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

Polyhedral design with concave and multi-connected faces

Jázmin Szörfi[1,2], Tamás Várady[1]

[1] Budapest University of Technology and Economics
[2] Shapr3D Ltd, Budapest

Abstract
Polyhedral design is an efficient technology to create complex free-form shapes by smoothing control polyhedra. Several well-known approaches exist, including recursive subdivision and direct algorithms. In this paper, we introduce a direct method where a collection of smoothly connected, multi-sided Bézier patches are stitched together. While traditional methods are based on control polyhedra with only convex faces, our approach is capable to handle faces with concave angles and multiple holes, using a special, topology preserving patchwork structure. The main steps of the algorithms: (i) compute an auxiliary polyhedron, (ii) define a general topology curve network and derive interior control points for generalized Bézier patches, (iii) compute the surfaces and (iv) adjust shape parameters, if needed. Several test examples will be given to demonstrate the capabilities of this new method.

Keywords: polyhedral design, curve networks, generalized Bézier patches, curved domains

1. Introduction

Polyhedral design is an important 3D technology that is widely applied in various areas of engineering, medicine, animation, etc. for creating complex free-form shapes. The basic idea is that nice, intuitive shapes can be obtained by smoothing a control polyhedron. When its control points are relocated, the model will also change in a natural and predictable manner.

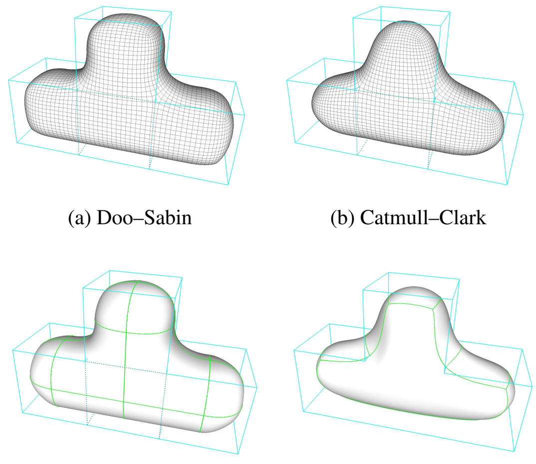
Approaches can be distinguished by the algorithms how the polyhedron is transformed into a collection of smoothly connected free-form surfaces (Figure 1). While *recursive subdivision* methods produce a sequence of refined polyhedra that converges to a limit surface, *direct methods* create patchworks that interpolate curve networks composed explicitly from the polyhedron. The majority of direct methods apply quadrilaterals or convex multi-sided surfaces, these being the most deeply studied, standard representations in CAGD. As it will be discussed later, providing models with convex faces can only be achieved by often inconvenient and artificial subdividing operations.

A recently published multi-sided Bézier surface representation over curved domains (called the CD-GB patch) is capable of handling faces with concave angles and interior holes (see Várady et al.[17]). This makes it possible to get rid of convex subdivision and build a patchwork that preserves the topological structure of the control polyhedron. In our paper we describe such an approach, which – according to our best knowledge – is the first of its kind. Currently, it ensures G^1 continuity only. After briefly presenting prior work (Section 2), we analyze why traditional methods may fail without convex division (Section 3). Then we describe the basic steps of the algorithm in Section 4, followed by the creation of *auxiliary polyhedra* (Section 5), curve networks and interior control points for the patches (Section 6). The equation of the generalized Bézier patches will be described in Section 7. Finally, by means of a few examples we demonstrate the capabilities of our method (Section 8). Suggestions for future work conclude the paper (Section 9).

(Note: this paper is a short version of a publication, being in preparation.)

2. Previous work

Polyhedral design is an essential chapter in Computer Aided Geometric Design[4] to model general topology surfaces that may have arbitrary n-sided faces, and vertices with arbitrary valency. The literature is dominated by *recursive subdivision* methods; the most influential and popular approaches were suggested by Doo–Sabin[3] and Catmull–Clark.[1] These produce a sequence of refined polyhedra that nicely converges to a smooth limit surface. It is well-known that these surfaces



(a) Doo–Sabin (b) Catmull–Clark

(c) Dual topology (d) Topology preserving

Figure 1: Different polyhedral design methods

define piecewise bi-quadratic or bi-cubic B-spline patches over regular subsets of the control points, respectively. At the same time, it is a hard problem to obtain an explicit surface representations at the extraordinary points, see a recent comprehensive survey by Peters.[12] Specific solutions include that of Loop and Schaefer[9] using S-patches, Hettinga and Kosinka[5] using generalized B-spline patches, and an extensive series of guiding surface based schemes, see e.g. recent publications by Karciauskas and Peters.[7,6]

Direct polyhedral methods – often called surface splines – emerged in the early 90s by various authors (see J. Peters' works[11]). The main problems to be solved included the split of n-sided patches into standard *quadrilateral patches* and ensuring G^n continuity with various constraints. This problem is still an area of active research.

An alternative concept is to replace a collection of quadrilateral patches by genuine *multi-sided patches*, see e.g. Malraison's survey.[10] In this case we do not need central splitting, and there are no internal discontinuities, but another deficiency emerges: these multi-sided patches cannot be generally represented in standard data formats. One family of multi-sided polyhedral design was proposed by Loop and DeRose[8] using S-patches. Rockwood applied transfinite patches.[13] The use of convex, multi-sided Bézier patches were suggested by Szörfi.[15] We remark that implicit multi-sided patches can also be used for polyhedral design, see a recent paper by Sipos et al.[14]

The majority of direct constructions heavily exploits the convex nature of the constituting surfaces. These are typically built over a *dual topological graph*, where the patch boundaries are constructed by connecting the centroids of adjacent faces. Unfortunately, this often forces an uncomfortable preprocessing related to the initial polyhedron, i.e., the faces with concave angles and internal hole loops must be *subdivided* in order to produce sensible convex faces.

The novelty of our *topology preserving approach* eliminates this inconvenience. The use of generalized Bézier patches[17] makes it possible to create a *one-to-one* correspondence between the topology of the patchwork and faces of the control polyhedron. In the next section we will point out its benefits when compared with traditional methods.

3. Problems of convex division

As discussed earlier, for the majority of traditional methods, including recursive subdivision and dual topology schemes, it is compulsory to insert artificial subdividing edges into the control polyhedron. While in many cases this is straightforward, in other cases difficulties may emerge, as follows.

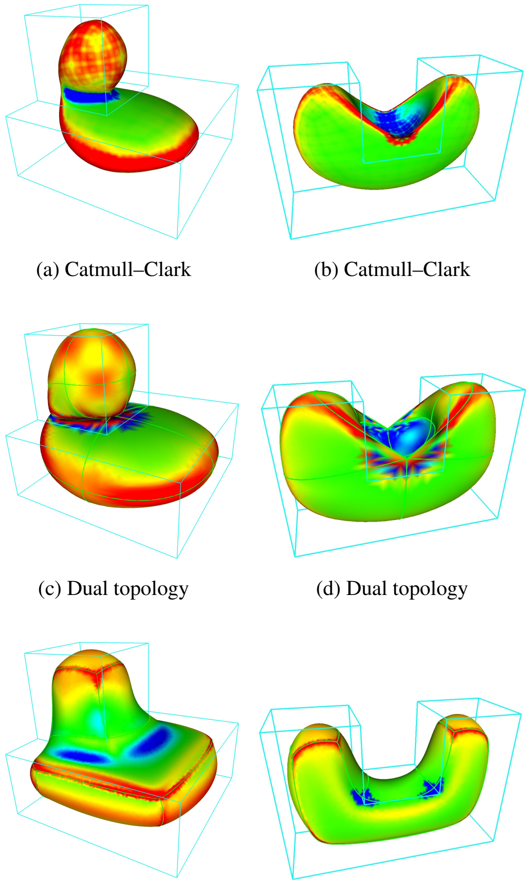
The Doo–Sabin method and the dual topology algorithms interpolate the *centroid* of the faces. The first step of the Catmull–Clark scheme also starts from the centroids of the faces. For concave or multi-connected faces the centroid may fall outside the face, which may cause artefacts. In Figure 2 simple examples (a, b, c, d) are shown to illustrate what would happen without inserting extra edges. Fortunately, the position of the centroid is irrelevant for the topology preserving method (e, f).

It can be a challenging task to compute a natural convex split for a complex face with holes; it is even more complicated to produce a fully quadrangular subdivision. Imagine what would be the best subdivision for our test object in Figure 16. Splitting complex faces may lead to splitting some edges (T-nodes), which may force further superfluous subdivisions on the adjacent faces. In Figure 3 we subdivided a concave T-shape face (black lines) and thus the bottom face (red lines) needed to be split into three subfaces. The example in Figure 4 shows that even relatively simple faces can be subdivided in various ways: (b) shows a set of convex faces and (c) quads only; in these cases T-nodes are generated and the side faces will be split, as well. Finally, (d) shows a subdivision without T-nodes, but several artificial subfaces were created. The topology preserving approach (a) avoids splitting and the insertion of T-nodes, and it defines a simple, unambiguous structure.

4. The algorithm and the flowchart of the process

The basic steps of the algorithm are depicted in Figure 5. We start from a general topology control polyhedron that may have faces with concave angles and interior hole loops.

1. First, we create an *auxiliary polyhedron*, which will be constructed by *shrinking* the original faces (FACE-faces), and *chamfering* the edges and the vertices, yielding (EDGE-faces) and (VERTEX-faces), respectively. (Hereinafter, we prefer to use the term *facet* instead of



(a) Catmull–Clark (b) Catmull–Clark

(c) Dual topology (d) Dual topology

(e) Topology preserving (f) Topology preserving

Figure 2: Failures (a, b, c, d) are due to lack of subdivisions

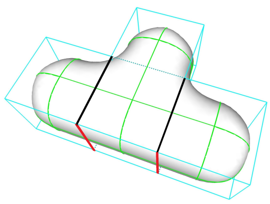
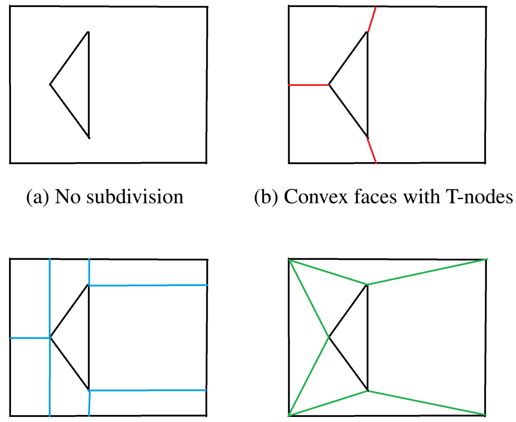


Figure 3: Extra edges inserted



(a) No subdivision (b) Convex faces with T-nodes

(c) Quad faces with T-nodes (d) Convex faces, no T-nodes

Figure 4: Subdivision for a face with a hole

the VERTEX-face.) This operation produces a polyhedron, somewhat similar to the Doo–Sabin first subdivision step. There are various parameters to adjust its shape, e.g. adjusting the depth of the edge-chamfer will indirectly tweak the depth of the vertex chamfer, and accordingly the curvatures of the final shape.

2. Next we construct a free-form curve network, where the individual curves are cubic Bézier curves. Each curve smoothly connects to the centroids of two adjacent facets. These edges correspond to the original edges of the control polyhedron, and constitute the boundary loops of the multi-sided patches, so we do reproduce the topology. There is another shape parameter to set the fullness of these curves, while restraining the tangential constraints at the ends.
3. For generating CD-GB patches, we need to determine further interior control points that will define the cross-derivatives of the patches for each boundary. These are the twist control points, placed initially into the vertices of the facets. We generally reposition these twist vectors in order to ensure G^1 continuity between the opposite boundary ribbons using the well-known direction blend method[2].
4. Now we are ready to compute the CD-GB patches; first the curved domain, and then a parameterization associated with each boundary. Once this is completed we can evaluate any point in the domain and map it to 3D, as defined by the parametric equation of the CD-GB patch. The individual patches are stitched together to form a global patchwork that smoothly mimics the shape of the given control polyhedron.

In the next sections we provide further details about the algorithm.

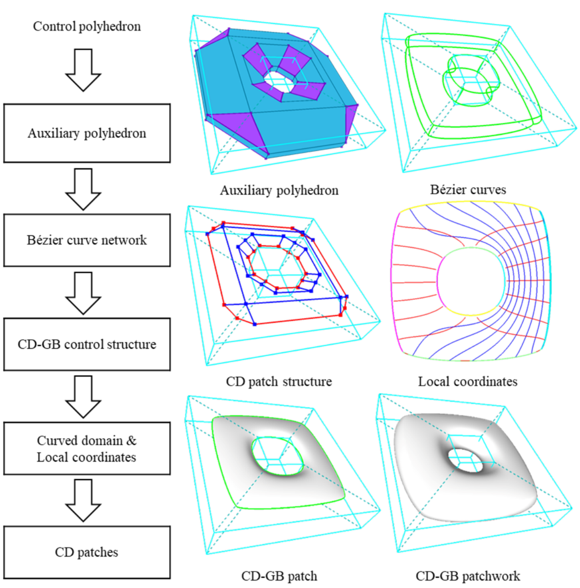


Figure 5: The flowchart of the algorithm

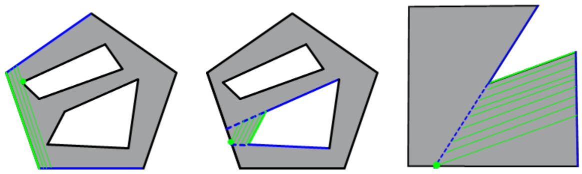
(a) Concave (b) With inner loop

Figure 6: Shrunken faces

5. Auxiliary polyhedra

As mentioned earlier, our algorithm first performs a similar operation like the refinement step of the Doo–Sabin subdivision. This generates a ‘shrunken polyhedron’ – with our terminology *auxiliary polyhedron* – this is the basis of our construction. While Doo–Sabin shrinks the individual faces by applying a convex combination of the vertices, in our case – as we need to handle faces with concave angles and holes – a different approach is needed. As shown in Figure 6, we propose to define shrunken faces by sliding the original edges ‘inwards’; thus chamfers will have boundaries *parallel* with the original edges.

First we take all faces of the control polyhedra, and for each edge we determine a *maximum offset line* of how far it can slide without intervening with another non-adjacent edge. Figure 7 shows different types of edges connecting (a) convex–convex, (b) concave–concave and (c) concave–convex corners, showing how they slide and run into obstacles. If a moving edge and its adjacent edge spans a concave angle, we slide on the extension of the adjacent edge.



(a) Convex (b) Concave (c) Mixed

Figure 7: Offset lines by corner types

Sliding determines a maximum offset for each moving edge, at the same time there is an ‘opposite edge’ that slides towards the current one. In order to avoid intersections we enforce that the actual chamfering boundary edge will not be placed beyond the half of its maximum region. We introduced a global *offset parameter* to control the depth of chamfering; it is a percentage of the maximal offset, retracting the offset lines towards the original edges. Of course, offset values can be set locally, as well. We remark that in special cases further constraints may need to be imposed for the widths of the chamfering faces, details can be found in Szörfi.[15]

Next we perform vertex chamfering to produce a collection of *facets*. Take an arbitrary vertex of the control polyhedron, and pick the neighbouring faces. In each face there are two offset lines, whose pairwise intersections determine retracted auxiliary vertices for the related facet. For faces with concave angles we have found that pairwise intersection would create shape artifacts, so we propose to generate *two* auxiliary vertices using edge extensions. This is illustrated by a simple example in Figure 8 (facets are colored purple). Here Ff1 shows a shrunken FACE-face; Ef3, Ef5 and Ef7 are EDGE-faces determined by various offset lines ol_i; and Vf2, Vf4 and Vf6 are VERTEX-faces, i.e., facets. Observe that while Vf2 is a three-sided facet with intersecting offset lines, Vf4 is four-sided with two auxiliary vertices, computed by the intersection of offset line ol_1 with the extended edge e_2, and ol_2 with the extended e_1. The significance of this will be explained in the next section, where curve network creation is described.

Figure 9 demonstrates the effect of changing the global offset parameter. While smaller offsets produce small chamfers and yield blended polyhedra with relatively high curvatures, large offsets push the chamfers inwards and yield globally smoothed polyhedra with more even curvature distribution. (See also Section 8, Case study 2.)

6. Curve network and interior control points

The auxiliary polyhedron defines a curve network of cubic Bézier curves in a natural manner. Take pairs of adjacent facets and connect their centroids in such a way that each curve is forced to be tangential to the corresponding local planes of their facets. An example is shown in Figure 10,

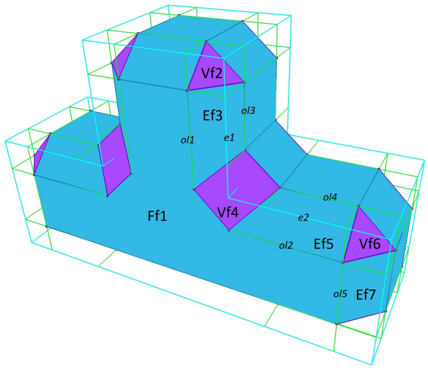
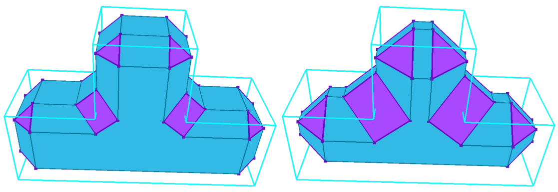
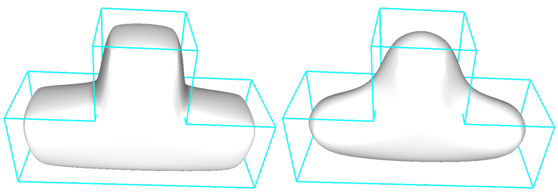


Figure 8: Creating an auxiliary polyhedron



(a) Polyhedron – small offset (b) Polyhedron – large offset



(c) Patchwork – small offset (d) Patchwork – large offset

Figure 9: Changing the global offset parameter

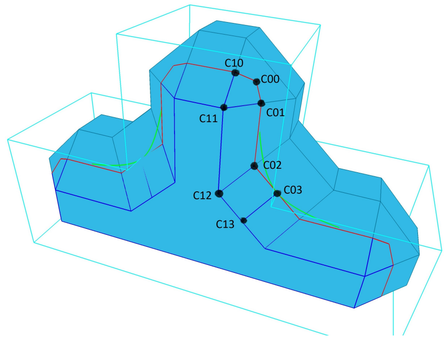
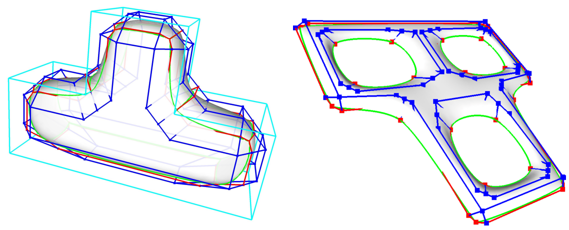


Figure 10: Curve and interior control points for a boundary segment



(a) Concave face (b) Face with three holes

Figure 11: Interior control points

where we connect the centroid of the top-right, 3-sided facet with the centroid of the middle-right, 4-sided facet. The cubic boundary curve is determined by four control points C00, C01, C02 and C03. Both C01 and C02, i.e., the tangent control points of the curve, are placed on their respective facets at the midpoints of the corresponding sides, thus the requested tangential constraints are satisfied. If we take an arbitrary facet and all the related curves starting from its centroid, smooth connection to the common tangent plane is automatically ensured.

The 4-sided facet in the middle represents a concave vertex. It can be observed that due to the symmetric placement of the control points, the left and right boundary segments meet with C^1 continuity there. In general, the adjacent boundary segments of all concave vertices will be smoothly connected and this property will be utilized when the CD-GB patches are constructed.

In order to define cross-derivatives for the individual boundary segments, we need a second row of control points. These can also be naturally derived from the auxiliary polyhedron. In our example, the control point C10 corresponds to the tangent control point of the adjacent curve starting from the centroid of the 3-sided facet. C13 represents a common tangent direction on the 4-sided facet, ensuring smoothly varying cross-derivatives between the segments of the smooth boundary. C11 and C12 are placed at the related vertices of the facets. These represent the well-known twist control points, associated with bicubic Bézier patches. The four-tuplet C10, C11, C12 and C13 will fully determine the cubic cross-derivative function associated with the selected boundary. The control structure of the full T-shaped patch can be depicted in Figure 11a, while another example with three hole loops is shown in 11b. As can be seen, the hole loops in the interior are represented by a sequence of four smoothly connected Bézier segments.

The cross-derivatives of the opposite ribbons will be inherited by the opposite multi-sided patches. For a tangent plane continuous connection the well-known direction blend methods can be applied, see details in Szörfi.[15]

We have freedom to set the magnitudes of the tangents of the cubic Bézier curves. Accordingly, we have introduced a second global shape parameter, called *tangent multiplier*, that controls the fullness of the boundary segments, and simultaneously sets the elevation of the curved patches from the faces of the control polyhedron. The endpoints at the cen-

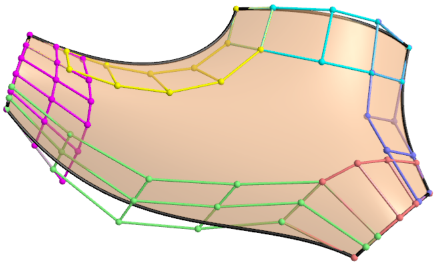
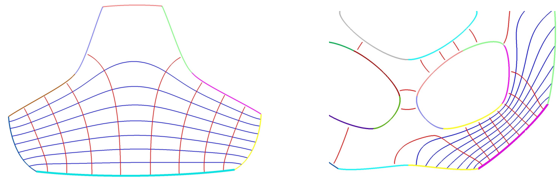


Figure 12: A CD-GB patch, different degrees and number of rows



(a) Concave face

(b) Face with three holes

Figure 13: s–h parameterizations

troids are fixed, but the tangent control points will be shifted, thus a more evenly curved patchwork can be obtained, without relatively flat surfaces. A simple example is shown in Section 8, see Case study 2.

7. Curved Domain Generalized Bézier patches

A detailed description of the CD-GB patch can be found in Várady et al.,[17] here we just summarize the basic concept. The patch is a normalized combination of n Bézier ribbons; an example is shown in Figure 12. The i-th ribbon R_i is defined by $(d_i+1)\times(e_i+1)$ control points, and its equation is given in the following form:

$$R_i(s_i,h_i) = \sum_{j=0}^{d_i}\sum_{k=0}^{e_i} C^i_{j,k}\cdot B^{d_i}_j(s_i)B^{e_i}_k(h_i). \tag{1}$$

Here $s_i, h_i \in [0,1]$ denote local parameters along the boundary and in the cross direction, respectively. $C^i_{j,k}$ refers to the j-th control point in the k-th row, being multiplied by associated Bernstein polynomials. In our current project, each ribbon is made of cubic boundaries with one additional row of control points to ensure G^1 continuity, thus $d_i = 3$ and $e_i = 1$, for all i.

The CD-GB patch is defined over a curved domain in the (u,v) plane, and for each side we compute a local parameterization, where the side parameter $s_i = s_i(u,v)$ varies linearly on side i between 0 and 1, while the distance parameter $h_i = h_i(u,v)$ vanishes on side i and increases monotonically within the domain, eventually reaching 1 as it gets to the 'distant' sides. Details of the domain generation and parameterization can be found in the above paper, here we just show two examples how the (s_i,h_i) isolines were generated for two test faces, see Figure 13.

When formulating the patch equation we combine modified ribbons R^*_i, using degree-raised Bernstein functions in the cross direction ($e^*_i = 2e_i+1$). We also introduce rational correction terms $\mu^i_j(h_i)$ to ensure that the multi-sided patch interpolates the individual Bézier ribbons:

$$R^*_i(s_i,h_i) = \sum_{j=0}^{d_i}\sum_{k=0}^{e_i} C^i_{j,k}\cdot \mu^i_j(h_i)B^{d_i}_j(s)B^{e^*_i}_k(h). \tag{2}$$

In fact, the correction terms $\mu^i_j(h_i)$ depend on the distance parameters of the adjacent sides h_{i-1} and h_{i+1}, as well. Here we apply circular indexing and it is sufficient to use squared terms for G^1 continuity. In the cubic case:

$$\mu^i_j = h^2_{i-1}/(h^2_{i-1}+h^2_i) \qquad \text{for } j\in\{0,1\}, \tag{3}$$

$$\mu^i_j = h^2_{i+1}/(h^2_{i+1}+h^2_i) \qquad \text{for } j\in\{2,3\}. \tag{4}$$

In order to ensure the convex combination property, we need to guarantee that the weighting functions sum up to 1 in each point of the curved domain. To this end, we normalize the patch equation by dividing with the sum of the control point weights:

$$S(u,v) = \frac{1}{B_\Sigma(u,v)}\cdot\sum_{i=1}^{n} R^*_i(s_i,h_i), \tag{5}$$

where

$$B_\Sigma(u,v) = \sum_{i=1}^{n}\sum_{j=0}^{d_i}\sum_{k=0}^{e_i} \mu^i_j(h_i)B^{d_i}_j(s_i)B^{e^*_i}_k(h_i). \tag{6}$$

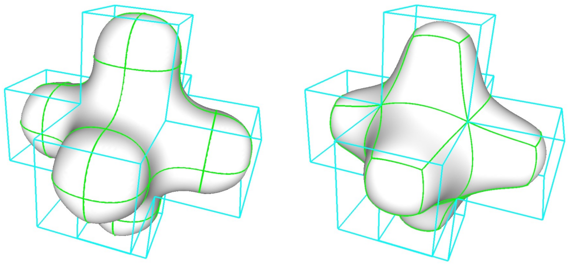
For multi-loop configurations the above formula can be generalized in a straightforward manner.

8. Discussion

In this section we try to demonstrate the capabilities of the topology preserving method by means of a few interesting examples. All the objects and images have been generated by aninteractive 3D test program[15] written in C++.

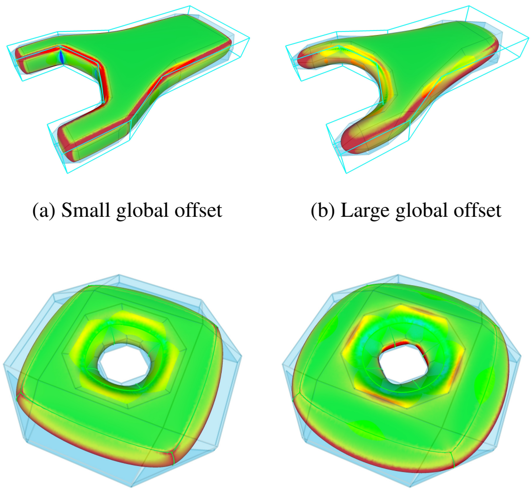
Case study 1. We can observe the difference between the dual and the topology preserving approach by a simple model that ha four-sided faces only. While the dual method yields 16 three-sided and 8 six-sided patches, our solution preserves the structure yielding only quadrilateral patches, see Figure 14.

Case study 2. One advantage of our approach is that – in fact – a family of objects can be produced by adjusting the shape parameters. Increasing the global offset yields smoother patches, but due to the enlarged chamfers it also pulls the patches inwards within the control polyhedron, see Figure 15b. When the tangent multipliers are increased, the auxiliary polyhedron is retained, but the fullness of the



(a) Dual (b) Topology preserving

Figure 14: Comparison



(a) Small global offset (b) Large global offset

(c) Small tangent multiplier (d) Large tangent multiplier

Figure 15: Changing the shape parameters

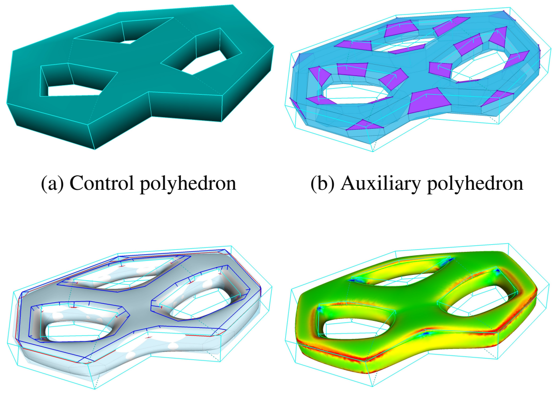
boundary segments (and thus the multi-sided patches) will be enlarged. A simple example is shown in Figure 15d.

Case study 3. Th strength of our solution is well demonstrated in Figure 16, where the algorithm successfully processed a model with three through holes. Note that the top of the patchwork is covered by a single patch containing three internal edge loops.

Case study 4. The complexity of the individual patches can be observed in Figure 17, where we display two interesting patches, separately.

9. Conclusion

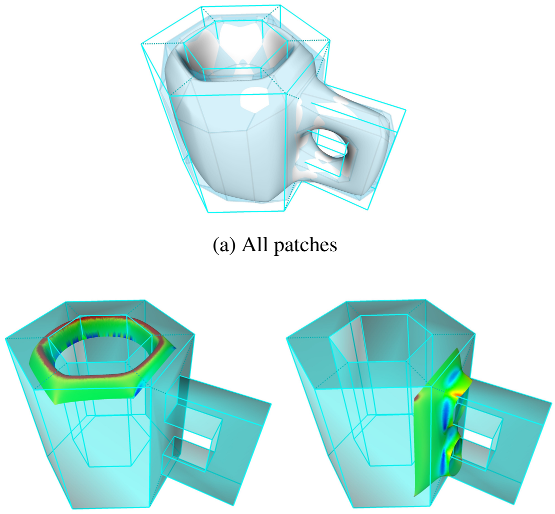
We have investigated a new modeling technique to create a free-form patchwork from a control polyhedron. The novelty of the approach is that the constituting surfaces are multi-sided, multi-connected Bézier patches over a topology preserving structure. First an auxiliary polyhedron is computed by special chamfering operations, then the control points of



(a) Control polyhedron (b) Auxiliary polyhedron

(c) Control structure (d) Patches

Figure 16: A model with several inner loops



(a) All patches

(b) A patch with one hole (c) A patch with two holes

Figure 17: Individual patches

the multi-sided patches are determined. We have also discussed shape variations using different parameters.

This scheme produces tangent plane continuous patches, and the generalization for curvature continuity is clearly a challenging task. The auxiliary polyhedron can be generated by many different rules, and further research is needed to analyze the 'pros and cons' of the alternative solutions.

Acknowledgements

This project has been partly supported by the Hungarian Scientific Research Fund (OTKA, No. 124727: Modeling general topology free-form surfaces in 3D). The authors acknowledge highly valuable technical discussions and contributions from Péter Salvi and Márton Vaitkus, in particular,

the 3D test environment and the curved domain library, respectively.

References

1. E. Catmull, J. Clark: Recursively generated B-spline surfaces on arbitrary topological meshes, *Computer-Aided Design*, **10**(6):350–355, 1978.

2. H. Chiyokra, F. Kimura: A new surface interpolation method for irregular curve models, *Computer Graphics Forum*, **3**:209-218, 1984

3. D. Doo: A subdivision algorithm for smoothing down irregularly shaped polyhederons, In: *Proceedings of the international conference on Interactive Techniques in Computer-Aided Design*, pp. 157–165, 1978.

4. G. Farin: Curves and Surfaces for CAGD – 5th Edition, *Elsevier*, 2002.

5. G. J. Hettinga, J. Kosinka: A multisided C^2 B-spline patch over extraordinary vertices in quadrilateral meshes, *Computer-Aided Design* **127**, #102855, 2020.

6. K. Karciauskas, J. Peters: Fair free-form surfaces that are almost everywhere parametrically C^2, *Journal of Computational and Applied Mathematics* **349**:470–481, 2019.

7. K. Karciauskas, J. Peters: A new class of guided C^2 subdivision surfaces combining good shape with nested refinement, *Computer Graphics Forum* **37**(6):84–95, 2018.

8. C. Loop, T. D. DeRose: Generalized B-spline surfaces of arbitrary topology, In: *Proceedings of the 17th annual conference on Computer Graphics and Interactive Techniques*, pp. 347–356, 1990.

9. C. Loop, S. Schaefer: Approximating Catmull-Clark subdivision surfaces with bicubic patches, *ACM Transactions on Graphics (TOG)*, **27**(1):1–11, 2008.

10. P. Malraison: N-sided surfaces: A survey, In: *Curve and Surface Design, P-J. Laurent, P. Sablonnière, L. L. Schumaker (eds.)*, pp. 247–255, Vanderbilt University Press, 1999.

11. J. Peters: Constructing C^1 Surfaces of arbitrary topology using biquadratic and bicubic splines, *Designing Fair Curves and Surfaces, N. S. Sapidis (ed.)*, pp. 277–294, SIAM, 1994.

12. J. Peters: Splines for meshes with irregularities, *SMAI Journal of Computational Mathematics*, **5**:161–183, 2019.

13. A. Rockwood, K. Gao: SuperD: Conceptual 3D modeling on mobiles, *ACM SIGGRAPH 2018 Appy Hour*, #7, 2018.

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

15. J. Szörfi: Kontrollpoliéder alapú 3D modellezés általánosított Bézier felületekkel (in Hungarian), Diploma Thesis, Budapest University of Technology and Economics, 2021.

16. T. Várady, P. Salvi, G. Karikó: Multi-sided Bézier Patch with a Simple Control Structure, *Computer Graphics Forum*, **35**(2):307–317, 2016.

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