Tenth Hungarian Conference on Computer Graphics and Geometry, Budapest, 2022

Implicit representations for multi-sided surface patches

Ágoston Sipos, Tamás Várady and Péter Salvi

Department of Control Engineering and Information Technology, Budapest University of Technology and Economics

Abstract

Many multi-sided surface representations exist that satisfy a given set of boundary constraints. Most of them are defined by means of parametric equations, i.e., there is a given domain and a function mapping it to a surface. Implicit surfaces represent an interesting alternative to avoid the difficulties of parametrization, although they also have their own problems. In this paper, we review various schemes for representing implicit patches in a general topology setting, showing their advantages and deficiencies.

Keywords: Implicit surfaces, Algebraic fitting, Multi-sided surfaces.

1. Introduction

Multi-sided surface modeling is dominated by parametric representations. While *control point based* surfaces[1, 2] are calculated as a weighted average of an editable point structure; *transfinite interpolation* surfaces[3, 4, 5] are constructed as a blend of ribbon surfaces, defined along individual boundaries. The difficulty of modeling with parametric surfaces is due to the necessity of an intrinsic parameterization to map a planar domain to 3D, which is a delicate issue. While tessellating parametric surfaces is easy, several geometric interrogations are computationally complex. Many of these can be performed more efficiently using implicit surfaces; for example, intersections, connecting trimmed patches, ray casting, and so on.

An implicit surface is interpreted as an isosurface of a scalar-valued function defined on the 3-dimensional space. This also defines two half-spaces, thus making point membership classification easy. Regular surfaces in CAD systems are generally represented in implicit form, however, extra care is needed when modeling free-form implicit surfaces with higher degrees, as these may produce various shape artifacts, such as singularities, self-intersections, and disconnected parts. Implicit surfaces are more rigid than parametric ones, and properly adjusting the shape interior without a geometric control structure may be a challenge.

Our current interest is to investigate multi-sided implicit surfaces, which can be used similarly to their parametric counterparts, while the advantages of using algebraic representations are exploited. We wish to define a finite portion of a surface that interpolates a collection of prescribed boundary curves and cross-derivatives, and simultaneously ensures a natural blend in the interior without awkward twists, holes, and self-intersections. We will revisit two main concepts for representing implicit patches. The first is *interpolation by algebraic surfaces*,[6, 7] where we directly compute a polynomial surface, satisfying positional and tangential constraints. The other approach is *ribbon-based surfacing*,[8, 9, 10] where we blend together surfaces that "carry" prescribed boundary constraints. This latter solution is somewhat similar to the construction of transfinite patches.

This paper is a follow-up to Sipos et al.,[11] elaborating more on analysis and comparison to other schemes, while leaving the exact details of our method in the original paper. The paper is structured as follows. After reviewing prior work in Section 2, we present potential difficulties that should be avoided for reasonable implicit patches (Section 3). In Sections 4 and 5 we examine algebraic interpolation and ribbon-based representations, respectively. Related problems and challenges will also be discussed.

2. Previous work

Multi-sided parametric surfaces have a wide presence in the literature. These include transfinite surfaces like the Charrot–Gregory patch,[3] Kato's patch,[12] the Overlap-patch,[13] the Generalized Coons Patch[14] and the Midpoint Patch;[15] as well as control point based surfaces like the S-patch,[1] toric patches[16] and the Generalized Bézier patch.[2] These patches are all parameterized over a planar polyg-

onal domain, although non-polygonal ones also exist, like that of the Curved Domain Bézier patch.[17] We also have to mention general curve network interpolation methods with energy minimization schemes, which can be solved both in parametric and discretized forms.[18]

Fitting implicit curves and surfaces to boundary conditions dates back to Liming's publication[19] where fitting conic segments (degree-two polynomials) on tangential conditions was discussed. It provided an implicit equation for conics in the form $C(\mathbf{p}) = (1-\lambda)\cdot L_1(\mathbf{p})\cdot L_2(\mathbf{p}) - \lambda\cdot L_3^2(\mathbf{p})$, where $L_1(\mathbf{p}) = 0$ and $L_2(\mathbf{p}) = 0$ are lines in implicit form representing tangential constraints. L_3 is a line through the endpoints of the conic segment and λ is a shape parameter between 0 and 1. The rest of this section will review methods achieving similar results for 3D surfaces with an arbitrary number of constrained boundaries.

Bajaj and Ihm[6] investigated the direct fit of an algebraic surface onto given curves and constrained normal fences. Based on Bézout's theorem, for any given polynomial degree, the coefficients arise as the solution of a linear system. A minimal degree, for which the system is feasible, can also be directly calculated beforehand. Such implicit surfaces are defined on the whole 3D space, often having additional branches or singular points, which complicate ray tracing, distance estimation and similar operations. Defining a bounding box that contains all the necessary and none of the undesirable parts of the surface is also not a trivial task. (See also Section 4.)

Another algebraic polynomial representation is the *A-patch*,[7] which is defined on a tetrahedral space partitioning. These patches are trivariate polynomials using Bézier-like control points and Bernstein basis functions. As a result, 2-, 3- or 4-sided patches are obtained, and it is guaranteed to exclude singularities or self-intersections; however, they have a fairly complicated control structure and it is hard to match boundary constraint equations even in low-degree cases. Later, rational A-patches were also introduced by Xu et al.[20] These representations were primarily used to create C^1-continuous surfaces from triangle meshes.

Functional splines, introduced by Li et al.[8] and Hartmann,[21] represent a direct generalization of Liming's method, defining the patch as a blend between a *base surface*, which is the product of an arbitrary number of ribbon surfaces for the patch to be smoothly connected to, and a *transversal surface*, which interpolates all the boundary curves. Functional splines may achieve any degree of geometric continuity, though they are usually not minimal degree solutions. It has been pointed out by Hartmann and Feng[22] that many configurations, where an inflection inside the patch is needed, cannot be correctly solved by functional splines, so a modified version, called *symmetric functional splines*, was introduced by Hartmann[9] and applied for vertex blending with good results. (See also Section 5.1.)

Várady et al.[10] described a different ribbon-based patch class, called the *I-patch*. Instead of a transversal surface, a pair of a ribbon and a bounding surface is associated with each side of the patch, thus yielding a natural local structure of constraints. I-patches have more degrees of freedom than functional splines; scalar weights are incorporated into the equation, which help to control the interior of the shape.

Recently current authors have been directing their research to deepen knowledge on I-patches and search for modeling applications; some related results can be found in Sipos et al.[11] (See also Section 5.2.)

3. Potential difficulties

In this section, we deal with the potential problems that an implicit surface representation may exhibit. A surface that satisfies certain boundary constraints is not necessarily acceptable, and special care is needed to avoid various inconsistencies, as follows.

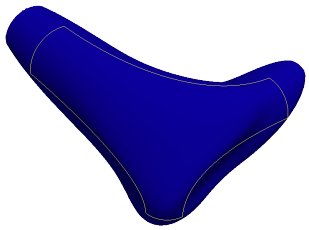
An implicit isosurface may consist of multiple branches. In our context, we only wish to deal with the portion that directly interpolates the boundaries. However, it is not guaranteed in general that other branches will not interact. In Figure 1b multiple disjoint components lie in the close proximity of the boundary curves. It is also possible that these branches intersect each other, leading to singularities along their intersection, these situations need to be identified and possibly eliminated.

When interpolating boundaries, the prescribed condition is generally that the gradient of the implicit patch is equal to the gradient of the ribbon surface multiplied by a scalar. When setting the constant parameters as the solution of equations or by other means, they may have an arbitrary sign. This means that the direction of normal vectors may flip along the border, so the interior of the patch cannot be both connected and nonsingular. In Figure 2 we demonstrate this phenomenon with implicit curves and in Figure 1c with an example surface. Consequently, we emphasize that setting consistent orientations for the gradient constraints is a crucial issue (see also Sipos et al.[11]).

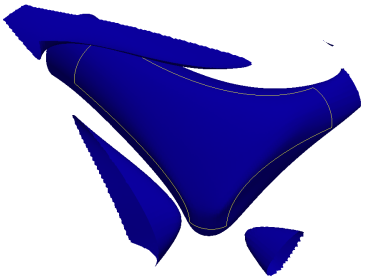
Even when the patch is locally correct in the vicinity of the boundaries, its interior may be unacceptable. Figure 1d shows an example where the isosurface has a hole in the middle; clearly, this should be detected and avoided accordingly.

4. Algebraic interpolation

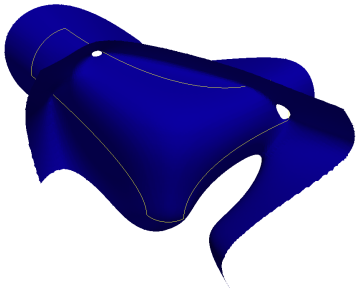
The method described by Bajaj and Ihm[6] deals with constructing a minimal degree algebraic surface satisfying given boundary constraints. These include points with an associated normal vector, or curves with a normal function along them.



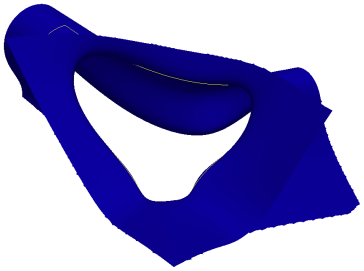
(a) Good surface



(b) Additional branches

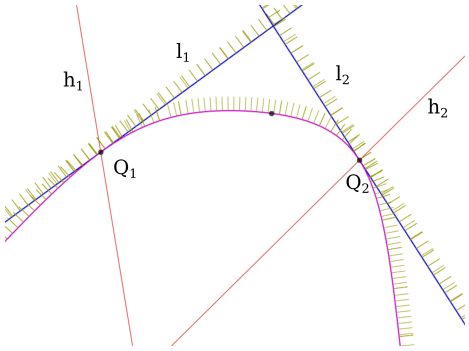


(c) Flipping

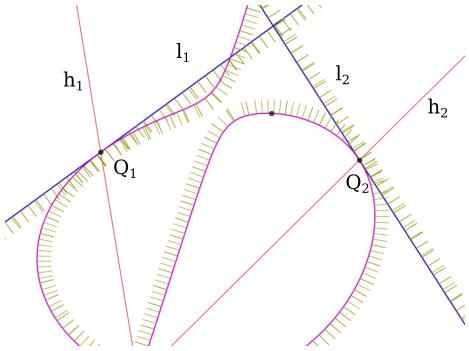


(d) Hole inside

Figure 1: Examples of surfaces fit on the same boundaries



(a) Well oriented constraints: good curve



(b) Badly oriented constraint: curve not connecting Q_1 and Q_2

Figure 2: Good and bad orientation of normal constraints; spikes show the orientation of the gradient on both the constraint lines (blue) and the resulting curves (purple)

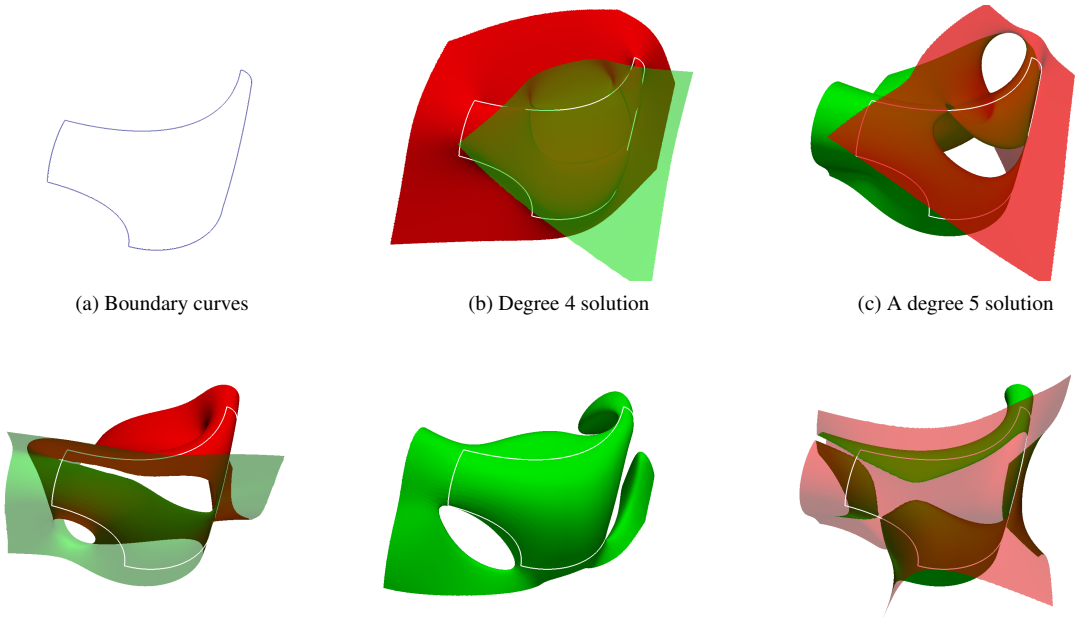
An algebraic surface defined by the function $f : \mathbb{R}^3 \to \mathbb{R}$ is said to interpolate the constraints with C^1 continuity, if

$$f(\mathbf{p}) = 0, \tag{1}$$
$$\nabla f(\mathbf{p}) = \alpha \cdot \mathbf{n}, \tag{2}$$

for all $\mathbf{p}$ specified points or all points of specified curves, where $\mathbf{n}$ is the normal vector at $\mathbf{p}$ and α is a nonzero real number.

For creating the equation system providing the surface coefficients, Bézout's theorem is used: a curve of degree d intersects a surface of degree n at most in $n \cdot d$ points, if the number of intersections is finite; otherwise, a component of the curve lies entirely in the surface. From this, it follows that if we choose $n \cdot d + 1$ points along the curve, then a surface fulfilling the constraints will interpolate the branches of the curve containing those points.

Now having reduced everything to point constraints, for



(a) Boundary curves (b) Degree 4 solution (c) A degree 5 solution

(d) Another degree 5 solution (e) Degree 6 solution (f) Another degree 6 solution

Figure 3: Algebraic surfaces of various degrees fit on boundary curves

all $(\mathbf{p},\mathbf{n})$ pair we have the following equations:[†]

$$f(\mathbf{p}) = 0, \tag{3}$$

$$\mathbf{n}_x \cdot \partial_y f(\mathbf{p}) - \mathbf{n}_y \cdot \partial_x f(\mathbf{p}) = 0, \tag{4}$$

$$\mathbf{n}_x \cdot \partial_z f(\mathbf{p}) - \mathbf{n}_z \cdot \partial_x f(\mathbf{p}) = 0. \tag{5}$$

The number of variables is the number of free coefficients of a degree n algebraic surface, which is $\binom{n+3}{3} - 1$. In the rare case when the rank of the equation system is equal to that, we have a unique solution. If there are fewer independent equations, the system is under-determined and there are infinitely many solutions. If there are more, the system is over-determined, so there are no exact solutions with that degree. The minimum degree solution(s) is the smallest degree for which the equation system is not over-determined.

We experimented, among others, with the setback vertex blend configuration shown in Figure 3. The curves are circular arcs and parabolas. The normal vectors at the corners are uniquely defined by the curves, while the normal fence function is a linear blend.

The minimal degree solution is a quartic surface which is unique (Figure 3b), but the resulting patch contains two components, none of them touching the whole boundary loop.[‡] Using quintic surfaces, the homogenous equation has a 6-dimensional nullspace, from which any element leads to a solution. However, in our investigation no nice surface inside that space has been found, only surfaces like Figures 3c and 3d. Finally, a degree-6 surface configuration instantly supplied a nice surface; Figure 3e shows a patch by taking the coefficient vector belonging to the numerically lowest singular value. We remark that it has been easy to find bad degree 6 surfaces, as well (Figure 3f).

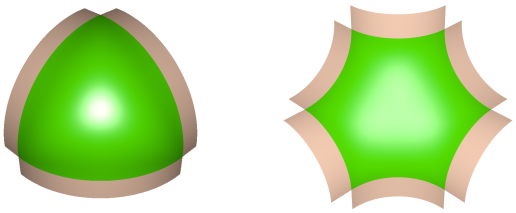
It would be useful to produce nice surfaces via optimization inside the nullspace using an objective function evaluated on the patch. Unfortunately, we are not aware of such a function that provides a good minimum and is computationally feasible.

5. Ribbon-based representations

The methods in this section are based on a common concept: for each boundary of the multi-sided patch, there is a given implicit surface R_i, to which the surface smoothly connects with a given order of continuity. The surface equation is a

[†] Here we suppose that $\mathbf{n}_x \neq 0$. Otherwise the equation $\mathbf{n}_z \cdot \partial_y f(\mathbf{p}) - \mathbf{n}_y \cdot \partial_z f(\mathbf{p}) = 0$ needs to be used instead of one mentioned, so that the coordinate included in both tangential equations is nonzero.

[‡] It is worth noting that a unique minimal degree solution is often good, for example Figure 1a is also such a surface.



(a) Simple 3-sided configuration (b) Simple 6-sided configuration

Figure 4: Sample inputs and patches for ribbon-based implicit surfaces

direct formula that includes the R_i-s and other auxiliary surfaces. Two examples are shown in Figure 4; the R_i surfaces are colored light brown, while the resulting patch is green.

5.1. Functional spline

The formula of functional splines, as given by Li et al.,[8] is

$$(1-\mu)\cdot f-\mu\cdot g^k, \tag{6}$$

where f is called *base surface* and g is called *transversal surface*, $0<\mu<1$ is a scalar parameter, $2\leq k$ is an integer parameter. The property of the functional spline surface is that it connects with G^{k-1} continuity to the base surface in the intersection points of the base and the transversal surface.

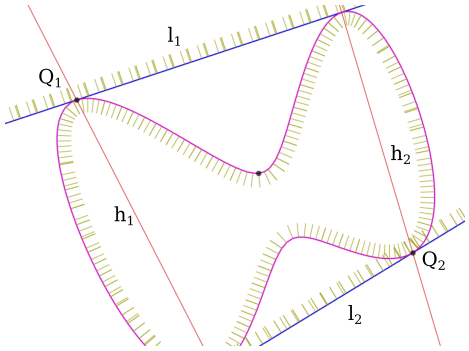
In a multi-sided setting, the base surface should be the union of the ribbon surfaces, which is simply achieved by $f=\prod R_i$. The transversal surface is a surface that interpolates all boundary curves of the patch. In special cases, a single surface can be provided, but the general solution is to create one *bounding surface* (B_i) for interpolating each boundary curve and to use their product as the transversal surface.

Simple 2D examples show that functional splines can only handle convex configurations, having problems with internal inflections. Take, for example, the absurd curve in Figure 5a that connects Q_1 and Q_2 within the area spanned by h_1 and h_2. This limitation has been formalized by Hartmann and Feng,[22] where the exact definition of convexity in this regard and the relevant theorems can be found.

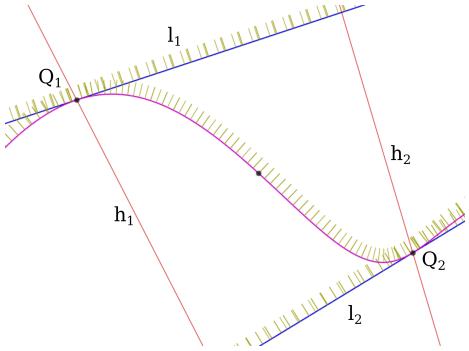
Focusing on 3D surfaces, this means that configurations like Figure 4a can be solved by functional splines, but ones like Figure 4b cannot (causing problems similar to Figure 1c). This problem motivated the introduction of the *symmetric parabolic functional splines*, as described by Hartmann.[9] The following equation has been proposed:

$$(1-\lambda)\cdot f_a\cdot g_b^k-\lambda\cdot f_b\cdot g_a^k=0, \tag{7}$$

where f_a is a surface touching all the locally convex boundaries, f_b is a surface touching all the locally concave boundaries, g_a and g_b intersect f_a and f_b respectively at the appropriate boundaries, and $0<\lambda<1$ is a scalar parameter.



(a) Functional spline with internal inflection



(b) I-patch with internal inflection

Figure 5: Handling inflections

Using the entities R_i–B_i, let $A\subset\{1,2,...,n\}$ be an index set denoting the primary surfaces with the same convexity. Then $f_a=\prod_{i\in A}R_i$, $f_b=\prod_{i\notin A}R_i$, $g_a=\prod_{i\in A}B_i$, $g_b=\prod_{i\notin A}B_i$.

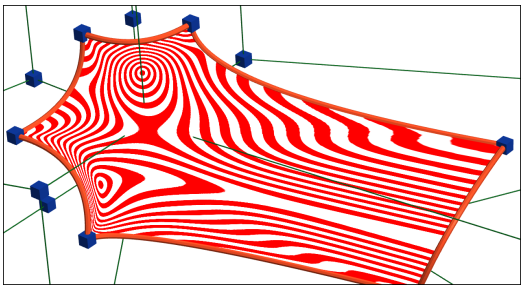
More complex boundary constraints can be solved using symmetric functional splines, see Figure 6a.

5.2. I-patch

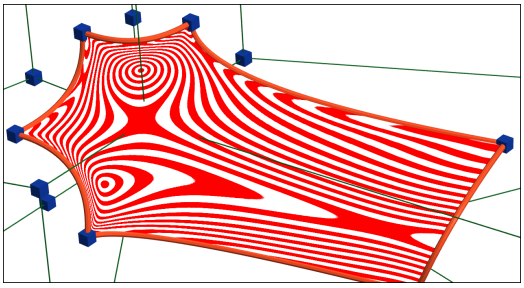
As mentioned in the Introduction, our current focus is to investigate and exploit the capabilities of I-patches. The original I-patch formula was published by Várady et al.[10] It is defined by the equation

$$\sum_{i=1}^{n} w_i R_i \prod_{\substack{j=1\\ j\neq i}}^{n} B_j^k + w_0 \prod_{j=1}^{n} B_j^k = 0, \tag{8}$$

where the boundary curves of the patch are defined as the intersection of associated ribbons R_i and bounding surfaces B_i, $(i=1\ldots n)$. The surface smoothly connects to the ribbons with geometric continuity G^{k-1}. The w_i weights are



(a) 6-sided symmetric functional spline



(b) 6-sided I-patch

Figure 6: Comparison of a symmetric functional spline and an I-patch

free scalar parameters to adjust the contribution of the individual components.

I-patches can also be given in a simple rational form; divide Eq. 8 by the term $\prod_{j=1}^{n} B_j^k$ then the patch equation becomes

$$\sum_{i=1}^{n} \frac{w_i R_i}{B_i^k} + w_0 = 0. \tag{9}$$

This can be interpreted as a combination of (algebraic) distances; i.e., the surface is the locus of points where the sum of the distances $\sum_{i=0}^{n} d_i$ equals to zero, using $d_0 = w_0$ and $d_i = \frac{w_i R_i}{B_i^k}$. It can be shown that distance-based I-patches can reproduce certain standard implicit surfaces, for example spheres, ellipsoids, etc. Note that the distance-based formula should not be evaluated in the very close vicinity of the already well-defined boundary curves, due to $0/0$ singularities. Further special considerations can be found in Sipos et al.[11]

I-patches can also be interpreted as a weighted sum of ribbons based on a blend function, similar to transfinite parametric patches.[4] If we consider the functions

$$F_m = w_m \prod_{\substack{j=1 \\ j \neq m}}^{n} B_j^k \Big/ \sum_{i=1}^{n} w_i \prod_{\substack{j=1 \\ j \neq i}}^{n} B_j^k, \tag{10}$$

it is easy to see that the isosurface of $\sum_{i=1}^{n} F_i R_i + F_0$ is the same as that of the I-patch function, while also $F_i = 1$ at the ith boundary and $F_i = 0$ at all other boundaries, which is generally a prescribed condition in case of parametric patches as well. In Figure 7 an I-patch is colored such that a color is assigned to each blend function, and in every surface point the colors are blended with the same functions.

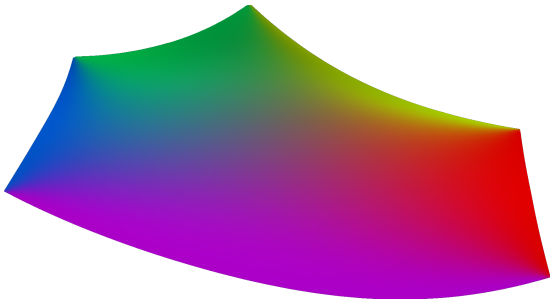


Figure 7: An I-patch colored by the weights of the individual blend functions

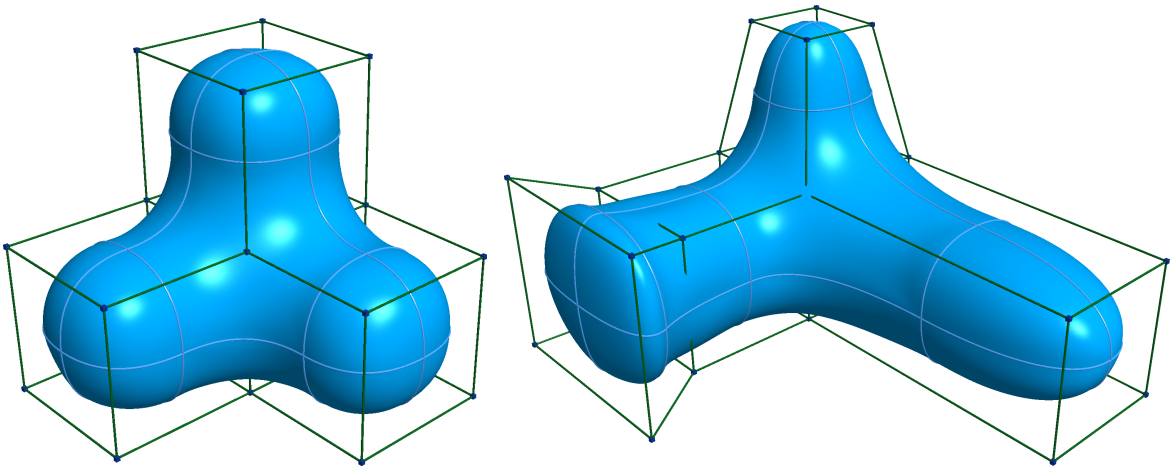
The number of independent scalar parameters in the I-patch equation is equal to the number of sides (we can always divide the equation with w_0), thus there are more possibilities to tweak the shape of the surface than with functional splines. In Figure 6b it can be seen that the moderately uneven curvature distribution of the functional spline is greatly enhanced via setting more optimal shape parameters.

In our previous paper[11] we defined a framework for using I-patches to generate complex continuous surfaces based on control polyhedra (Figure 8). Ribbons and bounding surfaces are generated automatically from the face midpoints and normals, while coefficients are set by a reference point at the middle of the patches assigning equal "algebraic distances" to the members of the above sum. There are special cases where further geometric constraints need to be enforced in order to obtain a valid collection of patches, see details in the paper.

We have also used I-patches for setback vertex blending,[23] where special multi-sided patches are created in order to smoothly connect converging edge blends and planar primary faces (Figure 9). As a result, these patches often have a high number of sides, being naturally handled by the I-patch construction.

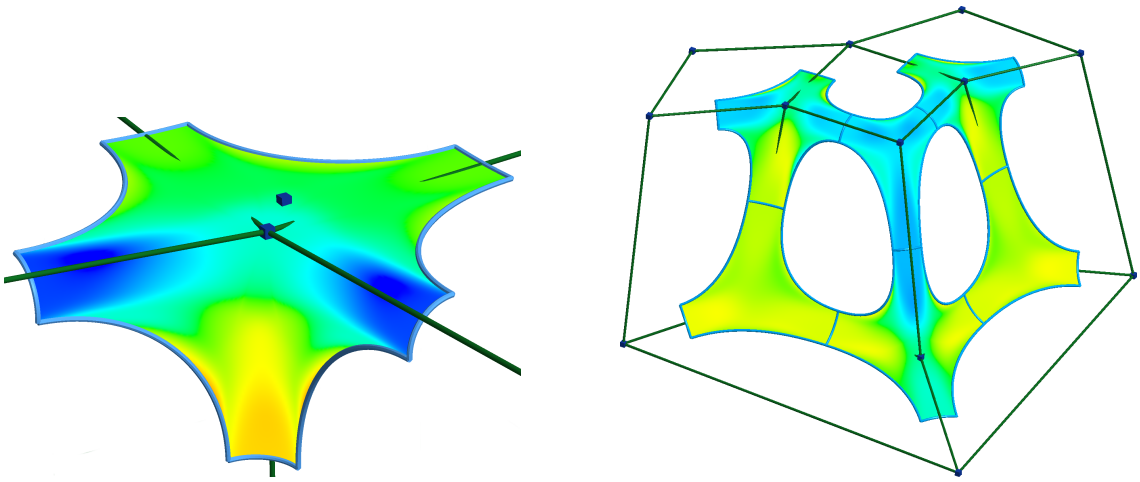
Conclusion

We have investigated a variety of multi-sided patches defined by implicit equations. These patches nicely connect to standard implicit surfaces, such as planes, cylinders, tori, etc., and provide several computational benefits for surface interrogations. We have found that the I-patch representation is presumably the most flexible for free-form modeling; since it is a natural geometric approach to constrain surfaces by means of implicit ribbons and bounding surfaces. There are various options to construct ribbons and adjust their shape



(a) Simple polyhedral blend (b) More complex polyhedral blend

Figure 8: Polyhedral blends created using I-patches



(a) A 10-sided setback vertex blend (b) Vertex blends defined by a control polyhedron

Figure 9: Setback vertex blends created using I-patches

parameters, but several open questions need to be answered before we can draw a final conclusion about the capabilities of multi-sided implicit modeling.

Acknowledgements

This project has been supported by the Hungarian Scientific Research Fund (OTKA, No. 124727: Modeling general topology free-form surfaces in 3D). Some images of this paper were generated using the Sketches system developed by ShapEx Ltd., Budapest; the contribution of György Karikó is highly appreciated. The authors acknowledge thought provoking discussions with Márton Vaitkus.

References

1. C. T. Loop and T. D. DeRose. A multisided generalization of Bézier surfaces. *ACM Transactions on Graphics*, 8(3):204–234, 1989.
2. T. Várady, P. Salvi, and Gy. Karikó. A multi-sided Bézier patch with a simple control structure. *Computer Graphics Forum*, 35(2):307–317, 2016.
3. P. Charrot and J. A. Gregory. A pentagonal surface patch for computer aided geometric design. *Computer Aided Geometric Design*, 1(1):87–94, 1984.
4. T. Várady, A. Rockwood, and P. Salvi. Transfinite sur-

face interpolation over irregular n-sided domains. *Computer Aided Design*, 43(11):1330–1340, 2011.

5. P. Salvi, I. Kovács, and T. Várady. Computationally efficient transfinite patches with fullness control. In *Proceedings of the Workshop on the Advances of Information Technology*, pages 96–100, 2017.

6. C. L. Bajaj and I. Ihm. Algebraic surface design with Hermite interpolation. *ACM Transactions on Graphics (TOG)*, 11(1):61–91, 1992.

7. C. L. Bajaj, J. Chen, and G. Xu. Modeling with cubic A-patches. *ACM Transactions on Graphics (TOG)*, 14(2):103–133, 1995.

8. J. Li, J. Hoschek, and E. Hartmann. G^{n-1}-functional splines for interpolation and approximation of curves, surfaces and solids. *Computer Aided Geometric Design*, 7(1-4):209–220, 1990.

9. E. Hartmann. Implicit G^n-blending of vertices. *Computer Aided Geometric Design*, 18(3):267–285, 2001.

10. T. Várady, P. Benkő, G. Kós, and A. Rockwood. Implicit surfaces revisited – I-patches. In *Geometric Modelling*, pages 323–335. Springer, 2001.

11. Á. Sipos, T. Várady, P. Salvi, and M. Vaitkus. Multi-sided implicit surfacing with I-patches. *Computers & Graphics*, 90:29–42, 2020.

12. K. Kato. Generation of n-sided surface patches with holes. *Computer-Aided Design*, 23(10):676–683, 1991.

13. T. Várady. Overlap patches: a new scheme for interpolating curve networks with n-sided regions. *Computer Aided Geometric Design*, 8(1):7–27, 1991.

14. P. Salvi, T. Várady, and A. Rockwood. Ribbon-based transfinite surfaces. *Computer Aided Geometric Design*, 31(9):613–630, 2014.

15. P. Salvi and T. Várady. Multi-sided surfaces with fullness control. In *Proceedings of the Eighth Hungarian Conference on Computer Graphics and Geometry*, pages 61–69, 2016.

16. R. Krasauskas. Toric surface patches. *Advances in Computational Mathematics*, 17(1):89–113, 2002.

17. T. Várady, P. Salvi, M. Vaitkus, and Á. Sipos. Multi-sided Bézier surfaces over curved, multi-connected domains. *Computer Aided Geometric Design*, 78:101828, 2020.

18. M. Vaitkus. Notes on curve network interpolation. In *Proceedings of the Workshop on the Advances of Information Technology*, pages 77–86, 2019.

19. R. A. Liming. Conic lofting of streamline bodies. *Aircraft Engineering and Aerospace Technology*, 19(7):222–228, 1947.

20. G. Xu, H. Huang, and C. L. Bajaj. C^1 modeling with A-patches from rational trivariate functions. *Computer Aided Geometric Design*, 18(3):221–243, 2001.

21. E. Hartmann. Blending of implicit surfaces with functional splines. *Computer-Aided Design*, 22(8):500–506, 1990.

22. E. Hartmann and Y. Y. Feng. On the convexity of functional splines. *Computer Aided Geometric Design*, 10(2):127–142, 1993.

23. T. Várady and A. Rockwood. Geometric construction for setback vertex blending. *Computer-Aided Design*, 29(6):413–425, 1997.