,i} = \mathbf{R}_{i-1} + \mathbf{R}_i - \mathbf{Q}_{i-1,i}$).

2.1.2 Control-Point–Based Surfaces

The S-patch [4] is a generalization of the Bézier triangle, see Fig. 5. For a point in the domain with (generalized) barycentric coordinates λ_i , it is defined as

$$S(\lambda_1, \dots, \lambda_n) = \sum_{\mathbf{J}} \mathbf{P}_{\mathbf{J}} \frac{p!}{\prod_{i=1}^n J_i!} \cdot \prod_{i=1}^n \lambda_i^{J_i}, \tag{5}$$

where $\mathbf{P}_{\mathbf{J}}$ are the control points, \mathbf{J} is an index vector with $|\mathbf{J}| = n$, and $\sum_i J_i = p$ is the degree or *depth* of the patch. (When $n = 2$ we get a Bézier curve with parameter $u = \lambda_2 = 1 - \lambda_1$.)

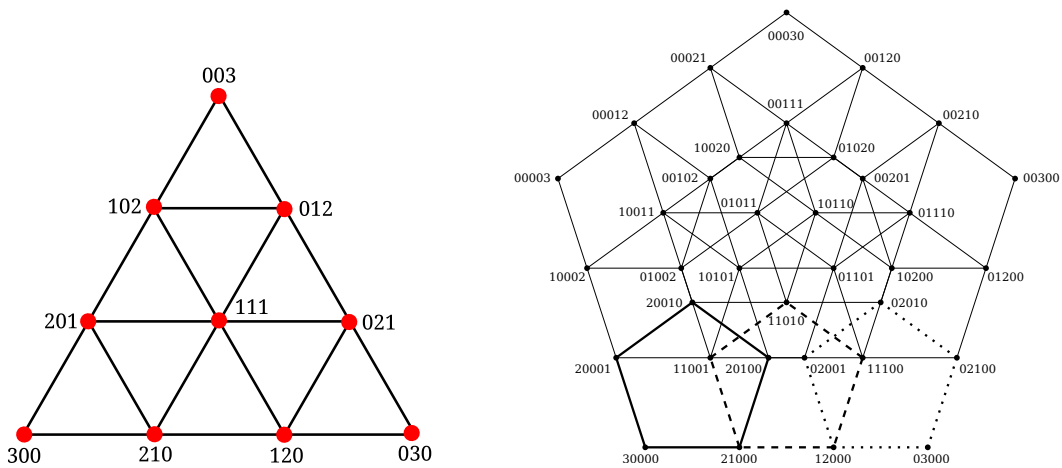


Figure 5: S-patch control net with $n = 3$ (left) and $n = 5$ (right).

A recent control-point–based surface representation is the Generalized Bézier or GB patch [12], which uses weighted Bernstein polynomials to satisfy the boundary constraints. Figure 6 shows its control structure and the weighting scheme. The black frames show which control points belong to the bottom and right-hand sides; some belong to both, and these are multiplied by both of the associated blending functions.

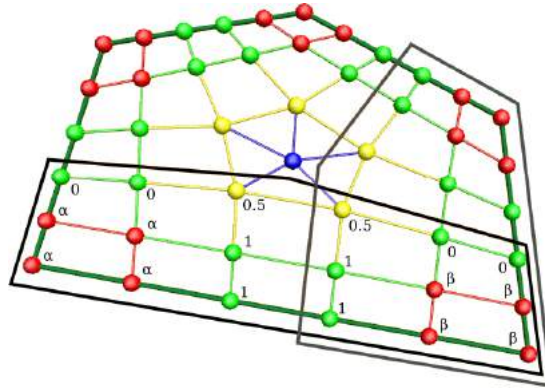


Figure 6: Control net of the Generalized Bézier patch.

The patch is defined as

$$\mathbf{S}(u, v) = \sum_{i=1}^n \sum_{j=0}^p \sum_{k=0}^{\lfloor \frac{p-1}{2} \rfloor} \mathbf{P}_{i,j,k} \cdot \mu_{i,j,k}(d_1, \dots, d_n) B_j^p(s_i) B_k^p(d_i) + \mathbf{P}_0(1 - B_\Sigma(u, v)), \quad (6)$$

where p is the degree, $\mathbf{P}_{i,j,k}$ is the j^{th} control point in the k^{th} row associated with the i^{th} side, and B_i^p is the i^{th} Bernstein polynomial of degree p . The $\mu_{i,j,k}$ functions denote the weights shown in Fig. 6 with

$$\alpha_i = \frac{d_{i-1}^2}{d_{i-1}^2 + d_i^2}, \quad \beta_i = \frac{d_{i+1}^2}{d_i^2 + d_{i+1}^2}. \quad (7)$$

The blends in the first term of Eq. (6) do not sum to 1; the second term associates the ‘weight deficiency’ with the central control point (\mathbf{P}_0), i.e.,

$$B_\Sigma(u, v) = \sum_{i=1}^n \sum_{j=0}^p \sum_{k=0}^{\lfloor \frac{p-1}{2} \rfloor} \mu_{i,j,k}(d_1, \dots, d_n) B_j^p(s_i) B_k^p(d_i). \quad (8)$$

This patch behaves at its boundaries (in a G^1 sense) as a quadrilateral Bézier surface created with the associated control points. Increasing the exponents in Eq. (7) allows for higher order interpolation.

2.2 Related Works

All representations reviewed above have been enhanced in various ways over the years in an effort to realize the goals outlined in the Introduction.

As indicated previously, ribbon-based surfaces handle boundary constraints well, but we have little command over the overall shape. In light of this, Kato’s patch [3] was supplemented with interior control [13] both by adding auxiliary curves and vertices as additional ribbon constraints, and by blending with another surface defining the central part of the patch. Also related is the recent ABC patch [5], which aims at improving trimmed B-spline surfaces with accurate boundary control, while preserving CAD-compatibility. While these are somewhat similar to our current approach, they lack a genuinely multi-sided way of editing the interior.

The Charrot–Gregory patch [1] was extended by a single extra degree of freedom [8], which can be used to set the fullness of the surface. Although this is sufficient in the majority of cases, sometimes more detailed control is required.

Shape adjustment with control-point-based representations is generally very flexible, but the admissible boundary constraints are limited. In the case of the S-patch [4], a good initial surface can be created based on G^1 cross-derivative constraints [7], but the boundaries are assumed to be polynomial.

The Generalized Bézier patch [12] has the same limitation; an extension to B-spline boundaries has also been published [11], but for the price of abandoning its uniform control structure, and thus, in a sense, transforming it into a ribbon-based representation.

Here we have focused on the drawbacks of other formulations—these are what our proposed surface is designed to solve; see also Section 4.1 for some of its limitations.

3 HYBRID PATCH

We have seen that Kato’s patch is a nice and simple ribbon-based transfinite interpolation surface, but lacks interior control. On the other hand, the Generalized Bézier patch has a very natural control structure allowing modification of the interior, but its boundary constraints are limited to Bézier curves. It is a natural idea to combine these two representations in a way that retains the favorable features of both.

The proposed surface formulation takes Eq. (6) and replaces the outer two control rows with linear ribbons multiplied by the weight sum of the two rows, which is a singular blending function (L_i^*) similar to L_i in Eq. (2):

$$S(u, v) = \sum_{i=1}^n \left[\sum_{j=0}^p \sum_{k=2}^{\lfloor \frac{p-1}{2} \rfloor} \mathbf{P}_{i,j,k} \cdot \mu_{i,j,k}(d_1, \dots, d_n) B_j^p(s_i) B_k^p(d_i) + \mathbf{R}_i(s_i, d_i) L_i^*(u, v) \right] + \mathbf{P}_0(1 - B_\Sigma(u, v)), \tag{9}$$

with

$$L_i^*(u, v) = \sum_{j=0}^p \sum_{k=0}^1 \mu_{i,j,k}(d_1, \dots, d_n) B_j^p(s_i) B_k^p(d_i). \tag{10}$$

The main difference between L_i and L_i^* is that the latter does not sum to 1 in the interior of the domain, which leaves weight for the control points—the same weight they would have in a GB patch.

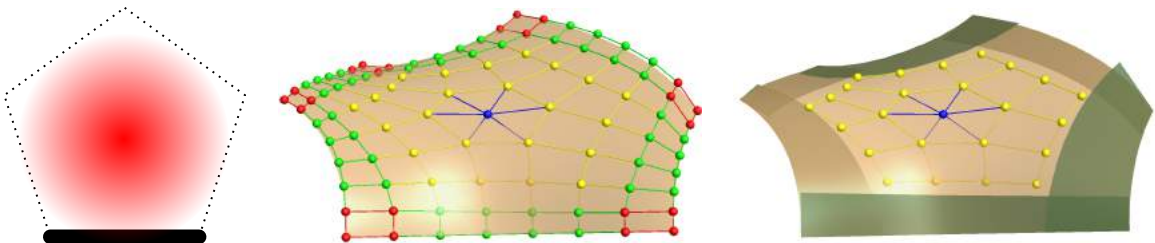


Figure 7: Construction of the hybrid patch. Left: blending scheme; Middle: GB patch; Right: hybrid patch.

Replacing three control rows and increasing the exponent to 3 in Eq. (7) can interpolate the ribbons in a G^2 sense. In this case we need curved ribbons, which can be inherited from adjacent surfaces, or generated from a curve network [9].

The new patch representation is an improvement over the one in Eq. (2), as it adds interior control, and does not exhibit the large curvature variations that often appear near the boundaries with the steeply falling inverse distance weights. It is also more general than the GB patch, since it is not limited to polynomial boundaries; in fact, it can handle procedural boundary constraints, as well, e.g. ribbons defined by rotation-minimizing frames [14] (see Fig. 9 below), or special boundaries like catenary curves, Euler spirals, etc.

In Figure 8, a 5-sided patch with B-spline boundaries is modified by moving its control points (the ribbons remain unchanged), resulting in a much more natural isophote line distribution.

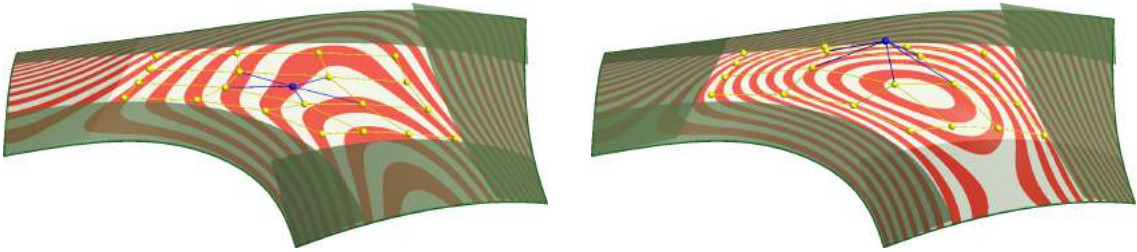


Figure 8: Modification of a hybrid patch, showing isophote lines.

Figure 9 shows a five- and a six-sided hybrid patch. At the shared boundary a rotation-minimizing frame defines the normal fence both surfaces are perpendicular to.

4 DISCUSSION

In this section we discuss some limitations and practical issues, such as the default positioning of control points.

4.1 Limitations

A limitation common with all other surfaces discussed above (with the exception of the ABC patch [5]) is that this is a non-standard representation that can be exported in a CAD-compatible format only by fitting on sampled points. Normal vectors and curvatures are hard to compute exactly, and usually mesh-based approximations are used instead.

Also, the patch formula makes use of Bernstein polynomials of a relatively high degree ($p = 2(k + \ell) + 1$ or $p = 2(k + \ell + 1)$ when using G^k cross-derivative constraints with ℓ inner control layers). For example, in Fig. 7 we have $k = 1$ and $\ell = 2$, and even this simple configuration results in degree 7. In our experience this has no negative implications, and note that one would use the same degree even with GB patches [12] (although not with S-patches [4]) for the same detail.

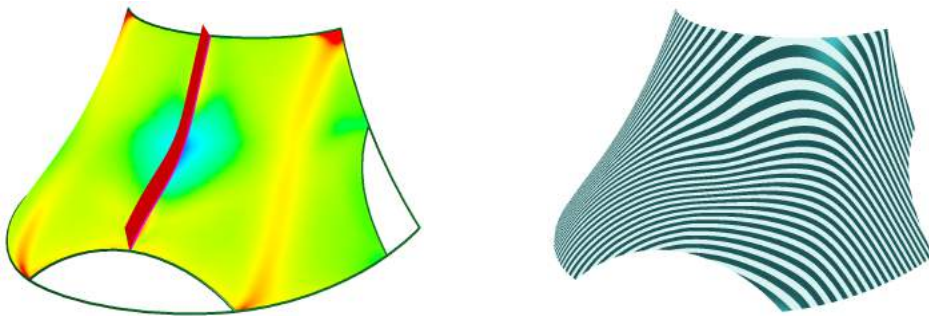


Figure 9: Two patches connected with G^1 continuity; mean curvature with common normal fence (left) and contouring (right). Full red/blue colors are set to ± 0.015 ; the bounding box diagonal is 350 units long.

Degree elevation can be used for increasing control in the interior (using the same algorithm as in [12]), but as with GB patches, this changes the patch slightly. Since designers would only add more control when they *do* want to change the shape, this limitation is mainly theoretical.

Finally, unlike GBS patches [11], this representation cannot handle hole loops or highly concave boundaries.

4.2 Practical Considerations

The patch defined in Section 3 has additional control points in the interior, but where should we place these by default? A natural idea is to find positions such that the resulting patch deviates minimally from the original surface. But since these control points are to be used for design, it is important to have a ‘nice’ control net without large undulations that often result from simple minimization schemes. Control structure fairness could be optimized by adding some energy to the system, but in practice the following simple heuristic works well:

1. Approximate each ribbon with a Bézier surface of the chosen degree (p by k)
2. Build the inner structure as in [12] (summarized in Appendix A for completeness) – generate a *base patch* of degree $2k + 1$ interpolating the derivatives at the corners, and then iteratively raise the degree and add displacements

Figure 10 shows an example, where cubic B-spline ribbons are approximated with two rows of octic-by-linear Bézier surfaces ($k = 1, \ell = 2$), then the control net is built by degree elevation, starting from a cubic base patch. Replacing the outer rows by the original ribbons gives the default hybrid patch.

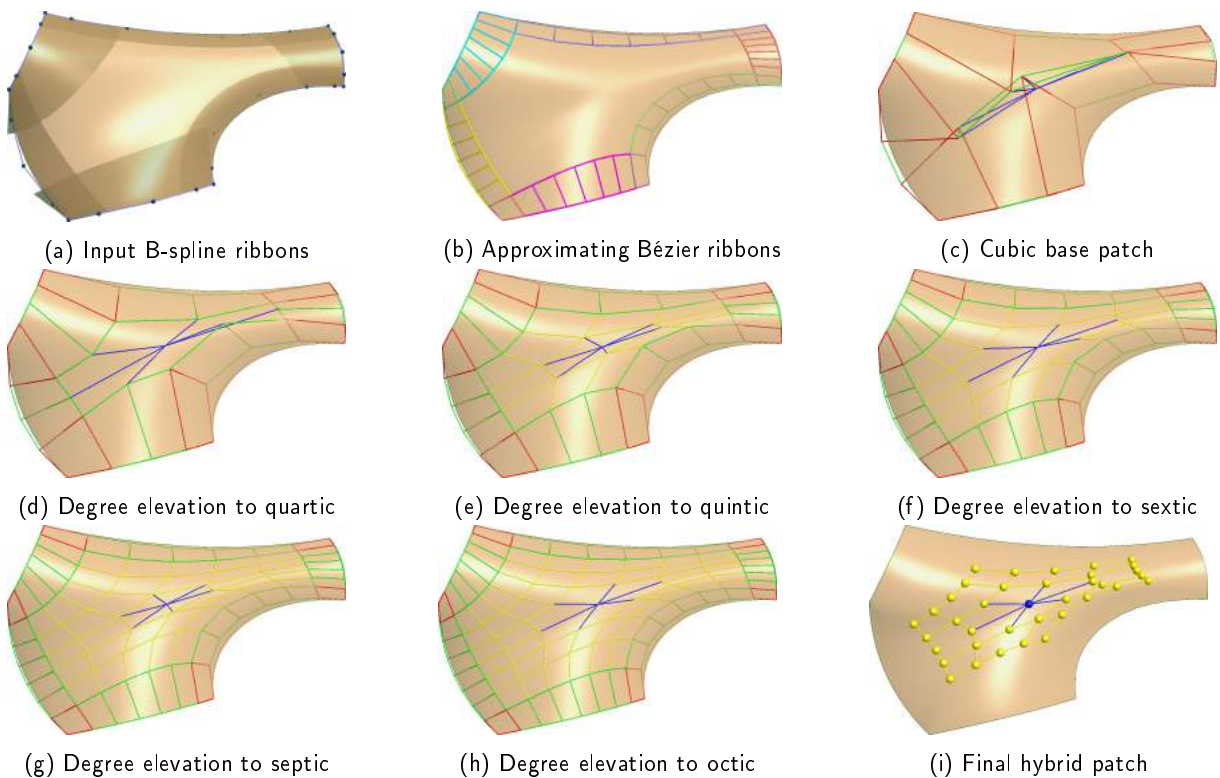


Figure 10: Generating default control point positions.

Figure 11 illustrates how these control points can be used to modify the surface interior. We can add or remove features by manual adjustment; alternatively, the additional degrees of freedom can be used to optimize an energy or fit data points.

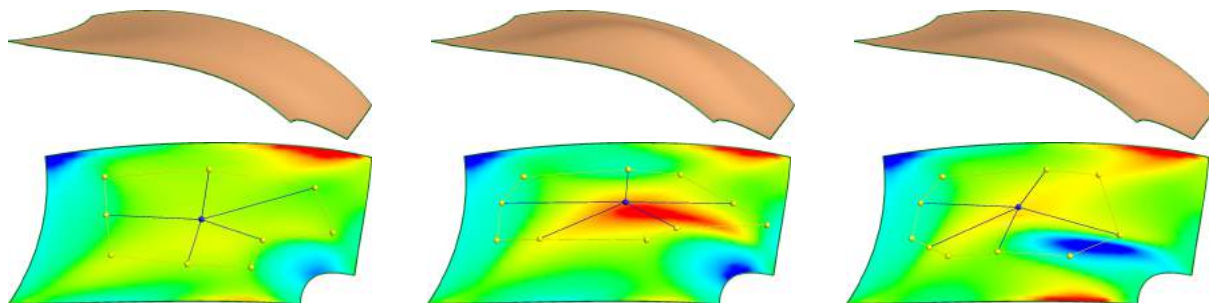


Figure 11: Adding different features by modifying the control net.

Figure 12 shows a larger model of 34 patches, with most of the surfaces having one inner layer ($\ell = 1$). The default positioning is not always perfect, some control points need adjustment to attain better quality. Long patches with large curvature, like the front fender in Fig. 12a, are notoriously difficult to represent with transfinite schemes, since the middle of the patch tends to be too flat or even ‘concave’. This artefact is easy to fix with the help of control points (Fig. 12b). The distribution of isophote lines on the hood can also be improved upon (Fig. 12c): the left surface shows the original, while the one on the right is after some adjustment.

Evaluation complexity is comparable to those of its predecessors, Kato’s patch and the Generalized Bézier patch, but slightly higher—this is the price to pay for additional freedom (in interior control and boundary representation, respectively). The actual speed ratio depends on many factors, including the number of interior control points, the number of sides, and the type of boundary constraints. In a typical case, generating a dense triangle mesh of around ten thousand vertices is computed well below 1 second on an Intel Core i5 CPU (without parallelization).

5 CONCLUSIONS

We have proposed a new multi-sided surface formulation that combines the advantages of ribbon- and control-point-based patches. It can handle positional and cross-derivative boundary constraints of any degree, while also providing intuitive control over the surface interior.

Future work will concentrate on generating natural control structures for surfaces with highly concave boundaries, or for patches with internal holes, probably building on results concerning quadmesh generation.

ACKNOWLEDGEMENTS

This project has been supported by the Hungarian Scientific Research Fund (OTKA, No. 124727).

Péter Salvi, <http://orcid.org/0000-0003-2456-2051>

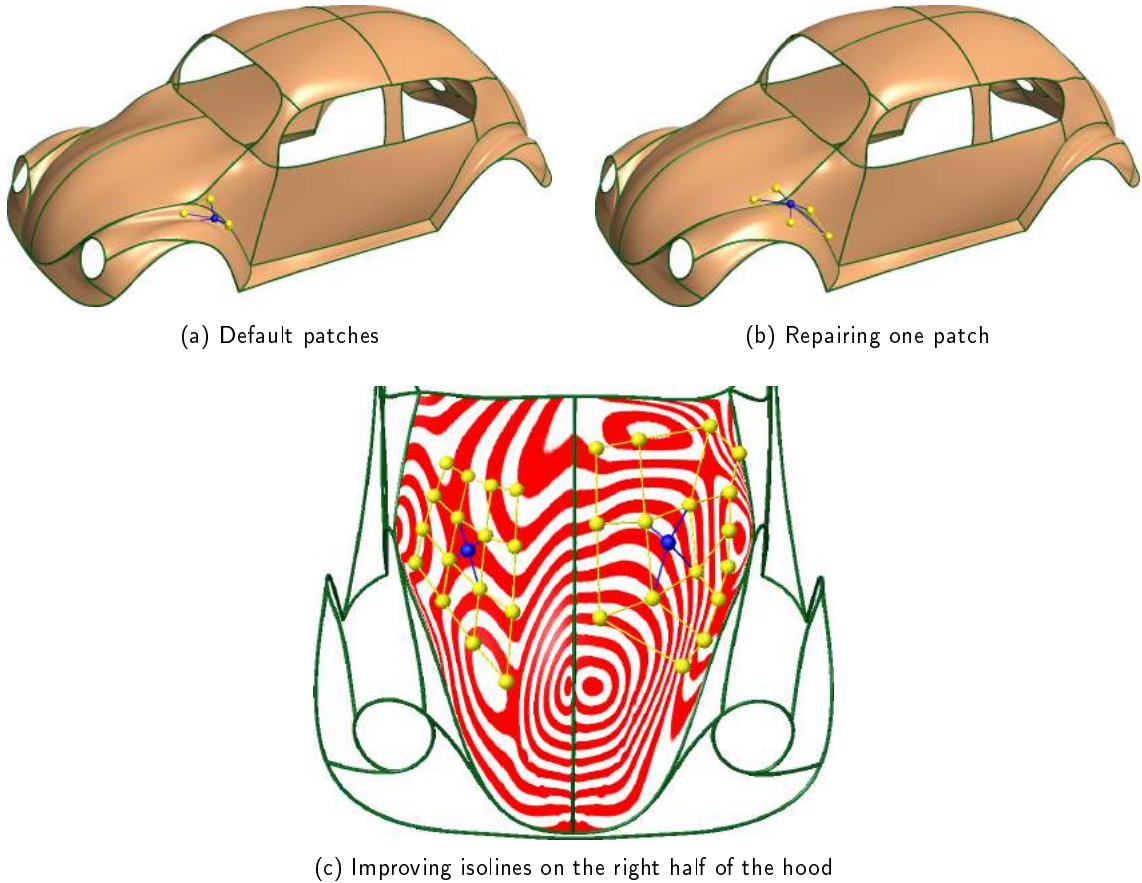


Figure 12: A complex model (based on the data in [6]).

A DEFAULT CONTROL POSITIONS

Here we summarize the algorithm [12] of generating default control point locations for GB patches. For more details, see Section 5 of the original paper. We will assume G^1 constraints for simplicity's sake; higher orders work similarly.

The input consists of the outer control rows defining the boundary constraints, i.e., $\mathbf{P}_{i,j,k}$ where $1 \leq i \leq n$, $0 \leq j \leq p$, and $0 \leq k \leq 1$ (cf. Eq. 6). The method relies on a formal degree reduction and degree elevation process (detailed below). We will use upper indices to denote the degree associated with a set of control points, so our input is $\mathbf{P}_{i,j,k}^{(p)} = \mathbf{P}_{i,j,k}$. Performing a series of degree reductions, saving all intervening states, we arrive at $\mathbf{P}_{i,j,k}^{(3)}$, the least degree allowing G^1 constraints (also called the *base patch*).

In the second phase, we iteratively raise the degree back to the original value of p , thereby generating the interior control points. After each step, however, control points in the outer rows (i.e., those that have been already computed once in the course of degree reduction) are returned to their previous values. This restores the original control points where they were available, but in the same time it also constructs a uniform control network that fits the constraints naturally.

For an even better default it is often practical to fix the central control point throughout the above process to a certain user-defined position, as this ensures the required fullness of the patch.

Degree Reduction

We generate the ‘reduced’ positions applying an inverse degree elevation, starting from the outside, and moving gradually inwards. The corner control points remain in place; as for those control points that are in the outermost row we have

$$\mathbf{P}_{i,j,0}^{(p)} = \frac{\mathbf{P}_{i,j,0}^{(p+1)} - \gamma_j \mathbf{P}_{i,j-1,0}^{(p)}}{1 - \gamma_j}, \quad 1 \leq j \leq m, \quad (11)$$

$$\mathbf{P}_{i,j-1,0}^{(p)} = \frac{\mathbf{P}_{i,j,0}^{(p+1)} - (1 - \gamma_j) \mathbf{P}_{i,j,0}^{(p)}}{\gamma_j}, \quad p - m + 1 \leq j \leq p, \quad (12)$$

where $m = \lfloor p/2 \rfloor$ and $\gamma_j = \frac{j}{p+1}$. In the odd-to-even case, averaging is needed at $\mathbf{P}_{i,m,0}^{(p)}$. The remaining control positions are generated similarly using the following equations:

$$\mathbf{P}_{i,j,k}^{(p)} = \frac{\mathbf{P}_{i,j,k}^{(p+1)} - \gamma_j \gamma_k \mathbf{P}_{i,j-1,k-1}^{(p)} - (1 - \gamma_j) \gamma_k \mathbf{P}_{i,j,k-1}^{(p)} - \gamma_j (1 - \gamma_k) \mathbf{P}_{i,j-1,k}^{(p)}}{(1 - \gamma_j)(1 - \gamma_k)}, \quad (13)$$

$$\mathbf{P}_{i,j-1,k}^{(p)} = \frac{\mathbf{P}_{i,j-1,k}^{(p+1)} - \gamma_j \gamma_k \mathbf{P}_{i,j-1,k-1}^{(p)} - (1 - \gamma_j) \gamma_k \mathbf{P}_{i,j,k-1}^{(p)} - (1 - \gamma_j)(1 - \gamma_k) \mathbf{P}_{i,j,k}^{(p)}}{\gamma_j(1 - \gamma_k)}. \quad (14)$$

Degree Elevation

Degree elevation is locally the same operation as in the case of quadrilateral Bézier patches. Corner control points are once again retained. For control points in the outermost row we have

$$\mathbf{P}_{i,j,0}^{(p+1)} = \gamma_j \mathbf{P}_{i,j-1,0}^{(p)} + (1 - \gamma_j) \mathbf{P}_{i,j,0}^{(p)}, \quad 1 \leq j \leq p. \quad (15)$$

The remaining control points are computed by

$$\mathbf{P}_{i,j,k}^{(p+1)} = \gamma_j \gamma_k \mathbf{P}_{i,j-1,k-1}^{(p)} + (1 - \gamma_j) \gamma_k \mathbf{P}_{i,j,k-1}^{(p)} + \gamma_j (1 - \gamma_k) \mathbf{P}_{i,j-1,k}^{(p)} + (1 - \gamma_j)(1 - \gamma_k) \mathbf{P}_{i,j,k}^{(p)}. \quad (16)$$

Note that for even-degree networks, references to $\mathbf{P}_{i,m,m}^{(p)}$ denote the central control point.

In all odd-degree networks, the central control point \mathbf{P}_0 (when not fixed) is set to the mass center of the innermost ring:

$$\mathbf{P}_0 = \frac{1}{n} \sum_{i=1}^n \mathbf{P}_{i,m,m}. \quad (17)$$

Odd-to-even-degree elevations keep the central control point position.

REFERENCES

- [1] Charrot, P.; Gregory, J.A.: A pentagonal surface patch for computer aided geometric design. *Computer Aided Geometric Design*, 1(1), 87–94, 1984. [http://doi.org/10.1016/0167-8396\(84\)90006-2](http://doi.org/10.1016/0167-8396(84)90006-2).
- [2] Hormann, K.; Sukumar, N.: *Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics*. CRC press, 2017. <http://doi.org/10.1201/9781315153452>.
- [3] Kato, K.: Generation of n -sided surface patches with holes. *Computer-Aided Design*, 23(10), 676–683, 1991. [http://doi.org/10.1016/0010-4485\(91\)90020-w](http://doi.org/10.1016/0010-4485(91)90020-w).
- [4] Loop, C.T.; DeRose, T.D.: A multisided generalization of Bézier surfaces. *ACM Transactions on Graphics (TOG)*, 8(3), 204–234, 1989. <http://doi.org/10.1145/77055.77059>.
- [5] Martin, F.; Reif, U.: Trimmed spline surfaces with accurate boundary control. In *Geometric Challenges in Isogeometric Analysis*, 123–148. Springer, 2022. http://doi.org/10.1007/978-3-030-92313-6_6.
- [6] Pan, H.; Liu, Y.; Sheffer, A.; Vining, N.; Li, C.J.; Wang, W.: Flow aligned surfacing of curve networks. *ACM Transactions on Graphics (TOG)*, 34(4), 1–10, 2015. <http://doi.org/10.1145/2766990>.
- [7] Salvi, P.: G^1 hole filling with S-patches made easy. In *Proceedings of the 12th Conference of the Hungarian Association for Image Processing and Pattern Recognition. KÉPAF*, 2019. <https://arxiv.org/abs/2002.11109>.
- [8] Salvi, P.; Kovács, I.; Várady, T.: Computationally efficient transfinite patches with fullness control. In *Proceedings of the Workshop on the Advances of Information Technology*, 96–100. BME, 2017. <https://arxiv.org/abs/2002.11212>.
- [9] Salvi, P.; Várady, T.: G^2 surface interpolation over general topology curve networks. *Computer Graphics Forum*, 33(7), 151–160, 2014. <http://doi.org/10.1111/cgf.12483>.
- [10] Salvi, P.; Várady, T.; Rockwood, A.: Ribbon-based transfinite surfaces. *Computer Aided Geometric Design*, 31(9), 613–630, 2014. <http://doi.org/10.1016/j.cagd.2014.06.006>.
- [11] Vaitkus, M.; Várady, T.; Salvi, P.; Sipos, Á.: Multi-sided B-spline surfaces over curved, multi-connected domains. *Computer Aided Geometric Design*, 89, 102019, 2021. <http://doi.org/10.1016/j.cagd.2021.102019>.
- [12] Várady, T.; Salvi, P.; Karikó, G.: A multi-sided Bézier patch with a simple control structure. *Computer Graphics Forum*, 35(2), 307–317, 2016. <http://doi.org/10.1111/cgf.12833>.
- [13] Várady, T.; Salvi, P.; Rockwood, A.: Transfinite surface interpolation with interior control. *Graphical Models*, 74(6), 311–320, 2012. <http://doi.org/10.1016/j.gmod.2012.03.003>.
- [14] Wang, W.; Jüttler, B.; Zheng, D.; Liu, Y.: Computation of rotation minimizing frames. *ACM Transactions on Graphics (TOG)*, 27(1), 1–18, 2008. <http://doi.org/10.1145/1330511.1330513>.
- [15] Zheng, J.; Ball, A.A.: Control point surfaces over non-four-sided areas. *Computer Aided Geometric Design*, 14(9), 807–821, 1997. [http://doi.org/10.1016/S0167-8396\(97\)00007-1](http://doi.org/10.1016/S0167-8396(97)00007-1).