A Parametric Hybrid Triangular Bézier Patch

Stephen Mann and Matthew Davidchuk

Abstract. We describe a parametric hybrid Bézier patch that, in addition to blending interior control points, blends boundary control points. This boundary blend is necessary to generalize a functional cross-boundary construction that relies on the natural parameterization of the functional setting. When interpolating irregularly scattered data and when increasing the tessellation of the data mesh, the new scheme shows improvement over representative parametric data fitting schemes.

§1. Introduction

A large number of local parametric triangular surface schemes have been developed over the past fifteen years (see [10] for a survey of such schemes). These schemes are local in that changes to part of the data only affect portions of the surface near the changed data. Surprisingly, all of these schemes exhibit similar shape defects. On closer inspection, it is seen that these schemes all have a large number of free parameters that are set using simple heuristics. By manually adjusting these parameters, one can improve the shape of the surfaces [7].

One way to improve automatically the shape of the constructed surfaces is to use variational methods. Several authors have used such schemes to improve the shape of the constructed surfaces, but usually at a high computational cost due to the global nature of the solution (e.g., [11]). Similarly, we can use *local optimization* methods to set the free parameters and improve the shape, although the results are not as good as the global methods [8].

In this paper, we will investigate a local, non-optimization method for improving the construction of the cross-boundary derivatives in a triangular parametric scheme. This is a generalization of the hybrid scattered data fitting scheme of Foley and Opitz [2] (their method is similar to a result independently developed by Goodman and Said [4]). The resulting surfaces show large improvement in shape over other local, parametric, triangular surface fitting techniques. More precisely, given a triangle of data (a set of three vertices with normals), our scheme constructs a hybrid, parametric patch that

Mathematical Methods for Curves and Surfaces II
Morten Dæhlen, Tom Lyche, Larry L. Schumaker (eds.), pp. 1–3.

ISBN 1-xxxxx-xxx-x.

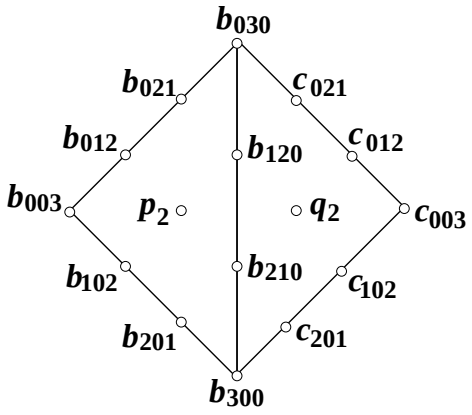

Fig. 1. Domain control net for the Foley-Opitz hybrid Bézier patch.

interpolates the positions and normals at the corners. When used to fill a triangular polyhedron, the resulting surface patches will meet with tangent plane continuity.

§2. The Foley-Opitz Scheme

Foley and Opitz [2] present a method for interpolation of scattered data above the plane using a "hybrid" cubic Bézier patch based on Nielson's scheme [12]. A hybrid cubic patch is similar to a cubic Bézier patch, except the interior control point is a rational blend of three points. With the Foley-Opitz method, the cubic patch boundaries are completely determined by the triangle vertices and normals. The three inner control points are constructed using a C^1 cross boundary construction that gives the hybrid patch cubic precision.

Figure 1 shows the domain control net for two neighboring triangles. $\boldsymbol{p_2}$ is one of the three interior control points associated with the left triangle and $\boldsymbol{q_2}$ is one of the three interior control points associated with the right triangle.

Foley and Opitz compute $\boldsymbol{p_2}$ as follows. Let r, s, and t be the barycentric coordinates of $\boldsymbol{c_{003}}$ with respect to $\boldsymbol{b_{003}}$, $\boldsymbol{b_{030}}$, and $\boldsymbol{b_{003}}$. If both patches if Figure 1 form a single cubic, then from subdividing Bézier cubics it can be shown that

$$\begin{aligned}\boldsymbol{p_2} = \big(&\boldsymbol{c_{102}} + \boldsymbol{c_{012}} - r^2(\boldsymbol{b_{300}} + \boldsymbol{b_{210}}) - 2rs(\boldsymbol{b_{210}} + \boldsymbol{b_{120}}) - 2rt\boldsymbol{b_{201}} - 2st\boldsymbol{b_{021}} \\ &- s^2(\boldsymbol{b_{120}} + \boldsymbol{b_{030}}) - t^2(\boldsymbol{b_{102}} + \boldsymbol{b_{012}})\big)/(2(r+s)t)\end{aligned}.$$

The point $\boldsymbol{q_2}$ is forced by continuity conditions to be

$$\boldsymbol{q_2} = r\boldsymbol{p_2} + s\boldsymbol{b_{120}} + t\boldsymbol{b_{210}}.$$

When applied to data that does not come from a cubic, the Foley-Opitz construction of $\boldsymbol{p_2}$ and $\boldsymbol{q_2}$ ensures that the two triangles have a C^1 join along their common border. Identical calculations would be made to ensure C^1 continuity across the remaining two edges giving three settings for the interior control points of each of the two patches.

Fig. 2. Isophotes of interpolants to the Franke data set.

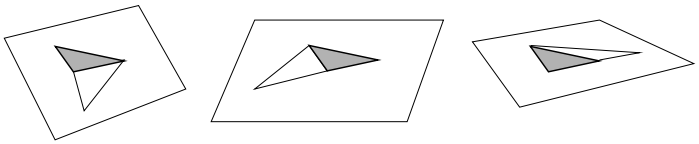

Fig. 3. One plane per patch pair.

The three interior points ($\boldsymbol{p_0}$, $\boldsymbol{p_1}$, $\boldsymbol{p_2}$) are blended with Nielson's rational blend functions,

$$a_i(t_0,t_1,t_2) = \frac{t_j t_k}{t_i t_j + t_i t_k + t_j t_k}, \quad i \neq j \neq k. \tag{1}$$

giving

$$\boldsymbol{b_{111}}(t_0,t_1,t_2) = a_0(t_0,t_1,t_2)\boldsymbol{p_0} + a_1(t_0,t_1,t_2)\boldsymbol{p_1} + a_2(t_0,t_1,t_2)\boldsymbol{p_2}.$$

After blending, we are left with a 10 point cubic Bézier patch, which is evaluated at (t_0,t_1,t_2) in the standard way.

Figure 2 shows isophote plots [5] of the Franke data set [3] (on the left), and isophote plots of this data interpolated with Clough-Tocher (middle) and Foley-Opitz (right) patches. The C^1 discontinuities in the isophote lines occur along patch boundaries, and are visible as shape artifacts in shaded images [9]. The Foley-Opitz interpolant is generally smoother than the Clough-Tocher interpolant.

§3. A Hybrid, Parametric, Cubic Scheme

Our goal is to create a parametric version of the Foley-Opitz scheme to bring its good surface quality to the parametric setting. There are two problems we must solve to create this parametric version of the Foley-Opitz method: Finding local parameterizations for the Foley-Opitz cross-boundary method, and finding blend functions for the resulting points that have the appropriate properties.

The Foley-Opitz cross-boundary construction relies on a natural parameterization between patch pairs since the barycentric coordinates of neighboring patches with respect to each other are key in determining the tangent plane fields. In the parametric setting, there is no predefined association between patch domains. Therefore, some association between neighboring patch domains must be made in order to use Foley's tangent plane field construction.

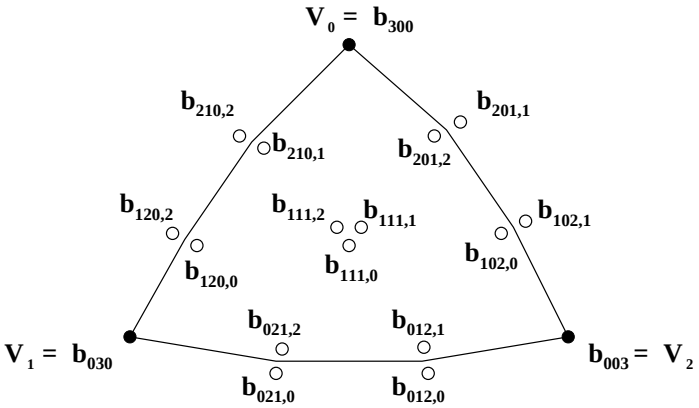

Fig. 4. Domain control net for parametric hybrid patch.

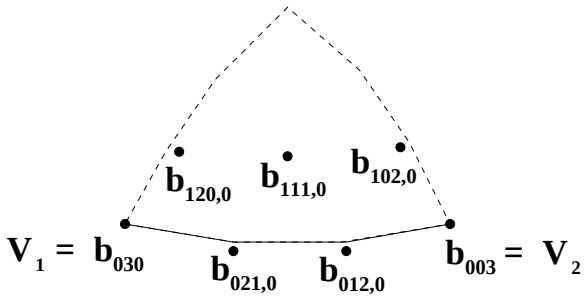

Fig. 5. Tangent plane field control points along the $b_{030}b_{003}$ edge.

Our approach is to choose a plane for each patch pair (Figure 3), project the corner points of both patches onto the plane, and then perform Foley's C^1 construction. Three sets of control points are calculated for each hybrid patch – each set representing a C^1 construction along one triangle edge. The three sets of control points will share the same triangle corner vertices but in general differ in the rest of the boundary and interior control points.

We must blend both boundary and interior control points to produce the final interpolant. Figure 4 shows the domain control net for a parametric version of Foley's scheme. The structure is similar to Foley's – the control points are organized like the control points of a cubic triangular Bézier patch except a group of control points correspond to a single, regular cubic control point.

When constructing the tangent plane field along a particular boundary, only two parametric hybrid patches are involved and consequently only two Bézier patches are needed for the construction. Each Bézier patch contributes seven control points to a parametric hybrid patch, as in Figure 5. These control points determine the tangent plane field along that boundary. To construct the seven points, a plane is chosen as a parameterization for the two Bézier patches allowing the construction as in Foley's functional scheme. Once the plane is chosen, the Bézier patch control points are completely determined by the triangle vertices and the associated normals – Hermite interpolation over a plane completely determines the cubic boundary curves and Foley's

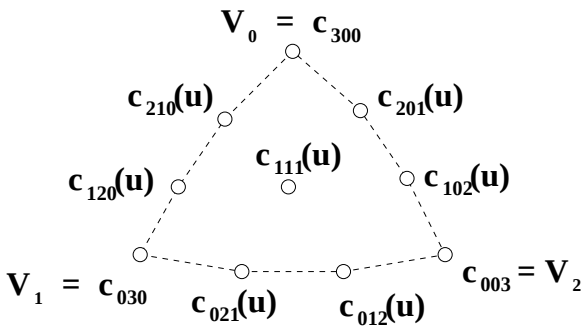

Fig. 6. Control net after blending parametric hybrid control points.

cross boundary construction determines the interior control points.

After the control points from three Bézier patches are calculated, they can be used to formulate the parametric hybrid patch. However, blending is not as straight forward as in the functional case – boundary points are included in the blend. Figure 6 illustrates the control net for the parametric patch. Each point $\boldsymbol{c_{ijk}}$ is a rational blend of the associated control points from the three Bézier patches shown in Figure 4. The points $\boldsymbol{c_{300}}, \boldsymbol{c_{030}}$, and $\boldsymbol{c_{003}}$ are constants, the remaining boundary control points are rational blends of two Bézier patch boundary control points, and the interior point $\boldsymbol{c_{111}}$ is a blend of three points.

The blend formulation must preserve the important properties of the three underlying Bézier patches, boundary curves and cross boundary derivatives, when evaluating along the edges. The corner control points are constants so no blending function is needed. The blending function for the interior control points is the same as used in Foley's functional construction, giving us

$$\boldsymbol{c_{111}(u)} = a_0(\boldsymbol{u})\boldsymbol{b_{111,0}} + a_1(\boldsymbol{u})\boldsymbol{b_{111,1}} + a_2(\boldsymbol{u})\boldsymbol{b_{111,2}}$$

where the a_i are defined in (1). The boundary control points are blended to give two properties:

1) When evaluated along a boundary the parametric Foley control points become the control points of one of the three Bézier patches
2) The tangent plane field of the parametric hybrid patch along a boundary matches the tangent plane field along the same boundary of one of the three Bézier patches.

An asymmetric blend of the following form has both of these properties

$$h_{ij}(u_0, u_1, u_2) = \frac{(1-u_i){u_j}^2}{(1-u_i){u_j}^2 + (1-u_j){u_i}^2} \tag{2}$$

This blending function is used to weight all the non-vertex boundary Bézier control points:

$$c_{ijk}(u_0, u_1, u_2) = h_{ij}(u_0, u_1, u_2)\boldsymbol{b_{ijk,i}} + h_{ji}b_{ijk,k}\boldsymbol{b_{ijk,j}},$$

for ijk being any permutation of 012. For example, the $\boldsymbol{V_1V_2}$ boundary control points would be

$$\boldsymbol{c_{012}}(u_0,u_1,u_2) = h_{01}(u_0,u_1,u_2)\boldsymbol{b_{012,0}} + h_{10}(u_0,u_1,u_2)\boldsymbol{b_{012,1}}$$

$$\boldsymbol{c_{021}}(u_0,u_1,u_2) = h_{02}(u_0,u_1,u_2)\boldsymbol{b_{021,0}} + h_{20}(u_0,u_1,u_2)\boldsymbol{b_{021,2}}$$

When evaluated along the $\boldsymbol{V_1V_2}$ edge $u_0 = 0$ we get

$$\boldsymbol{c_{012}}(0,u_1,u_2) = \boldsymbol{b_{012,0}}, \qquad \boldsymbol{c_{021}}(0,u_1,u_2) = \boldsymbol{b_{021,0}}$$

since Equation (2) gives $h_{0j}(0,u_1,u_2) = 1$ and $h_{i0}(0,u_1,u_2) = 0$. The $\boldsymbol{V_1V_2}$ boundary curve is the cubic Bézier curve given by the control points $\boldsymbol{b_{030}}$, $\boldsymbol{b_{021,0}}$, $\boldsymbol{b_{012,0}}$, and $\boldsymbol{b_{003}}$, which gives us C^0 continuity. This blend also gives us C^1 continuity, as described in Davidchuk's thesis [1].

The control points $\{\boldsymbol{b_{300}}, \boldsymbol{b_{030}}, \boldsymbol{b_{003}}, \boldsymbol{b_{\vec{i},j}}\}$ are blended to form the ten control points of a standard cubic Bézier patch, labeled as $\boldsymbol{c_{\vec{i}}}(\boldsymbol{u})$. The Bézier patch defined by the $\boldsymbol{c_{\vec{i}}}$ is then evaluated at $\boldsymbol{u}$. The concise definition is

$$\boldsymbol{F}(\boldsymbol{u}) = \sum_{|\vec{i}|=3} B^3_{\vec{i}}(\boldsymbol{u})\boldsymbol{c_{\vec{i}}}(\boldsymbol{u})$$

§4. Choice of plane

There is freedom in the choice of the projection plane, and there are some restrictions. The orientation, not the position, of the plane determines the positions of the control points thus giving two rotational degrees of freedom. The plane should be constructed geometrically from the given information – triangle vertices and normals. The two Bézier patches must not have overlapping domains on the plane so the orientation is restricted to being "underneath" both patches.

One failsafe method of choosing a plane is to take the plane perpendicular to the bisecting plane of the two neighboring triangles and that contains their common edge. With this choice of plane, the projection of the triangles will lie on opposite sides of the projection of their common edge, so the patch domains will never overlap.

However, a better choice is the plane perpendicular to the average of the normals at the two data points, and either date point. Although this construction is not guaranteed to give us a valid plane (i.e., the two projection of the two triangles along the edge may overlap), it does in general give us better shaped surfaces [1].

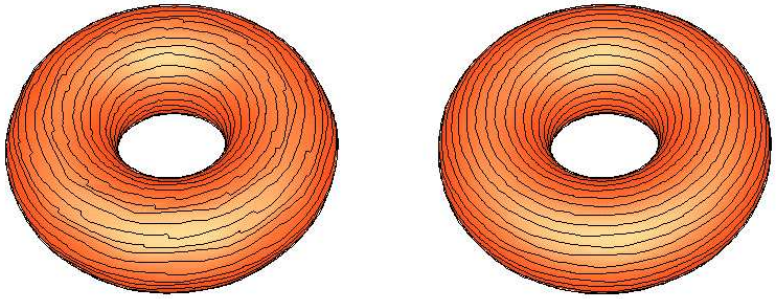

Fig. 7. Triangular Gregory patches and our scheme fit to a torus.

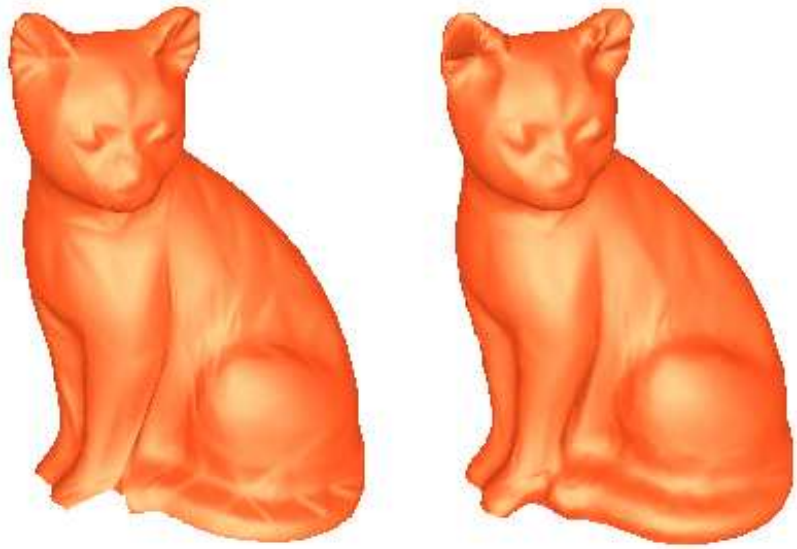

Fig. 8. Triangular Gregory patches and our scheme fit to a cat data set.

§Results

To test our surface construction method, we used it to fit patches to a cat data set (where normals are estimated) and to samplings of a torus (where normals come from the torus). We compared our method to Triangular Gregory patches [6]. In Figure 7, we see isophote plots of both methods fit to a 10x10 sampling of the torus. The isophotes for our method are noticeably smoother.

In Figure 8, we see shaded images of both methods fit to a cat data set. This data set has 366 vertices and 698 faces. Normals to the vertices were estimated by a simple averaging of face normals. Many of the shape artifacts that appear on triangular Gregory patches our not present on the surface constructed by our scheme.

One drawback to our scheme is that as the tangent planes on either side of an edge become perpendicular to the edge, the interior points of the boundary curve move towards infinity. Thus, normal estimation becomes a critical step.

Acknowledgments. This research was funded by the Natural Sciences and Engineering Research Council of Canada

References

1. Davidchuk, M., A Parametric Tybrid trianglar Bezier Patch, dissertation, Master's, University of Waterloo (Waterloo ON), 1997.
2. Foley, T. A. and K. Opitz, Hybrid Cubic Bezier Triangle Patches, in *Mathematical Methods in Computer Aided Geometric Design II*, T. Lyche and L. Schumaker (eds), Academic Press, New York, 1992, 275–286.
3. Franke, R., A critical comparison of some methods for interpolation of scattered data, Report NPS-53-79-003, Naval Postgraduate School, 1979.
4. Goodman, T. N. T. and H. B. Said, A C^1 triangular interpolant suitable for scattered data interpolation, Commun. Appl. Numer. Methods **7** (1991), 479–485.
5. Hagen, H., S. Hahmann, T. Schreiber, Y. Nakajima, B. Wördenweber, and P. Hollemann-Grundetedt, Surface interrogation algorithms, Comp. Graphics and Applics. **12** (1992), 53–60.
6. Longhi, L., Interpolating patches between cubic boundaries, Technical Report T.R. UCB/CSD 87/313, University of California, Berkeley, 1986.
7. Mann, S., Surface Approximation Using Geometric Hermite Patches, dissertation, Doctoral, University of Washington (Seattle, WA), 1992.
8. Mann, S., Using local optimization in surface fitting, in *Mathematical Methods for Curves and Surfaces*, Morten Dæhlen, Tom Lyche, Larry L. Schumaker (eds), Vanderbilt University Press, Nashville & London, 1995, 323–332.
9. Mann, S., Cubic precision Clough-Tocher interpolation, submitted for publication.
10. Mann, S., C. Loop, M. Lounsbery, D. Meyers, J. Painter, T. DeRose, and K. Sloan, A survey of parametric scattered data fitting using triangular interpolants, in *Curve and Surface Design*, H. Hagen (ed), SIAM Publications, SIAM, Philadelphia PA, 1992, 145–172.
11. Moreton, H. P. and C. H. Séquin, Functional Optimization for Fair Surface Design, Computer Graphics (ACM SIGGRAPH) **26** (1992), 167–176.
12. Nielson, G. M., A transfinite, visually continuous, triangular interpolant, in *Geometric Modeling: Algorithms and New Trends*, G. E. Farin (ed), SIAM Publications, Philadelphia, 1987, 235–245.

Stephen Mann and Matthew Davidchuk
Computer Science Department
University of Waterloo,
Waterloo, ON, N2L 2R7, CANADA
smann@cgl.uwaterloo.ca, mdavidch@cgl.uwaterloo.ca