Sixth Hungarian Conference on Computer Graphics and Geometry, Budapest, 2012

3D shape design using curve networks with ribbons

Tamás Várady[1], Péter Salvi[1], Alyn Rockwwod[2]

[1] Budapest University of Technology and Economics
[2] King Abdullah University of Science and Technology

Abstract

Curve network-based design is a challenging approach to supplement traditional trimmed surface modeling or recursive subdivision for defining complex free-form shapes. The input is a collection of feature curves, that can be directly created or extracted from manual sketches or images. The curves are built together in 3D and the network is interpolated by smoothly connected multi-sided patches. Each patch may have arbitrary number of free boundary curves; thus many difficulties of the traditional approaches can be avoided and there is no need to manipulate control grids or polyhedra. This paper focuses on multi-sided transfinite surface interpolation, where in addition to interpolating 3D curves, further design freedom is provided to control the interior of the patches. This is achieved by supplementing auxiliary vertices, curves and even control surfaces while the boundary constraints are retained. Different distance-based blending functions are discussed over non-regular, n-sided domains. The concept naturally extends to creating one- and two-sided patches as well. Curve network-based design will be demonstrated through several simple examples.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

1. Introduction

Creating complex free-form objects, composed of smoothly connected surface patches is a fundamental goal in Computer Aided Geometric Design. Aesthetic appearance is crucial for a wide variety of objects including cars, household appliances, office furniture, containers and many others. While the majority of such patches are four-sided, almost all industrial objects contain general n-sided patches that need to be inserted into some arrangement of quadrilaterals. A few examples are shown in Figure 1.

Nevertheless, general topology surfacing is a tough problem, and current techniques expose deficiencies for designers and CAD users. The standard approach is to combine a collection of biparametric (generally NURBS) surfaces, then create trimmed patches through a sequence of surface intersections, finally stitch these into a single model. The fundamental problem is that the original boundaries of the four-sided patch and the trimming curves have different representational form and design flexibility. For example, creating truly symmetric three-sided patches is not possible in a four-sided domain. Another simple example is the surface model in Figure 2, which contains three 3-sided and two 6-sided patches. It is not obvious at all, how the initial quadrilateral surfaces should be defined, how they can be stitched together (presumably, only with numerical continuity) and how the smoothly connected internal trimming curves can be modified, if some redesign is needed.

Another general topology surfacing approach, widely used in the animation industry, is based on recursively subdividing a control polyhedra[3]. This yields a set of smoothly connected quadrilaterals combined with n-sided surface patches, however, difficulties here include the "ab initio" creation of good control polyhedra. Take again the previous simple surface model: it would be hard to set up an appropriate control polyhedron, and practically impossible to directly interpolate and edit prescribed free-form curves with sharp edges, or smooth edges with tangential constraints.

In this paper we explore a third approach using 3D curve networks, where an arbitrary collection of feature curves is created, each can be manipulated directly and independently of the other ones. The network is automatically interpolated by multi-sided patches; their connection is watertight, and any boundary adjustment naturally invokes the modification of the surface model. In this way, users can focus on

Figure 1: Examples of nice free-form objects

shape concepts and aesthetic requirements, and do not need to fiddle with representational difficulties. We illustrate the paradigm using the same simple object shown in Figure 2. The first picture shows a manual sketch (a), which is the basis of creating a corresponding 3D network (b). The next stage is the automatic computation of ribbons (c), which determine the boundary constraints for multi-sided surface patches (d). In fact, this is the clue of the whole concept, as these patches must provide a natural blend between the boundaries and their interior must be predictable and controllable, when needed. The last two pictures show, that it is quite straightforward to modify the topology *and* geometry of the curve network, and the surface model will change accordingly (e,f).

The input of curve network-based design can be a set of manual sketches, or a set of images, when the user tries to reconstruct an existing object. Such a sequence is shown in Figure 3, where after aligning two photographs (a), they are mapped onto the side faces of a sketch box for 2D curve tracing, by means of which a 3D network is created that eventually defines the final model. Note, that the surface design may proceed without a 2D input — its starting point can also be an a priori defined curve network template, or the user can even design from scratch.

In this paper new techniques are proposed that extend the capabilities of conventional transfinite interpolation. We keep the transfinite nature of the surface patches, i.e. retain the boundary constraints, but provide additional degrees of freedom to perfect the interior of the surface patches, as well. Transfinite surface interpolation is a classical area of CAGD. Its origin goes back to the late 60's, when Coons formulated his Boolean sum surface[2]. In the next two decades, several papers were published, first on triangular patches (see summary in Farin[3]), and later on genuine n-sided patches, including the pioneering work of Charrot and Gregory[1, 4], Sabin[6], and Kato[5].

The alternatives of creating n-sided transfinite patches with different blending functions and parameterizations have been recently published by the current authors[8, 9]. There are also new emerging representational forms that truly generalize Coons' former approach, see also a companion paper[7] in this proceedings. Concerning interior shape control, it must be noted that the majority of papers in the literature deals with bi-parametric surfaces and shape deformation in approximating sense unlike our approach where the emphasis has been placed on multi-sided patches and interpolating auxiliary vertices and curves in the patch interior.

The outline of this paper is the following. In Section 2 we briefly revisit transfinite surface interpolation using tangential ribbons; as this will be the basis of the forthcoming discussions. In Section 3 we introduce the notion of auxiliary vertices and curves to adjust the shape interior, while in Section 4 the application of so-called interior surfaces will be discussed. Section 5 is devoted to describe one- and two-sided patches. Finally, examples and suggestions for future work conclude the paper.

2. Transfinite patches with ribbons

Multi-sided patches can be created by combining ribbon surfaces (see Fig. 2c) associated with the individual boundaries using special blending functions. For our forthcoming discussion, we present one particular scheme; however, these ideas can be expanded for other types of ribbons and parameterization schemes, as it is summarized, for example, in Várady et al.[8]

An *n-sided patch* is generally defined over a non-regular, convex polygonal domain as a convex combination of ribbon surfaces:

$$S(u,v) = \sum_{i=1}^{n} R_i(s_i,d_i)\mu_i(d_1,\ldots,d_n).$$

The polygonal domain is defined in the (u,v) plane, and the individual patch boundaries are the 3D images of the polygon sides.

A *ribbon surface* can be any bi-parametric surface with s_i and d_i being its local parameters. The simplest is using linear ribbons (i.e. linear by the d_i parametric direction), given as

$$R_i(s_i,d_i) = P_i(s_i) + d_iT_i(s_i),$$

where $P_i(s_i)$ is the boundary curve along the i-th side, and $T_i(s_i)$ is the cross-tangent function associated with the boundary. The local parameters of a ribbon depend on (u,v): $s_i = s_i(u,v)$ is the *side parameter*, and $d_i = d_i(u,v)$ is the *distance parameter*. d_i represents some distance measure, i.e., vaguely speaking, it is 0 on the i-th side and increases in a

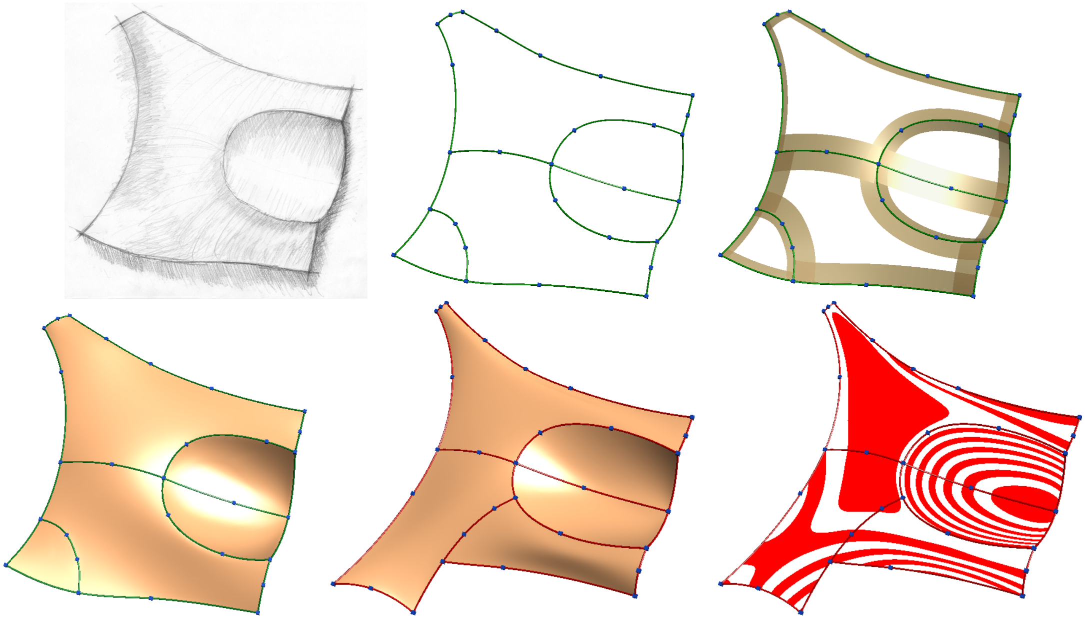


Figure 2: Test surface model - (a) sketches, (b) curve network, (c) ribbons, (d) multi-sided surfaces, (e) modified model, (f) slicing

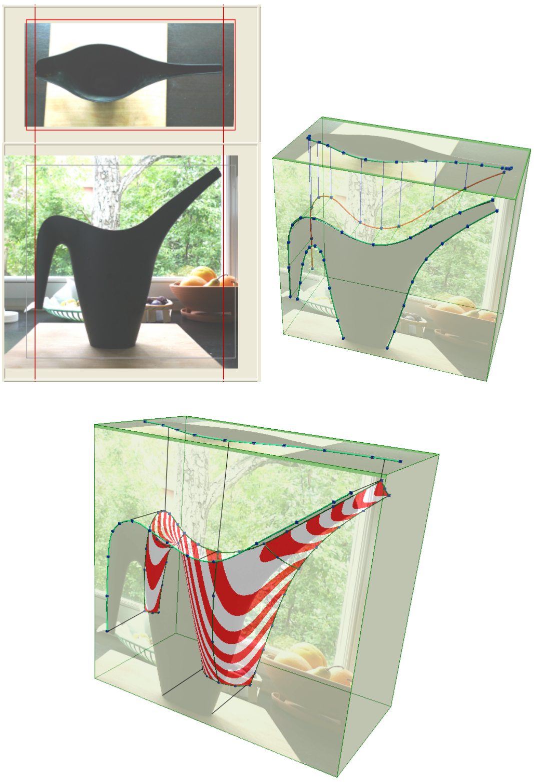


Figure 3: Shape definition from images

monotonic way as we move inwards. The above functions of s_i and d_i represent the so-called *ribbon mapping*, i.e. how the four-sided image of the ribbons are mapped onto the polygonal domain (for details see also Várady et al.[8]).

Blending functions. The ribbons are weighted by special blending functions $\mu_i(d_1 \dots, d_n)$ defined over the full domain. We present here rational polynomial functions. Let $D^n_{i1,i2,\dots,in}$ denote $\prod^n_{i \neq i1,i2,\dots,in} d_i^2$. Then

$$\mu_i(d_1,\dots,d_n) = \frac{D^n_i}{\sum^n_{j=1} D^n_j}.$$

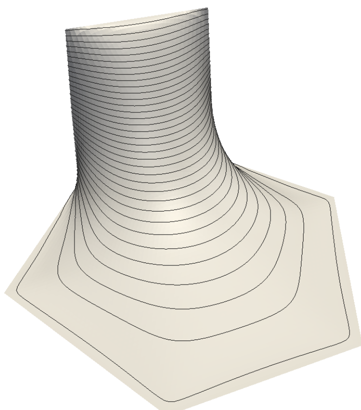
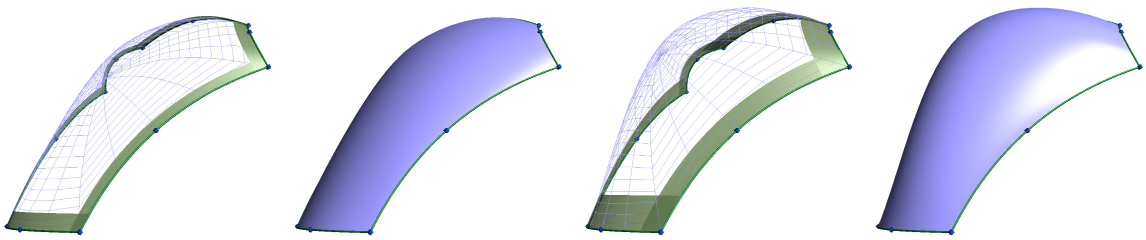


Figure 4: Edge blending function with contours



(a) Weight=0.25, spider lines (b) Weight=0.25, shaded (c) Weight=0.5, spider lines (d) Weight=0.5, shaded

Figure 5: Adjusting fullness for a six-sided patch.

μ_i is equal to 1 along side i, and 0 for all the remaining $n-1$ sides where $k \neq i$. For all domain points the μ_i-s have the partition of unity property. Such an edge-blending function is shown in Figure 4. These type of blending functions are singular at the corner points. For example, there is a jump between $\mu_1(0, d_2, \ldots, d_{n-1}, \varepsilon) = 1$ and $\mu_1(\varepsilon, d_2, \ldots, d_{n-1}, 0) = 0$. This singularity vanishes when two adjacent blending functions are added at a given corner:

$$\lim_{\substack{d_{i-1}\to 0,\\ d_i\to 0}} \mu_{i-1}(d_1, d_2, \ldots, d_n) + \mu_i(d_1, d_2, \ldots, d_n) = 1.$$

These blending functions ensure that the ribbons will be reproduced along the sides. Let us evaluate an arbitrary point of the i-th boundary as a function of d_i, i.e., $S_i(d_i) = R_i(d_i)\alpha_i(d_i)$. In order to interpolate the positional data, $\alpha_i(0)$ must be equal to 1. The derivative at the boundary is $\frac{\partial S_i}{\partial d_i} = R_i'(d_i)\alpha_i(d_i) + R_i(d_i)\alpha_i'(d_i)$, so in order to interpolate the tangential data, $\alpha_i'(0)$ must be zero. Thus for G^1-continuous cross-derivative constraints, it is sufficient to use quadratic terms; for G^2 constraints cubic terms are needed. It can easily be shown, that this also guarantees that the effect of the $k \neq i$ boundaries and their cross-derivative functions will vanish on the i-th side.

Take an example, for $n = 4,\ i = 1$,

$$\mu_1(d_1, d_2, d_3, d_4) = \frac{d_2^2 d_3^2 d_4^2}{d_1^2 d_2^2 d_3^2 + d_2^2 d_3^2 d_4^2 + d_3^2 d_4^2 d_1^2 + d_4^2 d_1^2 d_2^2},$$

i.e., if $d_1 = 0$, then $\mu_1 = 1$; if $d_2 = 0$ or $d_3 = 0$ or $d_4 = 0$, then $\mu_1 = 0$.

Note: an equivalent, but computationally more efficient formula can be used to evaluate the blending functions at the interior points of the domain:

$$\mu_i(d_1, \ldots, d_n) = \frac{d_i^{-2}}{\sum_{j=1}^{n} d_j^{-2}}.$$

This simpler equation is singular on the sides, so there the original formula must be used.

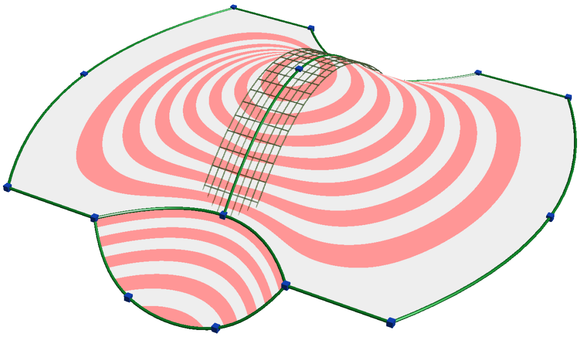
3. Simple shape modifications

Adjusting ribbons. In curve network-based design feature curves are the basic entities to define a shape, but the interior of the patches are not uniquely defined, and ribbons can provide further shape control. Assume that the boundaries and the cross-tangent directions are given. The most straightforward editing operation is to set the magnitudes of the ribbons, as these balance how much the surface patch is "glued" to the ribbons in the vicinity of the boundaries, and where convex combination starts to dominate as we are moving inwards. The simplest solution is to multiply the direction terms by w_i scalar values or scalar reparameterization functions, then

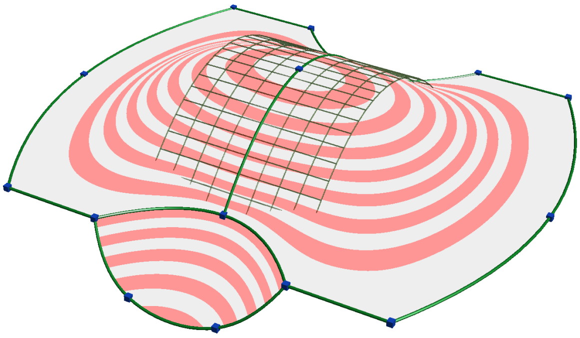
$$R_i(s_i, d_i) = P_i(s_i) + d_i w_i(s_i) T_i(s_i).$$

Adjusting simultaneously the magnitude of the ribbons yields a global change concerning the "fullness" of the patch, as it is shown in Figures 5a to 5d. Modifying the width of an individual ribbon creates a local effect as shown in Figure 6. By enlarging the magnitude of the top ribbons,n a different fullness is obtained — this is also illustrated by how the slices change their shape. Note, that by adjusting the widths of the ribbons, it is possible to optimize fairness energies for a collection of adjacent patches, as well.

Auxiliary vertices and curves. We wish to preserve the basic interpolation nature of curve network-based design, nevertheless it may be desirable to assign further entities to the interior of the patch to provide more shape control. Think of lifting certain interior vertices or prescribing interior feature curves, while the external ribbons are retained. Recall that the only property of the blending functions was that the i-th blend is 1 on the i-th boundary and zero elsewhere. By definition, an auxiliary element has an image within the domain, and a distance measure can be defined, which guarantees that it becomes zero on the image of the auxiliary element. Then the corresponding blending function will be 1 there and will vanish elsewhere. Such a vertex blending function is shown in Figure 7. This means that the patch equation needs to be modified, and $n+k$ entities will be blended together (k denotes the number of the auxiliary elements). The blending



(a) Initial ribbons



(b) Modified ribbons

Figure 6: Ribbon modification of six-sided patches.

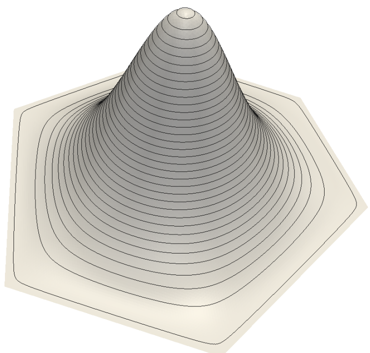
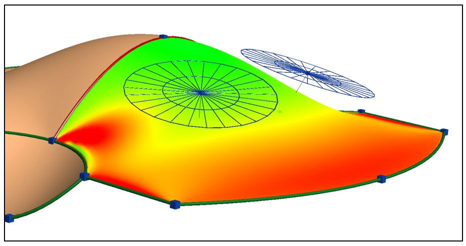
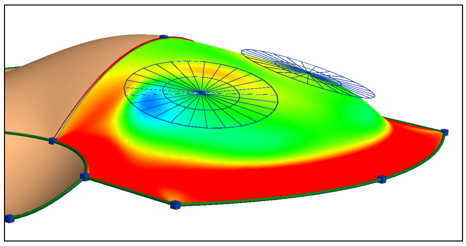


Figure 7: Vertex blending function with contours



(a) Two auxiliary vertices and circular ribbons



(b) After snapping

Figure 8: Six-sided patch modified by auxiliary vertices.

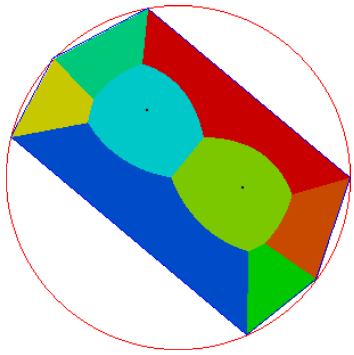


Figure 9: Blending function distributions in the domain

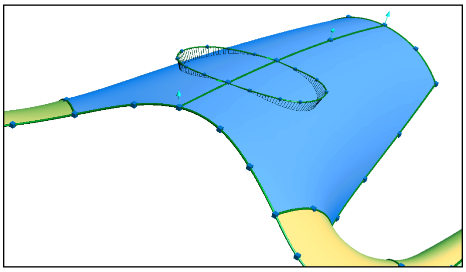
functions μ_i will also change and instead of the D_i^n terms, now D_i^{n+k} will be used combining $n+k$ distance values.

This concept is illustrated by two simple examples. In the first, two auxiliary vertices have been chosen on the surface, which define their parametric positions. Lifting the vertices and creating their circular ribbons determine the local properties of the modified surface after snapping; see Figures 8a and 8b. The distribution of the blending functions in the domain are illustrated in Figure 9, where the "strength" of the blends is shown. The areas show the dominance of the i-th blend, the boundaries between the areas show where adjacent blends have the same effect, thus providing a Voronoi-like structure in the domain.

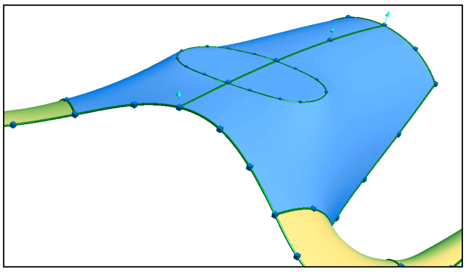
The second example shows two patches and two auxiliary curves. Initially, the curves are also defined on the surface, in order to obtain their parametric image in the domain. Lifting the curves leads to new features to be interpolated (Fig. 10a). As we snap the surfaces onto them the interiors change (Fig. 10b); keep in mind that these patches are still single surface entities.

4. Interior patches

In the previous section we have introduced auxiliary vertices and curves. We have increased the number of interpolants, but used the same family of blending functions with $n+k$ terms. Now we are going to introduce a so-called *interior*



(a) Auxiliary curve and its image on the surface



(b) After snapping

Figure 10: Two fives-sided patches modified by auxiliary curves.

surface $S_{\mathrm{int}}(u,v)$, which is defined over the same domain and serves to modify the interior of the original $S(u,v)$. Here we are going to apply alternative blending functions. Let

$$S^*(u,v) = \sum_{i=1}^{n} R_i(s_i,d_i)\nu_i(d_1,\ldots,d_n) + S_{\mathrm{int}}(u,v)\nu_0(d_1,\ldots,d_n).$$

We use the notations of Section 2 with the additional term of $D_0^n = \prod_{j=1,n} d_j^2$. Then the blending functions are defined as

$$\nu_i(d_1,\ldots,d_n) = \frac{D_i^n}{\sum_j D_j^n + wD_0^n}, \quad i = 1,\ldots,n,$$

and

$$\nu_0(d_1,\ldots,d_n) = \frac{wD_0^n}{\sum_j D_j^n + wD_0^n}.$$

Here w is a positive constant characterizing the blend family, this will be set later. As it can be seen, the side blends have an extended denominator, which will not change the basic properties, i.e., $\nu_i = 1$ on the i-th side and 0 on the other sides. The new blending function ν_0 is 0 on each side, which means the interior surface will have no effect on the boundaries.

The above surface equation can be formulated in another way:

$$S^*(u,v) = \alpha S(u,v) + (1-\alpha)S_{\mathrm{int}}(u,v) \qquad (1)$$

where

$$\alpha = \frac{\sum_j D_j^n}{\sum_j D_j^n + wD_0^n}.$$

This expression reproduces S along the edges and gives a weighted average of the original and the interior surface inside the patch. We define the constant w by means of α. Take a domain point c as center point; this is where the weighted average is prescribed. At point c let us evaluate all distances, thus we obtain constant terms

$$E_j^c = D_j^n(d_1,d_2,...,d_n),\ j = 1,\ldots,n.$$

Then $\alpha = \frac{\sum_j E_j^c}{\sum_j E_j^c + wE_0^c}$ and $w = \frac{(1-\alpha)\sum_j E_j^c}{wE_0^c}$. After dividing by E_0^c we obtain an alternative expression of $w = \sum_j \frac{1-\alpha}{d_j^2}$, where the non-zero distances d_j are determined by c.

$\alpha = 0.5$ will average the original and the interior surface at the center point. If we want to interpolate the interior surface at c and tightly approximate it in the vicinity of c, another surface, called auxiliary surface $S_{\mathrm{aux}}(u,v)$ needs to be used in Equation 1 above. Let us assume that

$$S_{\mathrm{int}}(u,v) = \alpha S(u,v) + (1-\alpha)S_{\mathrm{aux}}(u,v),$$

then

$$S_{\mathrm{aux}}(u,v) = \frac{S_{\mathrm{int}}(u,v) - \alpha S(u,v)}{1-\alpha}.$$

For example, at $\alpha = 0.5$

$$S_{\mathrm{aux}}(u,v) = 2 * S_{\mathrm{int}}(u,v) - S(u,v).$$

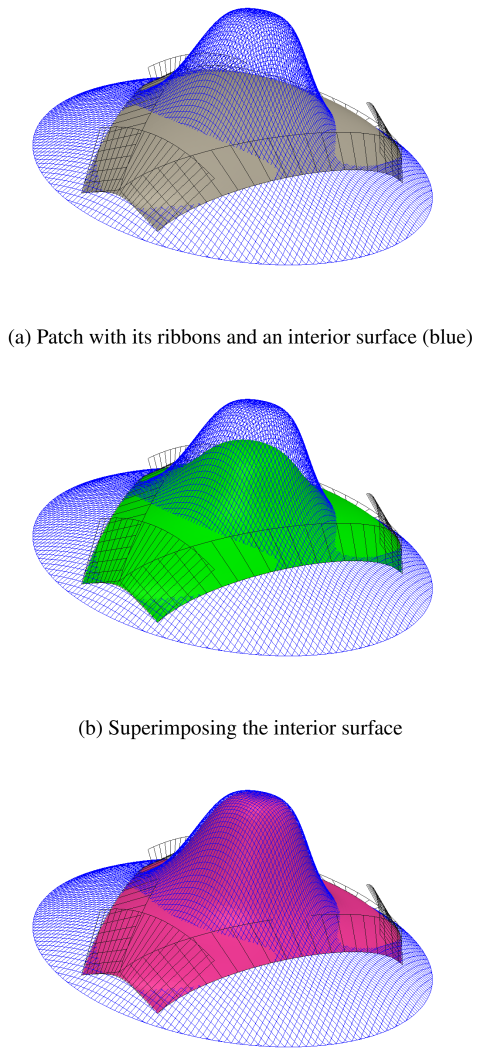
(Note, that instead of linear blending by α, Hermite functions can also be used.)

The parametric assignment of S and S_{int} can be realized in a projective sense, as earlier. Imagine that we have already computed the surface S; then any point on the interior surface with parameters (u^*,v^*) can be projected back to S, which will yield a parameter pair (u,v) to create a parametric assignment to combine the points of the two surface entities. The effect of using interior surfaces is demonstrated in Figure 11. The first one shows the input: a patch to be modified and the interior surface used for shape adjustment. The second picture shows the averaging effect, while the third one illustrates how we can reproduce the interior surface using a corresponding auxiliary surface. In both cases the original boundary constraints are retained.

5. One- and two-sided patches

In practical curve network-based design, one- and two-sided patches often occur. These also must interpolate the boundaries and match the ribbon surfaces determined by the network. Fortunately, sweeping line parameterization and distance-based blending can be applied in a similar way as before.

One-sided patches. Take a closed curve $r(t)$ and a center

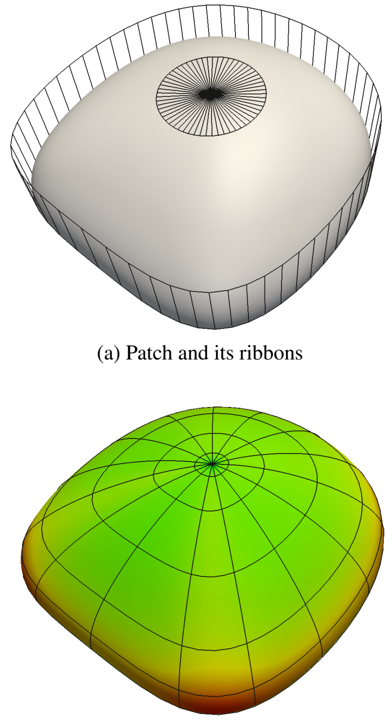


(a) Patch with its ribbons and an interior surface (blue)

(b) Superimposing the interior surface

(c) Superimposing the auxiliary surface

Figure 11: Reshaping the interior of a six-sided patch.



(a) Patch and its ribbons

(b) Mean curvature and spider lines

Figure 12: One-sided patch example "cap".

point c in 3D and associate an additional ribbon with it. Let us use a circle as domain with radial sweep-lines. We apply the same solution as for auxiliary points in Section 3 and combine the two ribbons by a simple blending function of the type

$$\mu_i(d_1,d_2) = \frac{d_j^2}{d_1^2+d_2^2},\ i,j\in\{1,2\}, i\neq j, \qquad (2)$$

where d_1 and d_2 represent the distances in the domain from the perimeter circle and from the center point, respectively. As an example, Figure 12a shows the defining ribbons and the corresponding cap-like surface patch. Figure 12b shows the curvature map of the surface together with spider-like constant parameter lines drawn on the surface in 3D.

One interesting issue is to find a good location for the center. In the majority of cases this will be set by the user, however, setting a good default may be necessary. One simple heuristic is to optimize the angles between the imaginary 3D sweeping lines and the tangents at sampled data points on the boundary, i.e.,

$$\sum_i (\langle c - r(t_i), \dot{r}(t)\rangle)^2 + \alpha|c - r(t_i)|^2 = \min .$$

The second term is needed to control the sum of the chord lengths between the center point and the points of the boundary; without this, the minimum found by the related system of equations would push the center point infinitely far from the closed boundary curve. α is a constant that can also be manually adjusted.

Two-sided patches. The domain of the two-sided patch is bounded by two parabolic arcs. As it was explained in Section 2, we search for a domain that mimics simultaneously the 3D angles between the two given boundary curves and their arc lengths. Figure 13 shows a simple, heuristic solution using a quadrangle. We inherit the 3D angles denoted by α and β, and define the parabolas in such a way, that their approximate arc lengths are proportional to the 3D boundaries.

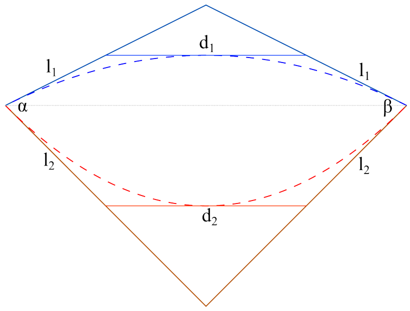


Figure 13: Computing a two-sided domain

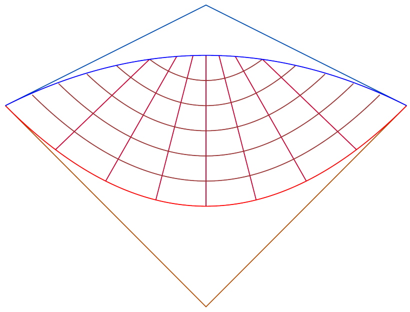


Figure 14: Sweeping lines in a parametric domain

The arc length of a parabolic arc is estimated by $2l_i + d_i$, as shown in the figure, and simple algebra leads to determine the missing parameters. Having the domain, sweeping lines are created by connecting the parabolas with the opposite corners of the quadrangle (see one set in Figure 14). Then normalized distances measured on the sweeping lines yield the distance parameters, which are used for the same type of blending functions as above in Equation 2 to combine the two ribbons.

A simple two-sided example defined by two ribbons has been shown earlier in Figures 6a and 6b.

6. Conclusion, future work

In this work we have focused on various aspects of curve network-based design, in particular, how to modify the interior of multi-sided transfinite surface patches. In addition to adjusting the widths of ribbons, additional entities — such as auxiliary vertices, curves and interior surfaces — were combined, applying variations of distance-based blending functions. Challenging future research topics include fairing operations for curve network-based models and point data approximation by transfinite patches, using free parameters of the ribbon interpolants.

Acknowledgements

This work was partially supported by the scientific program "Development of quality-oriented and harmonized R+D+I strategy and functional model at the Budapest University of Technology and Economics" (UMFT-TÁMOP-4.2.1/B-09/1/KMR-2010-0002) and a grant by the Hungarian Scientific Research Fund (No. 101845). The pictures in this paper were generated by the Sketches system developed by ShapEx Ltd, Budapest; the contribution of György Karikó is highly appreciated. The authors also acknowledge support from the Geometric Modeling and Scientific Visualization Research Center of KAUST, Saudi-Arabia.

References

1. P. Charrot, J. A. Gregory, A pentagonal surface patch for computer aided geometric design, Computer Aided Geometric Design 1 (1) (1984) 87–94.
2. S. A. Coons, Surfaces for computer-aided design of space forms, Tech. rep., Massachusetts Institute of Technology, Cambridge, MA, USA (1967).
3. G. Farin, Curves and surfaces for CAGD: a practical guide, 5th ed., Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.
4. J. A. Gregory, N-sided surface patches, in: The Mathematics of Surfaces, Oxford University Press, USA, 1986, pp. 217–232.
5. K. Kato, Generation of n-sided surface patches with holes, Computer-Aided Design 23 (10) (1991) 676–683.
6. M. Sabin, Transfinite surface interpolation, in: Proceedings of the 6th IMA Conference on the Mathematics of Surfaces, Clarendon Press, New York, NY, USA, 1996, pp. 517–534.
7. P. Salvi, T. Várady, A. Rockwood, New schemes for multi-sided transfinite surface interpolation, in: this proceedings.
8. T. Várady, A. Rockwood, P. Salvi, Transfinite surface interpolation over irregular n-sided domains, Computer Aided Design 43 (2011) 1330–1340.
9. T. Várady, P. Salvi, A. Rockwood, Transfinite surface interpolation with interior control, Graphical Models (under revision).