Transfinite surface interpolation over specific curvenet configurations

By Péter Salvi[†], Tamás Várady[†], Alyn Rockwood[‡]

[†]Budapest University of Technology and Economics
[‡]InterNext Graphics Institute, Utah

Abstract

In curve network-based design, objects consist of a collection of smoothly connected multi-sided patches. Transfinite surface interpolation is a favorable representation; these patches are determined exclusively by means of n boundary ribbon surfaces carrying positional and cross-derivative constraints to be interpolated. Interest in transfinite interpolation has been recently revitalized by new schemes to generalize Coons' four-sided patch to arbitrary n sides. While the majority of multi-sided surfaces are defined over convex domains combining linear ribbons, there are design situations, where applying convex domains is insufficient or impossible. The goal of our paper is to process curve networks with concave angles and internal holes. In the proposed scheme, subdividing curves are supplemented and parabolic ribbons are used to provide G^2 continuity. Gregory's generalized correction terms are applied to handle ribbons that are not compatible at the vertices. A simple construction to create parabolic ribbons is also discussed. A few examples illustrate the difficulties and the solutions.

1. Introduction

In curve network-based design, objects are directly defined by a collection of curves, arranged into a single 3D network with general topology. Curves may come from (i) sketch input, (ii) feature curves extracted from orthogonal views, (iii) curves traced on triangular meshes or (iv) direct 3D editing. It is assumed that the network automatically determines so-called ribbon surfaces that carry positional and cross-derivative constraints to be interpolated. The key to curve network-based design is transfinite surface interpolation, where no grid of control points is needed to define the shape interior, and all n boundaries are handled uniformly, unlike in the case of trimmed quadrilateral surfaces. The ability to naturally edit prescribed boundaries and cross-derivatives is also an advantage in contrast to recursive subdivision schemes.

The majority of multi-sided transfinite surfaces are defined over convex domains, combining linear ribbon surfaces and enabling G^1 continuity between the adjacent patches. At the same time, there are many practical design situations, where this approach may not work. In these cases additional curves need to be supplemented, which must be compatible with the already existing ribbons, and thus the extended network becomes suitable for applying convex methods. Typical examples include cases when two curves span a concave angle at a common vertex, or when disjoint loops with prescribed slopes need to be interpolated. It is particularly important to produce seamless connections within this sort of composite surfaces; this motivated our work to introduce parabolic ribbons for ensuring G^2 continuity.

The roots of transfinite surface interpolation go back to Coons' classical work [3], that generalizes for C^2, as well. Potential incompatibilities of adjacent ribbons meeting at a corner were also investigated. This problem can be resolved using Gregory's rational terms [6], which was later generalized to handle C^2 incompatible corners [1, 25]. Research related to non-four-sided transfinite schemes intensified in the eighties and the early nineties including various important publications [2, 17, 9, 16, 18, 11]. Multi-sided patches with G^2 continuity were also published, see [7, 8, 5] amongst others. Serious efforts were directed towards avoiding rational terms, when quadrilateral polynomial surfaces must satisfy compatibility constraints at n-valent vertices [14, 15]. As it will be discussed in Section 3, this question is also related to transfinite surface interpolation, as it determines whether ribbons are entirely unconstrained, or need to be made compatible through a set of algebraic equations. In the unconstrained case it is necessary to use rational correction terms.

An early paper on curvenet-based surfaces was published by [12]. A global fairness functional using the variation of curvature was proposed, yielding a surface model comprised of quadrilaterals (biquintic Bézier patches). Subdivision-based interpolation approaches have also been proposed, using extensions of the Catmull-Clark scheme [10, 21]. Recent advances include Coons patches bounded by geodesic curves [4].

Interest in transfinite surface interpolation has been revitalized recently by the current authors, proposing two new, multi-sided patch schemes [23, 19, 20]. This paper describes one of them, the so-called Generalized Coons (GC) surface, which expands the classical Boolean sum logic for n sides. In Section 2, three examples of specific curvenet configurations are presented, where ordinary transfinite interpolation would fail. In Section 3, we show how new interior curves can be added, and discuss the construction of G^2-continuous parabolic ribbons. In Section 4, we present the basic principle of the n-sided Coons approach, and discuss methods for constructing non-regular domains, blending functions, and ribbon parameterization. The computation of Gregory-style correction terms to blend independent linear and parabolic ribbons will also be described. Finally, we analyze the constructed composite transfinite patches, and discuss open problems for future research.

2. Motivation

We assume that there is a given network of free-form curves, represented by markers interpolated by B-splines. Based on the network, common tangent planes (and possibly best-fit surface curvatures, see [22]) are computed at the vertices, and bi-parametric ribbon interpolants are defined, that carry positional and cross-derivative information. A pair of linear ribbons, being on the opposite sides of a boundary, will satisfy G^1 continuity, if they share a common normal at each point of the boundary (e.g. using rotation minimizing frame, see [24]). A pair of parabolic ribbons with the same curvature will provide G^2 continuity across adjacent patches. Finally, multi-sided patches interpolating the given ribbons will be constructed.

2.1. *Three examples*

Our first case shows a loop of curves, where two consecutive edges span a concave angle (see Figure 1). Using a convex domain would yield a twisted patch, most likely with self-intersections, due to the distorted parameterization of the ribbons in the domain space. This case can be better modeled, if we insert a new edge, that connects the concave corner with a marker on the opposite boundary, as shown in the figure. This connecting

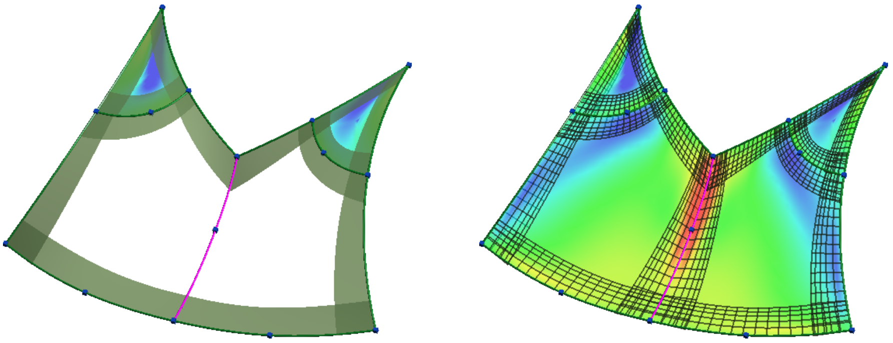
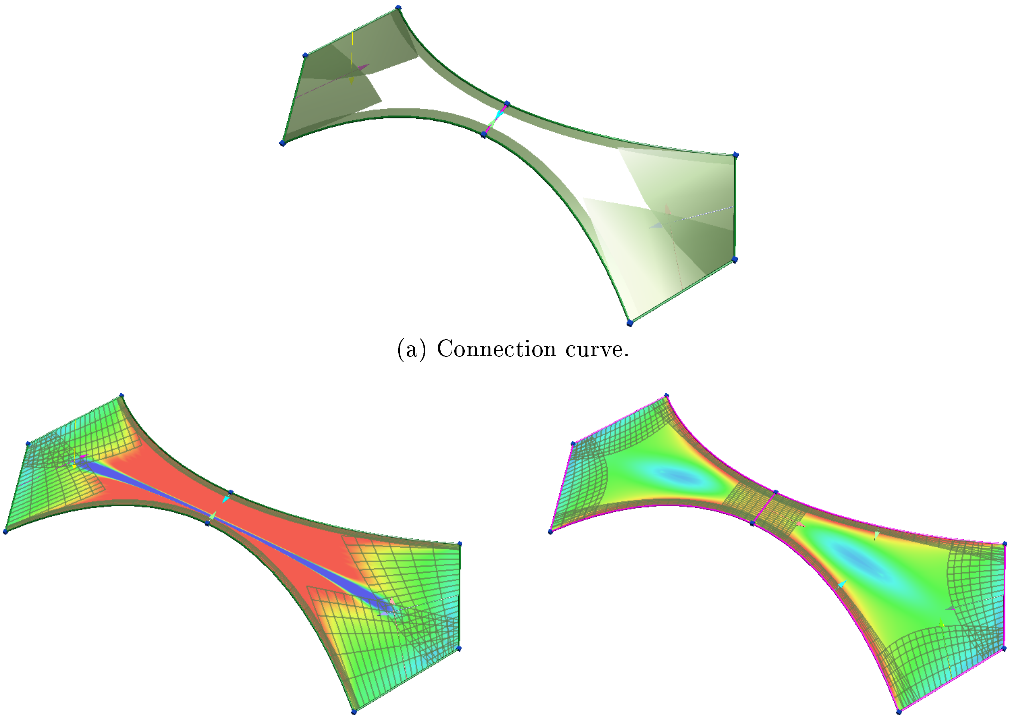


Figure 1: Subdividing a patch due to a concave angle.



(a) Connection curve.

(b) Original mean map.

(c) Mean map with interior ribbons.

Figure 2: Subdividing a patch due to poorly distorted domain.

curve is tangential to the plane of the concave corner and tangentially meets the ribbon at the marker.

The second example is a single patch, where all angles are convex, but the 3D boundary configuration significantly deviates from the shape of the 2D domain (see Figure 2). As a result, the interior of the patch does not produce a shape that meets our expectations. To overcome this problem, an artificial connecting curve is inserted in the middle, that tangentially connects the two opposite ribbons of the patch.

The third example shows two disjoint curve loops (see Figure 3) with already defined ribbons, but no appropriate domain exists for surface construction. As before, we insert

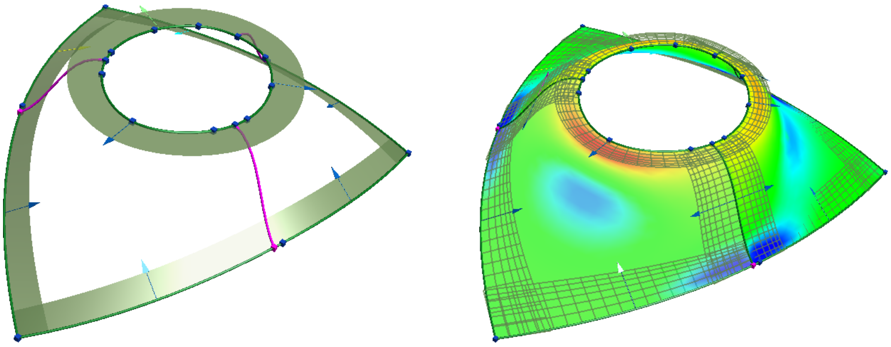


Figure 3: Adding connection curves to connect separate loops of edges.

three connection curves that provide three adjacent loops that now can be interpolated by multi-sided patches with convex domains.

3. Basic algorithms

This section deals with the heuristics of creating connection curves, as well as the definition of linear and parabolic ribbons with or without corner compatibility.

3.1. *Connection curves*

The applied connection curve algorithm can be briefly described as follows. Assume that we have two opposite curves from the network, $P_1(s_1)$ and $P_2(s_2)$, and their first derivatives and prescribed normal vectors are denoted by $D_i(s_i)$ and $N_i(s_i)$, respectively, $i \in \{1, 2\}$. Let us define a sequence of orthogonal sweeping planes $\pi_i(s_i)$ for each curve, which contain the points $P_i(s_i)$ and whose normal vectors are $D_i(s_i)$. The requested connection curve $K_{12}(t)$ connects two opposite curve points, and its tangent vectors are set perpendicular to the given curves, i.e., $\frac{\partial}{\partial t}K_{12}(0) = \pm D_1(s_1) \times N_1(s_1)$ and $\frac{\partial}{\partial t}K_{12}(1) = \pm D_2(s_2) \times N_2(s_2)$, where the signs are determined by the direction of the curves.

The connection curve may connect two fixed markers, i.e., two of the points defining the B-splines curves; then s_1 and s_2 are uniquely defined. Alternatively, the two end positions of the connecting curve are only loosely defined by user-defined sliding markers on the curves. In this case, the best end positions can be computed by a simple iterative algorithm. Substituting point $P_2(s_2)$ into the plane $\pi_1(s_1)$ gives a signed distance, that indicates the direction in which the first sliding marker and its plane should move on the first curve to contain the second point on the other side; this produces an enhanced $s_1^{(1)}$ value. Similarly, substituting point $P_1(s_1)$ into the plane $\pi_2(s_2)$ enhances the position of the sliding marker on the second curve. So we set point $P_2(s_2)$ to lie in $\pi_1(s_1)$, and vice versa. This iteration may not converge; in this case, we keep the initial positions. A simple example in Figure 4 shows how the sliding markers move to their final position after a few steps.

3.2. *Ribbons*

Linear ribbons are defined as $R(s, d) = P(s) + dT(s)$, where $T(s)$ is the cross-derivative; we assume that the functions $P(s)$ and $T(s)$ are given. Two linear ribbons enable G^1 continuity between two transfinite patches, if their normal vectors are parallel, i.e., $\frac{\partial}{\partial s_1}P_1(s_1) \times T_1(s_1) \parallel \frac{\partial}{\partial s_2}P_2(s_2) \times T_2(s_2)$ for each $P_1(s_1) = P_2(s_2)$ point of the common boundary curve.

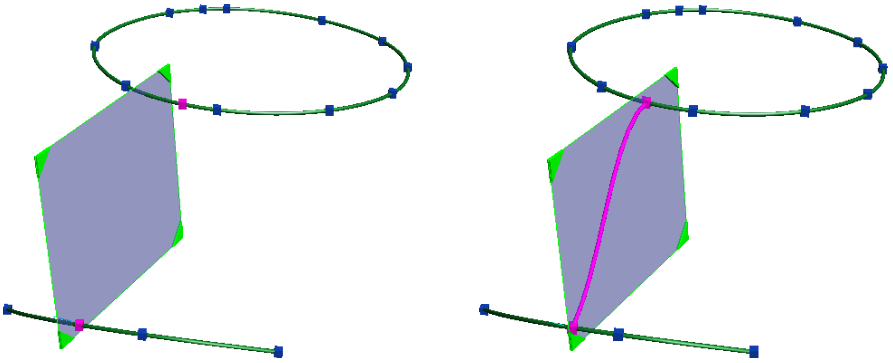


Figure 4: Sliding markers of a connection curve from initial to final positions.

Parabolic ribbons need a smooth second derivative function $C(s)$, as well:

$$R(s,d) = P(s) + dT(s) + \frac{1}{2}d^2C(s), \tag{3.1}$$

where we assume that both the position $P(s)$ and the derivative vectors $T(s)$ and $C(s)$ lie in the sweeping plane $\pi(s)$. Two parabolic ribbons enable G^2 continuity between two transfinite patches, if the corresponding parabolic arcs are contained in the same sweeping plane, and their normal curvature match for each point of the common boundary curve (due to the well-known Linkage Curve Theorem [13]), i.e., $\kappa_1(s_1) = \kappa_2(s_2)$ holds for all s_1, s_2, where $P_1(s_1) = P_2(s_2)$, and

$$\kappa_i(s_i) = \frac{\|T_i(s_i) \times C_i(s_i)\|}{\|T_i(s_i)\|^3}.$$

To compute common normal vector functions along a curve is relatively easy by interpolating the constrained normal vectors at the vertices of the network, however, to prescribe a common cross-curvature sweep along a curve is a difficult problem; poor values may destroy surface quality in the interior of the patches. In order to be on the safe side, we propose to compute parabolic ribbons by first creating G^1 transfinite patches, then computing local curvatures along the common boundary in given sweeping planes, and set a common target curvature as the average of the two opposite curvature values.

We can transform the parabolic arc into a local coordinate system, where the first point is the origin, and the tangent of the arc is the local x-axis. Then at a given boundary point the equation of a parabola can be written as a quadratic Bézier curve $R^*(d) = 2(1-d)d(\alpha l, 0) + d^2(l,h)$, yielding $\kappa = \frac{h}{2\alpha^2 l^2}$ (see Figure 5). Assuming that the width of the parabolic ribbon is the same as the corresponding linear one, l is already defined. Then the prescribed curvature κ can be set by means of α and h, which define the second and third control points of the parabolic arc. Transforming these points back into the 3D space defines the parabolic ribbon $R(s,d)$ at those positions.

3.3. *Ribbon compatibility at the corners*

In the previous paragraphs we have dealt with opposite ribbons sharing a common curve, now we look at two adjacent ribbons sharing a common vertex. As noted earlier, in order to obtain a valid transfinite patch, the ribbons must satisfy certain constraints, or must be made compatible. Let us denote the transfinite patch by $S(u,v)$, where (u,v) are the local parameters of the patch, and $S(0,0)$ is a corner point. Let us assume that we have two parabolic ribbons here, denoted by $U(u,v)$ and $V(u,v)$, parameterized in accordance

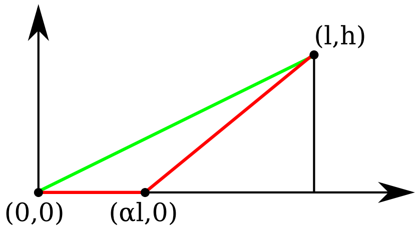


Figure 5: Setting the control points of a parabolic ribbon.

with $S(u,v)$. Let us assign $U(u,0)$ to $S(u,0)$, and $V(0,v)$ to $S(0,v)$. There are three basic cases:

(a) The ribbons are compatible, thus the transfinite patch equation is well-defined, all derivatives and mixed partial derivatives $\frac{\partial}{\partial u}$, $\frac{\partial}{\partial v}$, $\frac{\partial^2}{\partial u\partial v}$, $\frac{\partial^2}{\partial u^2}$, $\frac{\partial^2}{\partial v^2}$, $\frac{\partial^3}{\partial u^2\partial v}$, $\frac{\partial^3}{\partial u\partial v^2}$, $\frac{\partial^4}{\partial u^2\partial v^2}$ of U and V are equal at $(0,0)$. This means that the ribbons must satisfy a set of algebraic constraints [15]. To produce such compatible ribbons automatically from a given curve network is a hard task.

(b) Another case is when the ribbons are totally *independent,* and the only constraint is that the corner point at $S(0,0)$ is common. In this case we may have two different values for all the above derivatives, and in order to create a valid transfinite patch, we need to apply the Gregory-style correction terms $\frac{\partial^{i+j}}{\partial u^i\partial v^j}S(0,0) = W_{i,j}(u,v)$, where $W_{i,j}(u,v)$ is a rational function combining $\frac{\partial^{i+j}}{\partial u^i\partial v^j}U(0,0)$ and $\frac{\partial^{i+j}}{\partial u^i\partial v^j}V(0,0)$. (For details, see Section 4.4.)

(c) There exists a third case, which we call a *hybrid* solution. It is well-known, that if all curves match an associated surface curvature computed at a given vertex, this provides a sufficient condition to set compatible twists by means of a relatively easy local constraint. In this case the $\frac{\partial}{\partial u}$, $\frac{\partial}{\partial v}$, $\frac{\partial^2}{\partial u\partial v}$, $\frac{\partial^2}{\partial u^2}$, $\frac{\partial^2}{\partial v^2}$ derivatives of U and V are identical, and only the remaining three terms $\frac{\partial^3}{\partial u^2\partial v}$, $\frac{\partial^3}{\partial u\partial v^2}$, $\frac{\partial^4}{\partial u^2\partial v^2}$ must be computed by Gregory's correction.

In the next section we will describe how generalized Coons patches work. We will assume that the parabolic ribbons are independent. This helps optimizing ribbons without a highly complex algebraic system of equations, and it also facilitates interactive editing of transfinite patches.

4. Generalized Coons patches

The basic idea of Generalized Coons (GC) patches were first introduced in [23]. The necessary conditions for C^1 continuity, together with new, efficient parameterization schemes were elaborated in [19]. In this section, we recall important elements from the former publication, and also discuss some of the new enhancements. The original GC surfaces interpolated linear ribbons only; the scheme presented here further generalizes it to handling parabolic ribbons, as well.

Let $P_i(s_i)$, $T_i(s_i)$, $C_i(s_i)$, $1 \leq i \leq n$ $(n \geq 3)$ denote the position, first cross-derivative, and second cross-derivative of the boundaries, respectively. A quadrilateral ribbon surface can be written as

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

where γ and δ are rational reparameterization functions, and d_i is the *distance parameter* (as opposed to s_i, the *side parameter*). This is a more complex equation than the previous

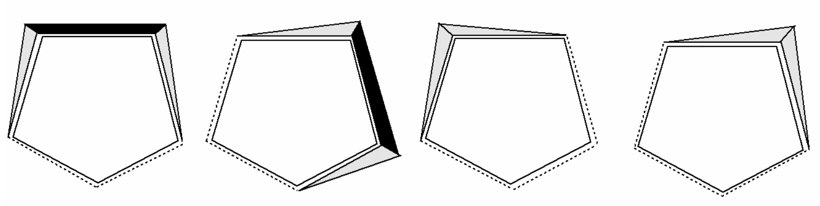


Figure 6: Schematic representation of two ribbon, and two corner interpolants.

one given in Equation (3.1). We were motivated to use these terms to obtain a multi-sided surface that matches the 4-sided C^2 Coons patch with the same ribbons; without these functions, the shape of the 4-sided GC would be a different patch with a much stronger fullness. The functions $\gamma(d)$ and $\delta(d)$ can be derived in a straightforward manner from the quintic Hermite functions:

$$\gamma(d) = \frac{H_1^5(d)}{H_0^5(d)} = \frac{3d^2+d}{6d^2+3d+1}, \quad \delta(d) = \frac{H_2^5(d)}{H_0^5(d)} = \frac{d^2}{12d^2+6d+2},$$

where $H_0^5(d) = 1 - 10d^3 + 15d^4 - 6d^5$, $H_1^5(d) = d - 6d^3 + 8d^4 - 3d^5$, and $H_2^5(d) = \frac{1}{2}d^2 - \frac{3}{2}d^3 + \frac{3}{2}d^4 - \frac{1}{2}d^5$ are the quintic Hermite blending functions. These reparameterization functions satisfy the following requirements: $\gamma(0) = \gamma''(0) = \delta(0) = \delta'(0) = 0$, and $\gamma'(0) = \delta''(0) = 1$.

We assume that the patch is defined over a convex polygonal domain Γ in the (u, v) parameter plane, and the sides of the polygon Γ_i are mapped to the boundaries of the patch. The local side and distance parameters of the ribbons are computed from (u, v), i.e., $s_i = s_i(u, v)$, and $d_i = d_i(u, v)$.

Following the idea of the Coons patch, the GC patch is also a Boolean sum of ribbons and correction surfaces, weighted by appropriate blending functions B_i (for sides) and $B_{i,i-1}$ (for corners):

$$S(u,v) = \sum_{i=1}^{n} R_i(s_i, d_i) \cdot B_i(d_1, \ldots, d_n) - \sum_{i=1}^{n} Q_{i,i-1}(s_i, s_{i-1}) \cdot B_{i,i-1}(d_1, \ldots, d_n),$$

where $Q_{i,i-1}(s_i, s_{i-1})$, $1 \le i \le n$ are correction patches. The basic idea is illustrated in Figure 6. Loosely speaking, the i-th ribbon interpolant, multiplied with its blend function, reproduces the prescribed boundary data (black area), and creates superfluous components (grey areas) along its adjacent sides $i-1$ and $i+1$. A corner interpolant, multiplied by its blend function, is to compensate the superfluous components (the grey areas) coming from the $(i-1)$-th and i-th ribbons, therefore in the end the patch will interpolate the boundary functions, as expected.

For creating a GC patch, the following constituents must be provided: (i) n ribbon surfaces, (ii) an n-sided domain polygon, (iii) blending functions, (iv) appropriate methods to parameterize the ribbons, and (v) n correction patches. We have already dealt with the creation of parabolic ribbons; all the other constituents will be explained briefly in the rest of this section.

4.1. *Domain polygon*

It is a widely accepted principle in CAGD, that the parameterization of the domain and the mapped 3D objects should have a "similar" shape. In other words, the domain should be mapped into 3D with minimal distortion. For example, non-uniform B-splines are defined in this fashion, thus avoiding undesirable overshoots.

Denote the arc lengths of the given three-dimensional boundary curves by L_i, and the angles between the end tangents of the $(i-1)$-th and i-th boundaries by ϕ_i. Denote the side lengths and the angles of the domain by l_i and α_i, respectively, then we seek to minimize the squared deviation of the chord lengths and the angles, i.e., $\sum_i (l_i - c_{\text{length}} L_i)^2 + \sum_i (\alpha_i - c_{\text{angle}} \phi_i)^2$, where c_{length} and c_{angle} are properly chosen constants. This is a non-linear expression, but practice shows that simple heuristic methods — such as circular polygonal domains or edge tweaking — can yield reasonable solutions, see details in [23]. The above minimization error characterizes the extent of domain distortion, thus it gives a hint about whether splitting the domain is likely to improve the shape (see the second example in Section 2).

4.2. *Blending functions*

The blending functions need to satisfy special interpolating properties. For each (u, v) point we determine n distance values d_i $(1 \le i \le n)$. 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:

$$B_{i,i-1}(d_1, \dots, d_n) = \frac{D_{i,i-1}}{\sum_j D_{j,j-1}}, \quad D_{i,i-1} = \prod_{j \notin \{i,i-1\}} d_j^3,$$
$$B_i(d_1, \dots, d_n) = B_{i,i-1}(d_1, \dots, d_n) + B_{i+1,i}(d_1, \dots, d_n).$$

The original G^1 patch contained quadratic distance terms, here cubic terms are needed for G^2 interpolation. It is easy to see, that B_i is equal to 1 on Γ_i, and vanishes on all non-adjacent sides Γ_j, where $j \notin \{i-1, i, i+1\}$. Due to the cubic terms, the related first and second partial derivatives of the blending functions vanish, i.e.,

$$\frac{\partial}{\partial d_k} B_i(d_1, \dots, d_j = 0, \dots, d_n) = 0, \qquad j \notin \{i-1, i+1\},\ k, l \in [1 \dots n]$$
$$\frac{\partial^2}{\partial d_k \partial d_l} B_i(d_1, \dots, d_j = 0, \dots, d_n) = 0.$$

4.3. *Parameterization*

The most crucial issue in transfinite surface generation 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. It is natural to require that each side parameter s_j $(j \in [1 \dots n])$ is linear, and for a point on Γ_i:

$$s_i \in [0, 1], \qquad d_i = 0, \qquad s_{i-1} = 1, \qquad s_{i+1} = 0. \tag{4.1}$$

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

$$d_{i-1} = s_i, \qquad d_{i+1} = 1 - s_i. \tag{4.2}$$

The following properties must also be satisfied for a point on Γ_i:

$$\frac{\partial d_{i-1}}{\partial u} = \frac{\partial s_i}{\partial u}, \quad \frac{\partial d_{i+1}}{\partial u} = -\frac{\partial s_i}{\partial u}, \quad \frac{\partial d_{i-1}}{\partial v} = \frac{\partial s_i}{\partial v}, \quad \frac{\partial d_{i+1}}{\partial v} = -\frac{\partial s_i}{\partial v}. \tag{4.3}$$

Roughly speaking, this means that the adjacent ribbon parameterizatons are identical in a differential sense, see Figure 8.

The following is a brief review of the so-called interconnected parameterization, which

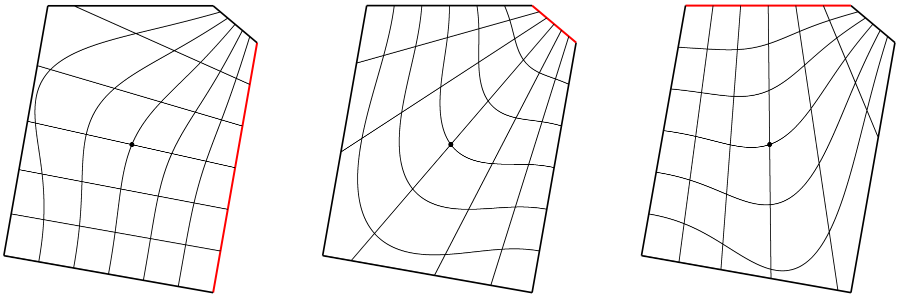


Figure 7: Constant parameter lines of the interconnected parameterization.

was found superior with respect to computational efficiency amongst the various ribbon mapping methods [19].

Take arbitrary functions $s_i(u,v)$ that give 0 for every point on side Γ_{i-1}, and 1 everywhere on Γ_{i+1}. For all other points inside the convex domain they return a value in $[0,1]$. For example, the s coordinates of the bilinear, radial or central line sweeps [23] are such functions. These naturally satisfy $s_i \in [0,1]$, $s_{i-1} = 1$ and $s_{i+1} = 0$. Let us define a blending function $\alpha(t) : [0,1] \to [0,1]$ with $\alpha(0) = 1$ and $\alpha(1) = \alpha'(0) = \alpha'(1) = 0$. Examples include the cubic Hermite blending function, or a rational blend function: $\alpha(t) = \frac{(1-t)^2}{t^2+(1-t)^2}$. Now we can define d_i by means of s_{i-1} and s_{i+1}, as follows:

$$d_i(u,v) = (1 - s_{i-1}(u,v)) \cdot \alpha(s_i(u,v)) + s_{i+1}(u,v) \cdot \alpha(1 - s_i(u,v)).$$

If we are on the i-th side, $s_{i-1} = 1$ and $s_{i+1} = 0$, so $d_i = 0$, satisfying (4.1). Straightforward algebra shows that the requirements (4.2) and (4.3) are also satisfied. A simplified view of this construction is that taking a (u,v) point, we determine three consecutive sweep lines that go through (u,v), and determine d_i as the weighted combination of $d_i = s_{i-1}$ on Γ_{i-1}, and $d_i = s_{i+1}$ on Γ_{i+1}, according to the middle coordinate s_i.

Figure 7 shows constant s and d lines for this parameterization using the central line sweep parameterization [23] for s. This method forces the $s_i = \frac{1}{2}$ lines to go through the center of the domain. The first image is based on the right side of the polygon; the second on the small side at the top-right; and the third on the top side. Observe that all lines of the second image start in the same way (in a differential sense) as their counterparts in the first and third images.

The above figures show that the d constant parameter lines may have inflections. To avoid this, an enhanced parameterization is suggested, using the quartic blend functions

$$\beta_0(t) = (1-t)^4 + 4(1-t)^3 t + c \cdot 6(1-t)^2 t^2,$$
$$\beta_1(t) = c \cdot 6(1-t)^2 t^2 + 4(1-t)t^3 + t^4,$$

where c is an appropriate shape parameter. Then we obtain

$$d_i(u,v) = (1 - s_{i-1}) \cdot \beta_0(s_i) + s_{i+1} \cdot \beta_1(s_i),$$

that results in smooth d constant parameter lines, see Figure 9.

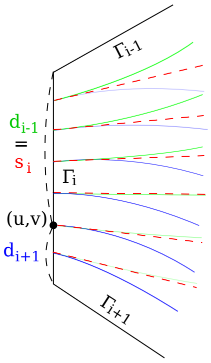


Figure 8: Constrained parameterization.

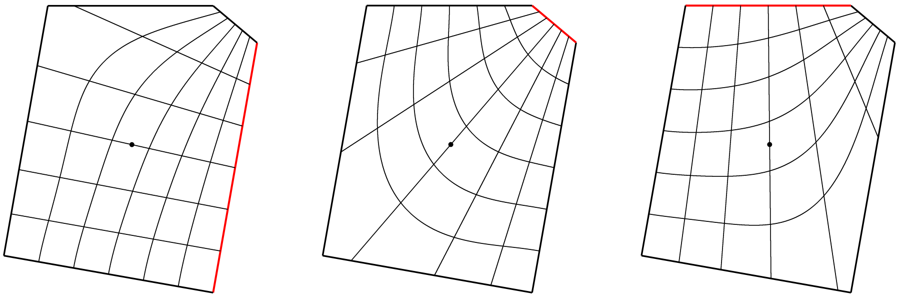


Figure 9: Constant parameter lines of the enhanced interconnected parameterization.

4.4. *Correction terms*

For G^1 and C^1 transfinite interpolation surfaces, it is customary to assume that the boundary curves are compatible, i.e., $\frac{\partial}{\partial s_i}P(0) = T_{i-1}(1)$, etc., and only the twist vectors differ. This is resolved by Gregory's rational twists. In our research, we assume that the boundary curves match only in position, and for all other terms rational expressions are used. This creates another layer of flexibility for shape optimization.

The correction term for the G^2 patch is somewhat complex, comprising nine terms. For ease of notation, let t_i denote $1 - s_{i-1}$. Then

$$\begin{aligned} Q_{i,i-1}(s_i, t_i) = {} & P_i(0) + \gamma(s_i)W_{1,0} + \gamma(t_i)W_{0,1} + \gamma(s_i)\gamma(t_i)W_{1,1} \\ & + \delta(s_i)W_{2,0} + \delta(t_i)W_{0,2} + \delta(s_i)\gamma(t_i)W_{2,1} \\ & + \gamma(s_i)\delta(t_i)W_{1,2} + \delta(s_i)\delta(t_i)W_{2,2}, \end{aligned}$$

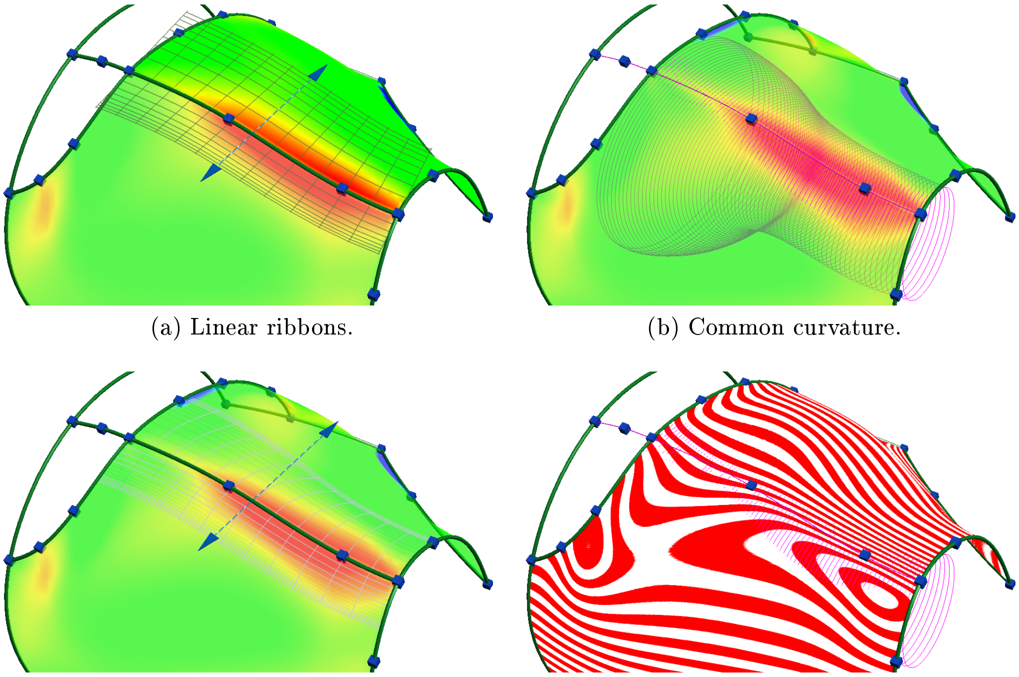
where each W is a rational function of s_i and t_i. The computation of these is fairly straightforward, see [25]. As an illustration, we show two such terms, the remaining ones are similar:

$$\begin{aligned} W_{1,0}(s_i, t_i) &= \frac{s_i^2 T_{i-1}(1) + t_i^3 \frac{\partial}{\partial s_i} P_i(0)}{s_i^2 + t_i^3}, \\ W_{1,2}(s_i, t_i) &= \frac{s_i^2 \frac{\partial^2}{\partial t_i^2} T_{i-1}(1) + t_i \frac{\partial}{\partial s_i} C_i(0)}{s_i^2 + t_i}. \end{aligned}$$

Conclusion

In this paper we have dealt with constructions for curve network-based design, where the initial ribbon information associated with the network is not sufficient to create transfinite patches with convex domains. To resolve this problem, additional connection curves are inserted, that are constrained to be compatible with the existing ribbons. For transfinite interpolation, we use a generalization of Coons' Boolean sum patches. In order to make seamlines practically invisible, parabolic ribbons have been introduced, thus G^2 continuity can be ensured across the boundaries. Figure 10a shows two patches with linear ribbons; this is the basis of computing an average curvature function along the boundary (Fig. 10b), which will be converted into parabolic ribbons (Fig. 10c) for the final patch interpolation — see the isophotes in Fig. 10d.

Quality improvement can also be assessed in Figure 11 by analyzing isophote lines. The first one shows a G^1 boundary with tangentially discontinuous isophotes, while the second one shows smooth isophote stripes indicating G^2 continuity.



(a) Linear ribbons.

(b) Common curvature.

(c) Parabolic ribbons.

(d) Isophotes

Figure 10: Test example.

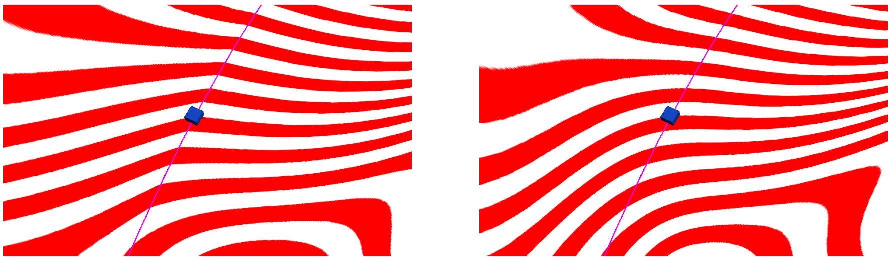


Figure 11: Adjacent patches with G^1 vs. G^2 continuity.

As it was noted earlier, to prescribe a curvature function based on the curve network or by user interaction is a delicate issue, and a poor choice may lead to nice connections along the boundaries, but poor shapes in the interior. There are several open issues in this area, and averaging the curvatures of the existing G^1 patches is only one possible solution. It was also pointed out, that even when the curvature is well-defined, there is some degree of freedom to construct the parabolic ribbons (see shape parameter α in Section 3.2); this is also a tool for further shape optimization to be explored.

Acknowledgements

This work was partially supported by the Budapest University of Technology and Economics (UMFT-TÁMOP-4.2.2.B-10/1-2010-0009), 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] R. Barnhill and W. Boehm, *Computer aided surface representation and design*, in Surfaces in Computer Aided Geometric Design, North-Holland, 1983, pp. 1–24.

[2] P. Charrot and J. A. Gregory, *A pentagonal surface patch for computer aided geometric design*, Computer Aided Geometric Design, 1 (1984), pp. 87–94.

[3] S. A. Coons, *Surfaces for computer-aided design of space forms*, Tech. Rep. MIT/LCS/TR-41, Massachusetts Institute of Technology, 1967.

[4] R. Farouki, N. Szafran, and L. Biard, *Existence conditions for coons patches interpolating geodesic boundary curves*, Computer Aided Geometric Design, 26 (2009), pp. 599–614.

[5] K. Gao and A. Rockwood, *Multi-sided attribute based modeling*, in Mathematics of Surfaces XI, Lecture Notes in Computer Science, Springer, 2005, pp. 219–232.

[6] J. A. Gregory, *Smooth interpolation without twist constraints*, in Computer Aided Geometric Design, R. E. Barnhill and R. F. Riesenfeld, eds., Academic Press, Inc., 1974, pp. 71–88.

[7] J. A. Gregory and J. M. Hahn, *A C^2 polygonal surface patch*, Computer Aided Geometric Design, 6 (1989), pp. 69–75.

[8] J. A. Gregory, V. K. H. Lau, and J. Zhou, *Smooth parametric surfaces and n-sided patches*, in Computation of Curves and Surfaces, W. Dahmen, M. Gasca, and C. A. Micchelli, eds., Kluwer Academic Publishers, 1990, pp. 499–528.

[9] K. Kato, *Generation of n-sided surface patches with holes*, Computer-Aided Design, 23 (1991), pp. 676–683.

[10] A. Levin, *Interpolating nets of curves by smooth subdivision surfaces*, in 26th Conference on Computer Graphics and Interactive Techniques, ACM, 1999, pp. 57–64.

[11] P. Malraison, *A bibliography for n-sided surfaces*, in Mathematics of Surfaces VIII, Information Geometers, 1998, pp. 419–430.

[12] H. P. Moreton and C. H. Séquin, *Functional optimization for fair surface design*, in 19th Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, ACM, 1992, pp. 167–176.

[13] J. Pegna and F.-E. Wolter, *Geometrical criteria to guarantee curvature continuity of blend surfaces*, Journal of Mechanical Design, 114 (1992), pp. 201–210.

[14] J. Peters, *Local smooth surface interpolation: a classification*, Computer Aided Geometric Design, 7 (1990), pp. 191–195.

[15] ———, *Smooth interpolation of a mesh of curves*, Constructive Approximation, 7 (1991), pp. 221–246.

[16] D. Plowman and P. Charrot, *A practical implementation of vertex blend surfaces using an n-sided patch*, in Mathematics of Surfaces VI, Clarendon Press, 1996, pp. 67–78.

[17] M. Sabin, *Some negative results in n-sided patches*, Computer-Aided Design, 18 (1986), pp. 38–44.

[18] M. Sabin, *Transfinite surface interpolation*, in Mathematics of Surfaces VI, Clarendon Press, 1996, pp. 517–534.

[19] P. Salvi, *Fair Curves and Surfaces*, PhD thesis, Eötvös Loránd University, Budapest, 2012.

[20] P. Salvi, T. Várady, and A. Rockwood, *Ribbon-based transfinite surfaces*, Computer Aided Geometric Design (submitted), (2013).

[21] S. Schaefer, J. Warren, and D. Zorin, *Lofting curve networks using subdivision surfaces*, in Symposium on Geometry Processing, ACM, Eurographics Association, 2004, pp. 103–114.

[22] T. Várady and T. Hermann, *Best fit surface curvature at vertices of topologically irregular curve networks*, in Mathematics of Surfaces VI, Clarendon Press, 1996, pp. 411–427.

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

[24] W. Wang, B. Jüttler, D. Zheng, and Y. Liu, *Computation of rotation minimizing frames*, Transactions on Graphics, 27 (2008), p. 2.

[25] A. Worsey, *A modified C2 Coons' patch*, Computer Aided Geometric Design, 1 (1984), pp. 357–360.