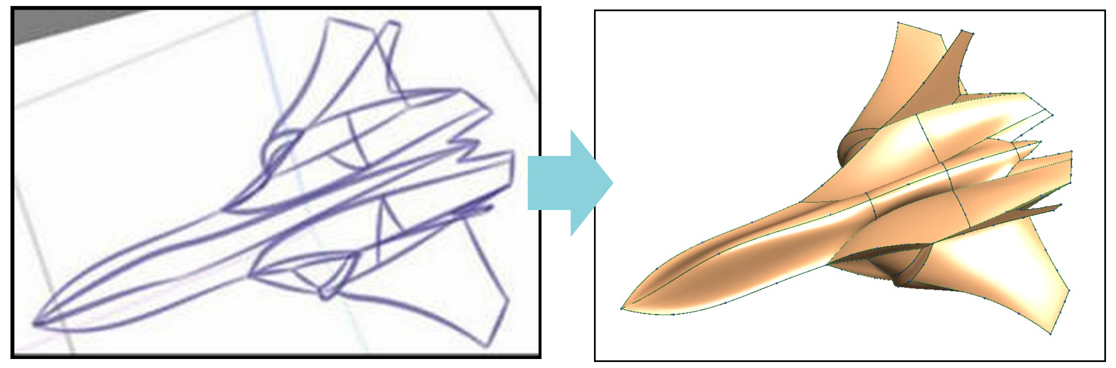
MŰEGYETEM 1782

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

Péter Salvi and Tamás Várady, Budapest University of Technology and Economics

Overview

Modeling complex 3D objects with free-form geometry is a challenging task. In curve network-based modeling, 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.



There are various choices for representing the surfaces in this scheme. Transfinite surface interpolation is especially appropriate, as it requires only boundary information.

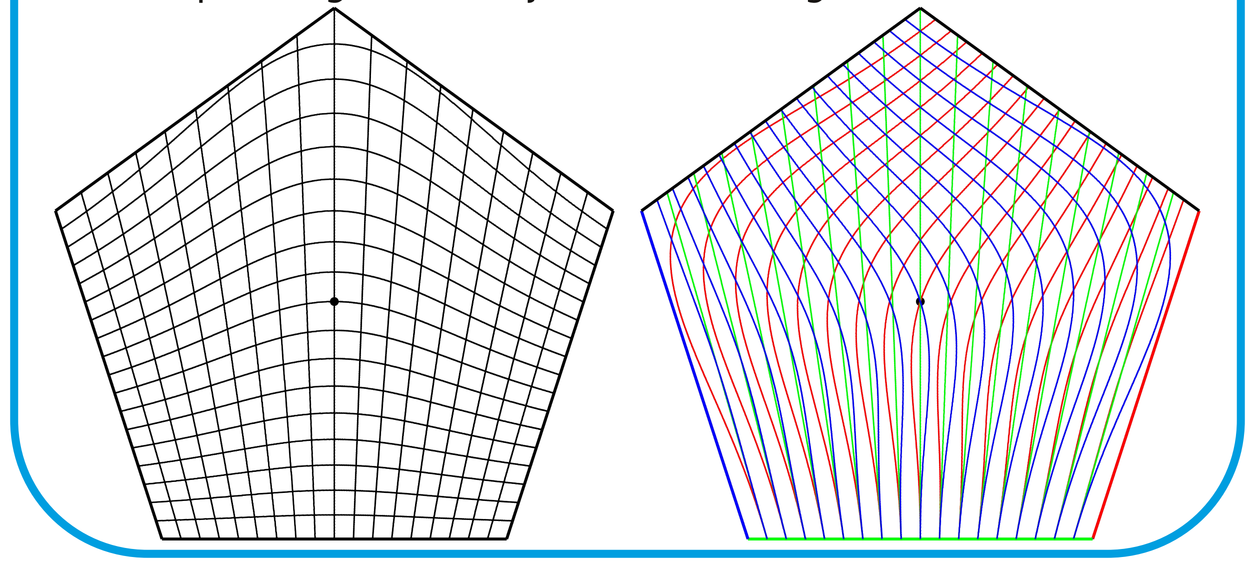
In a previous paper, the authors have introduced a generalization of the Coons patch, as well as a variant of the well-known Gregory patch. The aim of the current research is to compare these two representations.

Parameterizations

There are several parameterization requirements for the GC patch to interpolate correctly. One of these stipulates that s_i constant parameter lines should start with the same tangents as the adjacent d_{i-1} and d_{i+1} constant parameter lines.
This is achieved by the interconnected parameterization:

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

where $B(s_i)$ is a suitable blending function.
The images below show the (s_i, d_i) system [left], and s_i lines based on the bottom line, with d_i lines corresponding to the adjacent sides [right].

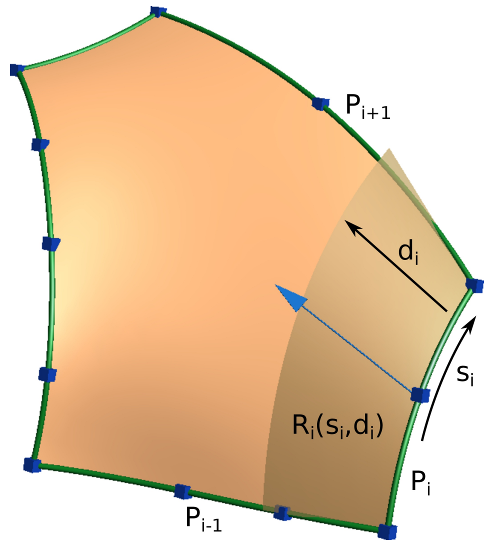


Transfinite Surface Interpolation

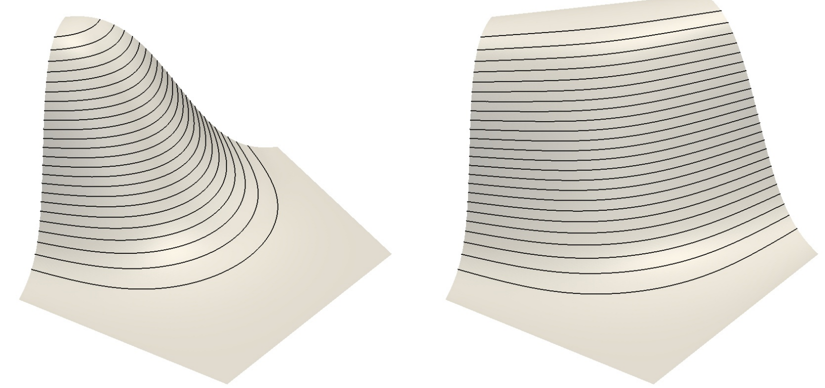
Given a loop of curves and corresponding cross-derivative functions, transfinite surface interpolation fills the loop with a surface exactly interpolating its boundaries while reproducing the cross-derivatives. In practical applications, the differential data are automatically generated from the boundary loop. The interpolation is based on a convex polygonal domain (usually a regular *n*-gon), and is composed of several components.

Components

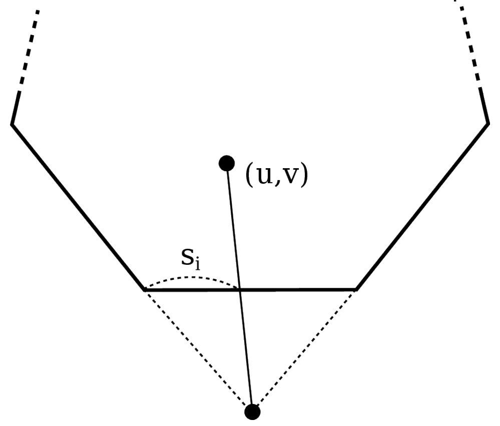
Linear ribbons: biparametric surfaces satisfying the boundary constraints.



Blending functions: weight functions that vanish on most edges of the domain, but evaluate to 1 at a given corner or side.



Parameterizations: mappings from domain space to ribbon space, such as the radial map. May involve two kinds of functions: *side-* (s_i) and *distance-parameters* (d_i).

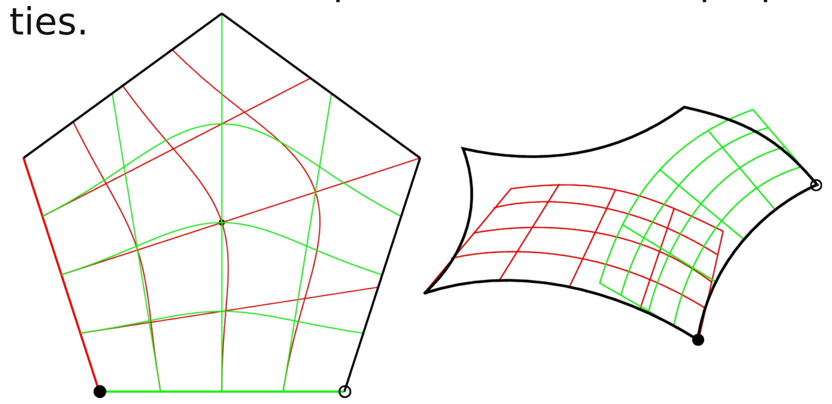


Generalized Coons Patch

The GC surface combines *side interpolants* (the linear ribbons R_i) using the side-based blending functions B_i, and subtracts the excess by applying a correction patch $Q_{i,i-1}$, similarly to the original Coons patch:

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

This scheme uses two parameterization functions with special differential properties.

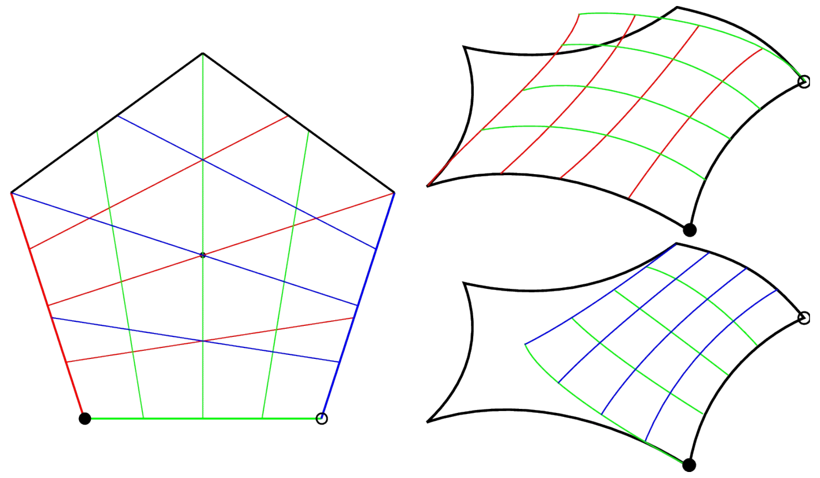


Corner-Based Patch

The CB surface combines *corner interpolants* $I_{i,i-1}$ (composed of two linear ribbons and a correction patch) using the corner-based blending functions $B_{i,i-1}$, similarly to the original Gregory patch:

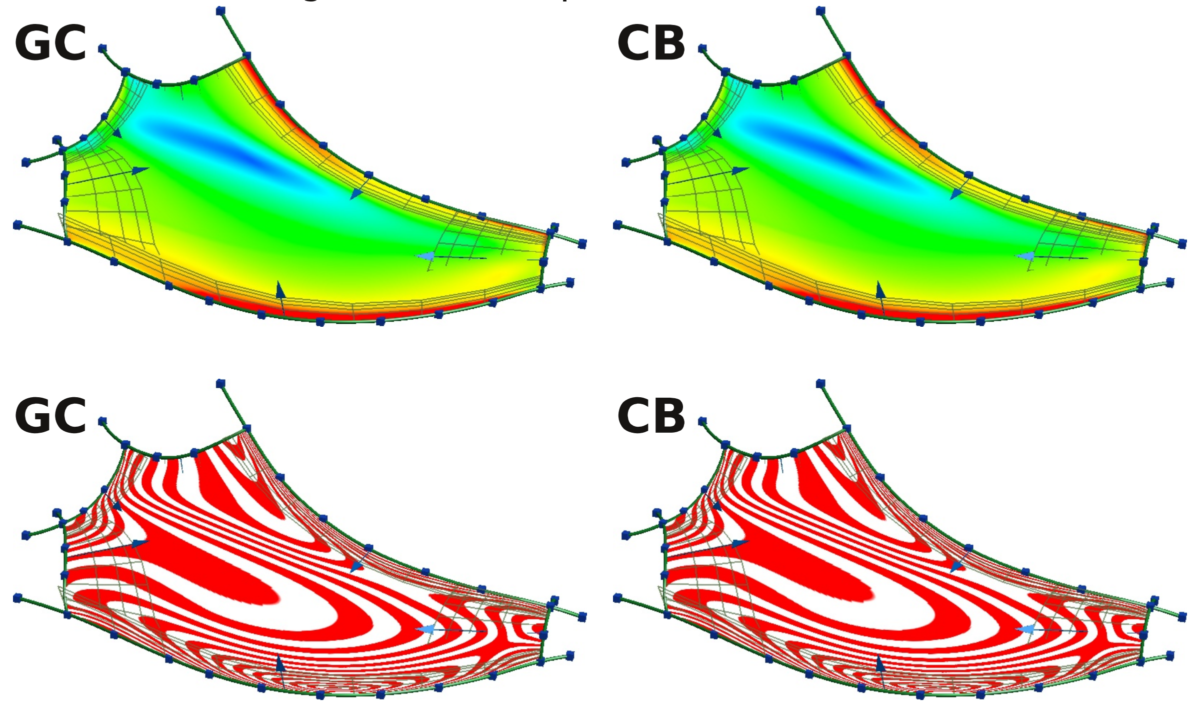
$$S_{CB}(u,v) := \sum_{i=1}^{n} I_{i,i-1}(s_i, 1-s_{i-1}) B_{i,i-1}(u,v).$$

This scheme uses only one domain map, the radial parameterization, so internally it has two parameterizations of R_i.

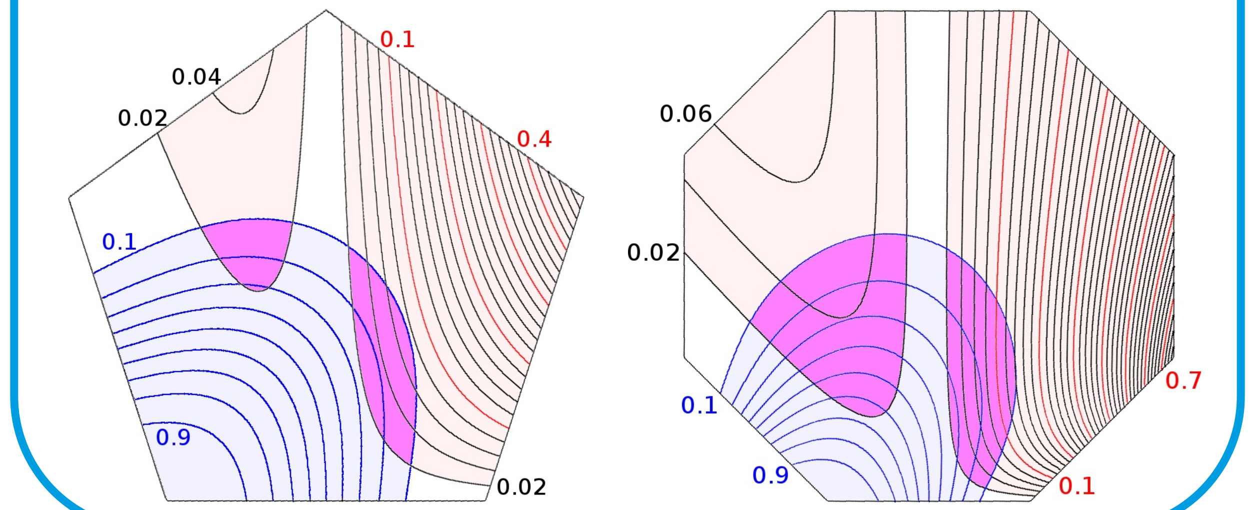


Comparison

Visual interrogation techniques show almost no difference:



Analysis of the formulae reveals that the only difference is in the parameterization. In the corner-based patch scheme, every domain point has two images on each ribbon surface. The extent of these deviations (red) is mostly cancelled by the blend functions (blue). The images below also show that the perceptible deviation (pink) is proportional to the number of sides.



Conclusion

We have compared two transfinite surface representations, a generalization of the Coons patch (GC) and a variant of the Gregory patch (CB).
It has been shown that the two methods are visually very similar, but due to their different parameterizations, the GC patch is more intuitive for curve network-based design, and it is also slightly more efficient to compute.