Eleventh Hungarian Conference on Computer Graphics and Geometry, Budapest, 2024

Polyhedral design with blended n-sided interpolants

Péter Salvi

Budapest University of Technology and Economics

Abstract

A new parametric surface representation is proposed that interpolates the vertices of a given closed mesh of arbitrary topology. Smoothly connecting quadrilateral patches are created by blending local, multi-sided quadratic interpolants. In the non-four-sided case, this requires a special parameterization technique involving rational curves. Appropriate handling of triangular subpatches and alternative subpatch representations are also discussed.

1. Introduction

Surface representations based on control polyhedra come in various guises, including recursive subdivision,[1] generalized splines[7] and 'manifold' approaches.[5] In this paper we examine a variation of the manifold-based construction exemplified by the works of Zorin,[13, 10] and recently by Fang.[3]

Figure 1 shows the basic idea. The input is a closed mesh. We assume that it only contains quads – if not, we can perform a central split on all faces (similarly to a Catmull–Clark subdivision step) to get rid of the multi-sided patches while retaining all original vertices. For each quad, the 1-rings around its corners define (multi-sided) quadratic control nets. We can generate four local patches interpolating these control points, and evaluate them in the parametric region associated with the quad (Fig. 2). The resulting surface is created by blending these together.

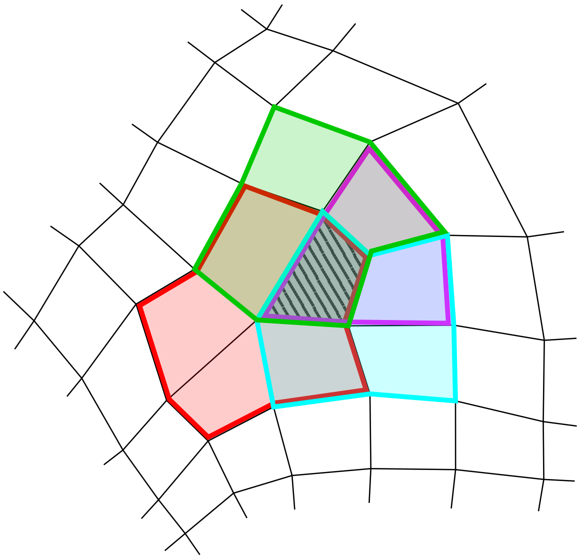


Figure 1: Overlapping quadratic nets around a quad.

Note that while most similar methods are approximative, here we focus on patches that interpolate the input points. The interpolation criterion makes the interpolants more similar in the given region, which may enhance the quality of the resulting surface.

The rest of the paper is organized as follows. Section 2 shows the construction in the simple case of regular quadmeshes. Handling of irregular vertices is explained in Section 3, with the definition of Quadratic Generalized Bézier (QGB) patches in Section 3.1 and a parameterization based on rational Bézier curves in Section 3.2. Some test results are presented in Section 4; notes on future work conclude the paper.

2. Regular meshes

In a regular (closed) mesh all vertices have a valency of 4, so all control nets around a quad will be quadrilateral. We use quadratic tensor-product Bézier patches for the interpolants.

Let $\mathbf{C}_i$ and $\mathbf{E}_i$ ($i = 1 \dots 4$) denote the corner and edge control points, with $\mathbf{E}_i$ being between $\mathbf{C}_{i-1}$ and $\mathbf{C}_i$ (with cyclical indexing). The middle control point is denoted by $\mathbf{M}$. Then

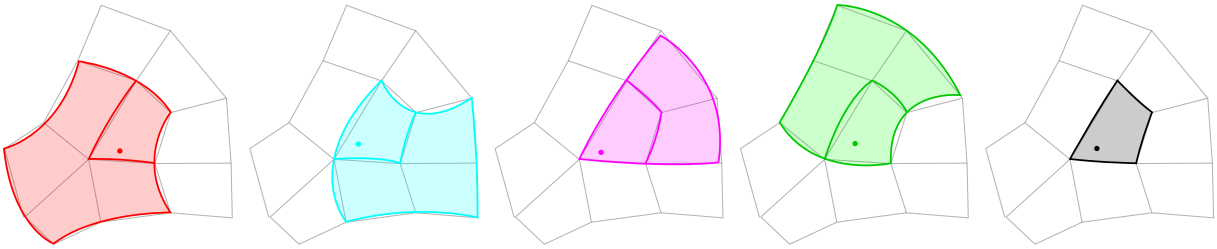


Figure 2: Blending the local interpolants. Dots indicate a point of evaluation in each; the last image shows the resulting patch.

the interpolant is defined as

$$\mathbf{I}(u,v) = \sum_{i=0}^{2}\sum_{j=0}^{2}\mathbf{P}_{ij}B_i^2(u)B_j^2(v), \tag{1}$$

where $B_k^d(t)$ is the k-th Bernstein polynomial of degree d at parameter t, and

$$\begin{aligned}
&\mathbf{P}_{00} = \mathbf{C}_3, && \mathbf{P}_{10} = \hat{\mathbf{E}}_4, && \mathbf{P}_{20} = \mathbf{C}_4,\\
&\mathbf{P}_{01} = \hat{\mathbf{E}}_3, && \mathbf{P}_{11} = \frac{1}{4}(16\mathbf{M} - \sum_{i=1}^{4}(\mathbf{C}_i + 2\hat{\mathbf{E}}_i)), && \mathbf{P}_{21} = \hat{\mathbf{E}}_1,\\
&\mathbf{P}_{02} = \mathbf{C}_2, && \mathbf{P}_{12} = \hat{\mathbf{E}}_2, && \mathbf{P}_{22} = \mathbf{C}_1.
\end{aligned} \tag{2}$$

Here $\hat{\mathbf{E}}_i = 2\mathbf{E}_i - \frac{1}{2}(\mathbf{C}_{i-1} + \mathbf{C}_i)$ is the control point position s.t. $[\mathbf{C}_{i-1}, \hat{\mathbf{E}}_i, \mathbf{C}_i]$ define a quadratic Bézier curve interpolating $\mathbf{E}_i$.

The parameterization of the quad is in $[0,1]^2$, and is rotated locally with the following interpolation constraints in mind ($\mathbf{S}$ denotes the patch to be created):

$$\begin{aligned}
\mathbf{S}(0,0) &= \mathbf{M}, & \mathbf{S}(1,0) &= \mathbf{E}_1,\\
\mathbf{S}(0,1) &= \mathbf{E}_2, & \mathbf{S}(1,1) &= \mathbf{C}_1.
\end{aligned} \tag{3}$$

The interpolant point corresponding to the (u,v) point in the quad's domain is then $((u+1)/2, (v+1)/2)$.

Consequently, the quad patch can be defined as the blend of the four interpolants:

$$\mathbf{S}(u,v) = \sum_{i=1}^{4}\mathbf{I}_i\left(\frac{u_i+1}{2}, \frac{v_i+1}{2}\right)\Phi(u_i,v_i). \tag{4}$$

The local parameterizations are defined as

$$\begin{aligned}
u_1 &= u, & v_1 &= v,\\
u_2 &= v, & v_2 &= 1-u,\\
u_3 &= 1-u, & v_3 &= 1-v,\\
u_4 &= 1-v, & v_4 &= u.
\end{aligned} \tag{5}$$

The blending function is the product of two 1-argument blends

$$\Phi(u,v) = \Psi(u)\cdot\Psi(v) \tag{6}$$

with the following constraints:

$$\Psi(0) = 1, \quad \Psi(1) = 0, \quad \Psi^{(k)}(0) = \Psi^{(k)}(1) = 0 \tag{7}$$

for some $k > 0$. For finite k, Hermite blends can be applied:

$$\Psi(t) = \sum_{i=0}^{k} B_i^{2k+1}(t). \tag{8}$$

For $k = \infty$ there are several options, including bump functions and expo-rational B-splines.[8] We have used the $k = 2$ Hermite blend for all examples in the paper.

3. Irregular vertices

An irregular vertex generates a non-quadrilateral control net. We interpolate these points by quadratic generalized Bézier patches, as defined below.

3.1. Quadratic Generalized Bézier (QGB) interpolants

Generalized Bézier (GB) patches[11] are normally defined only for cubic or higher degrees, but is easy to further generalize the construction to the quadratic case. As this will be a C^0 interpolant, its equation will be even simpler than that of the original.

The surface is defined over the regular n-sided polygon $\{(\cos\frac{2k\pi}{n}, \sin\frac{2k\pi}{n})\}$, $k = 0\dots n-1$, and for each side we create local coordinate mappings

$$s_i(u,v) = \lambda_i/(\lambda_{i-1}+\lambda_i), \quad d_i(u,v) = 1 - \lambda_{i-1} - \lambda_i, \tag{9}$$

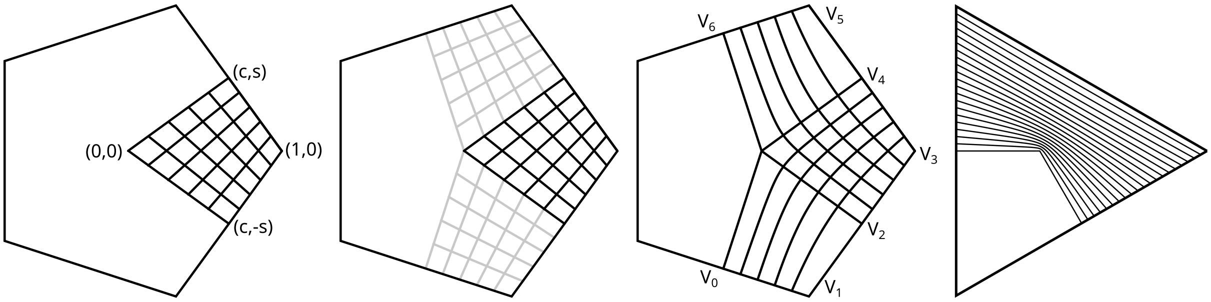
where $\{\lambda_i\}$ are the Wachspress coordinates[4] of (u,v) relative to the domain polygon.

Using the notations of the previous section,

$$\begin{aligned}
\mathbf{I}(u,v) = \sum_{i=1}^{n}\left(\mathbf{C}_{i-1}\frac{1}{2}B_0^2(s_i) + \hat{\mathbf{E}}_i B_1^2(s_i) + \mathbf{C}_i\frac{1}{2}B_2^2(s_i)\right)B_0^2(d_i)\\
+ \mathbf{P}_0 B_0(u,v).
\end{aligned} \tag{10}$$

Here B_0 denotes the weight deficiency

$$B_0(u,v) = 1 - \sum_{i=1}^{n}\left(\frac{1}{2}B_0^2(s_i) + B_1^2(s_i) + \frac{1}{2}B_2^2(s_i)\right)B_0^2(d_i), \tag{11}$$



(a) Bilinear map (b) Discontinuity problem (c) Rational curve map (d) Curves in a triangle

Figure 3: Quad → interpolant mappings.

and the central control point $\mathbf{P}_0$ is defined s.t. the patch interpolates $\mathbf{M}$:

$$\mathbf{P}_0 = \frac{n^2\mathbf{M} - \sum_{i=1}^{n}\left(\mathbf{C}_i + 2\hat{\mathbf{E}}_i\right)}{n(n-3)}. \tag{12}$$

Note that for $n = 4$ this is exactly the same as the tensor-product interpolant in Section 2. It is also easy to see that for $n = 3$ this is a quadratic triangular Bézier patch.

3.2. Parameterization

We still need a mapping between the quad domain $[0,1]^2$ and the interpolant's domain (a regular polygon inscribed in the unit circle). The associated part of the latter is a *kite* defined by the points $(0,0)$, $(c,-s)$, $(1,0)$ and (c,s), where $c = \frac{1}{2}(\cos(2\pi/n)+1)$ and $s = \frac{1}{2}\sin(2\pi/n)$, see Figure 3. A naïve approach would be to use a bilinear mapping (Fig. 3a). This presents a problem however: with this simple mapping adjacent parts of the same interpolant would be parameterized discontinuously (Fig. 3b). In the rest of this section, we will construct a mapping that is C^∞-continuous (except at the origin, which is singular).

The proposed method works by creating *pencils* of quadratic rational Bézier curves in the two parametric directions, and the mapping of (u,v) is defined to be the intersection of the u-curve with the v-curve – see Figure 3c, where the following notations are also shown:

$$\begin{aligned}
\mathbf{V}_0 &= (c+\hat{c}, -s-\hat{s}), & \mathbf{V}_1 &= (2c-1, -2s),\\
\mathbf{V}_2 &= (c, -s), & \mathbf{V}_3 &= (1,0),\\
\mathbf{V}_4 &= (c, s), & \mathbf{V}_5 &= (2c-1, 2s),\\
\mathbf{V}_6 &= (c+\hat{c}, s+\hat{s}), & &
\end{aligned} \tag{13}$$

where $\hat{c} = \frac{1}{2}(\cos(4\pi/n)-1)$ and $\hat{s} = \frac{1}{2}\sin(4\pi/n)$. The control points and rational weights of u-curves are given as

$$\begin{aligned}
\mathbf{R}_0 &= \mathbf{V}_0(1-u) + \mathbf{V}_1 u, & w_0 &= 1,\\
\mathbf{R}_1 &= \mathbf{V}_2 u \exp(u^2 - u), & w_1 &= 1/u,\\
\mathbf{R}_2 &= \mathbf{V}_4(1-u) + \mathbf{V}_3 u, & w_2 &= 1.
\end{aligned} \tag{14}$$

Similarly for v-curves:

$$\begin{aligned}
\mathbf{R}_0 &= \mathbf{V}_6(1-v) + \mathbf{V}_5 v, & w_0 &= 1,\\
\mathbf{R}_1 &= \mathbf{V}_4 v \exp(v^2 - v), & w_1 &= 1/v,\\
\mathbf{R}_2 &= \mathbf{V}_2(1-v) + \mathbf{V}_3 v, & w_2 &= 1.
\end{aligned} \tag{15}$$

Then the curves themselves can be evaluated with the formula

$$\mathbf{r}(t) = \frac{\sum_{i=0}^{2}\mathbf{R}_i w_i B_i^2(t)}{\sum_{i=0}^{2} w_i B_i^2(t)}. \tag{16}$$

The exponential term in the middle control point is there to accommodate for the triangular domain where the $\mathbf{V}_2 - \mathbf{V}_3 - \mathbf{V}_5 - \mathbf{V}_6 - (0,0)$ polygon is concave, and this helps pulling back the curve near the origin. Figure 3d shows v-curves in a triangle for $v = k/20$ values, $k = 0 \ldots 20$.

For a given (u,v) point in the quad domain, we need to intersect the corresponding u- and v-curves. This is easily done with nested golden section searches in the $[\frac{1}{2}, 1]$ interval.[6] While this is a bit expensive, it can (and should) be precomputed.

3.3. Triangular patches

The three-sided QGB patch is a quadratic Bézier triangle, and as such, does not have a central control point. Since we want to interpolate the middle point $\mathbf{M}$, we should degree-elevate the boundaries to cubic and use a cubic Bézier triangle, which has an extra degree of freedom:

$$\begin{aligned}
&\mathbf{P}_{300} = \mathbf{C}_1, \quad \mathbf{P}_{030} = \mathbf{C}_2, \quad \mathbf{P}_{003} = \mathbf{C}_3,\\
&\mathbf{P}_{210} = \frac{1}{3}(\mathbf{C}_1 + 2\hat{\mathbf{E}}_2), \quad \mathbf{P}_{120} = \frac{1}{3}(\mathbf{C}_2 + 2\hat{\mathbf{E}}_2),\\
&\mathbf{P}_{021} = \frac{1}{3}(\mathbf{C}_2 + 2\hat{\mathbf{E}}_3), \quad \mathbf{P}_{012} = \frac{1}{3}(\mathbf{C}_3 + 2\hat{\mathbf{E}}_3),\\
&\mathbf{P}_{102} = \frac{1}{3}(\mathbf{C}_3 + 2\hat{\mathbf{E}}_1), \quad \mathbf{P}_{201} = \frac{1}{3}(\mathbf{C}_1 + 2\hat{\mathbf{E}}_1),
\end{aligned} \tag{17}$$

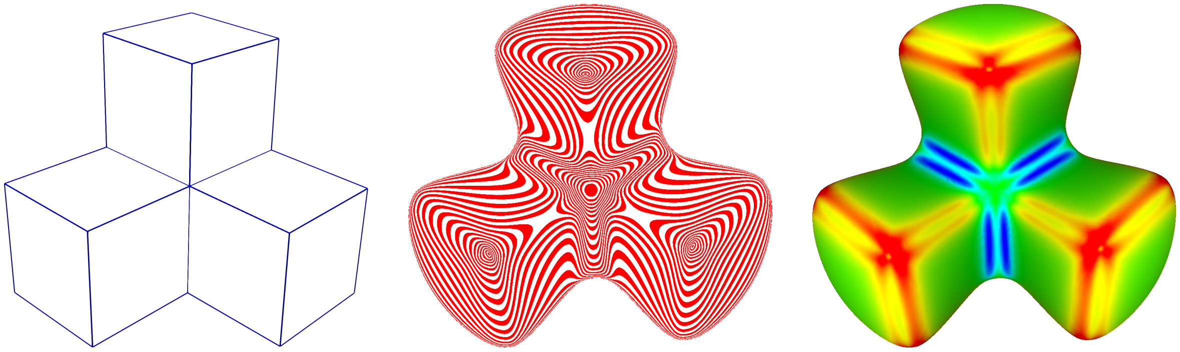


Figure 4: Trebol model (left: cage, center: isophote map, right: mean curvature map).

and

$$\mathbf{P}_{111} = \frac{1}{6}\Big(27\mathbf{M} - \sum_{\max(i,j,k)=3} \mathbf{P}_{ijk} - 3 \sum_{\max(i,j,k)=2} \mathbf{P}_{ijk}\Big). \quad (18)$$

Then the patch can be evaluated by

$$\mathbf{I}(u,v) = \sum_{i+j+k=3} \mathbf{P}_{ijk} \frac{6}{i!j!k!} \lambda_1^i \lambda_2^j \lambda_3^k. \quad (19)$$

3.4. Alternative representations

There are many multi-sided surface representations that we could use; for a survey, see our upcoming paper.[12] In particular, Midpoint and Midpoint Coons patches[9] are very well suited for this task. Our experiments have shown, however, little difference in the resulting models, while computationally QGB patches are simpler and more efficient.

4. Results

Our first model is a regular quadmesh in the shape of a torus, see Figure 5. The result is far from a torus, but the isophote lines flow smoothly on the surface.

The *trebol* object shown in Figure 4 is a commonly used test model as it has vertices of valences 3, 4, 5 and 6. The isophote lines indicate that the multi-sided patches are of good quality and connect continuously, while the mean map shows that edges tend to be a little flat.

The icosahedron model in Figure 6 is composed of triangles. As a first step, a Catmull–Clark step is performed, resulting in a quadmesh with 3-, 4- and 5-valent vertices. The mean curvature map suggests larger values near 5-valent vertices, but otherwise there are only small fluctuations, corresponding to the mesh edges.

Finally, Figures 7–8 show a more complex model[†] with environment mapping, contouring and mean curvature.

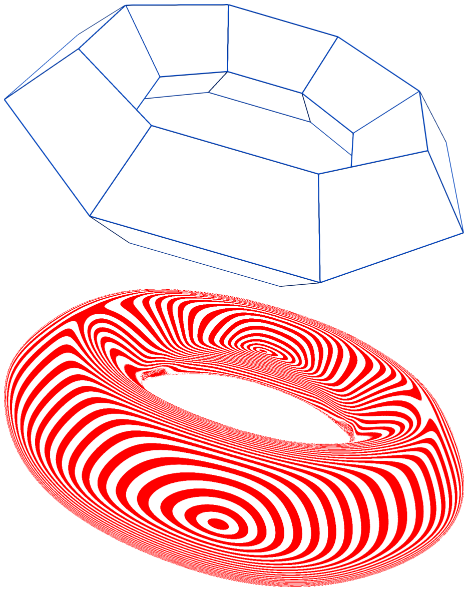


Figure 5: Torus model (top: cage, bottom: isophote map).

Conclusion and future work

We have proposed a new construction for a piecewise parametric surface interpolating a mesh of arbitrary topology. There are many avenues for further research. Currently our method works only for closed meshes; it should be generalized to meshes with boundary. Then continuous boundary constraints (positional and cross-derivative interpolation) may also be added. Extension with shape parameters can also be considered. Better quality may be achieved if normal vector interpolation is incorporated.

Acknowledgments

This project has been supported by the Hungarian Scientific Research Fund (OTKA, No. 145970).

[†] Taken from the code supplied to the paper on K-surfaces.[2]

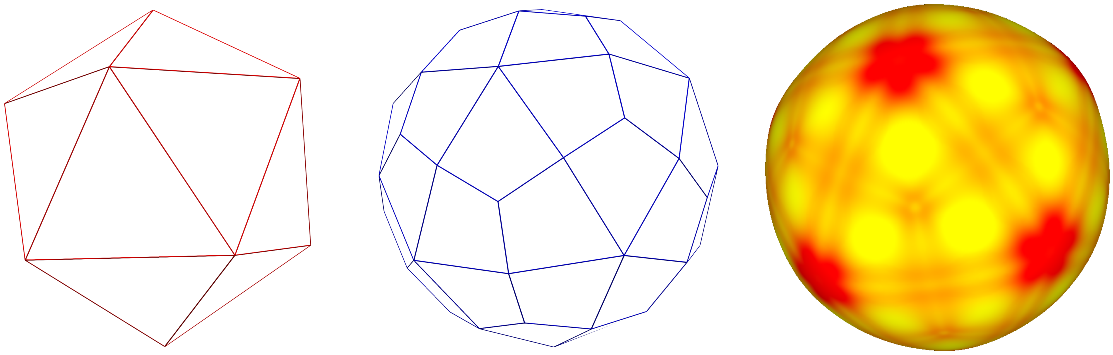


Figure 6: Icosahedron model (from left to right: triangular cage, cage after subdivision, mean curvature map).

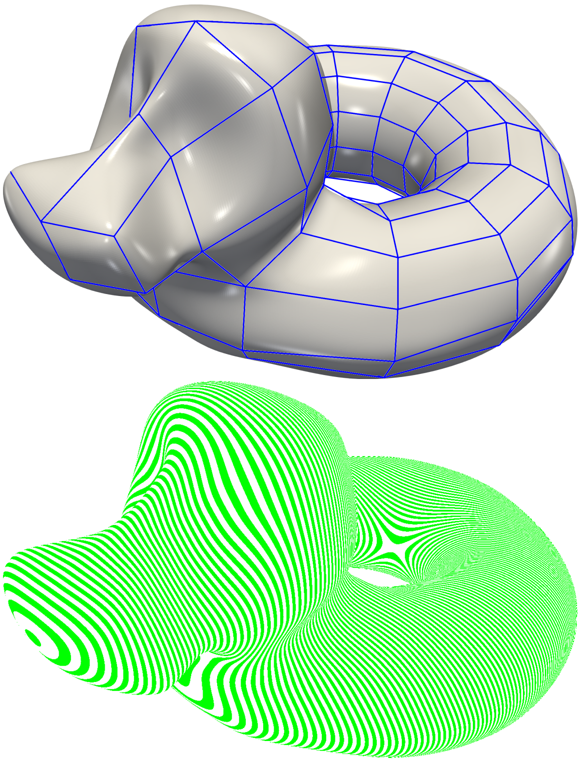


Figure 7: ‘Bob’ model (top: cage, bottom: contouring).

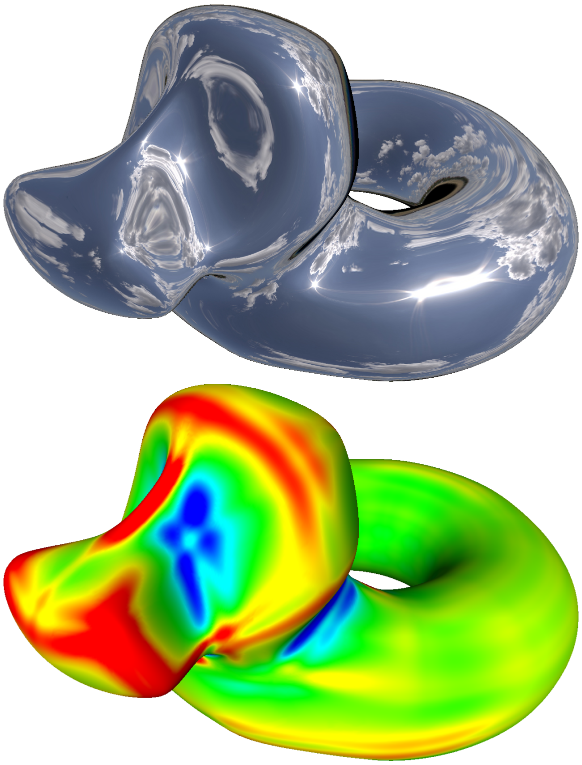


Figure 8: ‘Bob’ model (top: reflection, bottom: mean map).

References

1. Thomas J. Cashman. Beyond Catmull–Clark? A survey of advances in subdivision surface methods. *Computer Graphics Forum*, 31(1):42–61, 2012.

2. Tobias Djuren, Maximilian Kohlbrenner, and Marc Alexa. K-surfaces: Bézier-splines interpolating at Gaussian curvature extrema. *ACM Transactions on Graphics (TOG)*, 42(6):210, 2023.

3. Xiang Fang. *A generalized blending scheme for arbitrary order of continuity*. PhD thesis, University of Waterloo, Canada, 2023.

4. Michael S. Floater. Generalized barycentric coordinates and applications. *Acta Numerica*, 24:161–214, 2015.

5. Cindy M. Grimm and Denis Zorin. Surface modeling and parameterization with manifolds. In *SIGGRAPH ’06 Courses*, pages 1–81. ACM, 2006.

6. Mykel J. Kochenderfer and Tim A. Wheeler. *Algo-*

rithms for Optimization. The MIT Press, 2019.

7. Jörg Peters. Splines for meshes with irregularities. *The SMAI Journal of computational mathematics*, 5:161–183, 2019.

8. Péter Salvi. Editing the interior of arbitrary surfaces using C^∞ displacement blends. In *Proceedings of the Workshop on the Advances of Information Technology*, pages 35–38. BME, 2021.

9. Péter Salvi, István Kovács, and Tamás Várady. Computationally efficient transfinite patches with fullness control. In *Proceedings of the Workshop on the Advances of Information Technology*, pages 96–100. BME, 2017.

10. Elif Tosun and Denis Zorin. Manifold-based surfaces with boundaries. *Computer Aided Geometric Design*, 28(1):1–22, 2011.

11. Tamás Várady, Péter Salvi, and György Karikó. A multi-sided Bézier patch with a simple control structure. *Computer Graphics Forum*, 35(2):307–317, 2016.

12. Tamás Várady, Péter Salvi, and Márton Vaitkus. Genuine multi-sided parametric surface patches – a survey. *Computer Aided Geometric Design*, 2024 (accepted).

13. Lexing Ying and Denis Zorin. A simple manifold-based construction of surfaces of arbitrary smoothness. *ACM Transactions on Graphics (TOG)*, 23(3):271–275, 2004.