

Using Local Optimization in Surface Fitting

Computer Science Department
University of Waterloo
Research Report CS-94-33

Stephen Mann

University of Waterloo

Abstract

Local optimization is used to set the free parameters in a triangular surface fitting scheme, resulting in surfaces with better shape. While some of the free parameters can be set to match curvature information, other free parameters are independent of this information.

1 Introduction

A large number of local parametric triangular surface schemes have been developed over the past fifteen years (see [MLL⁺92, Man92] 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 [Man92].

One way to improve automatically the shape of the constructed surfaces is to use global optimization. Here, *global optimization* refers to using all the data to construct any portion of the surface. 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 [CG91, MS92].

In this paper, I will investigate using local optimization to set the free parameters in a local, triangular, split-domain, polynomial surface interpolation scheme. More precisely, given a triangle of data, this scheme constructs three triangular parametric polynomial patches, each of which interpolates two of the positions and normals at the three vertices. Further, the patches meet each other with G^1 continuity, and if used to fill a triangular polyhedron, the resulting surface will also be G^1 . Local optimization is

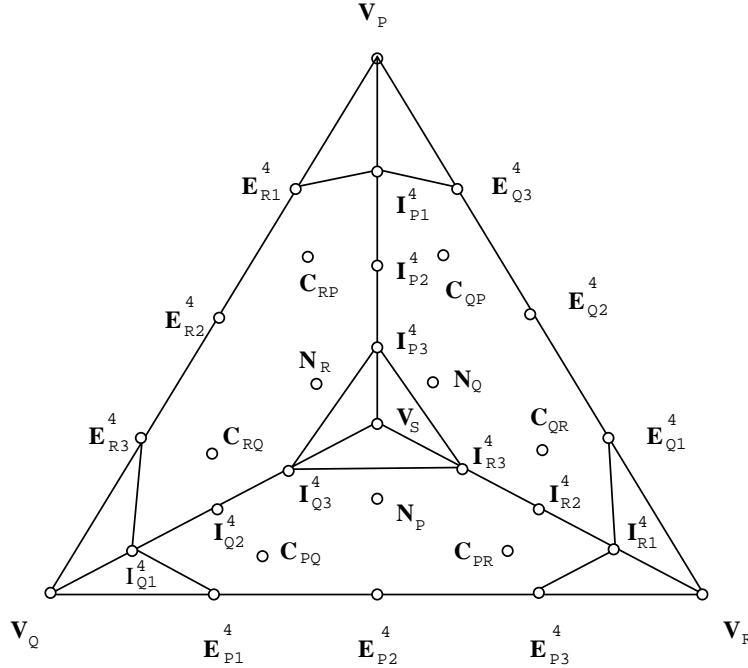


Figure 1: Shirman-Séquin Bézier control points.

used to set the remaining free parameters so as to minimize the error in interpolating second order data at the corners. Thus, our data is a set of three vertices with normals and second fundamental forms, $(p_i, \hat{n}_i, \mathbb{I}_i)$, for $i = P, Q, R$.

2 The Modified Shirman-Séquin Scheme

I will be working with a modified version of Shirman-Séquin's scheme [SS87, SS91]. They created a split domain scheme that fits three quartic patches to a triangle of data. These triangles meet each other G^1 and if used to fill a triangular polyhedron, the entire surface will be G^1 . Note that while the patches are quartic, the boundaries of the patches are only degree three.

The Bézier control points for the three patches are shown in Figure 1 (there are several references available for background on triangular Bézier patches; see, for example, Farin's book [Far93]). We will have use for both the cubic and quartic control points of the boundaries; thus, we shall denote the cubic control points with a superscript of 3 and the quartic with a superscript of 4. Using a modification of the Shirman-Séquin scheme, for $ijk \in \{PQR, QRP, RPQ\}$, we set these control points in the following manner:

$$V_i = P_i$$

$$\begin{aligned}
E_{i1}^3 &= P_i + \alpha_{i1} \vec{t}_{i1} \\
E_{i2}^3 &= P_j - \alpha_{i2} \vec{t}_{i2} \\
I_{i1}^3 &= (1 - \beta_i)P_i + \beta_i(E_{j2}^4 + E_{k1}^4)/2 \tag{1}
\end{aligned}$$

$$C_{ij} = \frac{1}{3}\{c_i(k_{ij} + k_{ik})\widehat{C}_{ij} + k_{ij}\widehat{C}_{ik} + 2h_{ij}(E_{i2}^3 - E_{i1}^3) + h_{ik}(E_{i1}^3 - V_j)\} + E_{i1}^3 \tag{2}$$

$$C_{ik} = \frac{1}{3}\{k_{ik}\widehat{C}_{ij} + c_i(k_{ij} + k_{ik})\widehat{C}_{ik} + h_{ij}(V_k - E_{i2}^3) + 2h_{ik}(E_{i2}^3 - E_{i1}^3)\} + E_{i2}^3 \tag{3}$$

$$I_{i2}^3 = \frac{3\beta_i}{16}V_i + \left(1 - \frac{27\beta_i}{16}\right)I_{i1}^3 + \frac{12\beta_i}{16}(C_{ji} + C_{ki})$$

$$N_i = \frac{1}{4}I_{i1}^3 - \frac{1}{4}(I_{j1}^3 + I_{k1}^3) + \left(\frac{4}{9\beta_i} - \frac{5}{4}\right)I_{i2}^3 + \left(\frac{-2}{9\beta_i} + \frac{5}{4}\right)(I_{k2}^3 + I_{j2}^3)$$

$$V_S = (I_{P2}^3 + I_{Q2}^3 + I_{R2}^3)/3$$

Here, the \vec{t}_{in} are vectors perpendicular to the normal at P_i . The α s and β s are free parameters; any positive setting of these free parameters will give us a G^1 surface scheme for most choices of the \vec{t}_{in} . The c_i are shape parameters used by Jensen [Jen87] in constructing the crossboundary functions (Shirman-Séquin used Chiyokura-Kimura's crossboundary function [CK83], which is given by Jensen's function when $c_i = 1$).

The \widehat{C} s, ks , and hs are formed by Chiyokura-Kimura's method: \widehat{C}_{ij} is perpendicular to both the normal \widehat{n}_j and to the vector $E_{i1}^3 - V_j$ with the sign of \widehat{C}_{ij} chosen to make k_{ij} positive. Similarly, \widehat{C}_{ik} is perpendicular to the normal \widehat{n}_k and $E_{i2}^3 - V_k$. The ks and hs are given by:

$$I_{i1}^4 - V_j = k_{ij} \cdot \widehat{C}_{ij} + h_{ij} \cdot (E_{i1}^3 - V_j) \tag{4}$$

$$I_{k1}^4 - V_k = k_{ik} \cdot \widehat{C}_{ik} + h_{ik} \cdot (V_k - E_{i2}^3) \tag{5}$$

The above ks and hs actually depend linearly on the β_i s. Since we want to optimize over the β_i , we will express the above equations in terms of the β_i by computing the ks and hs when the β_i s are 1.0. Call these \widehat{k}_{ij} , \widehat{h}_{ij} , \widehat{k}_{ik} , and \widehat{h}_{ik} . The above two equations then become

$$\begin{aligned}
I_{i1}^4 - V_j &= \beta_j \widehat{k}_{ij} \cdot \widehat{C}_{ij} + \beta_j \widehat{h}_{ij} \cdot (E_{i1}^3 - V_j) \\
I_{k1}^4 - V_k &= \beta_k \widehat{k}_{ik} \cdot \widehat{C}_{ik} + \beta_k \widehat{h}_{ik} \cdot (V_k - E_{i2}^3)
\end{aligned}$$

Many of the degrees of freedom occur in the boundaries of the three patches. In the discussion that follows, *exterior boundaries* refer to the boundaries given by the V_i and E_{in}^3 . *Interior boundaries* refer to the boundaries given by the V_i , I_{in}^3 and V_s , for $i \in \{P, Q, R\}$ $n \in \{1, 2, 3\}$.

Counting, we find a total of twelve shape parameters in the above equations. There are nine more shape parameters in the Shirman-Séquin scheme: we can vary the \vec{t}_{in} , and the I_{i1} could be allowed to vary in the tangent plane. We will set the \vec{t}_{in} by placing each boundary curve in a plane. The choice of planar boundaries uses three degrees

of freedom, while the particular planes chosen use another three. I chose the plane passing through the two data points and the average of the normals at the data points, which sets all six degrees of freedom. The I_{i1} s were restricted to a linear degree of freedom for reasons of symmetry.

3 Cubic Curve Optimization

In Pictures 1-3, we see three views of five surfaces. Each picture shows the five surfaces from one view. Three views are given to help the reader visualize the shade defects. In each picture, the surface on the left is an S-patch surface [LD89]. The second surface on the left is the same S-patch surface subdivided into triangular pieces. Each corner of these pieces has been sampled for position, normal, and second fundamental form. This particular sampling was created by an ε - G^1 scheme during its refinement process [DM92, Man92]. The surface in the center is a Shirman-Séquin surface fit to this data.

The original motivation for this work was to see if a G^1 scheme could create an approximation to the ring surface with quality comparable to the ε - G^1 scheme (the surface created by the ε - G^1 is indistinguishable from the S-patch surface in shaded images). To make the comparison reasonable, I used data set constructed by the ε - G^1 scheme so that both schemes would be working with the same data.¹ Thus, our goal is to improve the quality of this center surface. In particular, note the severity of the shape artifact at the “top” of the ring (this artifact is most visible in Picture 2). We will begin by improving the exterior boundary curves.

The left column of Figure 3 shows two of the exterior boundary curves constructed for this ring data set by the standard Shirman-Séquin scheme (these are the Bézier curves described by the control points V_i and E_{in}). The 'x's mark the Bézier control points of these cubic curves. Below each curve is a curvature plot of the curve (the curved line is the curvature plot; the straight line connects the curvature of the S-patch surface at the endpoints of the curve). While the curvature of the top curve is distributed relatively uniformly over the curve, the curvature of the bottom curve is seen to concentrate at the endpoints of the curve, leaving a flat region in the middle.

As discussed in [Man92], we can expect improved surface shape if we match second order data with the boundary curves. Thus, we would like a solution to the problem of interpolating the position, tangent direction, and curvature at two points with a cubic curve. Klass solved this problem numerically [Kla83], while de Boor et. al. found an analytic solution to the problem [dBHS87]. Note that a solution may not exist, and

¹This comparison is somewhat unfair to the G^1 scheme since the data was refined until the ε - G^1 scheme created a good surface, and in this sense is an “optimal” data set for the ε - G^1 scheme. However, this data suffices for initial comparisons.

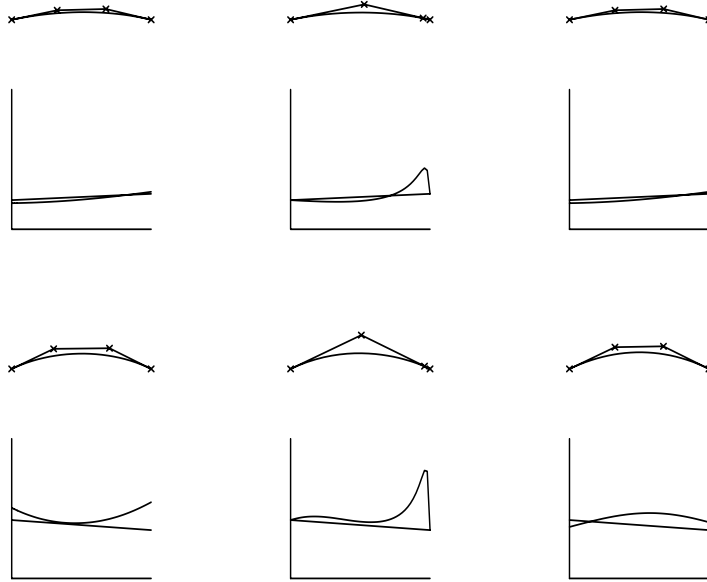


Figure 2: Boundary curves and their curvature plots.

when it does exist, it may not be unique.

I implemented the de Boor-Hollig-Sabin technique and integrated it into Shirman-Séquin's scheme. While some areas of the surface improve, there are regions where "lumps" appear in the surfaces as illustrated in the fourth surface of Pictures 1-3. Looking at the boundary curves again reveals the cause of the shape defect.

The center column of Figure 2 shows the two curves constructed by the deBoor-Hollig-Sabin method for the same data used in the first column. As can be seen in the figure, both curves have a short first derivative at the right end point and a spike in curvature near that end of the curve (note that the derivative is short enough that the right two control points almost coincide). This short derivative causes a problem for the surface construction scheme because short endpoint first derivatives lead to large values of the h_{ij} (Equations 4 and 5). These scalars are then used to weight vector quantities in the construction of the C_{ij} (Equations 2 and 3).

Thus, the result of short derivatives at the end of the exterior boundaries is that the C_{ij} and any interior control points dependent on the C_{ij} may be positioned far from the triangle of data, as illustrated in Figure 3. In this figure, there are two views of two patches constructed for the same three data points. Two of the exterior boundary curves are identical in both patches. However, the data along the third boundary has two cubic curves that interpolate the specified position, tangent directions, and curvatures. In the patch on the left, we see that both tangents at the ends of this boundary curve are of reasonable length, and that the interior control points are uniformly distributed. However, in the patch on the right, we see that one of the tangents at the

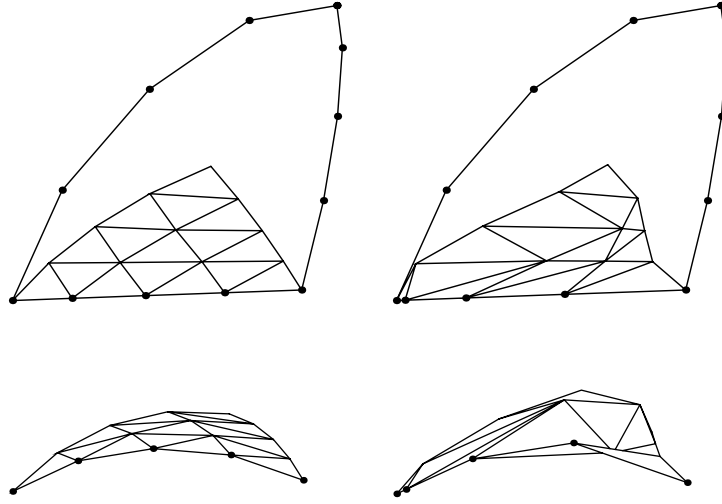


Figure 3: The patch on the left was constructed with a boundary curve that has end point first derivatives with reasonable lengths. The identical construction was used for the patch on the right, except that one of the boundary curves has a short first derivative at one end.

ends of this curve is short. The result is that the interior control points have a more chaotic placement, leading to the lumps seen in Pictures 1 and 3.

Thus, it is insufficient to construct exterior boundaries that match curvature at their endpoints. We must ensure that the derivatives at the ends of the curves are not too short. Therefore, we will solve a variation of the above curve construction problem. We will find a cubic curve that interpolates the position and tangent direction, and trades off interpolating the curvature with interpolating prescribed first derivatives at the endpoints.

We want our construction to be scale independent. To achieve this goal, we will always scale our problem to a “unit” problem by mapping the endpoints to be separated by unit distance. The first derivatives will scale “down” linearly, while the curvatures scale “up” linearly. So, if we wish to find f such that f interpolates our data at its endpoints as follows,

$$\begin{aligned} f(0) &= P_0, & f'(0) &= \vec{t}_0, & k(0) &= k_0 \\ f(1) &= P_1, & f'(1) &= \vec{t}_1, & k(1) &= k_1 \end{aligned}$$

then we will map it to

$$\begin{aligned} \overline{P}_0 &= P_0, & \overline{\vec{t}}_0 &= \vec{t}_0 / |P_0 - P_1|, & \overline{k}_0 &= k_0 \cdot |P_0 - P_1| \\ \overline{P}_1 &= P_0 + (P_1 - P_0) / |P_1 - P_0|, & \overline{\vec{t}}_1 &= \vec{t}_1 / |P_0 - P_1|, & \overline{k}_1 &= k_1 \cdot |P_0 - P_1| \end{aligned}$$

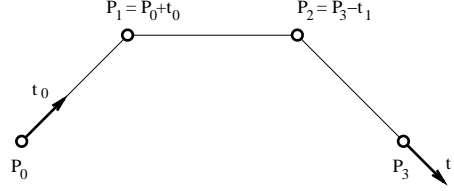


Figure 4: Control points of a cubic curve.

After constructing \bar{f} , we map the control points of \bar{f} through the inverse transformation, giving the control points of f . However, for notational simplicity, I will use the unbarred symbols, and use f to denote the scaled curve.

Our goal is to satisfy the following set of equations:

$$\begin{aligned} |f'(0) - \vec{t}_0| &= 0, & |f'(1) - \vec{t}_1| &= 0, & (6) \\ \frac{f'(0) \times f''(0)}{|f'(0)|^3} - k(0) &= 0, & \frac{f'(1) \times f''(1)}{|f'(1)|^3} - k(1) &= 0. & (7) \end{aligned}$$

In general we cannot expect to satisfy all four equations simultaneously. Thus, we will rephrase this as a minimization problem. First, we square all four equations to remove the derivative discontinuity. Then, we will rewrite these equations in terms of the control points. From Figure 4, we see that the derivatives f at the end points are:

$$\begin{aligned} f(0) &= P_0 \\ f'(0) &= 3(P_1 - P_0) = \alpha_0 \vec{t}_0 \\ f''(0) &= 6(P_2 - 2P_1 + P_0) \\ &= 6[(P_3 - P_0) - \alpha_1 \vec{t}_1 / 3 - 2\alpha_0 \vec{t}_0 / 3] \\ f(1) &= P_3 \\ f'(1) &= 3(P_3 - P_2) = \alpha_1 \vec{t}_1 \\ f''(1) &= 6(P_3 - 2P_2 + P_1) \\ &= 2[3(P_0 - P_3) + \alpha_0 \vec{t}_0 + 2\alpha_2 \vec{t}_1] \end{aligned}$$

We can rewrite Equations 6 in terms of the length of the first derivative, i.e., setting \vec{t}_{i1} and \vec{t}_{i2} to be the desired length. We set our goal first derivative to be of unit length for simplicity:

$$(\alpha_0 - 1.0)^2, \quad (\alpha_1 - 1.0)^2. \quad (8)$$

For the curvature equations (Equations 7), we square them to make them strictly non-negative:

$$\frac{(f'(0) \times f''(0))^2}{|f'(0)|^6} - \frac{2k_0(f'(0) \times f''(0))}{|f'(0)|^3} + k_0^2 \quad (9)$$

$$\frac{(f'(1) \times f''(1))^2}{|f'(1)|^6} - \frac{2k_1(f'(1) \times f''(1))}{|f'(1)|^3} + k_1^2 \quad (10)$$

I used an optimizer that needs the derivatives of the equations we optimize [MAB89]. The two partials of the first two equations are given by

$$\frac{\partial(\alpha_i - 1)^2}{\partial\alpha_j} = [2\alpha_i - 2]\delta_{ij},$$

where δ is the Kronecker delta function.

We will be optimizing Equations 8, 9, and 10 over α_0 and α_1 . To expand these equations in terms of the α s, we will rewrite their terms with respect to the α s:

$$\begin{aligned} f'(0) \times f''(0) &= \alpha_0 \vec{t}_0 \times 6[(P_3 - P_0) - \alpha_1 \vec{t}_1 / 3 - 2\alpha_0 \vec{t}_0 / 3] \\ &= 2\alpha_0 [3 \vec{t}_0 \times (P_3 - P_0) - \alpha_1 \vec{t}_0 \times \vec{t}_1] \\ \frac{\partial}{\partial\alpha_0}(f'(0) \times f''(0)) &= 2[3 \vec{t}_0 \times (P_3 - P_0) - \alpha_1 \vec{t}_0 \times \vec{t}_1] \\ \frac{\partial}{\partial\alpha_1}(f'(0) \times f''(0)) &= -2\alpha_0 \vec{t}_0 \times \vec{t}_1 \\ f'(1) \times f''(1) &= 2\alpha_1 [3 \vec{t}_1 \times (P_0 - P_3) + \alpha_0 \vec{t}_1 \times \vec{t}_0] \\ \frac{\partial}{\partial\alpha_0}(f'(1) \times f''(1)) &= 2\alpha_1 \vec{t}_1 \times \vec{t}_0 \\ \frac{\partial}{\partial\alpha_1}(f'(1) \times f''(1)) &= 2[3 \vec{t}_1 \times (P_0 - P_3) + \alpha_0 \vec{t}_1 \times \vec{t}_0] \end{aligned}$$

Now we'll take the partials of the equations we wish to minimize and expand in terms of our control points and free parameters:

$$\begin{aligned} &\frac{\partial}{\partial\alpha_0} \left(\frac{(f'(0) \times f''(0))^2}{|f'(0)|^6} - \frac{2k_0(f'(0) \times f''(0))}{|f'(0)|^3} + k_0^2 \right) \\ &= \frac{\partial}{\partial\alpha_0} \left(\frac{(f'(0) \times f''(0))^2}{\alpha_0^6} - \frac{2k_0(f'(0) \times f''(0))}{\alpha_0^3} + k_0^2 \right) \\ &= \frac{\partial}{\partial\alpha_0} \frac{(f'(0) \times f''(0))^2}{\alpha_0^6} - \frac{\partial}{\partial\alpha_0} \frac{2k_0(f'(0) \times f''(0))}{\alpha_0^3} + 0 \\ &= \alpha_0^{-6} \frac{\partial}{\partial\alpha_0} (f'(0) \times f''(0))^2 + (f'(0) \times f''(0))^2 \frac{\partial}{\partial\alpha_0} \alpha_0^{-6} - \\ &\quad \alpha_0^{-3} \frac{\partial}{\partial\alpha_0} 2k_0(f'(0) \times f''(0)) - 2k_0(f'(0) \times f''(0)) \frac{\partial}{\partial\alpha_0} \alpha_0^{-3} \\ &= \frac{2(f'(0) \times f''(0)) \frac{\partial}{\partial\alpha_0} (f'(0) \times f''(0))}{\alpha_0^6} - \frac{6(f'(0) \times f''(0))^2}{\alpha_0^7} \\ &\quad - \frac{2k_0 \frac{\partial}{\partial\alpha_0} (f'(0) \times f''(0))}{\alpha_0^3} + \frac{6k_0(f'(0) \times f''(0))}{\alpha_0^4} \end{aligned}$$

$$\begin{aligned}
&= \frac{8\alpha_0[3\vec{t}_0 \times (P_3 - P_0) - \alpha_1\vec{t}_0 \times \vec{t}_1]^2}{\alpha_0^6} - \frac{24\alpha_0^2[3\vec{t}_0 \times (P_3 - P_0) - \alpha_1\vec{t}_0 \times \vec{t}_1]^2}{\alpha_0^7} \\
&\quad - \frac{4k_0[3\vec{t}_0 \times (P_3 - P_0) - \alpha_1\vec{t}_0 \times \vec{t}_1]}{\alpha_0^3} + \frac{12k_0\alpha_0[3\vec{t}_0 \times (P_3 - P_0) - \alpha_1\vec{t}_0 \times \vec{t}_1]}{\alpha_0^4} \\
&= \frac{-16[3\vec{t}_0 \times (P_3 - P_0) - \alpha_1\vec{t}_0 \times \vec{t}_1]^2}{\alpha_0^5} + \frac{8k_0[3\vec{t}_0 \times (P_3 - P_0) - \alpha_1\vec{t}_0 \times \vec{t}_1]}{\alpha_0^3}
\end{aligned}$$

$$\begin{aligned}
&\frac{\partial}{\partial\alpha_1} \left(\frac{(f'(0) \times f''(0))^2}{\alpha_0^6} - \frac{2k_0(f'(0) \times f''(0))}{\alpha_0^3} + k_0^2 \right) \\
&= \frac{\partial}{\partial\alpha_1} \frac{(f'(0) \times f''(0))^2}{\alpha_0^6} - \frac{\partial}{\partial\alpha_1} \frac{2k_0(f'(0) \times f''(0))}{\alpha_0^3} \\
&= \frac{2(f'(0) \times f''(0)) \frac{\partial}{\partial\alpha_1}(f'(0) \times f''(0))}{\alpha_0^6} - \frac{2k_0 \frac{\partial}{\partial\alpha_1}(f'(0) \times f''(0))}{\alpha_0^3} \\
&= \frac{-8\alpha_0^2[3\vec{t}_0 \times (P_3 - P_0) - \alpha_1\vec{t}_0 \times \vec{t}_1][\vec{t}_0 \times \vec{t}_1]}{\alpha_0^6} - \frac{-4k_0\alpha_0[\vec{t}_0 \times \vec{t}_1]}{\alpha_0^3} \\
&= \frac{-8[3\vec{t}_0 \times (P_3 - P_0) - \alpha_1\vec{t}_0 \times \vec{t}_1][\vec{t}_0 \times \vec{t}_1]}{\alpha_0^4} + \frac{4k_0[\vec{t}_0 \times \vec{t}_1]}{\alpha_0^2}
\end{aligned}$$

The formulae for the partials in terms of f' is identical at 0 and 1. We just need to make the substitution. We will also convert $\vec{t}_1 \times \vec{t}_0$ to $-\vec{t}_0 \times \vec{t}_1$ for efficiency reasons.

$$\begin{aligned}
&\frac{\partial}{\partial\alpha_0} \left(\frac{(f'(1) \times f''(1))^2}{\alpha_1^6} - \frac{2k_1(f'(1) \times f''(1))}{\alpha_1^3} + k_1^2 \right) \\
&= \frac{2(f'(1) \times f''(1)) \frac{\partial}{\partial\alpha_0}(f'(1) \times f''(1))}{\alpha_1^6} - \frac{2k_1 \frac{\partial}{\partial\alpha_0}(f'(1) \times f''(1))}{\alpha_1^3} \\
&= \frac{8\alpha_1^2[3\vec{t}_1 \times (P_0 - P_3) + \alpha_0\vec{t}_1 \times \vec{t}_0][\vec{t}_1 \times \vec{t}_0]}{\alpha_1^6} - \frac{4\alpha_1 k_1[\vec{t}_1 \times \vec{t}_0]}{\alpha_1^3} \\
&= \frac{-8[3\vec{t}_1 \times (P_0 - P_3) - \alpha_0\vec{t}_0 \times \vec{t}_1][\vec{t}_0 \times \vec{t}_1]}{\alpha_1^4} + \frac{4k_1[\vec{t}_0 \times \vec{t}_1]}{\alpha_1^2}
\end{aligned}$$

$$\begin{aligned}
&\frac{\partial}{\partial\alpha_1} \left(\frac{(f'(1) \times f''(1))^2}{\alpha_1^6} - \frac{2k_1(f'(1) \times f''(1))}{\alpha_1^3} + k_1^2 \right) \\
&= \frac{2(f'(1) \times f''(1)) \frac{\partial}{\partial\alpha_1}(f'(1) \times f''(1))}{\alpha_1^6} - \frac{6(f'(1) \times f''(1))^2}{\alpha_1^7} \\
&\quad - \frac{2k_1 \frac{\partial}{\partial\alpha_1}(f'(1) \times f''(1))}{\alpha_1^3} + \frac{6k_1(f'(1) \times f''(1))}{\alpha_1^4}
\end{aligned}$$

$$\begin{aligned}
&= \frac{8\alpha_1[3\vec{t}_1 \times (P_0 - P_3) + \alpha_0\vec{t}_1 \times \vec{t}_0]^2}{\alpha_1^6} - \frac{24\alpha_1^2[3\vec{t}_1 \times (P_0 - P_1) + \alpha_0\vec{t}_1 \times \vec{t}_0]^2}{\alpha_1^7} \\
&\quad - \frac{4k_1[3\vec{t}_1 \times (P_0 - P_3) + \alpha_0\vec{t}_1 \times \vec{t}_0]}{\alpha_1^3} + \frac{12k_1\alpha_1[3\vec{t}_1 \times (P_0 - P_3) + \alpha_0\vec{t}_1 \times \vec{t}_0]}{\alpha_1^4} \\
&= \frac{-16[3\vec{t}_1 \times (P_0 - P_3) - \alpha_0\vec{t}_0 \times \vec{t}_1]^2}{\alpha_1^5} + \frac{8k_1[3\vec{t}_1 \times (P_0 - P_3) - \alpha_0\vec{t}_0 \times \vec{t}_1]}{\alpha_1^3}
\end{aligned}$$

Thus, the four partials are given by

$$\begin{aligned}
\frac{\partial}{\partial \alpha_0} \left(\frac{(f'(0) \times f''(0))^2}{|f'(0)|^6} - \frac{2k_0(f'(0) \times f''(0))}{|f'(0)|^3} + k_0^2 \right) = \\
\frac{-16[3\vec{t}_0 \times (P_3 - P_0) - \alpha_1\vec{t}_0 \times \vec{t}_1]^2}{\alpha_0^5} + \frac{8k_0[3\vec{t}_0 \times (P_3 - P_0) - \alpha_1\vec{t}_0 \times \vec{t}_1]}{\alpha_0^3}
\end{aligned} \tag{11}$$

$$\begin{aligned}
\frac{\partial}{\partial \alpha_1} \left(\frac{(f'(0) \times f''(0))^2}{\alpha_0^6} - \frac{2k_0(f'(0) \times f''(0))}{\alpha_0^3} + k_0^2 \right) = \\
\frac{-8[3\vec{t}_0 \times (P_3 - P_0) - \alpha_1\vec{t}_0 \times \vec{t}_1][\vec{t}_0 \times \vec{t}_1]}{\alpha_0^4} + \frac{4k_0[\vec{t}_0 \times \vec{t}_1]}{\alpha_0^2}
\end{aligned} \tag{12}$$

$$\begin{aligned}
\frac{\partial}{\partial \alpha_0} \left(\frac{(f'(1) \times f''(1))^2}{\alpha_1^6} - \frac{2k_1(f'(1) \times f''(1))}{\alpha_1^3} + k_1^2 \right) = \\
\frac{-8[3\vec{t}_1 \times (P_0 - P_3) - \alpha_0\vec{t}_0 \times \vec{t}_1][\vec{t}_0 \times \vec{t}_1]}{\alpha_1^4} + \frac{4k_1[\vec{t}_0 \times \vec{t}_1]}{\alpha_1^2}
\end{aligned} \tag{13}$$

$$\begin{aligned}
\frac{\partial}{\partial \alpha_1} \left(\frac{(f'(1) \times f''(1))^2}{\alpha_1^6} - \frac{2k_1(f'(1) \times f''(1))}{\alpha_1^3} + k_1^2 \right) = \\
\frac{-16[3\vec{t}_1 \times (P_0 - P_3) - \alpha_0\vec{t}_0 \times \vec{t}_1]^2}{\alpha_1^5} + \frac{8k_1[3\vec{t}_1 \times (P_0 - P_3) - \alpha_0\vec{t}_0 \times \vec{t}_1]}{\alpha_1^3}
\end{aligned} \tag{14}$$

To summarize, I used the optimizer of Mahdavi-Amiri and Bartels [MAB89] to build cubic curves by optimizing Equations 8, 9, and 10 using the first derivatives of these equations with respect to α_0 and α_1 (Equations 11-14) to drive the optimizer. Examples of the resulting curves can be seen in the right hand column of Figure 2. While these curves do not match the end point curvatures exactly, their end point curvatures are close to the desired curvatures, and none of the derivatives at the ends of the curves is short. Further, the curvature distribution throughout the curves is more uniform.

I integrated this curve construction into the surface construction technique as follows: For each $ijk \in \{PQR, QRP, RPQ\}$,

1. Pick a plane through V_j and V_k .
2. Construct the boundary $V_j, E_{i1}^3, E_{i2}^3, V_k$ using the curve construction technique detailed in this section.

Once we have constructed all the E_{ij} , we construct the interior control points using the remaining formulas given in Section 2.

The right most surface in Pictures 1-3 show the surface constructed by using these optimized curves in the Shirman-Séquin scheme. Several of the shape defects have disappeared, and no lumps have appeared. However, while this is a significant improvement, there are still several “wrinkles” visible in the surface. To attempt to remove these shape defects, we will look at adjusting other shape parameters in the next section.

4 Internal Degrees of Freedom

Internally, there are several degrees of freedom. There are degrees of freedom in constructing the cross boundary derivatives and in constructing the interior boundary curves. We would like to vary the degrees of freedom influencing the cross boundary derivatives to improve the shape of the surface patch. One choice is to use the degrees of freedom to improve the order of approximation. Another idea is to vary these degrees of freedom and try to further match the second fundamental forms at the data points. As discussed in this section, neither of these approaches works in this setting.

Using the Shirman-Séquin settings of the α s, β s, and c s, the construction discussed in Section 2 has linear precision (see Appendix A for details). If we construct boundary curves with quadratic precision, we could hope to achieve quadratic precision. However, using the Shirman-Séquin setting of the β s and c s does not have quadratic precision. Further, even if we use Jensen’s generalization of the crossboundaries, we are unable to achieve quadratic precision (Appendix B).

A second approach is to use these degrees of freedom to match the second fundamental forms at the data points. To match normal section curvature, the following equation must hold:

$$k_0 = \frac{\langle N_0, f''(0) \rangle}{\langle f'(0), f'(0) \rangle}.$$

If we assume the tangent direction \hat{t} is fixed, then we know that

$$C_1 = C_0 + 3\beta\hat{t}$$

We have three scalar degrees of freedom (the β s) to play with to try and match three curvature values. We’ll probably want to restrict the β_i in some way so they don’t get too long or too short. In any case, we have the following relationship between the second control point on the interior curve and the known control points:

$$I_{i2}^3 = \frac{3\beta_i}{16}V_i + \left(1 - \frac{27\beta_i}{16}\right)I_{i1}^3 + \frac{12\beta_i}{16}(C_{ji} + C_{ki})$$

We should expand I_{i1}^3 in this equation, since we wish to have an expression for I_{i2}^3 in terms of β_i and the fixed control points, but as we will see below, I_{i1} will cancel in the equations we minimize below.

We wish to minimize the following:

$$\begin{aligned} \left(\frac{\langle N_i, f_i''(0) \rangle}{\langle f_i'(0), f_i'(0) \rangle} - k_i \right)^2 &= \left(\frac{\langle N_i, 6(I_{i2}^3 - 2I_{i1}^3 + V_i) \rangle}{9\beta_i^2} - k_i \right)^2 \\ &= \left(\frac{2 \langle N_i, I_{i2}^3 - I_{i1}^3 \rangle}{3\beta_i^2} - k_i \right)^2 \end{aligned}$$

To further simplify this term, we need an expression for $\langle N_i, I_{i2}^3 - I_{i1}^3 \rangle$. The primary step used to simplify the equations is that $\langle N, P \rangle = \langle N, Q \rangle$ when $\langle N, P - Q \rangle = 0$. Thus, we see that

$$\begin{aligned} \langle N_i, I_{i2}^3 - I_{i1}^3 \rangle &= \langle N_i, \left[\frac{3\beta_i}{16} V_i + \left(1 - \frac{27\beta_i}{16} \right) I_{i1}^3 + \frac{12\beta_i}{16} (C_{ji} + C_{ki}) \right] - I_{i1}^3 \rangle \\ &= \langle N_i, \frac{3\beta_i}{16} (V_i - I_{i1}^3) \rangle + \langle N_i, \left(1 - \frac{27\beta_i}{16} \right) (I_{i1}^3 - I_{i1}^3) \rangle \\ &\quad + \langle N_i, \frac{12\beta_i}{8} \left(\frac{C_{ji} + C_{ki}}{2} - I_{i1}^3 \right) \rangle \\ &= \frac{3\beta_i}{2} \langle N_i, \left(\frac{C_{ji} + C_{ki}}{2} - I_{i1}^3 \right) \rangle \\ &= \frac{3\beta_i}{2} \langle N_i, \left(\frac{C_{ji} + C_{ki}}{2} - [(1 - \beta_i)P_i + \beta_i(E_{k2}^3 + E_{j1}^3)/2] \right) \rangle \\ &= \frac{3\beta_i}{2} \langle N_i, (1 - \beta_i) \left(\frac{C_{ji} + C_{ki}}{2} - P_i \right) \rangle \\ &\quad + \frac{3\beta_i}{2} \langle N_i, \beta_i \left(\frac{C_{ji} + C_{ki}}{2} - (E_{k2}^3 + E_{j1}^3)/2 \right) \rangle \\ &= \frac{3\beta_i(1 - \beta_i)}{4} \langle N_i, (C_{ji} + C_{ki} - 2P_i) \rangle \\ &\quad + \frac{3\beta_i^2}{4} \langle N_i, (C_{ji} + C_{ki} - E_{k2}^3 - E_{j1}^3) \rangle \\ &= \frac{3\beta_i(1 - \beta_i)}{4} \langle N_i, (C_{ji} + C_{ki} - 2P_i) \rangle \\ &\quad + \frac{3\beta_i^2}{4} \langle N_i, (C_{ji} + C_{ki} - 2P_i) \rangle \\ &= \frac{3\beta_i}{4} \langle N_i, (C_{ji} + C_{ki} - 2P_i) \rangle \\ &= \frac{3\beta_i}{4} \langle N_i, \frac{1}{3} \{ k_{ji} \widehat{C}_{jk} + c_j (k_{jk} + k_{ji}) \widehat{C}_{ji} + h_{jk} (V_i - E_{j2}^3) \\ &\quad + 2h_{ji} (E_{j2}^3 - E_{j1}^3) \} + E_{j2}^3 \\ &\quad + \frac{1}{3} \{ c_k (k_{ki} + k_{kj}) \widehat{C}_{ki} + k_{ki} \widehat{C}_{kj} + 2h_{ki} (E_{k2}^3 - E_{k1}^3) \\ &\quad + h_{kj} (E_{k1}^3 - V_i) \} + E_{k1}^3 - 2P_i \rangle \end{aligned}$$

$$\begin{aligned}
&= \frac{3\beta_i}{4} \langle N_i, \frac{1}{3}\{k_{ji}\widehat{C}_{jk} + 2h_{ji}(E_{j2}^3 - E_{j1}^3)\} + \\
&\quad \frac{1}{3}\{k_{ki}\widehat{C}_{kj} + 2h_{ki}(E_{k2}^3 - E_{k1}^3)\} \rangle \\
&= \frac{3\beta_i}{4} \langle N_i, \frac{1}{3}(\beta_i\widehat{k}_{ji}\widehat{C}_{jk} + 2\beta_i\widehat{h}_{ji}(E_{j2}^3 - E_{j1}^3) + \\
&\quad \beta_i\widehat{k}_{ki}\widehat{C}_{kj} + 2\beta_i\widehat{h}_{ki}(E_{k2}^3 - E_{k1}^3)) \rangle \\
&= \frac{\beta_i^2}{4} \langle N_i, \widehat{k}_{ji}\widehat{C}_{jk} + 2\widehat{h}_{ji}(E_{j2}^3 - E_{j1}^3) + \widehat{k}_{ki}\widehat{C}_{kj} + 2\widehat{h}_{ki}(E_{k2}^3 - E_{k1}^3) \rangle \\
&= \frac{\beta_i^2}{4} \langle N_i, \mathcal{X} \rangle
\end{aligned}$$

where

$$\mathcal{X} = \widehat{k}_{ji}\widehat{C}_{jk} + 2\widehat{h}_{ji}(E_{j2}^3 - E_{j1}^3) + \widehat{k}_{ki}\widehat{C}_{kj} + 2\widehat{h}_{ki}(E_{k2}^3 - E_{k1}^3).$$

Note that \mathcal{X} is independent of both the β_i and the c_i .

Substituting into the equation we wish to minimize, we see

$$\begin{aligned}
\left(\frac{\langle N_i, f_i''(0) \rangle}{\langle f_i'(0), f_i'(0) \rangle} - k_i\right)^2 &= \left(\frac{\langle N_i, I_{i2}^3 - I_{i1}^3 \rangle}{3\beta_i^2} - k_i\right)^2 \\
&= \left(\frac{\langle N_i, \mathcal{X} \rangle}{12} - k_i\right)^2.
\end{aligned}$$

Thus, we see that the normal section curvatures at the V_i endpoint of the internal boundary curves are independent of the β_i and the c_i .

Now let's look at the mixed partial derivatives at the corners. There will be six of these we wish to satisfy. For now, assume that we have a goal value for the mixed partials. Then

$$\begin{aligned}
(\langle N_p, D_{uv}P(i) \rangle - m_i)^2 &= \langle N_i, \frac{D_{uv}P(i)}{|D_uP||D_vP|} \rangle^2 - 2m_i \langle N_i, \frac{D_{uv}P(i)}{|D_uP||D_vP|} \rangle + m_i^2 \\
&= \frac{36}{81\beta_i^2|I_{i1} - V_i|^2} \langle N_i, C_{ji} - E_{j2}^3 + I_{i1}^3 - V_i \rangle^2 \\
&\quad - \frac{12m_i}{9\beta_i|I_{i1} - V_i|} \langle N_i, C_{ji} - E_{j2}^3 + I_{i1}^3 - V_i \rangle + m_i^2 \\
&= \frac{36 \langle N_i, C_{ji} - E_{j1}^3 \rangle^2}{81\beta_i^2|I_{i1} - V_i|^2} - \frac{12m_i \langle N_i, C_{ji} - E_{j2}^3 \rangle}{9\beta_i|I_{i1} - V_i|} + m_i^2
\end{aligned}$$

Again, we look at $\langle N_i, C_{ji} - E_{j2}^3 \rangle$:

$$\begin{aligned}
\langle N_i, C_{ji} - E_{j2}^3 \rangle &= \langle N_i, \frac{1}{3}(k_{ji}\widehat{C}_{jk} + c_j(k_{jk} + k_{ji})\widehat{C}_{ji} + h_{jk}(V_i - E_{j2}^3) + 2h_{ji}(E_{j2}^3 - E_{j1}^3)) \\
&\quad + E_{j2}^3 - E_{j2}^3 \rangle \\
&= \frac{1}{3} \langle N_i, k_{ji}\widehat{C}_{jk} + 2h_{ji}(E_{j2}^3 - E_{j1}^3) \rangle
\end{aligned}$$

$$\begin{aligned}
&= \frac{\beta_i}{3} \langle N_i, \widehat{k}_{ji}\widehat{C}_{jk} + 2\widehat{h}_{ji}(E_{j2}^3 - E_{j1}^3) \rangle \\
&= \frac{\beta_i}{3} \langle N_i, \mathcal{Y} \rangle,
\end{aligned}$$

where

$$\mathcal{Y} = \widehat{k}_{ji}\widehat{C}_{jk} + 2\widehat{h}_{ji}(E_{j2}^3 - E_{j1}^3).$$

And looking at the equation to minimize, we again see that it is independent of our minimization variables β_i and c_i :

$$\begin{aligned}
(\langle N_p, D_{uv}P(i) \rangle - m_i)^2 &= \langle N_i, \frac{D_{uv}P(i)}{|u||v|} \rangle^2 - 2m_i \langle N_i, \frac{D_{uv}P(i)}{|u||v|} \rangle + m_i^2 \\
&= \frac{36 \langle N_i, C_{ji} - E_{j1}^3 \rangle^2}{81\beta_i^2|I_{i1} - V_i|^2} - \frac{12m_i \langle N_i, C_{ji} - E_{j2}^3 \rangle}{9\beta_i|I_{i1} - V_i|} + m_i^2 \\
&= \frac{36\beta_i^2 \langle N_i, \mathcal{Y} \rangle^2}{9 \cdot 81\beta_i^2|I_{i1} - V_i|^2} - \frac{12m_i\beta_i \langle N_i, \mathcal{Y} \rangle}{27\beta_i|I_{i1} - V_i|} + m_i^2 \\
&= \frac{4 \langle N_i, \mathcal{Y} \rangle^2}{81|I_{i1} - V_i|^2} - \frac{4m_i \langle N_i, \mathcal{Y} \rangle}{9|I_{i1} - V_i|} + m_i^2
\end{aligned}$$

5 Conclusions and Future Work

We have seen that improved settings of degrees of freedom in a G^1 surface fitting scheme yields a surface of better quality. Thus far, we have only found improved settings for degrees of freedom on the boundaries of the surface patches. Improved settings for the remaining degrees of freedom should be found (the β_i and the c_i of the previous section).

The degrees of freedom along the boundary were set by matching second fundamental forms at the data points. We have shown that the internal degrees of freedoms are independent of such data. Thus, an alternate way of setting them needs to be found. Further, we would prefer to construct surfaces for a triangulated set of points without normals and second fundamental forms specified at these points. Techniques that do not require such data should also be investigated.

A Proof of reproduction of linear patches

In this section, I will show that Shirman-Séquin's scheme reproduces linear surfaces. The data to be interpolated are three points with normals perpendicular to the plane containing the three points. Without loss of generality, I will assume that these three points lie in the x-y plane and that the normals are parallel to the z axis.

I will assume that the boundary curve construction algorithm reproduces linear boundaries for this data. Thus, the V_i and E_{in} are set with linear precision. This is

equivalent to saying the α_{in} are one third the length of the chord V_iV_j . Further note that the Shirman-Séquin scheme sets $\beta_i = 1/3$ and $c_i = 1$. Then using Equation 1 sets the I_{i1} with linear precision.

Now note that choosing any specific instance of the problem shows that the control points are set with linear precision for that particular instance. If the entire construction were affine, then we would be done. However, the Chiyokura-Kimura's technique uses perpendicular vectors, a non-affine construction. Thus, we need to show that the placement of the C_{ij} are made with linear precision.

Consider the Chiyokura-Kimura construction along one edge. First note that \widehat{C}_{ij} is equal to \widehat{C}_{ik} since the end tangents of the boundary are parallel. This also means that k_{ij} and k_{ik} are equal since the I_{j1} and I_{k1} are the same distance from the edge V_jV_k . Further note that

$$(E_{i1}^3 - V_i) = (E_{i2}^3 - E_{i1}^3) = (V_j - E_{i2}^3).$$

Then the Equations 2 and 3 simplify to

$$C_{ij} = k_{ij}\widehat{C}_{ij} + (2h_{ij} + h_{ik})(E_{i1}^3 - V_i)/3 + E_{i1}^3$$

and

$$C_{ik} = k_{ij}\widehat{C}_{ij} + (2h_{ik} + h_{ij})(E_{i1}^3 - V_i)/3 + E_{i2}^3.$$

We know that if our patch is linear, then $C_{ij} = I_{i1}^4 + (I_{j1} - I_{i1})/3$. The following derivation shows that this is the setting used by Shirman-Séquin:

$$\begin{aligned} I_{i1}^4 + (I_{j1} - I_{i1})/3 &= V_i + k_{ij}\widehat{C}_{ij} + h_{ij}\overline{E} + (V_j + h_{ik}\overline{E} - V_i - h_{ij}\overline{E})/3 \\ &= E_{i1} + k_{ij}\widehat{C}_{ij} + (2h_{ij} + h_{ik})(E_{i1} - V_i)/3. \end{aligned}$$

A similar derivation shows that C_{ik} is set with linear precision.

Since the remaining equations are affine, and we have one instance of the problem where the control points are set with linear precision, we have shown that Shirman-Séquin's scheme has linear precision.

A.1 Specific Example

In this example, our data points are $(0, 0, 0)$, $(1, 0, 0)$, and $(0, 1, 0)$. All three normals are $(0, 0, 1)$. Shirman-Séquin constructs the following three patches, which are seen to be linear:

$$\begin{array}{ccccc} (1, 0, 0) & (0.75, 0, 0) & (0.5, 0, 0) & (0.25, 0, 0) & (0, 0, 0) \\ (0.8\overline{3}, 0.08\overline{3}, 0) & (0.58\overline{3}, 0.08\overline{3}, 0) & (0.\overline{3}, 0.08\overline{3}, 0) & (0.08\overline{3}, 0.08\overline{3}, 0) & \\ (0.6\overline{6}, 0.1\overline{6}, 0) & (0.41\overline{6}, 0.1\overline{6}, 0) & (0.1\overline{6}, 0.1\overline{6}, 0) & & \\ (0.50, 0.25, 0) & (0.25, 0.25, 0) & & & \\ (0.\overline{3}, 0.\overline{3}, 0) & & & & \end{array}$$

$(1, 0, 0)$ $(0.8\bar{3}, 0.08\bar{3}, 0)$ $(0.\bar{6}, 0.1\bar{6}, 0)$ $(0.5, 0.25, 0)$ $(0.\bar{3}, 0.\bar{3}, 0)$
 $(0.75, 0.25, 0)$ $(0.58\bar{3}, 0.\bar{3}, 0)$ $(0.41\bar{6}, 0.41\bar{6}, 0)$ $(0.25, 0.5, 0)$
 $(0.5, 0.5, 0)$ $(0.\bar{3}, 0.58\bar{3}, 0)$ $(0.1\bar{6}, 0.\bar{6}, 0)$
 $(0.25, 0.75, 0)$ $(0.08\bar{3}, 0.8\bar{3}, 0)$
 $(0, 1, 0)$

$(0.\bar{3}, 0.\bar{3}, 0)$ $(0.25, 0.25, 0)$ $(0.1\bar{6}, 0.1\bar{6}, 0)$ $(0.08\bar{3}, 0.08\bar{3}, 0)$ $(0, 0, 0)$
 $(0.25, 0.5, 0)$ $(0.1\bar{6}, 0.41\bar{6}, 0)$ $(0.08\bar{3}, 0.\bar{3}, 0)$ $(0, 0.25, 0)$
 $(0.1\bar{6}, 0.\bar{6}, 0)$ $(0.08\bar{3}, 0.58\bar{3}, 0)$ $(0, 0.5, 0)$
 $(0.08\bar{3}, 0.8\bar{3}, 0)$ $(0, 0.75, 0)$
 $(0, 1, 0)$

B Proof that reproduction of quadratics is not possible

The easiest way to show that our version of the modified Shirman-Séquin scheme can not reproduce quadratics is by counter example. For this question to even make sense, we will assume that the boundary curve construction technique does reproduce quadratics. I will then give a quadratic for which neither Chiyokura-Kimura's scheme nor Jensen's scheme can reproduce the cross boundary behavior.

Consider the quadratic patch given by the following control points:

$(0., 0., 0.)$ $(1.5, 0., 3.)$ $(3., 0., 0.)$
 $(0.75, 2., 0.)$ $(2.25, 2., 0.)$
 $(1.5, 4., 0.)$

If we degree raise this patch twice (to fourth degree), then " \widehat{C}_{ij} " and " \widehat{C}_{ik} "² are given by $(1.125, 1, 1)$ and $(1.875, 1, 1)$ while Chiyokura-Kimura's construction yields the points $(1.125, 1, 1.8)$ and $(1.875, 1, 1.8)$. This is for the boundary between the points $(0, 0, 0)$ and $(3, 0, 0)$. Now note that if we vary Jensen's shape parameter, we will be changing these points by scale factors of the vectors $(.28, 0.95, -0.14)$ and $(-.28, .95, -.14)$ respectively, and thus we can not reproduce the cross boundary behavior of this quadratic with Jensen's crossboundary scheme.

²I put the names of these control points in quotes because here I'm only running the Chiyokura-Kimura construction and not the Shirman-Séquin construction; however, it is these quoted points that Chiyokura-Kimura construct in the Shirman-Séquin scheme. The above example could be reworked into the complete Shirman-Séquin setting if desired.

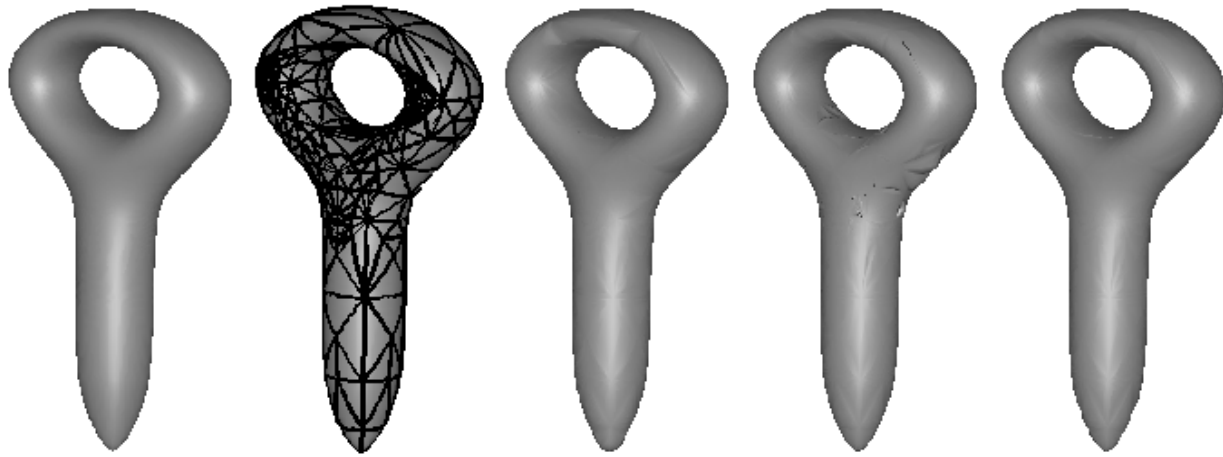
References

- [CG91] George Celniker and Dave Gossard. Deformable curve and surface finite-elements for free-form shape design. In *SIGGRAPH 1991*, July 1991.
- [CK83] Hiroaki Chiyokura and Fumihiko Kimura. Design of solids with free-form surfaces. *Computer Graphics*, 17(3):289–298, 1983.
- [dBHS87] Carl de Boor, Klaus Höllig, and Malcolm Sabin. High accuracy geometric Hermite interpolation. *Computer Aided Geometric Design*, 4(4):269–278, December 1987.
- [DM92] Tony DeRose and Stephen Mann. An approximately G^1 cubic surface interpolant. In T. Lyche and L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design II*. Academic Press, 1992.
- [Far93] Gerald Farin. *Curves and Surfaces for CAGD*. Academic Press, 1993.
- [Jen87] Thomas Jensen. Assembling triangular and rectangular patches and multivariate splines. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 203–220. SIAM, 1987.
- [Kla83] Reinhold Klass. An offset spline approximation for plane cubic splines. *Computer Aided Design*, 15(5):297–299, September 1983.
- [LD89] Charles T. Loop and Tony D. DeRose. A multisided generalization of Bézier surfaces. *TOG*, 8(3):204–234, July 1989.
- [MAB89] Nezam Mahdavi-Amiri and Richard H. Bartels. Constrained nonlinear least squares: An exact penalty approach with projected structured quasi-newton updates. *ACM Transactions on Mathematical Software*, 15(3):220–242, September 1989.
- [Man92] Stephen Mann. *Surface Approximation Using Geometric Hermite Patches*. PhD thesis, University of Washington, 1992.
- [MLL⁺92] Stephen Mann, Charles Loop, Michael Lounsbery, David Meyers, James Painter, Tony DeRose, and Kenneth Sloan. A survey of parametric scattered data fitting using triangular interpolants. In H. Hagen, editor, *Curve and Surface Design*, chapter 8, pages 145–172. SIAM, 1992.
- [MS92] Henry P. Moreton and Carlo H. Séquin. Functional optimization for fair surface design. In *SIGGRAPH 1992*, July 1992.
- [SS87] Leon A. Shirman and Carlo H. Séquin. Local surface interpolation with Bézier patches. *Computer Aided Geometric Design*, 4(4):279–295, December 1987.

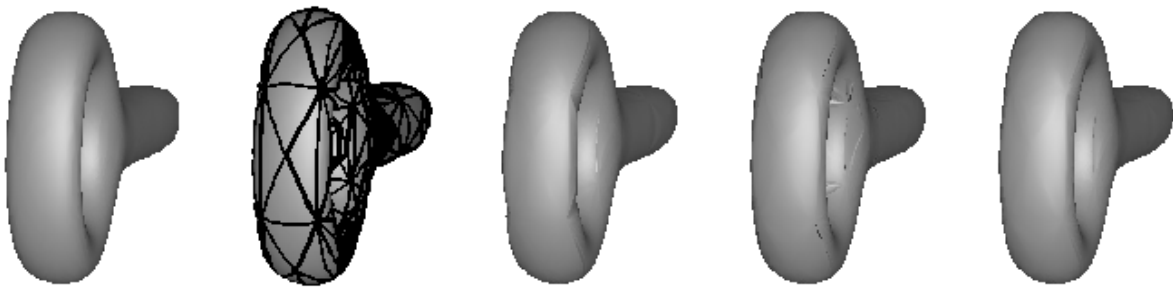
- [SS91] Leon A. Shirman and Carlo H. Séquin. Local surface interpolation with Bézier patches: errata and improvements. *Computer Aided Geometric Design*, 8(3):217–221, August 1991.

The Computer Graphics Laboratory at the University of Waterloo is a group of students and faculty doing research within the Computer Science Department. Interests include most areas of computer graphics, document preparation and man-machine interfaces. CGL is affiliated with the Institute for Computer Research. Graduate work at the masters and doctoral level leads to a Computer Science degree from the Faculty of Mathematics. Further information may be obtained by writing to the following address:

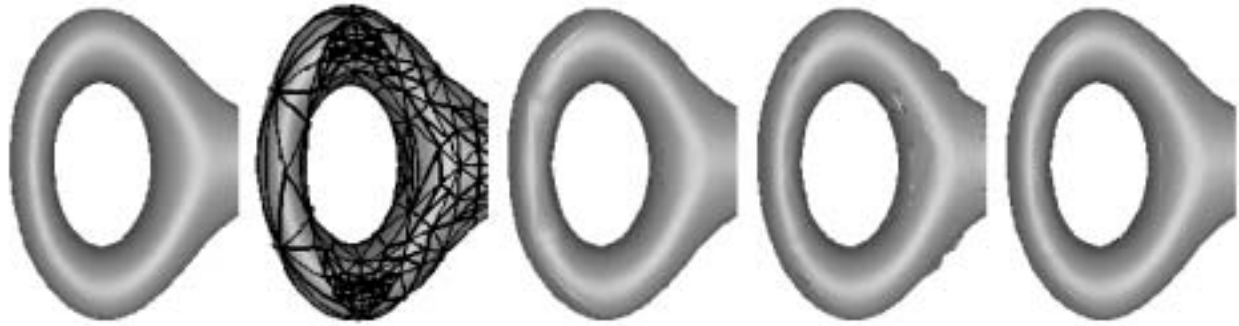
*Computer Graphics Laboratory
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
(519) 888-4534*



Picture 1



Picture 2



Picture 3