

Ribbon-based Transfinite Surfaces

Péter Salvi^a, Tamás Várady^a, Alyn Rockwood^b

^a*Budapest University of Technology and Economics*

^b*King Abdullah University of Science and Technology*

Abstract

One major issue in CAGD is to model complex objects using free-form surfaces of general topology. A natural approach is curvenet-based design, where designers directly create and modify feature curves. These are interpolated by smoothly connected, multi-sided patches, which can be represented by transfinite surfaces, defined as a combination of side interpolants or *ribbons*. A ribbon embeds Hermite data, i.e., prescribed positional and cross-derivative functions along boundary curves.

The paper focuses on two transfinite schemes: the first is an enhanced and extended variant of a multi-sided generalization of the classical Coons patch [43]; the second one is based on a new concept of combining doubly curved composite ribbons, each one interpolating three adjacent sides. Main contributions include various ribbon parameterizations that surpass former methods in quality and computational efficiency. It is proven that these surfaces smoothly interpolate the prescribed ribbon data. Both formulations are based on non-regular convex polygonal domains and distance-based, rational blending functions. A few examples illustrate the results.

Keywords: transfinite surface interpolation, n -sided surfaces, ribbons, distance-based blending functions, parameterization

1. Introduction

Curvenet-based design is a natural approach to create complex free-form models in three dimensions. The process begins with creating a curve network representing the boundaries and feature lines of the object. These curves may come from several sources, such as traditional blueprints, 2D sketches, or directly through some sophisticated 3D graphical user interface, such as [1, 18], then the interpolating surfaces are expected to be automatically generated over the curve network. *Transfinite surfaces* are particularly suitable for this approach, since they are determined solely by boundary curves and cross-derivatives.

It is hard to compare transfinite schemes with other types of general topology surfaces, such as (i) trimmed parametric surfaces, (ii) subdivision surfaces or (iii) split n -sided surfaces. Trimmed surfaces are typically defined by a grid of control points, in contrast to transfinite patches that require only the boundary information. Certain symmetric n -sided configurations cannot be achieved by trimmed surfaces, for example a regular 3- or 5-sided patch cannot be exactly represented over a 4-sided domain. While the boundary edges over the four sides of the parametric domain can be directly edited in 3D, the internal trimming curves cannot. As for continuity, stitching parametric surfaces along general trimming curves can be achieved only in a numerical sense, controlled by tolerances; however, for transfinite patches accurate “watertight” stitching is possible.

Subdivision surfaces offer a highly intuitive technique to produce smoothly connected, general topology surfaces derived from a control polyhedra, but interpolating a network of prescribed boundaries and cross-derivatives is a difficult task. To match a given network, first a good initial control polyhedron must be found, then the subdivision rules need to be modified to obtain an interpolating limit surface. Related work on interpolating curve networks by subdivision can be found in e.g. [24, 29, 26].

There are CAD systems that use only stitched quadrilaterals, and when an n -sided patch needs to be represented, it is subdivided — in most cases — using the central split scheme. This may seem to be a good solution, but it is not easy to determine optimal splitting curves and a good center point, or to resolve compatibility constraints [32].

T-splines also represent surfaces in this category [40], where the connection of adjacent patches with different knot vectors is efficiently solved. Recent work on n -sided patches composed of quadrilaterals include [21, 25, 3].

Our work is motivated by designers and stylists, who want to focus on controllable boundary curves and ribbons. They expect nice, predictable and smoothly connected surfaces, and would like to avoid dealing with intricacies not related to design intent. Transfinite patches are attractive for real-time curve editing with immediate surfacing response. It is a disadvantage, however, that these surfaces are defined by high-degree rational polynomials in a non-standard format. Strictly local control can be an advantage, though some applications may demand for a global optimization over large curve networks; this may be the subject of future research.

This paper expands former work by the same authors and also introduces a new transfinite formulation using composite ribbons. The concept of multi-sided generalization of Coons patches appeared in [43], while composite ribbons were recently outlined in a short paper [41]. Here we present the necessary boundary conditions and proofs of continuity for both formulations in details, which was formerly impossible due to space limitations. We also introduce new ribbon parameterizations that surpass former methods in terms of quality and computational efficiency. After reviewing related work in Section 2, we revisit classical topics, such as Hermite curves and Coons patches, that will help to reformulate the equations leading to the new schemes (Section 3). We summarize the basic elements of multi-sided surface constructions in Section 4, and discuss various ribbon parameterizations with and without constraints. The actual surface formulations are described in Sections 5 and 6. We analyze our results in Section 7, then suggestions for future work conclude the paper.

2. Previous work

There is an extensive and diverse literature on transfinite methods. These differ in several aspects, including (i) the number of boundaries, (ii) the cross-derivative constraints (C^1 , G^1 or G^2) between adjacent surfaces, (iii) the definition of the interpolants, (iv) the type of blending functions, and (v) the method of parameterization, i.e., how the interpolants are mapped onto the domain and vice versa.

In the classical work of Coons [6] four boundaries were interpolated. This was followed by Gordon's extension to interpolate a grid of curves by quadrilaterals [14]. Then transfinite patches with three sides were investigated [2, 30] followed by 5-sided [5] and later general n -sided schemes [16]. Inserting two-sided or even one-sided patches has also become important, in order to complete general topology curve networks, see e.g. [38, 44]. Initially, domains were regular polygons, but later non-convex polygons and domains with internal hole loops were also proposed [22, 35]. The use of non-regular 2D domains was suggested in [43].

While Coons' 4-sided patches can be extended to C^1 or C^2 in a relatively simple manner by raising the degrees of the Hermite blending functions, most n -sided methods achieve only G^1 continuity. Extension to G^2 in a general way is not so easy (see e.g. [17, 13]), and it generally requires further constraints related to the defining curve network. Important contributions concerning minimum energy curve networks, compatibility constraints, and conditions for G^2 curvenets can be found in [28, 31] and [19], respectively.

At the beginning side-based interpolations dominated the field (see Coons-like patches and surfaces in [22, 37, 23]). Later corner interpolants were suggested as well, e.g. [5, 33, 42]. Many schemes, including Coons patches, need correction terms in the formulae to satisfy the interpolating constraints.

An interesting concept of permanence was suggested in [9], which requires a shrinking domain polygon with edges parallel to the original sides to be mapped onto an n -sided subpatch of the same kind. This is a challenging idea — currently no solution is known for $n \neq 4$.

There is a fairly wide variety of blending functions and parameterization methods [43]. For general n -sided patches, mostly “tricky” rational functions based on special distances are computed within the domain, for example in [16, 22, 43]. In Sabin's early paper [36] it is shown that rational polynomial parameterization by two domain variables is not possible for $n \geq 6$.

Generalized barycentric coordinates [11, 20] provide a good basis for parameterizing polygonal domains (see also Section 4.5). Recent publications extend this theory to interpolate Hermite data — continuous functions and derivatives — over polygons and curved boundaries (see moving least squares coordinates [27] and transfinite mean value interpolation [7, 4]).

Recent developments of transfinite surfaces include Coons patches having geodesic boundaries [10], and methods to adjust the shape interior of transfinite patches while retaining their boundaries [44].

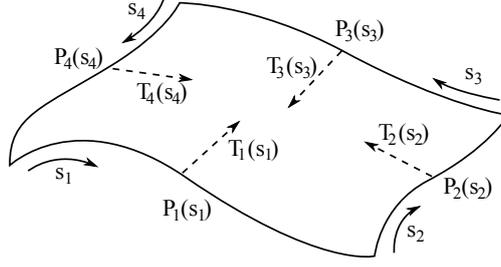


Figure 3.1: Coons patch with circular indexing.

3. Preliminaries

First we revisit and reformulate basic curve and surface equations to introduce our new n -sided surface schemes. Instead of separate positional and tangential data, we combine ribbons, i.e., continuous quadrilaterals having their own local parameterization. Generally ribbons are parameterized linearly by a distance parameter; in this paper we propose a rational ribbon reparameterization that reproduces Hermite cubics and Coons patches.

3.1. Parametric cubic curves

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

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

so

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

A similar curve equation can be formulated by combining two continuous, parametric straight line segments, which we call *2D ribbons* (analogously to *3D ribbons* used later). For each 2D ribbon let us introduce a local parameter d_i , and define the segment with endpoint P_i and tangent T_i 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 with properties $\gamma(0) = 0$ and $\gamma'(0) = 1$. Then the curve can be written as

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

The local ribbon parameters are computed from the parameter of the curve by $d_1(u) = u$, $d_2(u) = 1 - u$. Here only the first Hermite blending function needs to be used, and T_2 is reversed. $\gamma(d_i) = d_i$ would yield a quartic interpolating curve, however, if we apply a rational $\gamma(d_i) = \beta_0(d_i)/\alpha_0(d_i) = \frac{d_i}{2d_i+1}$, a curve identical to the original cubic Hermite is obtained.

3.2. C^1 Coons patches

The C^1 Coons patch [6] 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 $T_1(u), T_3(u), T_2(v), T_4(v)$ (Fig. 3.1). 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.

To reformulate the surface equation based on sides, 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, as shown in Fig. 3.1. 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.,

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

It is well-known, that when the twist vectors are not compatible, Gregory's rational twists need to be used [15, 8].

Now let us construct a Coons patch using *3D ribbons*, i.e., instead of blending eight one-dimensional vector functions, let us combine four biparametric surfaces, in a similar fashion as for the cubic curve. The i -th ribbon is defined as

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

Here we use the other local parameter, 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

$$S(u, v) = \sum_{i=1}^4 R_i(s_i, d_i)\alpha_0(d_i) - \sum_{i=1}^4 Q_{i,i-1}(s_i, s_{i-1})\alpha_0(s_i)\alpha_1(s_{i-1}), \quad (3.1)$$

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, the correction patches are doubly curved. If we apply the previous rational $\gamma(d_i)$ functions, this formula will be *identical* to the original C^1 Coons patch. This is the basis of the multi-sided generalization presented in Section 5.

4. Multi-sided patches

We investigate multi-sided patches that interpolate n three-dimensional curves $P_i(s_i)$, $1 \leq i \leq n$, and related cross-derivative functions $T_i(s_i)$ for an arbitrary $n \geq 3$. We focus on ribbon-based patches, where ribbons are biparametric surfaces $R_i(s_i, d_i)$ that reproduce prescribed positional and tangential constraints at $d_i = 0$. For creating a multi-sided patch, the following must be provided: (i) n ribbon surfaces, (ii) an n -sided domain polygon, (iii) blending functions, and (iv) appropriate methods to parameterize the ribbons.

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., $s_i = s_i(u, v)$, $d_i = d_i(u, v)$, and for each side there is an associated blending function $B_i(u, v) = B_i(d_1, \dots, d_n)$. We investigate *two* different schemes based on the weighted combination of ribbon surfaces. The first is a generalization of Coons' Boolean sum scheme with linear ribbons (Eq. 3.1), that also requires a correction surface (see Section 5), while the second is a combination of doubly curved ribbons (see Section 6), each interpolating three consecutive sides.

4.1. Ribbons

We briefly describe how ribbons can be created. We assume that for each vertex of the network the crossing curves define a local tangent plane. For each boundary $P_i(s_i)$ of a given patch, there exists a normal vector function $N_i(s_i)$

(often called the *normal fence*) that interpolates the normals at the related corners and minimizes its rotation along the boundary (see RMF frames [46]). The cross-derivatives of the ribbons are composed by the well-known technique of

$$\frac{\partial}{\partial d_i} R_i(s_i, 0) = \alpha(s_i) \left(N_i(s_i) \times \frac{\partial}{\partial s_i} R_i(s_i, 0) \right) + \beta(s_i) \frac{\partial}{\partial s_i} R_i(s_i, 0),$$

where $\alpha(s_i)$ and $\beta(s_i)$ are scalar functions. Thus, if two adjacent patches share a common normal fence, then both ribbons, and consequently both multi-sided patches, will be connected with G^1 continuity. The scalar functions are to satisfy end conditions at the corner points ($s_i = 0$ and $s_i = 1$), but there is further freedom to define these in order to optimize the shape of the ribbons. For example, by means of the so-called ribbon-handles, which are prescribed cross-derivatives at the middle point of the boundary ($s_i = 0.5$), users can interactively edit ribbons, if needed. Alternatively, these can be optimized by different fairing algorithms; which is the subject of ongoing research. (Typically, we apply cubic B-spline boundaries with quadratic or cubic scalar functions.) From now on we will assume that the ribbons have already been defined, and focus only on the patch formulation.

4.2. Domain polygon

Let L_i denote the arc-lengths of the boundary curves, and l_i the side lengths of the domain. Also let φ_i denote the corner angles in 3D space, and α_i those in the 2D domain. Introducing the constants $c_{length} = \sum L_i / \sum l_i$ and $c_{angle} = \sum \varphi_i / \sum \alpha_i$, where $\sum \alpha_i = (n - 2)\pi$, we propose to minimize the expressions

$$\frac{1}{n} \sum \left(\frac{L_i}{l_i c_{length}} - 1 \right)^2 \quad \text{and} \quad \frac{1}{n} \sum \left(\frac{\varphi_i}{\alpha_i c_{angle}} - 1 \right)^2,$$

i.e., the arc length and angle distortions between the surface and the domain.

Our experience shows that we can apply a regular polygon, if the above measures remain below prescribed thresholds. However, if the length or angle distortion is high, we strongly recommend using *non-regular* convex polygons to avoid unexpected shape artifacts. Instead of solving a non-linear optimization, simple heuristic methods for creating non-regular domains have been suggested in [43].

4.3. Rational blending functions

Assume that we have a polygonal domain, and for each (u, v) point we determine an n -tuple 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, we create distance-based rational blending functions to combine ribbons. We introduce the notation $D_{i_1 \dots i_k} = \prod_{j \in \{i_1 \dots i_k\}} d_j^2$, where squared distance terms are needed to satisfy G^1 constraints.

The simplest blending function proposed by Kato and other authors [22, 43] is

$$B_i^*(d_1, \dots, d_n) = \frac{D_i}{\sum_j D_j} \left(= \frac{1/d_i^2}{\sum_j 1/d_j^2} \right), \quad (4.1)$$

with the property that $B_i^* = 1$ on Γ_i , and 0 on all other sides Γ_j , $j \neq i$. (The formula in the parenthesis is equivalent to the one on the left. It is more efficient to evaluate the reciprocal terms when all d_i -s are larger than a given threshold. At the same time, very small d_i -s can produce numerical instability, when we need to return to the original evaluation with the D_i terms.) This blending function is singular at the corners — fortunately the corner positions and tangents are uniquely defined by the boundary constraints. Nevertheless, this sort of blending may create uneven curvature distributions in the vicinity of the boundaries.

Avoiding singular blending functions while retaining the side-based concept was one of the key motivations in our research to generalize Coons patches. In the four-sided case, the cubic Hermite blending functions ensure interpolation on Γ_i and a “gradual” $1 \rightarrow 0$ transition on the sides Γ_{i-1} and Γ_{i+1} , as we move from Γ_i to the opposite side of the domain. For general n this is impossible by means of the Hermite functions, but it is possible using distance-based blending functions. For a general n the basic requirement is that the *side 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 4.1 (left).

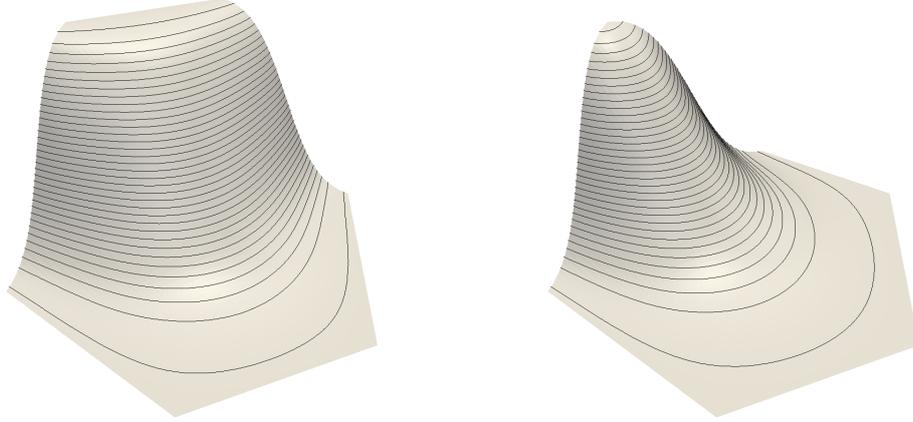


Figure 4.1: Blending functions B_i and $B_{i,i-1}$ over a six-sided domain.

Before going any further, let us introduce *corner blending functions* that provide a convenient way to define *side blending functions*, and will be later used for correction patches, as well:

$$B_{i,i-1}(d_1, \dots, d_n) = \frac{D_{i,i-1}}{\sum_j D_{j,j-1}} \left(= \frac{1/(d_i d_{i-1})^2}{\sum_j 1/(d_j d_{j-1})^2} \right).$$

These functions have also been proposed earlier in the transfinite surface patches of [16, 33], where corner interpolants are combined. The corner blend $B_{i,i-1}$ yields 1 at the $(i-1, i)$ corner, and ensures a “gradual” $1 \rightarrow 0$ transition on sides Γ_{i-1} and Γ_i as we move towards the $i-1$ and $i+1$ domain vertices; $B_{i,i-1} = 0$ on all the remaining sides Γ_j , $j \notin \{i-1, i\}$. See Figure 4.1 (right).

It is easy to show, that by adding together two adjacent corner blends, we obtain the requested properties of side blending:

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

Due to the squared terms, most partial derivatives of the blending functions vanish, i.e.,

$$\begin{aligned} \frac{\partial}{\partial d_k} B_{i,i-1}(d_1, \dots, d_j = 0, \dots, d_n) &= 0, \quad j \notin \{i-1, i\}, k \in [1 \dots n], \\ \frac{\partial}{\partial d_j} B_{i,i-1}(d_1, \dots, d_j = 0, \dots, d_n) &= 0, \quad j \in [1 \dots n], \end{aligned}$$

and consequently $\frac{\partial}{\partial d_k} B_i(d_1, \dots, d_j = 0, \dots, d_n) = 0$ for $j \notin \{i-1, i+1\}$, $k \in [1 \dots n]$. (For $k \neq i$, we get $\frac{\partial}{\partial d_k} B_i(d_1, \dots, d_i = 0, \dots, d_n) = 0$ from $B_i(d_1, \dots, d_i = 0, \dots, d_n) = 1$.)

Both the side and corner blending functions will be used in the multi-sided Coons patch (Section 5), while only the side blends are needed for the composite ribbon patch (Section 6). These blending functions ensure that the ribbons naturally vanish on the “opposite” sides; the number of terms to be evaluated increases with the number of sides.

4.4. Ribbon parameterization overview

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 (see Figure 4.2). This is what determines the associated points of the ribbons and thus has a critical effect on the shape. There are infinitely many possible solutions to map a four-sided parametric quadrilateral onto an n -sided domain, and our experiments produced a wide variety of visually similar surfaces with different interior curvatures. We will describe a few interesting solutions.

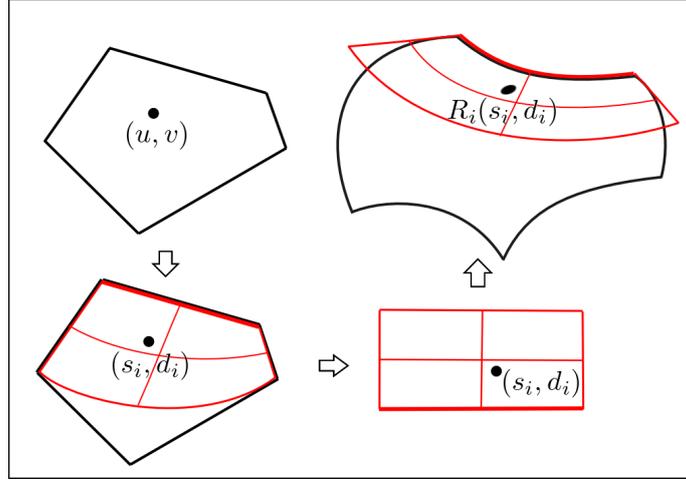


Figure 4.2: Ribbon evaluation.

Two groups of ribbon parameterizations are considered. In *simple parameterizations* (Section 4.5) 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

$$s_i \in [0, 1], \quad d_i = 0, \quad s_{i-1} = 1, \quad s_{i+1} = 0. \quad (4.2)$$

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

$$d_{i-1} = s_i, \quad d_{i+1} = 1 - s_i. \quad (4.3)$$

For *constrained parameterizations* (Section 4.6) further properties must also be satisfied for a point on Γ_i :

$$\frac{\partial d_{i-1}}{\partial u} = \frac{\partial s_i}{\partial u}, \quad \frac{\partial d_{i+1}}{\partial u} = -\frac{\partial s_i}{\partial u}, \quad \frac{\partial d_{i-1}}{\partial v} = \frac{\partial s_i}{\partial v}, \quad \frac{\partial d_{i+1}}{\partial v} = -\frac{\partial s_i}{\partial v}. \quad (4.4)$$

Roughly speaking, this means that in the simple case the parameterizations of the adjacent ribbons match only in a positional sense. For the constrained case these are identical in a differential sense, as well (see Figure 4.3). These properties are trivially satisfied for the four-sided Coons patch, but for n -sided domains special constructions are needed. In the evaluation of parameterization methods, there are two main issues: (i) the constant s_i, d_i parameter lines should have an even distribution in the domain, and (ii) the $(u, v) \rightarrow (s_i, d_i)$ mappings must be simple and computationally efficient. New parameterization schemes will be introduced in the next two sections.

4.5. Simple parameterizations

A family of simple parameterizations is defined when the $s_i = \text{const}$ 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} . Methods differ in how the directions change. The simplest is *bilinear mapping* defined by a quadrangle, spanned by sides Γ_{i-1}, Γ_i and Γ_{i+1} , see Figure 4.4a. Alternative sweeping line methods include the radial sweep [5] or the central line sweep [43]. The former constrains all sweep lines to go through a given point outside the domain, the latter forces the $s_i = \frac{1}{2}$ isolines to go through the center point of the domain.

For a (u, v) point, the related sweep lines determine the s_i coordinates. In the cases $n = 3, 4$ it is straightforward to use bilinear mapping for computing the d_i

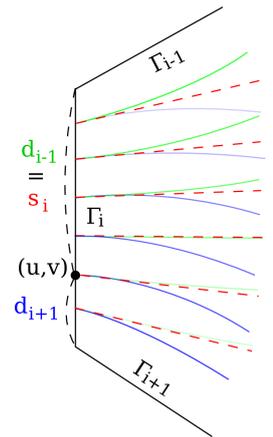


Figure 4.3: Constrained parameterization.

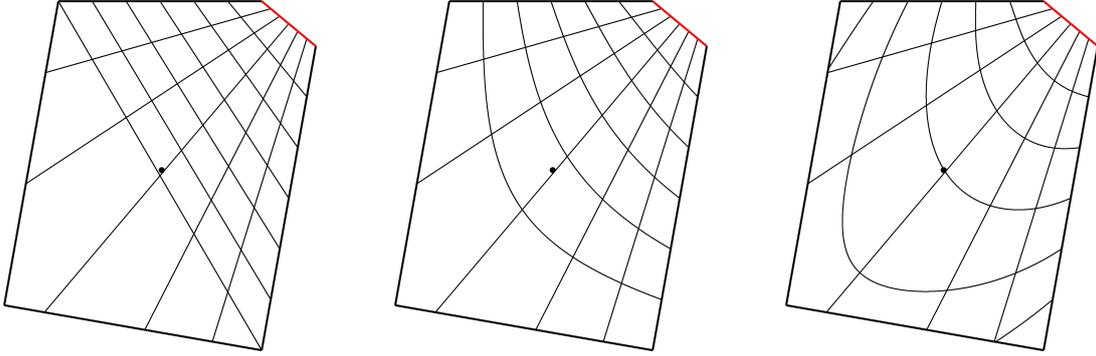


Figure 4.4: A five-sided domain with different parameterizations: (a) bilinear, (b) barycentric by Wachspress coordinates, (c) barycentric by modified Wachspress coordinates.

coordinates, as well. For $n \geq 5$, the well-known Wachspress coordinates [45, 12] turned out to be a good solution. 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 $A_{i-1} = \Delta(p_{i-1}, (u, v), p_i)$, $A_i = \Delta(p_i, (u, v), p_{i+1})$ and $C_i = \Delta(p_{i-1}, p_i, p_{i+1})$ represent triangles [20], whose areas are incorporated into the above formula. See Figure 4.5.

Then the d_i distance parameter is computed as

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

that satisfies equations (4.2) and (4.3) and edge linearity, due to the properties of Wachspress coordinates. This is a simple construction, however, the use of more general barycentric coordinates, such as Mean Value Coordinates [11] or Moving Least Squares [27] is also possible.

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 ensures that the midpoints of the ribbons are mapped to the midpoint of the domain. Assume that we have central line sweep parameterization, i.e., $s_i = \frac{1}{2}$ holds for the center point. Let us precompute the Wachspress coordinate of the center point by the original mapping: $m_i = d_i(\text{center}_u, \text{center}_v)$. Then, to obtain the (s_i, \hat{d}_i) coordinates of an arbitrary point, apply a quadratic reparameterization, using the original (s_i, d_i) :

$$\hat{d}_i(s_i, d_i) = d_i \cdot (1 + (1 - s_i)s_i \cdot (1 - d_i)d_i \cdot w_i), \quad (4.5)$$

see Figure 4.4c.

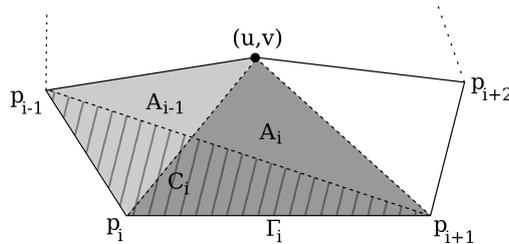


Figure 4.5: Triangle-based construction of Wachspress coordinates.

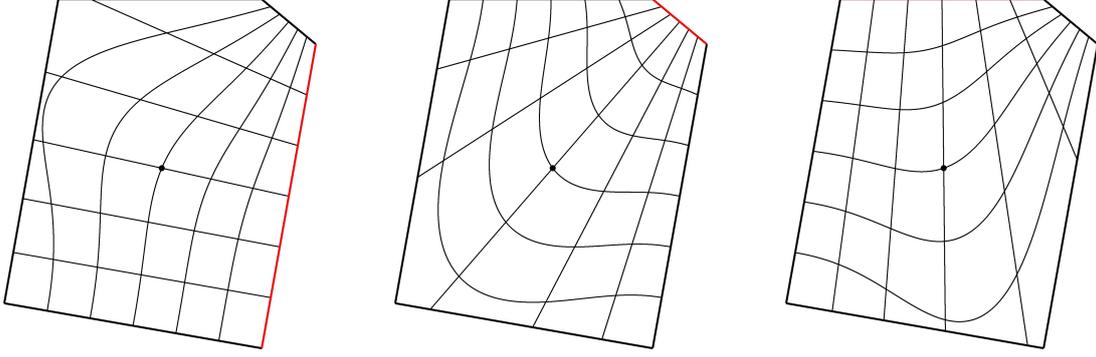


Figure 4.6: Constant parameter lines of the interconnected parameterization.

Here the unknown w_i can be determined by substituting the center point $(0.5, m_i)$ into the equation:

$$\hat{d}_i(0.5, m_i) = m_i \cdot (1 + 0.25 \cdot (1 - m_i)m_i \cdot w_i) = 0.5,$$

thus

$$w_i = \frac{2 - 4m_i}{(1 - m_i)m_i^2}.$$

Note that \hat{d}_i must not become negative. Looking at equation (4.5), our condition is that $(1 - s_i)s_i \cdot (1 - d_i)d_i \cdot w_i \geq -1$, where $(1 - s_i)s_i \leq \frac{1}{4}$ and $(1 - d_i)d_i \leq \frac{1}{4}$. So $w_i \geq -16$ is sufficient, resulting in the inequality $8m_i^3 - 8m_i^2 + 2m_i - 1 \leq 0$, which limits the range of m_i to $[0, \approx 0.877]$. In the rare case of higher values, the above improvement cannot be applied.

4.6. Constrained parameterizations

In the following subsections we will propose two parameterizations that satisfy all three requirements (4.2)–(4.4).

4.6.1. Interconnected parameterization

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 values in $[0, 1]$. For example, the s coordinates of the bilinear, radial or central line sweeps are such functions. These naturally satisfy $s_i \in [0, 1]$, $s_{i-1} = 1$ and $s_{i+1} = 0$ for a point on Γ_i . Let us define a blending function $\alpha(t) : [0, 1] \rightarrow [0, 1]$ with $\alpha(0) = 1$ and $\alpha(1) = \alpha'(0) = \alpha'(1) = 0$, for example the Hermite function $\alpha_0(t)$ from Section 3, or a variation of the rational blend function presented in Section 4.3: $\alpha(t) = \frac{(1-t)^2}{t^2+(1-t)^2}$. Now we can define d_i by means of s_{i-1} and s_{i+1} as follows:

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

If we are on the i -th side, $s_{i-1} = 1$ and $s_{i+1} = 0$, so $d_i = 0$, satisfying (4.2).

Furthermore on the i -th side, d_{i-1} and its derivative are the same as that of s_i :

$$\begin{aligned} d_{i-1} &= (1 - s_{i-2}) \cdot \alpha(s_{i-1}) + s_i \cdot \alpha(1 - s_{i-1}) = s_i, \\ \frac{\partial d_{i-1}}{\partial u} &= \frac{\partial}{\partial u}(1 - s_{i-2}) \cdot \alpha(s_{i-1}) + \frac{\partial}{\partial u} s_i \cdot \alpha(1 - s_{i-1}) = \frac{\partial s_i}{\partial u}, \end{aligned}$$

because the derivatives of the blend function vanish. The same reasoning works for the derivative by v . Similarly

$$\begin{aligned} d_{i+1} &= (1 - s_i) \cdot \alpha(s_{i+1}) + s_{i+2} \cdot \alpha(1 - s_{i+1}) = 1 - s_i, \\ \frac{\partial d_{i+1}}{\partial u} &= \frac{\partial}{\partial u}(1 - s_i) \cdot \alpha(s_{i+1}) + \frac{\partial}{\partial u} s_{i+2} \cdot \alpha(1 - s_{i+1}) = \frac{\partial}{\partial u}(1 - s_i) = -\frac{\partial s_i}{\partial u}, \end{aligned}$$

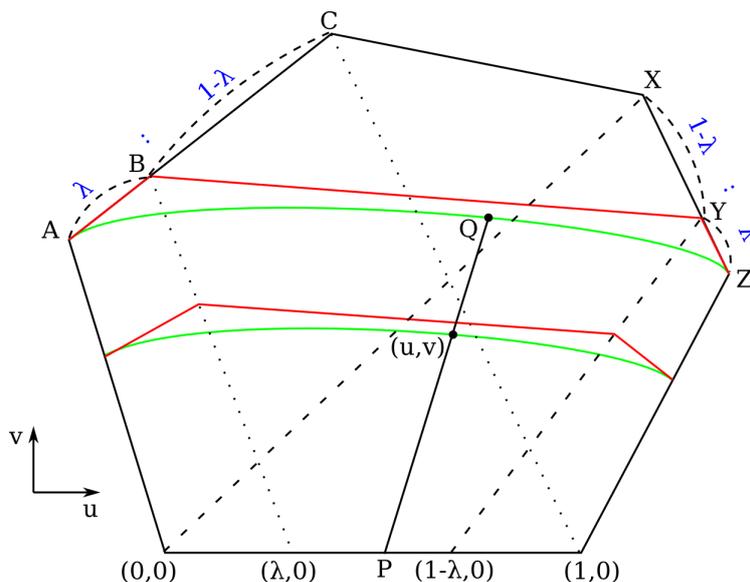


Figure 4.7: Construction of the cubic parameterization.

so the requirements (4.3) and (4.4) are 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 4.6 shows constant s and d lines for this parameterization using the central line sweep parameterization as a basis [44]. 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 the in same way (in a differential sense) as their counterparts in the first and third images.

4.6.2. Cubic parameterization

While the interconnected parameterization scheme satisfies all the requirements, its d -constant parameter lines have inflections. In this section we will show a different parameterization where the d -constant parameter lines are cubic curves. This computation is not direct, as it requires the solution of fourth- or sixth-degree equations. (Cubic parameter lines are the simplest, but quartics can also be used by extending the method below to spread the d constant parameter lines more evenly for large n -s, e.g. $n \geq 8$.)

Similarly to the previous method, this parameterization also builds on a sweep line parameterization. For simplicity's sake, we will use the bilinear sweep, but the same can be done for radial or other parameterizations, that lead to higher degree equations. For ease of computation, we will assume that the base side lies on the u axis, with its endpoints at the origin and $(1, 0)$. Points A and Z are the distant endpoints of the left and right polygon sides Γ_{i-1} and Γ_{i+1} , respectively. See Figure 4.7.

Let us define the d isocurves in such a way, that they “inherit” the same positions and tangents from the left and right side as the corresponding bilinear sweeping lines of those sides. We place a point B as shown in the figure at a λ proportion of the AC side, Y is defined symmetrically on the ZX side. λ is a *fullness parameter*, that controls the shape of the d isocurves to be more flat or curved. The $ABYZ$ polygon defines a cubic Bézier curve, drawn in green. Note that the quadrilateral $[A \sim C \sim (0, 0) \sim (1, 0)]$ produces a sequence of tangents corresponding to the bilinear parameterization based on the left side of the polygon, and similarly the quadrilateral $[Z \sim X \sim (1, 0) \sim (0, 0)]$ produces a sequence of tangents corresponding to the bilinear parameterization based on the right side of the polygon. In other words, the d_i isolines will start and end in the same way as the sweep lines at sides $i - 1$ and $i + 1$.

Thus the left and right sweep lines define a family of Bézier curves, the two inner points of the Bézier control polygon are on the chords defined by $[B \sim (\lambda, 0)]$ and $[Y \sim (1 - \lambda, 0)]$, respectively. One of these curves goes through the (u, v) point at parameter \hat{s} . Let Q be the point on the top curve defined by the $ABYZ$ control polygon at \hat{s} . The line

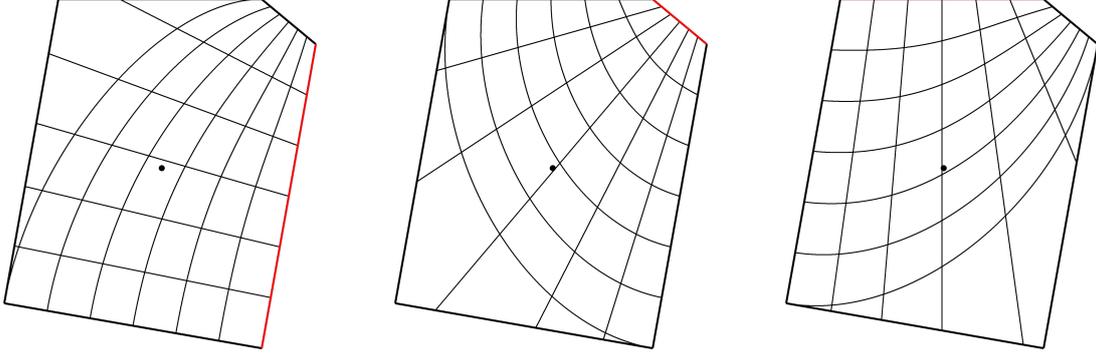


Figure 4.8: Constant parameter lines of the cubic parameterization, with $\lambda = \frac{1}{2}$.

through Q and (u, v) intersects the base side at P .

Obviously

$$v = \frac{u - p^u}{q^u - p^u} \cdot q^v,$$

where upper indices denote coordinates, e.g. $Q = (q^u, q^v)$. This can also be written as

$$(q^u - p^u)v + q^v(p^u - u) = 0.$$

From the definition of Q , we have

$$Q = (1 - \hat{\delta})^3 A + 3(1 - \hat{\delta})^2 \hat{\delta} B + 3(1 - \hat{\delta}) \hat{\delta}^2 Y + \hat{\delta}^3 Z,$$

and similarly, $P = (p^u, 0)$, where

$$p^u = (1 - \hat{\delta})^3 \cdot 0 + 3(1 - \hat{\delta})^2 \hat{\delta} \cdot \lambda + 3(1 - \hat{\delta}) \hat{\delta}^2 \cdot (1 - \lambda) + \hat{\delta}^3 \cdot 1.$$

This leads to a sixth-degree equation in $\hat{\delta}$, that can be solved by various numerical methods, see [34]. (The sweeping line construction guarantees that there exists a unique line that goes through (u, v) , thus we need to compute only the real root in the $[0, 1]$ interval, and can ignore the other ones.) Now we define d as

$$d = \frac{v}{q^v} = v / \left((1 - \hat{\delta})^3 \cdot a^v + 3(1 - \hat{\delta})^2 \hat{\delta} \cdot b^v + 3(1 - \hat{\delta}) \hat{\delta}^2 \cdot y^v + \hat{\delta}^3 \cdot z^v \right).$$

Note, that using $\frac{1}{3}$ for the fullness parameter λ , the parameterization of the base side becomes linear, and the equation simplifies to fourth degree; however, the isoparameter lines become somewhat flat. A larger λ gives more fullness, but requires solving the above sixth-degree equation. This construction also satisfies the requirements of constrained parameterization (see Figure 4.8).

5. Generalized Coons patch

This section deals with the direct generalization of the Coons patch to n sides following the original idea of summing up linear side interpolants and subtracting corner correction patches (see also Section 3). Having a general polygonal domain, we define side and corner blending functions, and apply constrained parameterization for the individual ribbons. Either the interconnected or the cubic method can be used, or anything else that satisfies the requirements (4.2)–(4.4). Running the indices from 1 to n , the following surface equation is obtained:

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

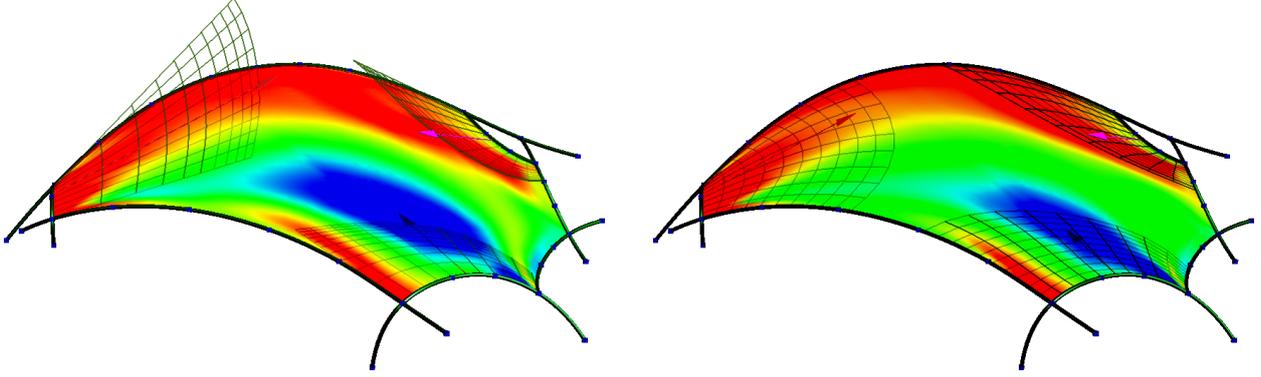


Figure 6.1: Linear and curved ribbons of a seven-sided patch.

Along a given side, similarly to the reformulated Coons patch, five surface interpolants — three linear ribbons and two corner correction patches — are combined to evaluate a point on the boundary. As these satisfy the rules of constrained parameterization, the patch will interpolate the related boundary and cross-derivative functions. A short proof can be found in Appendix A; for more details see [39]. A few examples are shown in Section 7.

Note, that for $n = 4$ this equation reproduces the Coons patch when the cubic Hermite blending functions are used, but only a close approximation of the original patch can be obtained when rational distance-based blending functions are applied.

6. Composite ribbon patches

The patch defined in the previous section is based on linear ribbons. These are very simple and can be computed efficiently. On the other hand, linear ribbons may strongly deviate from highly curved n -sided surfaces, and their combination may cause unexpected curvature changes, when the points in the interior are computed by the affine combination of relatively “distant” positions. This motivated us to introduce a special *curved ribbon* that actually interpolates three consecutive linear ribbons. The curved ribbons align much closer to the final n -sided patch, and their combination, even in extreme cases, yields more predictable interior shapes (see Figure 6.1). Using curved ribbons leads to a new multi-sided patch formulation, called *composite ribbon patches*.

6.1. Curved ribbons

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. 6.2):

$$C_i(s, d) = R_i^l(s, d)H(s) + R_i(s, d)H(d) + R_i^r(s, d)H(1-s) - Q_i^l(s, d)H(s)H(d) - Q_i^r(s, d)H(1-s)H(d),$$

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

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

and apply an $H(t)$ blend function, such as the Hermite blend function or the rational blend function, that creates a special C^1 Coons patch over a four-sided domain.

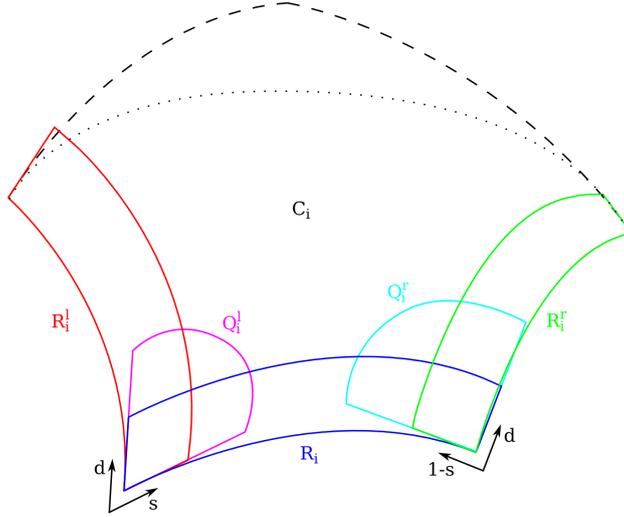


Figure 6.2: Construction of a curved ribbon.

Due to the above construction, d is constrained to lie in $[0, 1]$. This holds for the Wachspress and interconnected parameterizations, but not for the bilinear and cubic parameterizations.

Note that the fourth side of the curved ribbons is “floating”; it comes as a by-product of the three interpolating linear ribbons. Constraining the fourth side can be an advantage, as it offers further degrees of freedom to adjust the shape. At the same time, this fourth side, as in the case of simple linear ribbons, has a relatively small influence due to the blending functions.

Also, using our notation, the classical Gregory patch [5] can be interpreted as a combination of *two-sided curved corner interpolants*, that are made up of two linear ribbons minus a single correction term, as follows:

$$R_{i,i-1}^{\text{Gregory}}(s_i, s_{i-1}) = R_{i-1}(s_{i-1}, s_i) + R_i(s_i, 1 - s_{i-1}) - Q_{i,i-1}(s_i, s_{i-1}).$$

In the next sections we will present new schemes that combine curved ribbons.

6.2. The C^1 composite ribbon patch

Curved ribbons can be used as side interpolants for simple transfinite interpolation patches using the singular blending function (Equation 4.1). Unfortunately, the direct generalization of Coons patches, introduced in Section 5, does not work with curved ribbons. Therefore, we propose an alternative representation that combines curved ribbons in a different way, and eliminates correction patches; thus yielding a simple formula:

$$S(u, v) = \frac{1}{2} \sum_{i=1}^n C_i(s_i(u, v), d_i(u, v)) B_i(d_1(u, v), \dots, d_n(u, v)).$$

In this case, we use a constrained parameterization as defined earlier in Section 4.6. According to the characteristics of the B_i blend function, for any point on the i -th boundary all addends of the sum vanish except for 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

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

hence the division by two in the surface equation. This patch also satisfies the boundary constraints. A short proof can be found in Appendix B; for more details see [39].

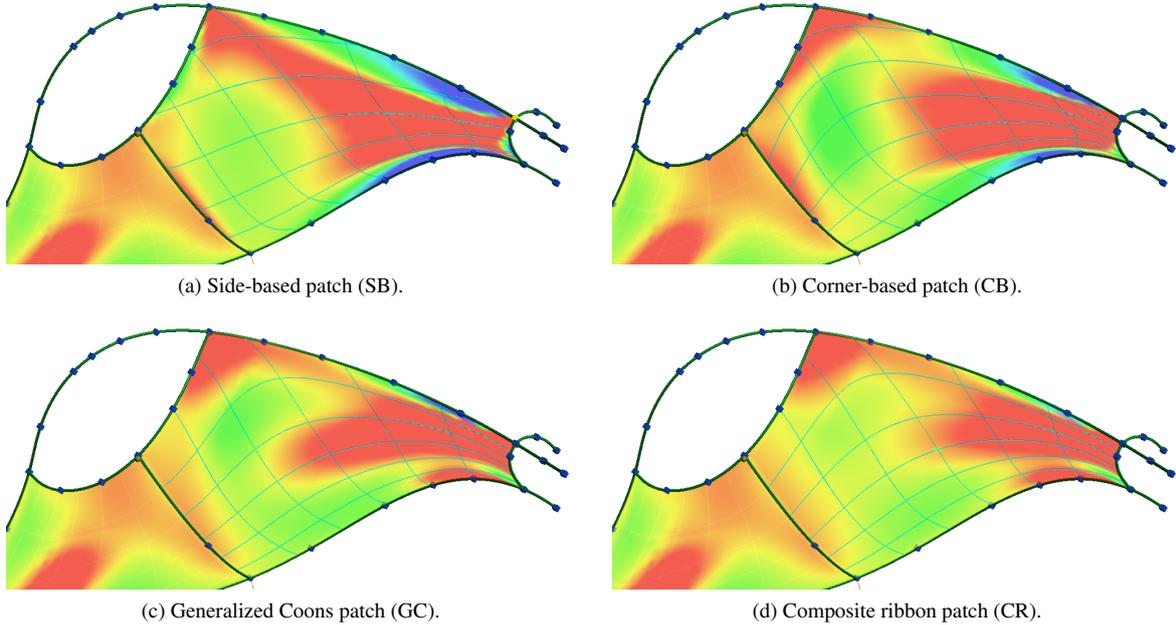


Figure 7.1: Comparing four transfinite patches (Example 1).

6.3. The G^1 composite ribbon patch

It can be proven that the ribbon interpolation property remains valid, if we ignore the last parameterization requirement (4.4). Then we have the same tangent plane for every boundary point as the respective curved ribbon, but not the same tangent vector. This enables the use of simple parameterizations, such as the Wachspress method, which is computationally more efficient. The resulting patch reproduces the boundary constraints in a G^1 sense, as shown in Appendix B. The visual difference between the C^1 and G^1 patches can be noticed only at extreme boundary configurations.

To sum it up, in this section we have created new transfinite surface schemes that (i) use curved side interpolants, (ii) have roughly the same computational complexity as other transfinite methods, and (iii) avoid using singular blending functions.

7. Examples

In this section we show a few examples to evaluate our results. Where needed, we will refer to two prior schemes. The Side-based (SB) patch suggested by Kato in [22] applies singular blending functions. The Corner-based (CB) patch proposed by Gregory in [16] combines corner interpolants. We will use the abbreviations GC and CR for the Generalized Coons and the Composite Ribbon patches, respectively.

Example 1

Even curvature distribution and predictable patch interior is of great importance. In other words, curvature artifacts must be avoided. Our experience shows that in this respect CR patches seem to be the best, in particular for difficult boundary configurations. For simple cases, the curvature distributions are similar. This similarity is strong between the GC and CB patches, which are “close relatives” in generalizing the Coons patch, in spite of the fact that the GC patch is a combination of side interpolants, while the CB patch uses corner interpolants.

The images in Figure 7.1 show mean curvature maps for SB, CB, GC and CR patches. Observe the undesirable narrow curvature variations that occur along some edges of the SB patch; this problem disappears when using GC or CR patches, yielding more natural transitions between the ribbons. Table 1a shows numerical values of mean curvature values including its minimum, maximum, average and standard deviation for all four schemes.

	Min	Max	Average	Std. Deviation
SB	-48.36	294.82	27.92	1.261
CB	-56.15	224.44	22.50	0.728
GC	-37.89	192.27	22.48	0.678
CR	-23.70	192.26	22.64	0.648

(a) Curvature values in Example 1.

	Min	Max	Average	Std. Deviation
SB	0.9963	1.0098	1.0040	3.41e-3
CB	0.9942	1.0082	0.9990	3.13e-3
GC	0.9960	1.0082	1.0014	3.02e-3
CR	0.9960	1.0057	1.0007	2.77e-3

(b) Radii in Example 3.

Table 1: Numeric evaluation.

SB	CB	GC	CR
0.4	0.65	0.45	1.3

Table 2: Average computation times per surface point (in milliseconds).

These values were computed using a dense triangular mesh; if exact values are needed the derivatives are computed using the formulae in the Appendices, though this is computationally much more demanding.

Example 2

Computational efficiency is determined by three components: (i) evaluating the interpolants, (ii) computing the blending functions, and (iii) computing the ribbons' parameterization. Assuming cubic B-spline boundary curves, the basic interpolants are cubic-by-linear for SB and GC, bicubic for CB, and biquartic for CR. The blend functions are rational functions of the distances with degree $2(n-1)$ for SB, and $2(n-2)$ for the other patches. Finally, the mapping between the domain and the ribbons can be characterized by a bilinear polynomial for SB and CB, a quintic for GC, and a trigonometric parameterization for CR.

Our experience shows that 97% of the evaluation time is used for ribbons, and only 3% is devoted to parameterization and blending functions. Table 2 shows the computation times for a single point on a 2.5 GHz processor. As expected, the SB patch is the fastest, and the CR patch is the slowest. The ratios of computing different parameterizations are the following: interconnected \rightarrow 1, cubic \rightarrow 1.78, Wachspress \rightarrow 1.28. (Computations can be made more efficient with extensive caching.)

Example 3

Another interesting question is how these rational transfinite patches can approximate regular shapes, such as an octant of a sphere. We have used approximate B-spline boundaries for the circular arcs with a constraint that the

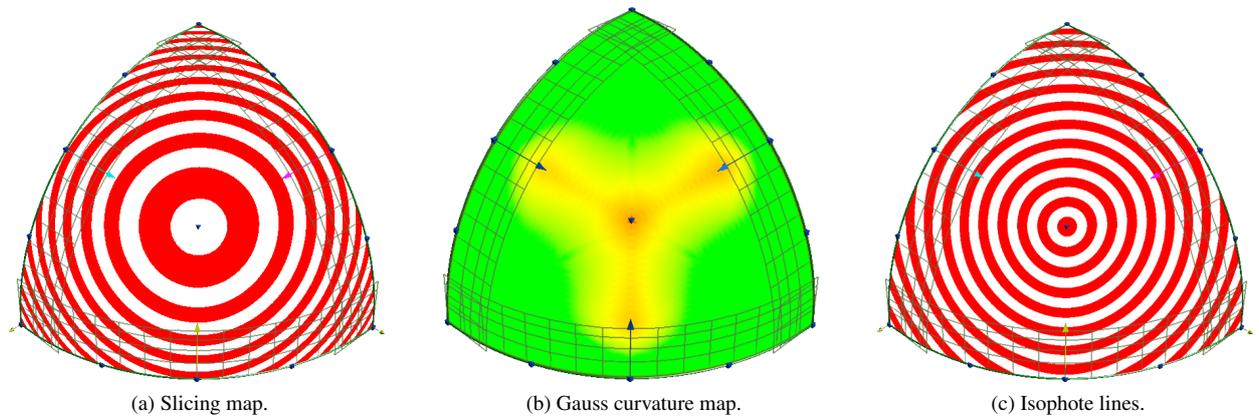


Figure 7.2: Sphere reproduction by a CR patch (Example 3).

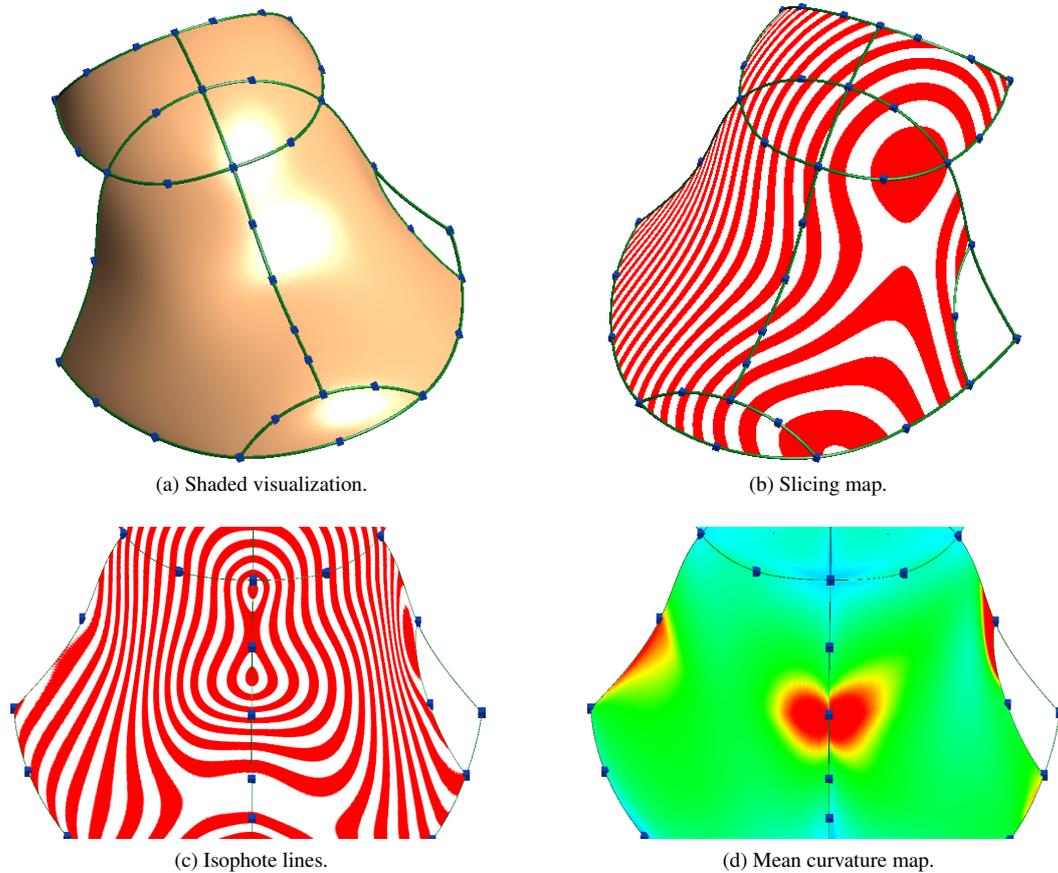


Figure 7.3: Stability — inserting a new edge (Example 4).

middle point of the patch lies exactly on the ideal sphere. In this test CR patches proved to be the best; in Figure 7.2 planar contours, a sensitively scaled Gaussian curvature map and isophotes illustrate the quality of the approximation. See Table 1b for numerical comparison of the radius values for all four schemes.

Example 4

Stability is a qualitative characteristic, that is hard to define for multi-sided schemes. Assuming that we are editing one of the boundaries, we would expect that the shape surrounded by the other sides is generally unaffected. Similarly, if we insert a new edge, i.e., we create an $(n + 1)$ -sided patch from an n -sided, we expect that the parts of the patch that are not directly adjacent will remain unchanged. Our experiment shows that in this respect CR patches are good, CB and GC patches are reasonable, and SB patches are somewhat weaker.

Figure 7.3 shows a somewhat artificial model that interpolates a general topology curve network. It contains one 2-sided, two 3-sided, two 4-sided, one 5-sided and one 6-sided patch, all smoothly connected. For the 4-sided loops ordinary Coons patches were used, for the 2-sided a special formula shown in [44] was used, and the rest was represented by CR patches. The cross-derivative functions were extracted from the curve network, and the ribbons satisfy tangent plane constraints at the vertices. See the shaded visualization (a) and the contours (b). This is an almost symmetric surface configuration, except that a corner was cut off at the right side, which converted the five-sided patch at the left into a six-sided patch. The isophote lines (c) and mean curvature map (d) figures show that the patch has changed only to a minimal extent in the middle. It can also be observed that these patches provide visual G^2 continuity in non-extreme configurations, see the smooth variation of the isophote lines across the boundaries.

It is a particular advantage of transfinite schemes that it ensures a very concise representation for complex shapes.

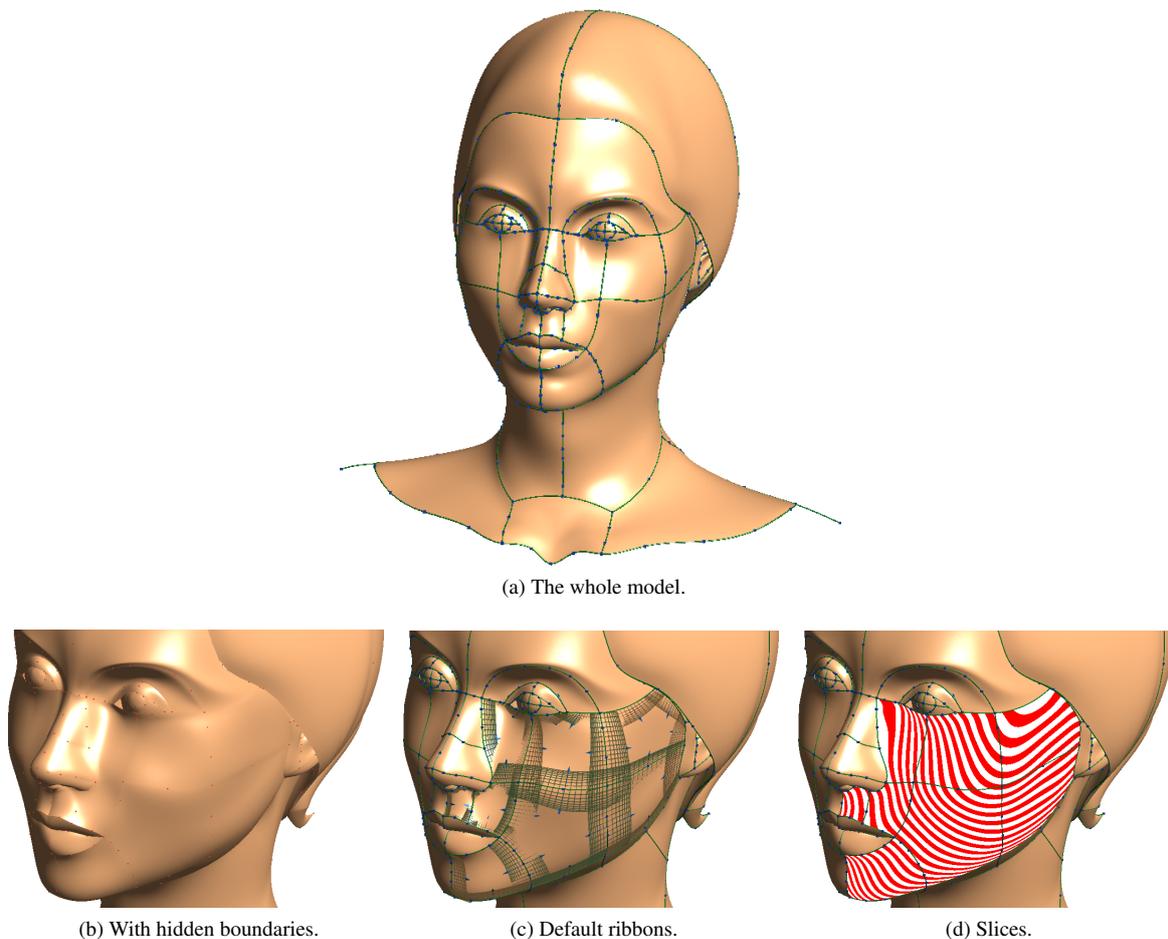


Figure 7.4: Woman's head (Example 5).

For example, the above model can be fully defined by 5 KB, while using quadrilateral B-splines would require more than 100 KB.

Example 5

Finally, a more complex example is shown in Figure 7.4. The three close-up views show the face (a) without boundaries, (b) with the default ribbons and also with a (c) slicing map to show G^1 continuity. This collection of faces consists of six 5-sided, and two 4-sided patches.

Conclusion

Two new surface representations for transfinite surface interpolation have been investigated. Both schemes are based on ribbon surfaces, composed of boundary curves and cross-derivative functions, ensuring normal plane continuity between adjacent patches. The first scheme can be considered a true generalization of the Coons patch, following the classical Boolean sum logic. The second is a transfinite surface combining doubly curved side interpolants. Ribbons have been reparameterized to reproduce the Coons patch for $n = 4$. Various mappings of the ribbons onto the n -sided polygonal domain were suggested, defining simple and constrained parameterizations that are necessary for ensuring the interpolation property. These schemes inherit the continuity of the boundaries, e.g. having cubic B-splines will produce a C^2 continuous patch in the interior. Future research work is going to be directed towards fairing, data approximation by ribbon-based patches, and using ribbons over non-convex parametric domains.

Acknowledgments

The authors acknowledge many constructive suggestions by the anonymous reviewers. This work was partially supported by a grant from the Hungarian Scientific Research Fund (No. 101845) and the Budapest University of Technology and Economics. The pictures were generated by the Sketches system (ShapEx Ltd, Budapest); György Karikó implemented a significant part of this test environment; his valuable contribution is highly appreciated. The authors also acknowledge support from the Geometric Modeling and Scientific Visualization Research Center of KAUST, Saudi Arabia. Thanks are due to Supriya Chewle (KAUST), who has created the head model.

Appendix A. Generalized Coons Patch

Here we prove the interpolating properties of the GC patch using the notations of Section 4. Take a (u, v) point on Γ_i , i.e., let $d_i = 0$. For such a point, most of the blend functions vanish, leaving only

$$S(u, v) = (R_{i-1} - Q_{i,i-1}) B_{i,i-1} + R_i + (R_{i+1} - Q_{i+1,i}) B_{i+1,i}.$$

(We have dropped the arguments to simplify the notation.) It is easy to see that $R_{i-1} = Q_{i,i-1}$ and $R_{i+1} = Q_{i+1,i}$, which proves positional interpolation.

As for the tangential part, let us compute the partial derivative by u :

$$\begin{aligned} \frac{\partial}{\partial u} S(u, v) &= \sum_{j=1}^n \left[\left(\frac{\partial}{\partial s_j} R_j \frac{\partial s_j}{\partial u} + \frac{\partial}{\partial d_j} R_j \frac{\partial d_j}{\partial u} \right) B_j + R_j \frac{\partial}{\partial u} B_j \right] \\ &\quad - \sum_{j=1}^n \left[\left(\frac{\partial}{\partial s_j} Q_{j,j-1} \frac{\partial s_j}{\partial u} + \frac{\partial}{\partial s_{j-1}} Q_{j,j-1} \frac{\partial s_{j-1}}{\partial u} \right) B_{j,j-1} + Q_{j,j-1} \frac{\partial}{\partial u} B_{j,j-1} \right]. \end{aligned}$$

Using the properties of the blend functions and the parameterization, we can simplify this expression, as well:

$$\begin{aligned} \frac{\partial}{\partial u} S(u, v) &= \left(\frac{\partial}{\partial s_{i-1}} R_{i-1} - \frac{\partial}{\partial s_{i-1}} Q_{i,i-1} \right) \frac{\partial s_{i-1}}{\partial u} B_{i,i-1} + \left(\frac{\partial}{\partial d_{i-1}} R_{i-1} - \frac{\partial}{\partial s_i} Q_{i,i-1} \right) \frac{\partial d_{i-1}}{\partial u} B_{i,i-1} \\ &\quad + (R_{i-1} - Q_{i,i-1}) \frac{\partial}{\partial u} B_{i,i-1} + \frac{\partial}{\partial s_i} R_i \frac{\partial s_i}{\partial u} + \frac{\partial}{\partial d_i} R_i \frac{\partial d_i}{\partial u} + (R_{i+1} - Q_{i+1,i}) \frac{\partial}{\partial u} B_{i+1,i} \\ &\quad + \left(\frac{\partial}{\partial s_{i+1}} R_{i+1} - \frac{\partial}{\partial s_{i+1}} Q_{i+1,i} \right) \frac{\partial s_{i+1}}{\partial u} B_{i+1,i} + \left(\frac{\partial}{\partial d_{i+1}} R_{i+1} + \frac{\partial}{\partial s_i} Q_{i+1,i} \right) \frac{\partial d_{i+1}}{\partial u} B_{i+1,i}. \end{aligned}$$

In the above equation, all pairs of the R - and Q -derivatives cancel out each other, leaving

$$\frac{\partial}{\partial u} S(u, v) = \frac{\partial}{\partial s_i} R_i \frac{\partial s_i}{\partial u} + \frac{\partial}{\partial d_i} R_i \frac{\partial d_i}{\partial u} = \frac{\partial}{\partial s_i} P_i \frac{\partial s_i}{\partial u} + T_i \frac{\partial d_i}{\partial u},$$

which is what we wanted to show. The same reasoning works for the v -derivative.

Appendix B. Composite Ribbon Patch

Here we prove the interpolating properties of the CR patch. The positional constraints are trivially satisfied, as explained briefly in Section 6. The tangential proof is similar to the one in Appendix A. The derivative at a (u, v) point on Γ_i is (dropping the arguments for ease of notation):

$$\frac{\partial}{\partial u} S(u, v) = \frac{1}{2} \sum_{j=1}^n \left[\left(\frac{\partial}{\partial s_j} C_j \frac{\partial s_j}{\partial u} + \frac{\partial}{\partial d_j} C_j \frac{\partial d_j}{\partial u} \right) B_j + C_j \frac{\partial}{\partial u} B_j \right],$$

which can be expanded into

$$\begin{aligned} \frac{\partial}{\partial u} S(u, v) = & \frac{1}{2} \left[\left(\frac{\partial}{\partial s_{i-1}} C_{i-1} \frac{\partial s_{i-1}}{\partial u} + \frac{\partial}{\partial d_{i-1}} C_{i-1} \frac{\partial d_{i-1}}{\partial u} \right) B_{i-1} + C_{i-1} \frac{\partial}{\partial u} B_{i-1} + \right. \\ & \left(\frac{\partial}{\partial s_i} C_i \frac{\partial s_i}{\partial u} + \frac{\partial}{\partial d_i} C_i \frac{\partial d_i}{\partial u} \right) B_i + \\ & \left. \left(\frac{\partial}{\partial s_{i+1}} C_{i+1} \frac{\partial s_{i+1}}{\partial u} + \frac{\partial}{\partial d_{i+1}} C_{i+1} \frac{\partial d_{i+1}}{\partial u} \right) B_{i+1} + C_{i+1} \frac{\partial}{\partial u} B_{i+1} \right]. \end{aligned}$$

On Γ_i , where $d_i = 0$, we have $C_{i-1} = C_{i+1}$, so the terms

$$C_{i-1} \frac{\partial}{\partial u} B_{i-1} + C_{i+1} \frac{\partial}{\partial u} B_{i+1} = \frac{1}{2} (C_{i-1} + C_{i+1}) \frac{\partial}{\partial u} (B_{i-1} + B_{i+1}) = \frac{1}{2} (C_{i-1} + C_{i+1}) \frac{\partial}{\partial u} B_i = 0$$

vanish. Straightforward calculation leads to

$$\frac{\partial}{\partial u} S(u, v) = \frac{1}{2} \left[\left(\frac{\partial}{\partial s_i} P_i \frac{\partial s_i}{\partial u} + T_i \frac{\partial s_{i-1}}{\partial u} \right) B_{i-1} + \left(\frac{\partial}{\partial s_i} P_i \frac{\partial s_i}{\partial u} + T_i \frac{\partial d_i}{\partial u} \right) B_i + \left(\frac{\partial}{\partial s_i} P_i \frac{\partial s_i}{\partial u} + T_i \frac{\partial s_{i+1}}{\partial u} \right) B_{i+1} \right],$$

i.e., the derivative vector will be a combination of the s_i - and d_i -derivatives of R_i , so it will be in the tangent plane of R_i (G^1 continuity). This holds for simple parameterizations, as well.

If we have a constrained parameterization, the expression can be simplified further:

$$\frac{\partial}{\partial u} S(u, v) = \left(\frac{\partial}{\partial s_i} P_i \frac{\partial s_i}{\partial u} + T_i \frac{\partial d_i}{\partial u} \right) \frac{B_{i-1} + B_i + B_{i+1}}{2} = \frac{\partial}{\partial s_i} P_i \frac{\partial s_i}{\partial u} + T_i \frac{\partial d_i}{\partial u},$$

satisfying C^1 continuity. The same reasoning works for the v -derivative.

References

- [1] Bae, S.H., Balakrishnan, R., Singh, K.: EverybodyLovesSketch: 3D sketching for a broader audience. In: 22nd Symposium on User Interface Software and Technology. pp. 59–68. ACM (2009)
- [2] Barnhill, R., Birkhoff, G., Gordon, W.: Smooth interpolation in triangles. *Journal of Approximation Theory* 8(2), 114–128 (1973)
- [3] Bommers, D., Lempfer, T., Kobbelt, L.: Global structure optimization of quadrilateral meshes. *Computer Graphics Forum* 30(2), 375–384 (2011)
- [4] Bruvold, S., Floater, M.S.: Transfinite mean value interpolation in general dimension. *Journal of Computational and Applied Mathematics* 233(7), 1631–1639 (2010)
- [5] Charrot, P., Gregory, J.A.: A pentagonal surface patch for computer aided geometric design. *Computer Aided Geometric Design* 1(1), 87–94 (1984)
- [6] Coons, S.A.: Surfaces for computer-aided design of space forms. Tech. Rep. MIT/LCS/TR-41, Massachusetts Institute of Technology (1967)
- [7] Dyken, C., Floater, M.S.: Transfinite mean value interpolation. *Computer Aided Geometric Design* 26(1), 117–134 (2009)
- [8] Farin, G.: Curves and surfaces for CAGD: a practical guide. Morgan Kaufmann Publishers Inc., 5th edn. (2002)
- [9] Farin, G., Hansford, D.: Discrete Coons patches. *Computer Aided Geometric Design* 16(7), 691–700 (1999)
- [10] Farouki, R., Szafran, N., Biard, L.: Existence conditions for Coons patches interpolating geodesic boundary curves. *Computer Aided Geometric Design* 26(5), 599–614 (2009)
- [11] Floater, M.S.: Mean value coordinates. *Computer Aided Geometric Design* 20(1), 19–27 (2003)
- [12] Floater, M.S., Hormann, K.: Personal communication (2012)
- [13] Gao, K., Rockwood, A.: Multi-sided attribute based modeling. In: *Mathematics of Surfaces XI*, pp. 219–232. Lecture Notes in Computer Science, Springer (2005)
- [14] Gordon, W.J.: Spline blended surface interpolation through curve networks. *Journal of Mathematics and Mechanics* 18(10), 931–952 (1969)
- [15] Gregory, J.A.: Smooth interpolation without twist constraints. In: Barnhill, R.E., Riesenfeld, R.F. (eds.) *Computer Aided Geometric Design*. pp. 71–88. Academic Press, Inc. (1974)
- [16] Gregory, J.A.: N-sided surface patches. In: *Mathematics of Surfaces I*. pp. 217–232. Oxford University Press (1986)
- [17] Gregory, J.A., Hahn, J.M.: A C^2 polygonal surface patch. *Computer Aided Geometric Design* 6(1), 69–75 (1989)
- [18] Grimm, C., Joshi, P.: Just Draw It: a 3D sketching system. In: *Symposium on Sketch-Based Interfaces and Modeling*. pp. 121–130. Eurographics Association (2012)
- [19] Hermann, T., Peters, J., Strotman, T.: A geometric criterion for smooth interpolation of curve networks. In: *Conference on Geometric and Physical Modeling*. pp. 169–173. ACM (2009)
- [20] Hormann, K., Floater, M.S.: Mean value coordinates for arbitrary planar polygons. *Transactions on Graphics* 25(4), 1424–1441 (2006)

- [21] Karčiauskas, K., Peters, J.: Bicubic polar subdivision. *Transactions on Graphics* 26(4), 14 (2007)
- [22] Kato, K.: Generation of n -sided surface patches with holes. *Computer-Aided Design* 23(10), 676–683 (1991)
- [23] Kato, K.: N -sided surface generation from arbitrary boundary edges. In: *Curve and Surface Design*, pp. 173–181. *Innovations in Applied Mathematics*, Vanderbilt University Press (2000)
- [24] Levin, A.: Interpolating nets of curves by smooth subdivision surfaces. In: *26th Conference on Computer Graphics and Interactive Techniques*. pp. 57–64. ACM (1999)
- [25] Loop, C., Schaefer, S.: Approximating Catmull–Clark subdivision surfaces with bicubic patches. *Transactions on Graphics* 27(1), 8 (2008)
- [26] Ma, W., Wang, H.: Loop subdivision surfaces interpolating B-spline curves. *Computer-Aided Design* 41(11), 801–811 (2009)
- [27] Manson, J., Schaefer, S.: Moving least squares coordinates. *Computer Graphics Forum* 29(5), 1517–1524 (2010)
- [28] Moreton, H., Séquin, C.: Surface design with minimum energy networks. In: *1st Symposium on Solid Modeling Foundations and CAD/CAM Applications*. pp. 291–301. ACM (1991)
- [29] Nasri, A.H., Abbas, A.: Designing Catmull–Clark subdivision surfaces with curve interpolation constraints. *Computers & Graphics* 26(3), 393–400 (2002)
- [30] Nielson, G.M.: The side-vertex method for interpolation in triangles. *Journal of Approximation Theory* 25(4), 318 – 336 (1979)
- [31] Peters, J.: Joining smooth patches around a vertex to form a C^k surface. *Computer Aided Geometric Design* 9(5), 387–411 (1992)
- [32] Peters, J.: Smooth free-form surfaces over irregular meshes generalizing quadratic splines. *Computer Aided Geometric Design* 10(3), 347–361 (1993)
- [33] Plowman, D., Charrot, P.: A practical implementation of vertex blend surfaces using an n -sided patch. In: *Mathematics of Surfaces VI*. pp. 67–78. Clarendon Press (1996)
- [34] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press (1992)
- [35] Sabin, M.: Further transfinite surface developments. In: *Mathematics of Surfaces VIII*. pp. 161–173. *Information Geometers* (1998)
- [36] Sabin, M.: Some negative results in n -sided patches. *Computer-Aided Design* 18(1), 38–44 (1986)
- [37] Sabin, M.: Transfinite surface interpolation. In: *Mathematics of Surfaces VI*. pp. 517–534. Clarendon Press (1996)
- [38] Sabin, M.: Two-sided patches suitable for inclusion in a B-spline surface. In: *Conference on Mathematical Methods for Curves and Surfaces*. pp. 409–416. Vanderbilt University (1998)
- [39] Salvi, P.: *Fair Curves and Surfaces*. Ph.D. thesis, Eötvös Loránd University, Budapest (2012)
- [40] Sederberg, T.W., Zheng, J., Bakenov, A., Nasri, A.: T-splines and T-NURCCs. *Transactions on Graphics* 22(3), 477–484 (2003)
- [41] Várady, T., Salvi, P., Rockwood, A.: Transfinite surface patches using curved ribbons. In: *EuroGraphics 2013 — Short Papers*. pp. 5–8. Eurographics Association (2013)
- [42] Várady, T.: Overlap patches: a new scheme for interpolating curve networks with n -sided regions. *Computer Aided Geometric Design* 8(1), 7–27 (1991)
- [43] Várady, T., Rockwood, A., Salvi, P.: Transfinite surface interpolation over irregular n -sided domains. *Computer Aided Design* 43(11), 1330–1340 (2011)
- [44] Várady, T., Salvi, P., Rockwood, A.: Transfinite surface interpolation with interior control. *Graphical Models* 74(6), 311–320 (2012)
- [45] Wachspress, E.L.: *A rational finite element basis*. Academic Press (1975)
- [46] Wang, W., Jüttler, B., Zheng, D., Liu, Y.: Computation of rotation minimizing frames. *Transactions on Graphics* 27(1), 2 (2008)