New multi-sided patches for curve network-based design

Tamás Várady, Péter Salvi

Budapest University of Technology and Economics

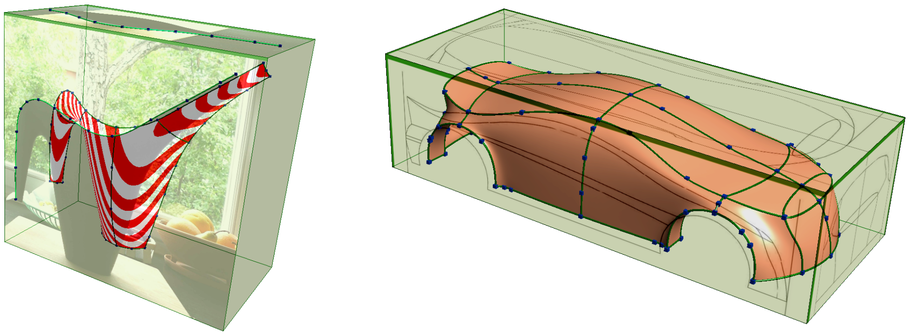
Abstract. An important problem in Computer Aided Design is to create digital representations for complex free-form objects that produce nice, predictable shapes and facilitate real-time editing in 3D. Curve network-based design is such an approach; the clue to this technology is the construction of smoothly connected multi-sided patches. A new type of transfinite surface, called *Composite Ribbon* (CR) patch, has been proposed recently in [12], that combines *curved ribbon surfaces* over a non-regular, convex polygonal domain. This paper explores important aspects of this formulation by introducing an enhanced ribbon construction with a new parameterization scheme using Wachspress coordinates. The new scheme will be demonstrated and evaluated through a few simple examples.

1 Introduction

Creating general topology free-form objects, composed of smoothly connected surface patches, is a fundamental problem in Computer Aided Geometric Design. Aesthetic appearance is crucial for a wide variety of models including cars, household appliances, office furniture, containers and many others. While the majority of such patches are four-sided, almost all industrial objects contain *general n-sided patches* that need to be inserted into some arrangement of quadrilaterals. Two examples are shown in Figures 1a and 1b.

General topology surfacing is a tough problem, and the most widely used techniques expose certain deficiencies for designers and CAD users. First we compare the "pros and cons" of these with the multi-sided approach. A very simple example is shown in Figure 2a, where a 3D curve network of adjacent 5- and 6-sided patches is given to be directly edited and then interpolated.

The standard approach is to combine a collection of bi-parametric (generally NURBS) surfaces, then create *trimmed patches* through a sequence of surface intersections, finally stitch these into a single model, providing tolerance-controlled numerical continuity. The fundamental problem is that the original boundaries of the four-sided patch and the trimming curves have different representational form and design flexibility. For example, creating a truly symmetric three-sided patch is not possible in the four-sided domain. Or take the example in Figure 2b: it is not obvious at all, how to obtain four boundaries by extending the given curve segments. There is no known technique to smoothly connect two



(a) Design by 2D and 3D curves (b) Concept car defined by orthogonal sketches

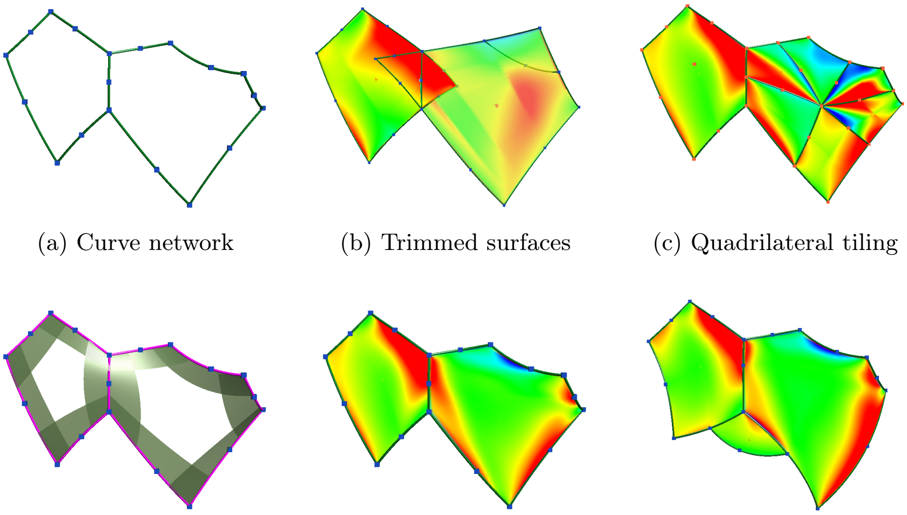
Fig. 1: Models containing general n-sided patches.

parametric surfaces along their "common" trimming curve. Shape editing of the common trimming curves is practically impossible, as that would modify the whole surface representation with their related cross-continuity constraints.

Another approach is splitting n-sided regions into smaller *quadrilateral tiles*, as shown in Figure 2c. Here the main difficulty is to find an appropriate center point in 3D, adding suitable subdividing curves and producing smoothness between the tiles. The figure shows the uneven curvature distribution of this approach. It is an advantage that the network can be edited and the n-patches can be connected, but the price is that surface quality is likely to be disappointing.

A third, well-known and popular surfacing approach, widely used in the animation industry, is based on *recursively subdividing* general topology control polyhedra [3]. This yields a set of smoothly connected quadrilaterals combined with n-sided surface patches, however, difficulties here include the "ab initio" creation of a good control polyhedra and directly interpolating and editing prescribed free-form curves with tangential constraints.

In this paper we explore a fourth approach, where the network automatically spans a collection of *multi-sided transfinite patches*. Feature curves may come from 2D sketches or images, or may be directly defined in 3D. These can be edited in a straightforward manner without topological limitations. Any boundary adjustment naturally modifies the adjacent patches, which are automatically connected in a watertight manner by the ribbon construction (see a trivial example in 2f). In this way, users can focus on shape concepts and aesthetic requirements, and do not need to fiddle with representational difficulties. The main advantage of the transfinite approach is that the interior of the shape is solely defined by boundary ribbons, i.e., there is no need to deal with a grid of interior control points. At the same time, transfinite patches also have their deficiencies — the generated ribbons may not always meet user expectations, and surface portions on extruded or rotational surfaces can only be reproduced in approximate sense.



(a) Curve network (b) Trimmed surfaces (c) Quadrilateral tiling

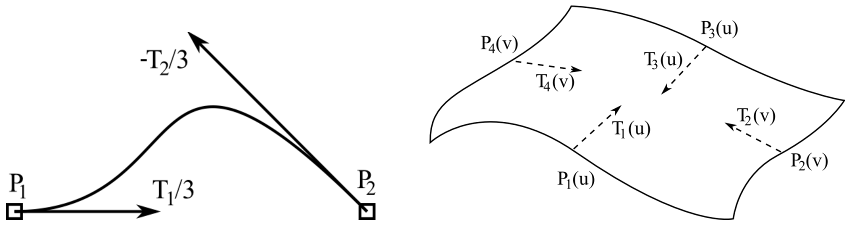
(d) Ribbon surfaces (e) Transfinite surfaces (f) Modified network

Fig. 2: Comparison of different modelling approaches.

Figure 2e illustrates the simplicity of the curve network-based paradigm. From a given curve network, ribbon surfaces are derived for each curve (Fig. 2d), and from these, transfinite patches are computed. The two previous examples (Figures 1a and 1b) have also been defined over general topology 3D networks. When needed, these curves can be modified or their topology redesigned, and the surface geometry will adjust accordingly.

Transfinite surface interpolation is a classical area of CAGD. Its origin goes back to the late 60's, when Coons formulated his Boolean sum surface [2]. In the next two decades, several papers were published, first on triangular patches (see summary in [3]), and later on genuine n-sided patches, including the pioneering work of Charrot and Gregory [1,6], Sabin [11], and Kato [9]. The alternatives of creating n-sided transfinite patches with different blending functions and parameterizations have been recently reviewed by the current authors [13]. New transfinite representations that truly generalize Coons' approach for n-sides have been recently proposed in [12].

In this paper we present enhancements concerning one of these new representations, called *Composite Ribbon* or CR patches. In Section 2 we briefly revisit the classical Coons patch formula, as this will be the starting point of our n-sided surface construction. In Section 3 we introduce the basic constituents of the scheme with emphasis on the construction of curved ribbons and preferred ribbon parameterizations. In Section 4 a few examples demonstrate the natural behavior of CR patches. Suggestions for future work conclude the paper.



(a) Hermite curve (b) Coons patch

Fig. 3: Boundary constraints.

2 Revisiting Coons patches

First we revisit and reformulate basic curve and surface equations in order to introduce our new n-sided surface scheme.

2.1 Hermite curves reformulated

A cubic Hermite curve interpolates four discrete quantities: two endpoints P_1, P_2, and two tangent vectors T_1, T_2, multiplied by the Hermite blending functions

$$\alpha_0(u) = 2u^3 - 3u^2 + 1, \qquad \alpha_1(u) = -2u^3 + 3u^2,$$
$$\beta_0(u) = u^3 - 2u^2 + u, \qquad \beta_1(u) = u^3 - u^2,$$

so

$$r(u) = P_1\alpha_0(u) + P_2\alpha_1(u) + T_1\beta_0(u) + T_2\beta_1(u),$$

see Figure 3a.

A similar curve equation can be formulated by combining two continuous, parametric straight line segments, which we will call *linear ribbons*. A ribbon has its own parameter d_i, and represents a segment with endpoint P_i and tangent T_i as follows:

$$R_i(d_i) = P_i + d_iT_i, \quad d_i \in [0, 1],$$

then the curve can be written as

$$r(u) = \sum_{i=1,2} R_i(d_i)\alpha_0(d_i),$$

using local ribbon parameters computed from the parameter of the curve by $d_1(u) = u$, $d_2(u) = 1 - u$. Note, that only the first Hermite blending function needs to be used and T_2 has been reversed. The above curve is a quartic vector polynomial, that satisfies the end constraints and looks "similar" to the Hermite curve.

Here we propose an enhanced ribbon with non-linear parametrization, as follows:

$$R_i(d_i) = P_i + \gamma(d_i)T_i, \quad d_i \in [0, 1],$$

where $\gamma(d_i)$ is a scalar function. To retain the interpolation properties it is necessary that $\gamma(0) = 0$ and $\gamma'(0) = 1$. If we choose γ as the rational function $\gamma(d_i) = \beta_0(d_i)/\alpha_0(d_i) = \frac{d_i}{2d_i+1}$, then $r(u)$ reproduces the original Hermite curve. This property will be exploited later, as it can be shown that the four-sided CR patch runs very close to the original Coons patch.

2.2 Coons patches reformulated

It is known that the C^1 Coons patch [2] is a four-sided surface S, parameterized in the (u, v) plane $(u, v \in [0, 1])$. It interpolates four boundaries $P_1(u)$, $P_3(u)$, $P_2(v)$, $P_4(v)$ and four cross-derivative functions $T_1(u)$, $T_3(u)$, $T_2(v)$, $T_4(v)$, see Figure 3b. In Coons' original Boolean sum formulation there are three constituents: an interpolant connecting side 1 and 3, another interpolant connecting side 2 and 4, and a term that corrects unwanted artifacts of the two side-to-side interpolants. The correction surface contains a combination of constant vectors at the corners, such as $P_i(0)$, $P_i'(0)$, $T_i(0)$ and $T_i'(0)$. In the C^1 Coons patch cubic Hermite blending functions are used.

In order to reformulate the equation based on the sides of the surface, we introduce cyclic indices (with 1 coming after 4), and so-called *side parameters* $s_i = s_i(u, v)$. The s_i-s are associated with the i-th side of the domain $(i = 1, \dots, 4)$ and take the values of u, v, $1 - u$ and $1 - v$, respectively. For symmetry reasons, the parameterization of the boundaries are reversed along sides 3 and 4. Grouping the positional and tangential constraints of $P_i(s_i)$ and $T_i(s_i)$, and splitting the correction surface into four parts (each one corresponding to one corner), we can rewrite the Coons patch as the composition of four side-based terms minus four corner-based correction terms, as follows:

$$S(u,v) = \sum_{i=1}^{4} \begin{bmatrix} \alpha_0(s_{i+1}) \\ \beta_0(s_{i+1}) \end{bmatrix}^T \begin{bmatrix} P_i(s_i) \\ T_i(s_i) \end{bmatrix} - \sum_{i=1}^{4} \begin{bmatrix} \alpha_0(s_{i+1}) \\ \beta_0(s_{i+1}) \end{bmatrix}^T \begin{bmatrix} P_i(0) & P_i'(0) \\ T_i(0) & T_i'(0) \end{bmatrix} \begin{bmatrix} \alpha_0(s_i) \\ \beta_0(s_i) \end{bmatrix}.$$

The derivative of $T_i(0)$ is the twist vector, which is a corner-specific quantity, also denoted by $W_{i,i-1}$. We assume that the twist vectors related to the $(i-1)$-th and i-th cross-derivative functions are compatible, i.e.,

$$W_{i,i-1} = \frac{\partial}{\partial s_i} T_i(0) = -\frac{\partial}{\partial s_{i-1}} T_{i-1}(1).$$

If the twist vectors are not compatible, Gregory's rational twists need to be used [5,3].

Now let us construct a Coons patch using enhanced linear ribbons (as in Section 2.1), i.e., instead of blending eight one-dimensional vector functions, let us combine four biparametric surfaces similarly to the cubic curves in the previous section:

$$R_i(s_i, d_i) = P_i(s_i) + \gamma(d_i) T_i(s_i).$$

Here we introduce another local parameter, the so-called *distance parameter* $d_i = d_i(u, v)$, that measures a distance from the i-th boundary; for $d_i = 0$

the positional and tangential constraints are satisfied. In the four-sided case $d_i = 1 - s_{i-1} = s_{i+1}$ is an obvious choice. The resulting patch equation is

$$S(u,v) = \sum_{i=1}^{4} R_i(s_i, d_i)\alpha_0(d_i) - \sum_{i=1}^{4} Q_{i,i-1}(s_i, s_{i-1})\alpha_0(s_i)\alpha_1(s_{i-1}), \qquad (1)$$

where the corner correction patches $Q_{i,i-1}$ are given as

$$Q_{i,i-1}(s_i, s_{i-1}) = P_i(0)+\gamma(1-s_{i-1})T_i(0)+\gamma(s_i)T_{i-1}(1)+\gamma(s_i)\gamma(1-s_{i-1})W_{i,i-1}.$$

The ribbons are ruled surfaces, and the correction patches are bilinear surfaces. This formula is identical to the original C^1 Coons patch.

3 Composite ribbon patches

The CR patch is a transfinite surface interpolating $n \geq 3$ three-dimensional boundaries $P_i(s_i)$, $1 \leq i \leq n$, and related cross-derivative functions $T_i(s_i)$. We define the CR patch as a combination of special *curved ribbons*, which, in turn, are composed of the previously introduced linear ribbons. (For the construction of curved ribbons, see Section 3.3.)

Let Γ be a convex polygonal domain in the (u, v) parameter plane, and map the sides of the polygon, Γ_i, onto the boundaries of the patch (see Figure 4). The local side and distance parameters of the ribbons can be computed from (u, v), i.e., $s_i = s_i(u, v)$, $d_i = d_i(u, v)$, and we associate a blending function $B_i(u, v) = B_i(d_1, \ldots, d_n)$ to each side. The CR patch is given as

$$S(u,v) = \frac{1}{2}\sum_{i=1}^{n} C_i(u,v)B_i(u,v), \qquad (2)$$

where $C_i(u, v) = C_i(s_i, d_i)$ denote the curved ribbons. The patch formulation will be explained in Section 3.5.

In order to create this surface, the following constituents must be provided: (i) an n-sided domain polygon, (ii) blending functions, (iii) n ribbon surfaces and (iv) appropriate methods to parameterize the ribbons. The following sections will treat these one by one.

3.1 Domain polygon

Our experience shows that it is preferable to use *non-regular* convex domains to avoid undesirable shape artifacts in extreme geometric configurations. The 2D polygon is supposed to mimic the 3D boundary configuration by arc lengths and angles at the corners. The specifics of domain creation does not concern us here; the reader can find details on different methods in [13].

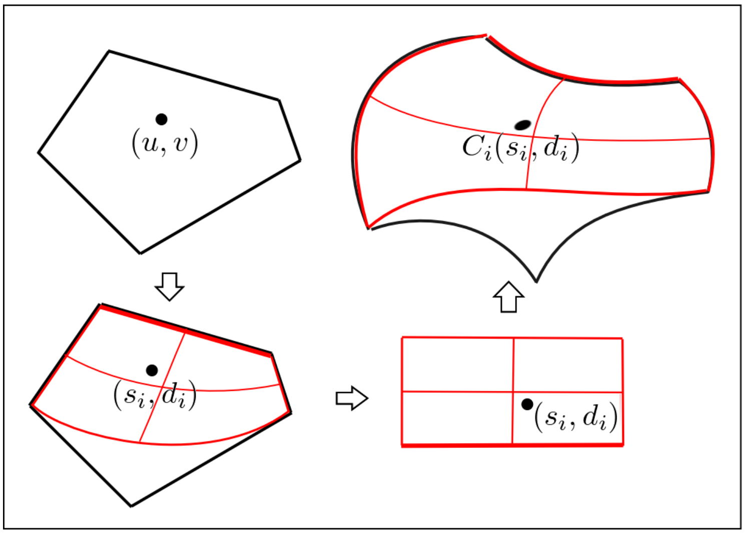


Fig. 4: CR patch: domain — ribbon mapping.

3.2 Generalized blending functions

Assume that we have a polygonal domain, and we need blending functions that reproduce the ribbons along their boundaries. These need to satisfy special interpolating properties as will be shown below. For each (u, v) point we determine an n-tuplet of distance values. Each d_i is associated with the i-th side: d_i is equal to 0 on side Γ_i, and it increases monotonically as we move away from Γ_i. In our patch formulations distance-based rational blending functions are used to combine ribbons.

The basic requirement is that the blending function B_i is equal to 1 on Γ_i, and vanishes on all non-adjacent sides Γ_j, where $j \notin \{i-1, i, i+1\}$, see Figure 5. We propose the rational function

$$B_i(d_1, \dots, d_n) = \frac{D_{i,i-1} + D_{i+1,i}}{\sum_j D_{j,j-1}},$$

where $D_{i_1 \dots i_k} = \prod_{j \notin \{i_1 \dots i_k\}} d_j^2$. Due to the squared terms, the related partial derivatives of the blending functions vanish, i.e., $\frac{\partial}{\partial d_k} B_i(d_1, \dots, d_j = 0, \dots, d_n) = 0$ for $j \notin \{i-1, i+1\}$, $k \in [1 \dots n]$.

3.3 Ribbon surfaces

Let us assume that the tangential boundary information has already been specified by the user, or computed automatically based on the given curve network, i.e., in addition to P_i, the T_i functions are also given. We have already seen how these can constitute linear ribbons. Nevertheless, it can be observed that

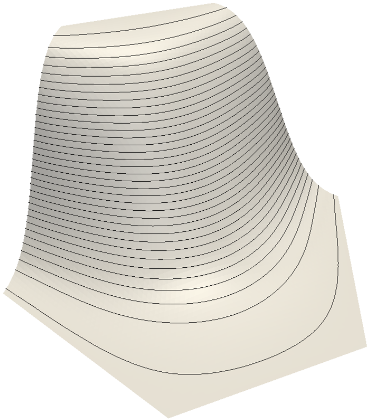


Fig. 5: A blending function over a six-sided domain.

linear ribbons may largely deviate from the expected surface, which spoils the predictability of the shape interior, and in unfortunate cases may bring in unexpected surface artifacts. This motivated us to apply curved interpolants in the CR scheme, as these are supposed to run much closer to the final n-sided patch, and their combination produces better shapes in strongly asymmetric boundary configurations.

A *curved ribbon* is defined as the combination of three consecutive linear ribbons, and it is actually a *Coons patch with three of its four sides given,* defined over a local rectangular domain. Let $C_i(s_i, d_i)$ denote the curved ribbon for the i-th side. We simplify the notation and drop the indices of s and d, as it does not cause any ambiguity. The definition of C_i is as follows (see Fig. 6):

$$\begin{aligned} C_i(s,d) &= R_i^l(s,d)\alpha_0(s) + R_i(s,d)\alpha_0(d) + R_i^r(s,d)\alpha_1(s) \\ &\quad - \left[Q_i^l(s,d)\alpha_0(s)\alpha_0(d) + Q_i^r(s,d)\alpha_1(s)\alpha_0(d)\right] \end{aligned}$$

where $R_i^l(s,d)$, $R_i^r(s,d)$, $Q_i^l(s,d)$ and $Q_i^r(s,d)$ denote the ribbons and the correction patches on the left and right sides, respectively. We parameterize these by the local coordinates of the i-th side as follows:

$$\begin{aligned} R_i^l(s,d) &= R_{i-1}(1-d,s) = P_{i-1}(1-d) + \gamma(s)T_{i-1}(1-d), \\ R_i^r(s,d) &= R_{i+1}(d,1-s) = P_{i+1}(d) + \gamma(1-s)T_{i+1}(d), \\ Q_i^l(s,d) &= Q_{i,i-1}(s,1-d) \\ &= P_i(0) + \gamma(s)T_{i-1}(1) + \gamma(d)T_i(0) + \gamma(s)\gamma(d)W_{i,i-1}, \\ Q_i^r(s,d) &= Q_{i+1,i}(d,s) \\ &= P_{i+1}(0) + \gamma(d)T_i(1) + \gamma(1-s)T_{i+1}(0) + \gamma(d)\gamma(1-s)W_{i+1,i}. \end{aligned}$$

Note that due to the above construction, d is constrained to lie in $[0,1]$.

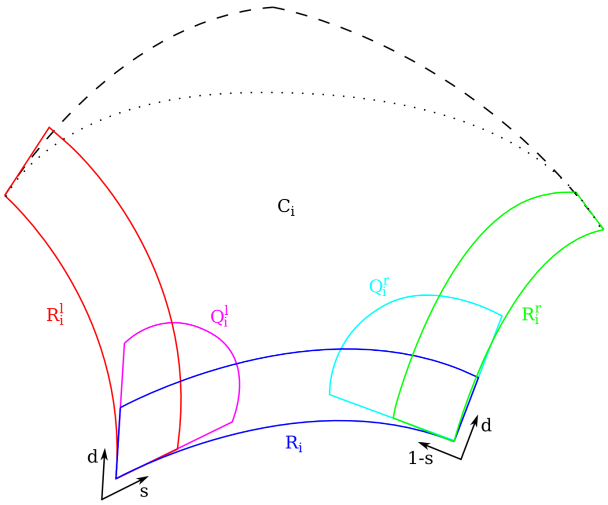


Fig. 6: Construction of a curved ribbon.

Controlling the fourth side. The above equation shows that the fourth side of the curved ribbons is "floating" — it is a by-product of the three interpolating linear ribbons. While this is satisfactory in the majority of cases, it may be advantegeous to explicitly constrain the fourth side, as it offers further degrees of freedom to adjust the shape. A possible solution is to indirectly define the fourth side by an appropriate parametric cubic curve $P_i^{\mathrm{aux}}(s)$. This can be implemented by adding a correction term to the curved ribbon equation:

$$C_i^*(s,d) = C(s,d) + \left(P_i^{\mathrm{aux}}(s) - C(s,1)\right) 4\alpha_0(s)\alpha_1(s)\alpha_1(d).$$

$P_i^{\mathrm{aux}}(s)$ has the same endpoints as the default fourth boundary $C(s,1)$ and shares common slopes at the distant corners of the ribbon.

3.4 Ribbon parameterization overview

The most crucial issue in all transfinite surface schemes is ribbon parameterization, i.e., how to compute the local side and distance parameters (s_i, d_i) from a given (u,v) domain point, see Figure 4. This determines the associated points of the ribbons and thus has an essential effect on the shape.

We have seen the requirement that $d_j \in [0,1]$ $(j \in [1 \dots n])$; it is also natural to require that each side parameter s_j is linear, and for a point on Γ_i

$$s_i \in [0,1], \qquad d_i = 0, \qquad s_{i-1} = 1, \qquad s_{i+1} = 0. \tag{3}$$

The distance parameters d_j $(j \in [1 \dots n])$ also change linearly along the sides, so on the i-th side

$$d_{i-1} = s_i, \qquad d_{i+1} = 1 - s_i. \tag{4}$$

In the evaluation of parameterization methods there are two main issues: (i) the constant s_i, d_i parameter lines must have an even distribution in the domain, and (ii) the $(u, v) \to (s_i, d_i)$ mappings must be simple and computationally efficient. Let us deal with the s_i and d_i parameters separately.

In the so-called *linear sweep* parameterizations the s_i = const. isolines are straight lines in the domain space; as s_i varies from 0 to 1 these lines sweep from side Γ_{i-1} to side Γ_{i+1}, for example using a linear mapping between them.

As for the d_i = const. isolines, applying Wachspress coordinates [14,4] turned out to be a good solution. Originally these assign weights to the corners of a polygon, but it is possible to compute distance isolines by them, as follows. The barycentric coordinates λ_i are defined as

$$\lambda_i(u, v) = w_i(u, v) / \sum_k w_k(u, v),$$

where the individual weights are computed by

$$w_i(u, v) = C_i/(A_{i-1}(u, v) \cdot A_i(u, v)).$$

Here $A_{i-1} = \triangle(p_{i-1}, (u, v), p_i)$, $A_i = \triangle(p_i, (u, v), p_{i+1})$ and $C_i = \triangle(p_{i-1}, p_i, p_{i+1})$ represent triangle areas [8], see Figure 7.

Then the distance d_i is computed as

$$d_i(u, v) = 1 - (\lambda_{i-1}(u, v) + \lambda_i(u, v)),$$

which satisfies equations (3) and (4) and edge linearity, due to the properties of Wachspress coordinates.

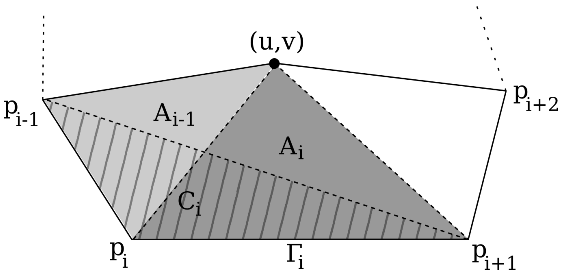


Fig. 7: Triangle-based construction of Wachspress coordinates.

An example using the above construction of s_i and d_i is shown in Figure 8a. As a side note, the distribution of the d_i isolines can be improved, if we force the $d_i = \frac{1}{2}$ isolines to go through the center point of the domain. This can be done by a quadratic reparameterization, as follows (see Figure 8b):

$$\hat{d}_i = d_i \cdot \left[(1 - s_i)^2 + 2(1 - s_i)s_i \cdot (2\mu_i - 1) + s_i^2\right],$$

where the μ_i-s are scaling weights assigned to each side. These can be precomputed using a binary search, such that d_i equals to $\frac{1}{2}$ at the center of the domain polygon.

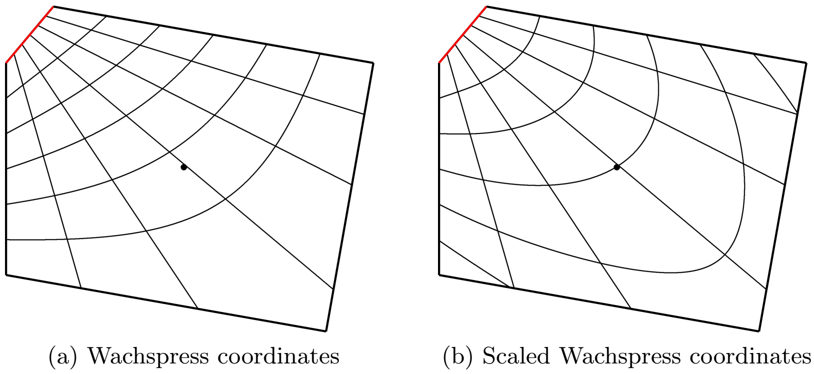

(a) Wachspress coordinates (b) Scaled Wachspress coordinates

Fig. 8: A five-sided domain with linear s_i and barycentric d_i parameter lines.

3.5 Assembling the composite ribbon patch

The CR patch has the simple formula already shown in (2):

$$S(u, v) = \frac{1}{2} \sum_{i=1}^{n} C_i(u, v) B_i(u, v).$$

According to the characteristics of the B_i blend functions, for any point on the i-th boundary all addends of the sum vanish except C_{i-1}, C_i and C_{i+1}. Since each of these ribbons also interpolates the corresponding three boundaries, the related three points on these ribbons are the same. Their cumulative blend is

$$B_{i-1} + B_i + B_{i+1} = (B_{i-1} + B_{i+1}) + B_i = 1 + 1 = 2,$$

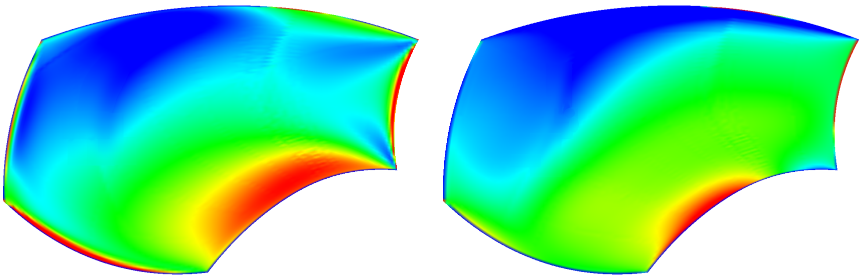
which explains the division by two in the surface equation.

This patch does not satisfy the boundary derivative constraints in strict parametric sense (C^1), i.e., the multi-sided patch and the ribbon may have different tangent vectors at a given boundary point. However, it does share a common tangent plane with the ribbon along the boundary, so G^1 continuity remains valid (see proof in [12]), which is sufficient for surface generation.

4 Examples

Former side-based transfinite schemes combined linear ribbons and applied different blending functions, see for example [10]. While these patches are computationally simple, they could produce uneven curvatures in the vicinity of the boundaries. In fact, the main motivation to develop our new schemes — such as the CR patch — was to avoid these artifacts, see Figure 9.

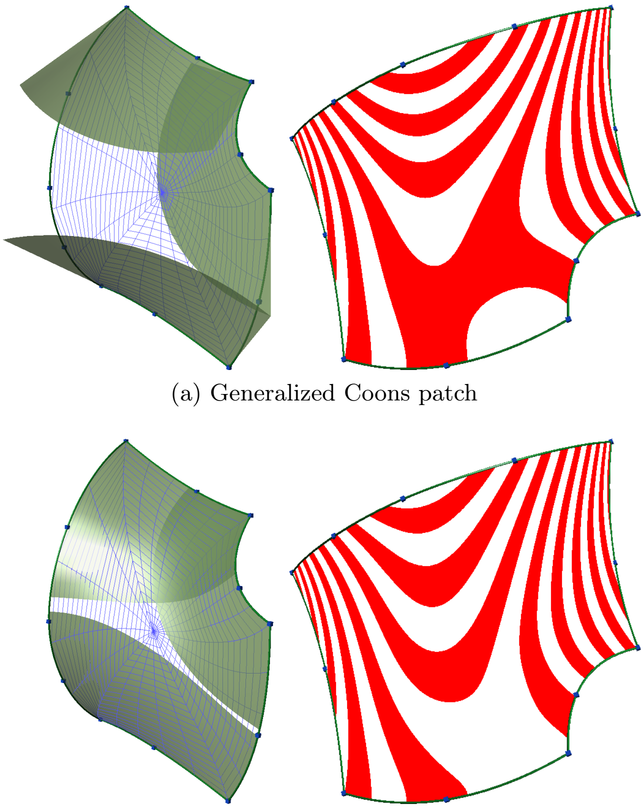
Figure 10 shows two patches wih the same boundary constraints, where linear and curved ribbons were combined to obtain multi-sided patches, respectively.



(a) Conventional patch

(b) Composite ribbon patch

Fig. 9: Mean map comparison of a surface using different patch types.



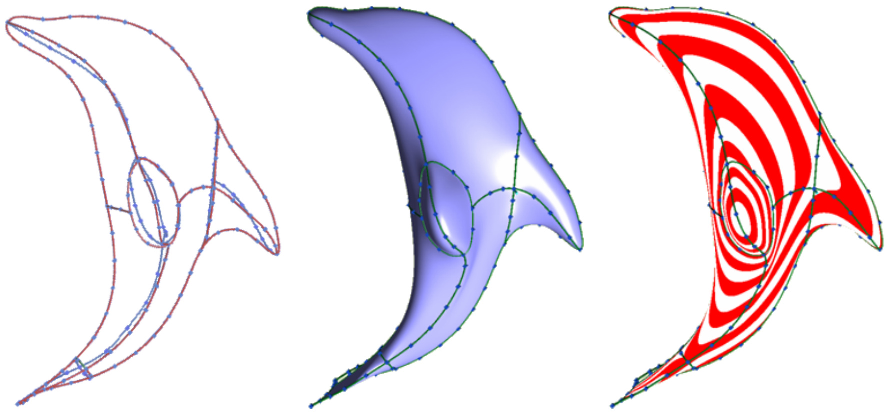
(a) Generalized Coons patch

(b) Composite ribbon patch

Fig. 10: Ribbons and slicing map of a five-sided boundary configuration.

The first figure shows an alternative scheme (directly generalized Coons patches, see [12]), while the second shows CR patches with curved ribbons.

The curve network in the third example comes from a 3D drafting system (courtesy of Cindy Grimm [7]). The network was interpolated by our CR patches, yielding the model in Figure 11.



(a) Curve network (b) Shading (c) Contouring

Fig. 11: Dolphin test model.

Conclusion

In this paper we have focused on the most crucial part of curve network-based design, i.e., how to represent collections of multi-sided transfinite surface patches that naturally fit onto general topology networks, *and* — at the same time — make shape editing easy and naturally predictable. In the CR approach curved ribbons comprising positional and tangential constraints are combined. We have presented an enhanced ribbon construction together with a simple and efficient parameterization using barycentric coordinates. Challenging future research topics include fairing operations for curve network-based models and polygonal mesh approximation by means of transfinite patches, exploiting the free parameters of the ribbon interpolants.

Acknowledgments

This work was partially supported by the scientific program "Development of quality-oriented and harmonized R+D+I strategy and functional model at the Budapest University of Technology and Economics" (UMFT-TÁMOP-4.2.1/B-09/1/KMR-2010-0002) and a grant by the Hungarian Scientific Research Fund

(OTKA, No. 101845). The pictures in this paper were generated by the Sketches system developed by ShapEx Ltd, Budapest; the contribution of György Karikó is highly appreciated.

References

1. Peter Charrot and John A. Gregory. A pentagonal surface patch for computer aided geometric design. *Computer Aided Geometric Design*, 1(1):87–94, 1984.
2. S. A. Coons. Surfaces for computer-aided design of space forms. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1967.
3. Gerald Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition, 2002.
4. Michael S. Floater and Kai Hormann. personal communication, 2012.
5. J. A. Gregory. Smooth interpolation without twist constraints. In Robert E. Barnhill and Richard F. Riesenfeld, editors, *Computer Aided Geometric Design*, pages 71–88. Academic Press, Inc., 1974.
6. J. A. Gregory. N-sided surface patches. In *The Mathematics of Surfaces*, pages 217–232, USA, 1986. Oxford University Press.
7. Cindy Grimm and Pushkar Joshi. Just drawit: a 3d sketching system. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, SBIM '12, pages 121–130, Aire-la-Ville, Switzerland, 2012. Eurographics Association.
8. Kai Hormann and Michael S. Floater. Mean value coordinates for arbitrary planar polygons. *ACM Trans. Graph.*, 25:1424–1441, 2006.
9. K. Kato. Generation of n-sided surface patches with holes. *Computer-Aided Design*, 23(10):676–683, 1991.
10. K. Kato. n-sided surface generation from arbitrary boundary edges. In P.-J. Laurent, P. Sablonnière, and L. L. Schumaker, editors, *Curve and Surface Design: Saint-Malo 1999*, Innovations in Applied Mathematics, pages 173–181. Vanderbilt University Press, Nashville, TN, 2000.
11. Malcolm Sabin. Transfinite surface interpolation. In *Proceedings of the 6th IMA Conference on the Mathematics of Surfaces*, pages 517–534, New York, NY, USA, 1996. Clarendon Press.
12. Péter Salvi. *Fair Curves and Surfaces.* PhD thesis, Eötvös Loránd University, Budapest, 2012.
13. Tamás Várady, Alyn Rockwood, and Péter Salvi. Transfinite surface interpolation over irregular n-sided domains. *Computer Aided Design*, 43:1330–1340, 2011.
14. Eugene L. Wachspress. *A rational finite element basis.* Academic Press, New York, 1975.