Comparison of Two n-Patch Representations in Curve Network-Based Design

Péter Salvi and Tamás Várady

Budapest University of Technology and Economics

Abstract. Transfinite surface interpolation is a natural representation for curve network-based design, where a shape is determined by its feature curves. This paper contrasts the corner-based Gregory patches [5] with the side-based generalized Coons patches [7] in this context. After a brief introduction to transfinite methods, the pros and cons of the above two constructions are analyzed considering surface quality and computational efficiency.

1 Introduction

Modeling complex 3D objects with free-form geometry is a challenging task. Several competing approaches are known, including trimmed surface modeling, modeling by quadrilaterals, subdivision surfaces and—last, but not least—*curve network-based design*. In this paradigm, the input is a connected collection of characteristic curves or *feature curves*, coming from a variety of media, including hand-drawn sketches, a set of images, or spline curves built on existing mesh data. The output is a smoothly connected collection of surfaces filling the loops of the network. This approach to modeling has attracted considerable attention in recent years, due to novel 3D curve drawing methods [1,6].

The natural decomposition of free-form shapes into disjoint regions along feature lines requires patches that are often not four-sided, take, for example, various car body parts, household appliances, furnitures or other "stylish" everyday objects, where aesthetic appearance is particularly important. The traditional representation of trimmed biparametric surfaces may cause asymmetries between those sides that are trimmed and those that are not. The central split representation, connecting quadrilaterals whose sides match the respective half-segments of the boundaries, causes difficulties due to its weakly defined internal subdividing curves. Both of these representations can be adequate at the end of the pipeline of standard CAD/CAM applications, but their drawbacks appear when the model needs to be interactively modified or redesigned.

The obvious strength of the curve network-based approach is that designers do not need to think about intricate mathematical surface notions and various artificial constraints, rather they can focus on the direct expression of the design. A good curvenet-based user interface is supposed to be very responsive to editing operations, assuming the curves and their associated cross-derivatives naturally define a smoothly connected patchwork. This motivates our research into transfinite surface representations [8], as they can directly interpolate general topology curvenets, and adjust their shapes as the feature curves and other boundary conditions are being edited. As an application example, Figure 1 shows a model consisting of smoothly connected transfinite surfaces that interpolate a curve network drawn in 3D using special gestures [1].

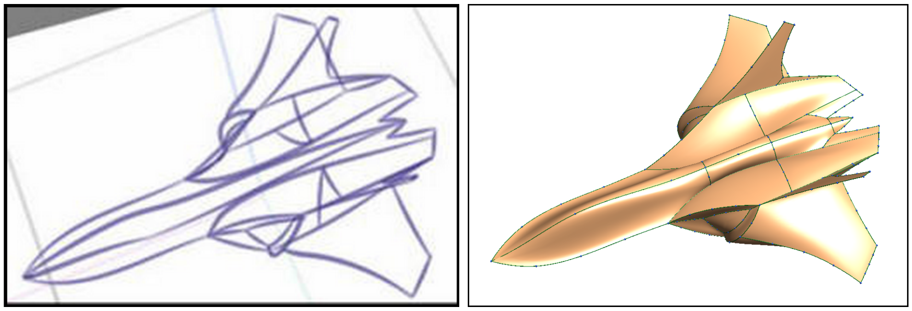


Fig. 1: (a) curve network sketched in 3D, (b) collection of transfinite surfaces

In a recent paper by Salvi et al. [7], the authors proposed two new transfinite surface representations, a multi-sided generalization of the Coons patch [3], and a patch combining Coons patch-like interpolants called "composite ribbons". We will limit our investigation here to the *generalized Coons* or GC patch, and its relation to the well-known Gregory patch [5]. These two patches, as we will see, are very similar in many respects, and deserve a thorough comparison.

In Section 2 we review the basics of transfinite surface interpolation, followed by our formulation of the classical Gregory patch in Section 3. The derivation of the GC patch is briefly summarized in Section 4, and a contrastive analysis of the two surfaces is given in Section 5, supported by both qualitative and quantitative examples.

2 Preliminaries

Given a 2D domain polygon Γ with sides Γ_i $(i = 1 \dots n)$, an n-sided surface $S(u, v)$ is defined by mapping the points of the domain $(u, v) \in \Gamma$ into 3D. Each side of the domain is mapped onto a 3D boundary curve. Assume that there is a continuous local *side parameterization* function $s_i : \Gamma \to [0, 1]$, that maps Γ_i onto the whole $[0, 1]$ interval in an anti-clockwise fashion. Also, let us denote the prescribed boundary curves and normal vector functions by $P_i(s_i(u, v))$ and $N_i(s_i(u, v))$, respectively. For convenience, we will omit the real parameters, leaving only the mapping function, e.g. $P_i(s_i)$. We investigate two surface schemes, one combining side interpolants, and the other corner interpolants.

A *side interpolant* $I_i(s_i, d_i)$ is a biparametric surface corresponding to one side of the domain polygon. Here $d_i : \Gamma \to \mathbb{R}_{\geq 0}$ is the *distance parameterization*

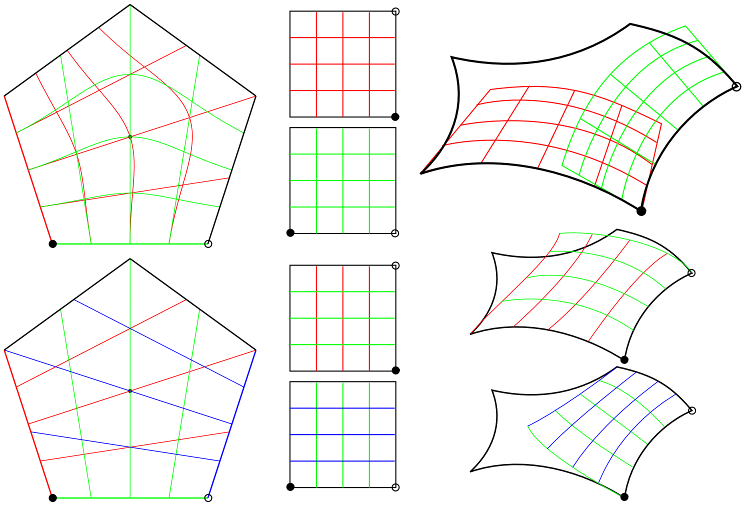


Fig. 2: Side and corner interpolants, and their mappings to the domain polygon.

function, signifying a kind of distance from the boundary, thus for points $(u,v) \in \Gamma_i$, we have $d_i(u,v) = 0$. Then the surface satisfies the given boundary constraints along Γ_i, both positionally and tangentially:

$$I_i(s_i,0) = P_i(s_i), \quad \frac{\partial}{\partial s_i} I_i(s_i,0) \times \frac{\partial}{\partial d_i} I_i(s_i,0) \parallel N_i(s_i), \tag{1}$$

where partial derivatives are interpreted as directional derivatives along constant parameter lines of the given mapping.

We can build an n-sided surface by blending these interpolants together:

$$S_{\text{side}}(u,v) := \sum_{i=1}^{n} I_i(s_i,d_i) B_i(u,v) + C(u,v), \tag{2}$$

where $B_i(u,v)$ are *side blending functions*, and $C(u,v)$ is a *correction patch*. The side blending functions $B_i(u,v)$ must satisfy the following properties: (i) evaluates to 1 along the i-th edge, (ii) gradually vanishes on the adjacent sides, and (iii) gives 0 on all other sides. The correction patch is needed to ensure the interpolation property (see Section 3).

A *corner interpolant* $I_{i,i-1}(s_i,\delta_i)$ is also a biparametric surface. Here the mapping $\delta_i : \Gamma \to [0,1]$ is a different type of distance parameterization—like d_i, it vanishes on Γ_i, but it also satisfies $\delta_i(u,v) = 1 - s_{i-1}$ for points $(u,v) \in \Gamma_{i-1}$ (where $s_i(u,v) = 0$). This interpolant meets the given boundary constraints along sides Γ_i and Γ_{i-1}:

$$I_{i,i-1}(s_i, 0) = P_i(s_i), \tag{3}$$

$$I_{i,i-1}(0, 1 - s_{i-1}) = P_{i-1}(s_{i-1}), \tag{4}$$

$$\frac{\partial}{\partial s_i} I_{i,i-1}(s_i, 0) \times \frac{\partial}{\partial \delta_i} I_{i,i-1}(s_i, 0) \parallel N_i(s_i), \tag{5}$$

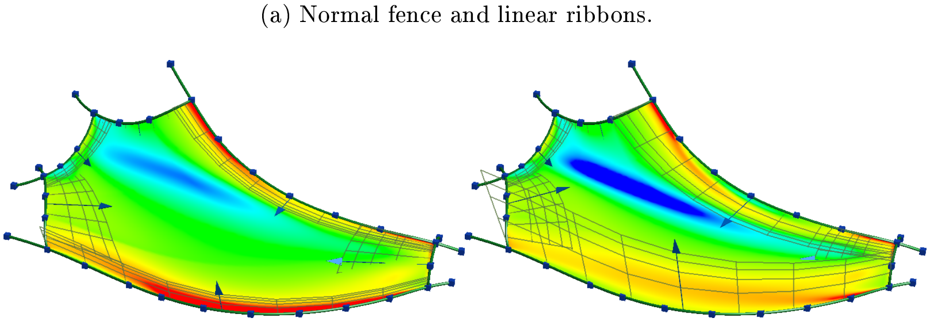
$$\frac{\partial}{\partial s_i} I_{i,i-1}(0, 1 - s_{i-1}) \times \frac{\partial}{\partial \delta_i} I_{i,i-1}(0, 1 - s_{i-1}) \parallel N_{i-1}(s_{i-1}). \tag{6}$$

In this case the n-sided surface is a blended sum of corner interpolants:

$$S_{\text{corner}}(u, v) := \sum_{i=0}^{n} I_{i,i-1}(s_i, \delta_i) B_{i,i-1}(u, v), \tag{7}$$

where $B_{i,i-1}(u, v)$ are *corner blending functions*, that satisfy the following properties: (i) evaluates to 1 at the $(i, i-1)$ corner, (ii) gradually vanishes on the adjacent sides, and (iii) gives 0 on all other sides.

Figure 2 illustrates the two types of interpolants and their mappings to the domain. Formal definitions, with concrete details of parameterization, blending functions, side and corner interpolants, and correction patches are presented in Sections 3 and 4. The remainder of this section deals with some concepts common to both representations.



(a) Normal fence and linear ribbons.

(b) Modifying boundary conditions using ribbon handlers.

Fig. 3: Linear ribbons.

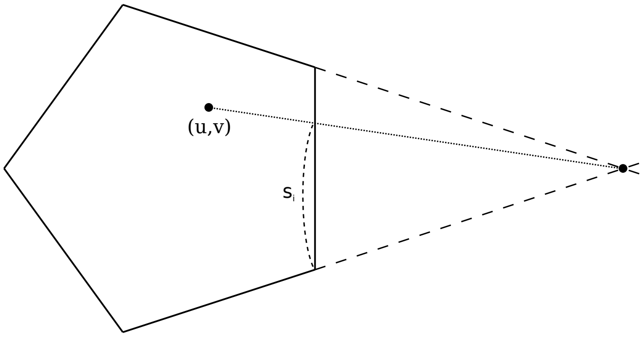


Fig. 4: Construction of the radial parameterization.

Throughout this paper we use regular polygons as domains. Irregular domains can be useful when handling highly curved, complex surfaces, see [8].

We introduce the notion of the simple *linear ribbon* surface (see Figure 3a) as follows:

$$R_i(s_i, d_i) := P_i(s_i) + d_i T_i(s_i), \tag{8}$$

where $T_i(s_i)$ is a cross-derivative perpendicular to the *normal fence* $N_i(s_i)$, defined by

$$T_i(s_i) := \alpha_i(s_i) N_i(s_i) \times P_i'(s_i) + \beta_i(s_i) P_i'(s_i). \tag{9}$$

Here $\alpha_i(s_i)$ and $\beta_i(s_i)$ are scalar functions, representing the two degrees of freedom (the angle and length of the derivative in the tangent planes along the boundary). These can be indirectly accessed by the designer using intuitive controls called *ribbon handlers.* For example, in Figure 3b, the magnitude of the ribbons is modified to give a greater fullness to the patch interior.

3 Corner-Based Patch

The original Charrot–Gregory patch [2] was a five-sided patch, defined over a regular five-sided polygonal domain. Its boundary conditions were given as functions over this domain. It was later generalized for n-sided surfaces [5]; here we will examine its variant, the *CB patch,* presented in [8,7]. Its main feature is that the surface is defined by means of corner interpolants composed of simple ribbons:

$$S_{\mathrm{CB}}(u, v) := \sum_{i=1}^{n} I_{i,i-1}(s_i, 1 - s_{i-1}) B_{i,i-1}(u, v). \tag{10}$$

Comparing with Eq. 7, we can see that this scheme uses the side parameter of the previous side for defining δ_i, so we do not need two coordinates for each side, as only the side parameters s_i are sufficient.

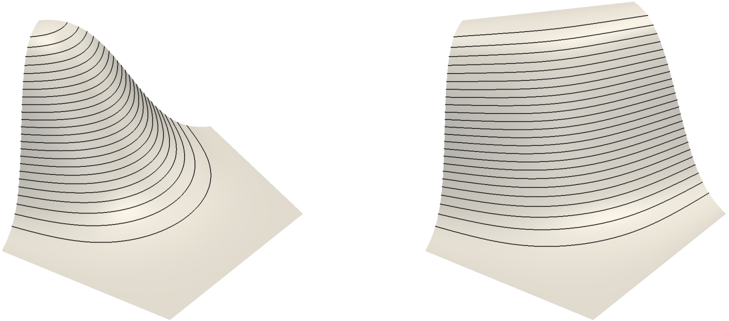
These $s_i(u, v)$ mapping functions are computed by intersecting the i-th domain edge with a line through the (u, v) point and the intersection of the adjacent edges. This is the radial parameterization, see Figure 4. The constant parameter lines are depicted in Figure 6a.

Blending functions operate using a *distance measure* $\Delta_i(u,v)$ that represents a notion of distance from the i-th domain edge. For our purposes it is sufficient, that it vanishes on the edge, and increases monotonically as we move away from it. A practical choice is the Euclidean distance from the line of the i-th edge, i.e., the perpendicular distance.

Let $D_{i_1\dots i_k} = \prod_{j\notin\{i_1\dots i_k\}} \Delta_j^2$, then the corner blend is defined as

$$B_{i,i-1}(u,v) := \frac{D_{i,i-1}}{\sum_j D_{j,j-1}} \left(= \frac{1/(\Delta_i\Delta_{i-1})^2}{\sum_j 1/(\Delta_j\Delta_{j-1})^2} \right). \tag{11}$$

See Figure 5a. Note, that the expression in the parentheses is a more efficient way to compute this function, but it is unstable near the boundaries.



(a) Corner blend.

(b) Side blend.

Fig. 5: Blending functions.

In curve network-based design, the defining quantities—including the boundaries and the cross-derviatives—are associated with the edges of the related patches. Thus it would be hard and somewhat artificial to directly define editable corner interpolants. At the same time, since each corner interpolant is supported by two boundary curves we can create these by combining two simple ribbons and applying the Boolean sum technique. Adding two adjacent ribbons would result in some excess term that must be compensated with a *local correction patch* $Q_{i,i-1}$ (see the definition below).

Based on the linear ribbons, we can write the corner interpolant as

$$\begin{aligned} I_{i,i-1}(s_i, 1-s_{i-1}) := {} & R_{i-1}(s_{i-1}, s_i) + R_i(s_i, 1-s_{i-1}) \\ & - Q_{i,i-1}(s_i, s_{i-1}). \end{aligned} \tag{12}$$

Note, that if we also consider $I_{i+1,i}(s_{i+1}, 1-s_i)$, there will be two different parameterizations of R_i, which we denote by $R_i^-(s_i, 1-s_{i-1})$, and $R_i^+(s_i, s_{i+1})$.

The local correction patch guarantees that the interpolation properties are the same as for the generating ribbons, i.e.,

$$I_{i,i-1}(s_i,0) = R_i(s_i,0), \tag{13}$$
$$I_{i,i-1}(0,1-s_{i-1}) = R_{i-1}(s_{i-1},0), \tag{14}$$
$$\frac{\partial}{\partial s_{i-1}} I_{i,i-1}(s_i,0) = \frac{\partial}{\partial d_i} R_i^-(s_i,0), \tag{15}$$
$$\frac{\partial}{\partial s_i} I_{i,i-1}(0,1-s_{i-1}) = \frac{\partial}{\partial d_{i-1}} R_{i-1}^+(s_{i-1},0). \tag{16}$$

A simple computation leads to

$$Q_{i,i-1}(s_i,s_{i-1}) := P_i(0) + s_i P_i'(0) - (1-s_{i-1})P_{i-1}'(1) + s_i(1-s_{i-1})W_{i,i-1}, \tag{17}$$

where $W_{i,i-1}$ is the common twist vector. If such a common twist does not exist, Gregory's rational twists can be used instead [4]. The above equation also assumes that

$$T_{i-1}(1) = P_i'(0) \quad \text{and} \quad T_i(0) = -P_{i-1}'(1) \tag{18}$$

hold. This is easily enforced, but similar rational expressions can be applied here, as well.

4 Generalized Coons Patch

The generalization follows the original idea of the Coons patch, i.e., computing a surface by a Boolean sum of interpolants and a correction patch. This involves first rearranging the terms of the Coons patch into the form

$$S_{\text{CP}}(u,v) := \sum_{i=1}^{4} R_i(s_i,d_i)\alpha_0(d_i) - \sum_{i=1}^{4} Q_{i,i-1}(s_i,s_{i-1})\alpha_0(s_i)\alpha_1(s_{i-1}), \tag{19}$$

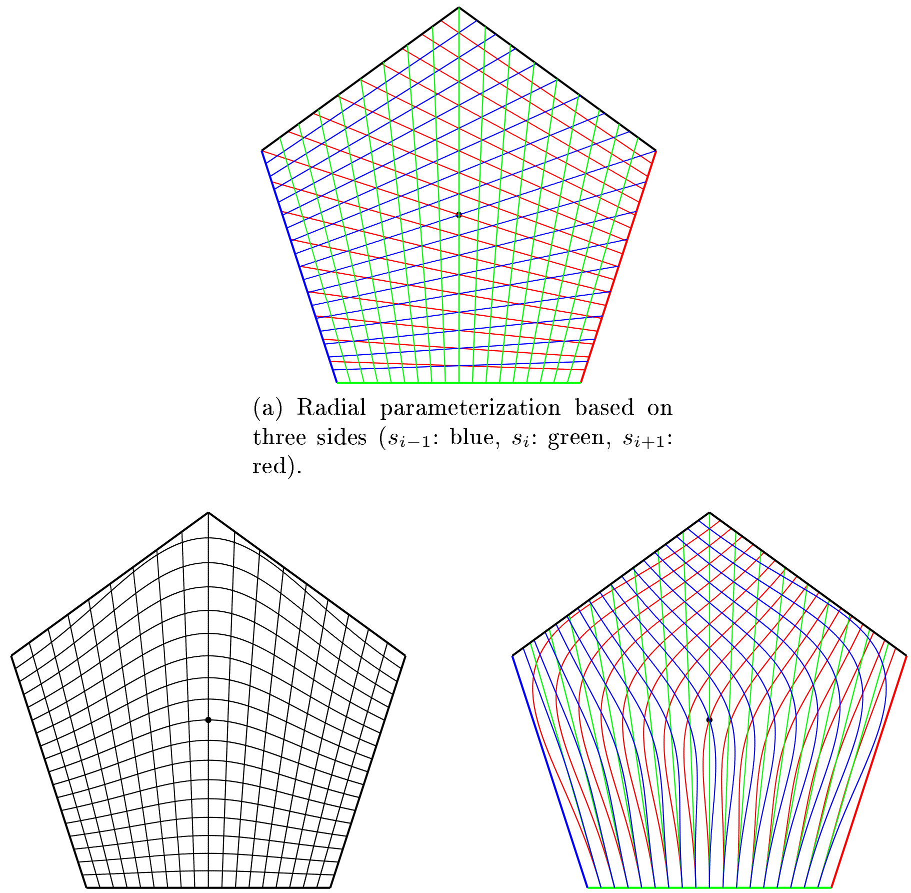
where $\alpha_0(u) = 2u^3 - 3u^2 + 1$ and $\alpha_1(u) = -2u^3 + 3u^2$ are Hermite polynomials, and the distance parameter is defined by $d_i = 1 - s_{i-1} = s_{i+1}$. This can be adapted for n-sided surfaces, resulting in the equation

$$S_{\text{GC}}(u,v) := \sum_{i=1}^{n} R_i(s_i,d_i) \cdot B_i(u,v) - \sum_{i=1}^{n} Q_{i,i-1}(s_i,s_{i-1}) \cdot B_{i,i-1}(u,v). \tag{20}$$

Comparing this to Eq. 2, we can see the substitutions

$$I_i(s_i,d_i) := R_i(s_i,d_i), \quad C(u,v) := \sum_{i=1}^{n} Q_{i,i-1}(s_i,s_{i-1}) \cdot B_{i,i-1}(u,v). \tag{21}$$

Here $B_i(u,v) = B_{i,i-1}(u,v) + B_{i+1,i}(u,v)$, which satisfies the required properties, see Figure 5b.



(a) Radial parameterization based on three sides (s_{i-1}: blue, s_i: green, s_{i+1}: red).

(b) Interconnected parameterization based on the bottom side.

(c) Interconnected parameterization: constant s lines of the bottom side (green), constant d lines of the adjacent sides (d_{i-1}: blue, d_{i+1}: red).

Fig. 6: Parameterizations.

The last piece of the puzzle is the parameterization. We can compute s_i the same way as before, but for d_i we have some restrictions. For a point on the i-th side, the following requirements must be met:

$$d_i = 0, \qquad d_{i-1} = s_i, \qquad d_{i+1} = 1 - s_i, \tag{22}$$

$$\frac{\partial d_{i-1}}{\partial u} = \frac{\partial s_i}{\partial u}, \quad \frac{\partial d_{i+1}}{\partial u} = -\frac{\partial s_i}{\partial u}, \quad \frac{\partial d_{i-1}}{\partial v} = \frac{\partial s_i}{\partial v}, \quad \frac{\partial d_{i+1}}{\partial v} = -\frac{\partial s_i}{\partial v}. \tag{23}$$

A simple solution for these is the *interconnected parameterization*, which combines the side parameters of adjacent sides to compute a distance parameter:

$$d_i(u,v) := (1 - s_{i-1}(u,v)) \cdot B(s_i) + s_{i+1}(u,v) \cdot (1 - B(s_i)), \tag{24}$$

where $B(s_i)$ is a blending function, such as the rational blend

$$B(s_i) := \frac{(1-s_i)^2}{s_i^2 + (1-s_i)^2}.$$

This blending function has the same properties as that of the cubic Hermite function, i.e., $B(0) = 1$, and $B'(0) = 0$. Figure 6b shows the constant parameter lines of this mapping. The effect of the parameterization restrictions are clearly visible in Figure 6c.

5 Comparison of CB and GC Patches

At first sight, the two patches in this paper do not show much similarity. Almost all components—the interpolants, the blending functions, the parameterization—are different. Closer inspection, however, reveals that the two representations are closely related. In Section 5.1 we examine their relationship from a purely algebraic point of view, then in the rest of the section a numerical analysis and two test surfaces are presented.

5.1 Theoretical Differences

From the definitions of the CB patch and the corner interpolant, we have

$$\begin{aligned} S_{\mathrm{CB}}(u,v) &= \sum_{i=1}^{n} I_{i,i-1}(s_i, 1 - s_{i-1}) B_{i,i-1}(u,v) \\ &= \sum_{i=1}^{n} \left[R_{i-1}(s_{i-1}, s_i) + R_i(s_i, 1 - s_{i-1}) - Q_{i,i-1}(s_i, s_{i-1})\right] B_{i,i-1}(u,v) \\ &= \sum_{i=1}^{n} R_i(s_i, 1 - s_{i-1}) B_{i,i-1}(u,v) + R_i(s_i, s_{i+1}) B_{i+1,i}(u,v) \\ &\quad - \sum_{i=1}^{n} Q_{i,i-1}(s_i, s_{i-1}) B_{i,i-1}(u,v). \end{aligned} \tag{25}$$

Comparing this with the definition of the GC patch

$$\begin{aligned} S_{\mathrm{GC}}(u,v) &= \sum_{i=1}^{n} R_i(s_i, d_i) \cdot B_i(u,v) - \sum_{i=1}^{n} Q_{i,i-1}(s_i, s_{i-1}) \cdot B_{i,i-1}(u,v) \\ &= \sum_{i=1}^{n} R_i(s_i, d_i) B_{i,i-1}(u,v) + R_i(s_i, d_i) B_{i+1,i}(u,v) \\ &\quad - \sum_{i=1}^{n} Q_{i,i-1}(s_i, s_{i-1}) \cdot B_{i,i-1}(u,v), \end{aligned} \tag{26}$$

we can see that the only difference is that of parameterization. Let us examine the differing parts, i.e., the blended contribution of the i-th ribbon:

$$\begin{aligned}R_i^{\mathrm{CB}} &= R_i(s_i, 1-s_{i-1})B_{i,i-1}(u,v) + R_i(s_i, s_{i+1})B_{i+1,i}(u,v)\\ &= P_i(s_i)\left[B_{i,i-1}(u,v) + B_{i+1,i}(u,v)\right]\\ &\quad + \left[(1-s_{i-1})B_{i,i-1}(u,v) + s_{i+1}B_{i+1,i}(u,v)\right]T_i(s_i) \qquad (27)\\ R_i^{\mathrm{GC}} &= R_i(s_i, d_i)B_{i,i-1}(u,v) + R_i(s_i, d_i)B_{i+1,i}(u,v)\\ &= P_i(s_i)\left[B_{i,i-1}(u,v) + B_{i+1,i}(u,v)\right]\\ &\quad + \left[(1-s_{i-1})B(s_i) + s_{i+1}(1-B(s_i))\right]\\ &\quad\quad \cdot \left[B_{i,i-1}(u,v) + B_{i+1,i}(u,v)\right] \cdot T_i(s_i). \qquad (28)\end{aligned}$$

The difference of the two formulas is $R_i^{\mathrm{CB}} - R_i^{\mathrm{GC}} = \xi_i \cdot T_i(s_i)$, where

$$\begin{aligned}\xi_i &= (1-B(s_i)) \cdot (1-s_{i-1})B_{i,i-1}(u,v) + B(s_i) \cdot s_{i+1}B_{i+1,i}(u,v)\\ &\quad - B(s_i) \cdot (1-s_{i-1})B_{i+1,i}(u,v) - (1-B(s_i)) \cdot s_{i+1}B_{i,i-1}(u,v) \qquad (29)\\ &= (s_{i+1} - (1-s_{i-1})) \cdot \left[B(s_i)B_{i+1,i}(u,v) - (1-B(s_i))B_{i,i-1}(u,v)\right]. \qquad (30)\end{aligned}$$

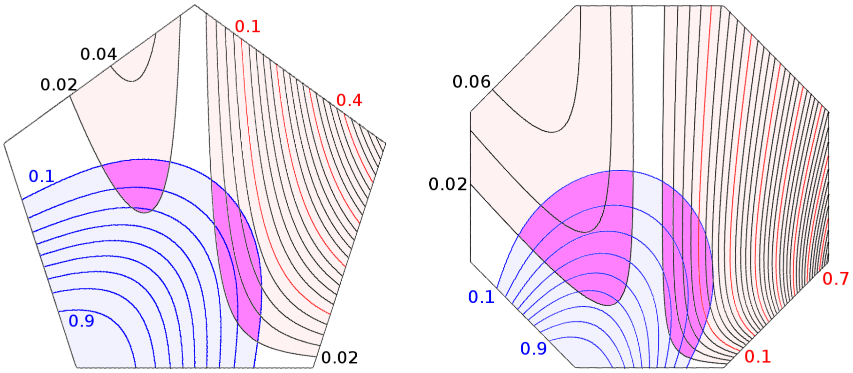
It is easy to see that ξ_i vanishes on Γ_i. This is to be expected, as both patches interpolate the boundaries. See Section 5.2 for a numerical evaluation of this measure.

Following the notations of Section 3, the CB patch has R_i^- and R_i^+, two different interpolants, where the GC patch has R_i. This means that evaluating a point of an n-sided surface requires $2n$ ribbon evaluations, instead of only n, as in the case of the GC patch. On the other hand, the parameterization function is simpler, since we do not need distance parameters. Computing the (s_i, d_i) parameters of a domain point requires $3n$ mappings, considering that each distance parameter is derived from two side parameters (although this can be overcome by caching).

The implications of this difference go far beyond efficiency concerns. Having two parameterizations for the same interpolant means that in evaluating a point on the surface, the contribution of a given ribbon comes from two different points. This is very counterintuitive, as the modification of linear interpolants has only an indirect effect on the final surface.

5.2 Examples

As we have seen above, CB and GC patches evaluate the linear ribbons with different parameterizations. But how different are these? How much does it matter, if we compute $R_i(s_i, d_i)$ instead of $R_i(s_i, 1-s_{i-1})$? Figure 7 tries to answer these questions. Placing side i at the bottom, the black and red lines show the contours of $|d_i - (1-s_{i-1})|$. Since these ribbons are then multiplied by the corner blend $B_{i,i-1}(u,v)$ associated with the lower left corner of the images, it is instructive to examine its contours (shown in blue), as well. We can see that larger parametric deviations are virtually cancelled by the blend function, and only the regions in pink may have a slightly detectable effect on the resulting surface. Comparing Figures 7a and 7b, we can see that the deviations grow with the number of sides.



(a) Five-sided domain.

(b) Eight-sided domain.

Fig. 7: Parametric deviations between CB and GC patches.

Numerical tests show, that ξ_i defined in Section 5.1 practically vanishes on the whole domain. Indeed, the maximal value of

$$|B(s_i)B_{i+1,i}(u,v) - (1 - B(s_i))B_{i,i-1}(u,v)| \tag{31}$$

is numerically negligable. Modifying any of the variables fixed above, such as using irregular domains, or defining $B(s_i) := \alpha_0(s_i)$, employing different parameterizations, results in more substantial divergences.

At the end of this section we show a six-sided patch using both representations, with the same boundary constraints, see Figure 8. The two curvature maps are visually identical, though the extremely sensitive isophotes show some tiny differences. There is a noticable difference in computation time between the two surfaces, due to reasons mentioned in Section 5.1. In our prototype system, evaluating one point of an n-sided surface took around $n \cdot 0.16$ms for the CB patch, and $n \cdot 0.12$ms for the GC patch on a 2.8GHz machine, showing a 25% speedup. Computation of the GC patch could be done even faster, if symmetries of the regular polygon are taken into account.

Conclusion

Research on transfinite surface interpolation has been recently revitalized thanks to new approaches to 3D curve sketching. In this paper we have analyzed and compared two new surface representations: the corner-based patch (a variant of the Gregory patch), and the generalized Coons patch. We have shown that the

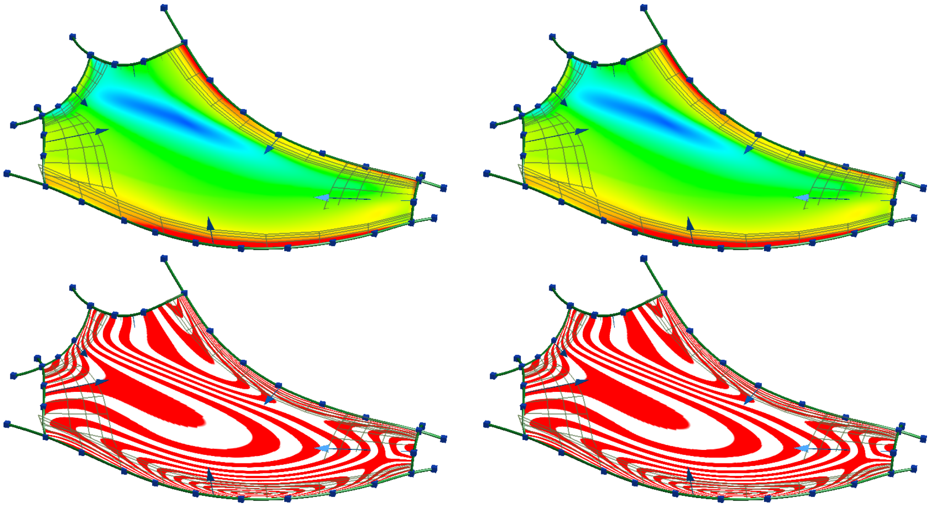


Fig. 8: Visual comparison of CB and GC patches.

equations of the two representations are similar, except for their parameterization. We have also pointed out, that the GC patch is more intuitive for curvenet-based design and is computationally more efficient. Visually—inspecting curvature maps or isophotes—one can hardly find differences between the two formulations.

Acknowledgements

This work was supported 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 to develop this prototype system is highly appreciated.

References

1. S-H. Bae, R. Balakrishnan, and K. Singh. EverybodyLovesSketch: 3D sketching for a broader audience. In *22nd Symposium on User Interface Software and Technology*, pages 59–68. ACM, 2009.
2. P. Charrot and J. A. Gregory. A pentagonal surface patch for computer aided geometric design. *Computer Aided Geometric Design*, 1(1):87–94, 1984.
3. S. A. Coons. Surfaces for computer-aided design of space forms. Technical Report MIT/LCS/TR-41, Massachusetts Institute of Technology, 1967.
4. J. A. Gregory. Smooth interpolation without twist constraints. In R. E. Barnhill and R. F. Riesenfeld, editors, *Computer Aided Geometric Design*, pages 71–88. Academic Press, Inc., 1974.
5. J. A. Gregory. N-sided surface patches. In *Mathematics of Surfaces I*, pages 217–232. Oxford University Press, 1986.

6. C. Grimm and P. Joshi. Just Draw It: a 3D sketching system. In *Symposium on Sketch-Based Interfaces and Modeling*, pages 121–130. Eurographics Association, 2012.
7. P. Salvi, T. Várady, and A. Rockwood. Ribbon-based transfinite surfaces. *Computer Aided Geometric Design*, 31(9):613–630, 2014.
8. T. Várady, A. Rockwood, and P. Salvi. Transfinite surface interpolation over irregular n-sided domains. *Computer Aided Design*, 43(11):1330–1340, 2011.