Triangular B-Splines for Blending and Filling of Polygonal Holes

Ron Pfeifle
Alias|Wavefront
110 Richmond St. E.
Toronto, Ontario
Canada, M5P 1C1
email: rpfeifle@aw.sgi.com

Hans-Peter Seidel
Computer Graphics Group
University of Erlangen
Am Weichselgarten 9
D-91058 Erlangen, Germany
email: seidel@informatik.uni-erlangen.de

Abstract

Triangular B-splines lend themselves naturally to problems such as blending and the filling of polygonal holes. Here we present an automatic method for smoothly blending piecewise polynomial surfaces using triangular B-splines.

The method proceeds in two phases. In the first phase, the domain of the region to be blended or filled is triangulated and populated with basis functions. In the second phase, coefficients for these basis functions are found by minimizing a functional that measures the curvature of the blending or filling surface.

Examples are provided that show the use of this method for a number of blending and filling problems.

Keywords: Blending, Filling Polygonal Holes, DMS splines, Triangular B-splines, Surface Smoothing

1 Introduction

Triangular B-splines or DMS splines [DMS92] are a new tool for the modeling of complex objects with non-rectangular topology. Since any piecewise polynomial can be represented in the new scheme, DMS splines offer themselves naturally for problems such as blending and the filling of polygonal holes. Although this fact has been pointed out before, all previous solutions have required considerable user-interaction, and an automatic algorithm for blending and/or the filling of polygonal holes based on DMS splines has not been available.

In this paper, we present an automatic method for solving blending and filling problems using DMS splines. We presume that the surfaces that are to be blended or filled are already expressed in DMS form, with appropriate domain triangulations. There is no restriction, since every piecewise polynomial can be represented as a DMS spline [Sei91]. The method proceeds in two phases. In the first phase, the triangulation of the domain of the existing surface or surfaces is extended to encompass the region of the domain corresponding to the intended blend or fill. Vertices for the extended triangulation are then found with the help of techniques from the finite-elements literature. New vertices are successively added and the Delaunay triangulation of those points formed. Together with the automatic assignment of knot clouds, this procedure yields the basis functions of the new blending or filling surface.

In the second phase, the control points of the new surface are selected in such a way as to smoothly connect the new surface with preexisting patches and to minimize surface curvature. The control points are chosen such that a quadratic functional approximating the curvature of the surface is minimized. The minimization problem can thus be expressed as a system of linear equations and solved by standard methods.

The paper is organized as follows: Section 2 reviews the definitions of bivariate DMS splines and highlights their major attributes. Section 3 shows how we construct the extended triangulation needed to form a blending or filling surface. Section 4 describes the minimization problem used to set the values of new control points arising from the extended triangulation. Section 5 summarizes the entire procedure and gives examples showing the use of this technique for a number of blending and filling problems. Finally, we present our conclusions and suggestions for further work in Section 6.

2 Triangular B-Splines

The triangular B-splines of [DMS92] are a flexible extension of B-spline curves to the surface case. These surfaces offer properties such as affine invariance and local control, and exhibit the Convex Hull Property. Moreover, every degree n piecewise polynomial over a triangulation T can be represented as a DMS spline surface.

The following discussion reviews the mathematics of DMS splines. A broader introduction can be found in [Sei91].

2.1 Simplex Splines

Simplex splines [Mic79] form the individual basis functions for the DMS spline surface. A degree n simplex spline is defined recursively as follows: Let u and *knots* $t_0, \ldots, t_{n+2}$ be points in $\mathbb{R}^2$, and let V be the set $\{t_0, \ldots, t_{n+2}\}$. Furthermore, select an affinely independent set $W = \{t_{i_0}, t_{i_1}, t_{i_2}\}$ from V. Then, for $n > 0$, the degree n simplex spline $M(u|V)$ is defined as

$$M(u|V) = \sum_{j=0}^{2} \lambda_j(u|W) M(u|V \backslash \{t_{i_j}\}) \qquad (1)$$

where $\lambda_j(u|W)$ are the barycentric coordinates of u with respect to W.

Let $d(t_i, t_j, t_k)$ be twice the signed area of $\Delta(t_i, t_j, t_k)$. When $n = 0$, we define

$$M(u|t_0, t_1, t_2) = \frac{\chi_{[t_0,t_1,t_2)}(u)}{|d(t_0, t_1, t_2)|} \qquad (2)$$

where $\chi_{[t_0,t_1,t_2)}$ is the characteristic function on the *half-open convex hull* [1] $[t_0, t_1, t_2)$.

2.2 DMS Splines

A DMS spline surface [DMS92] is formed by considering the vertices of a triangulation augmented with additional points, *knot clouds*, from which we build a collection of simplex splines to act as basis functions for our surface. Let T be an arbitrary proper triangulation of some bounded domain $D \subset \mathbb{R}^2$, with vertices $t_0, \ldots, t_l$. "Proper" means that every pair of domain triangles are either disjoint, share exactly one edge, or share exactly one vertex.

A sequence of knots $t_{i,0}, \ldots, t_{i,n}$ called a *knot cloud* is assigned to each vertex t_i, with $t_{i,0} \equiv t_i$. For each triangle $\Delta(t_0, t_1, t_2) \in T$, we place the restriction that $(t_{0,i}, t_{1,j}, t_{2,k})$ must always form a proper triangle. We then define, for each Δ, the knot sets

$$V^{\Delta}_{ijk} = \{t_{0,0}, \ldots, t_{0,i}, t_{1,0}, \ldots, t_{1,j}, t_{2,0}, \ldots, t_{2,k}\} \quad (3)$$

[1] u is in $[t_0, t_1, t_2)$ if we can find a wedge-shaped triangle $\Delta(u, u + (0, s), u + (t, 0))$, $s, t > 0$ that is contained within the convex hull of $\{t_0, t_1, t_2\}$ [Sei91].

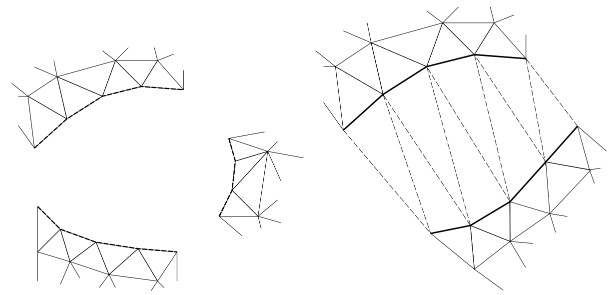

Figure 1: Issues involved in triangulating a blend or fill region. Left, specifying which edges to blend only defines part of the parametric boundary of the region. Right, using only boundary vertices to extend a triangulation can yield elongated triangles.

where $i + j + k = n$, which yields the simplex splines $M(u|V^{\Delta}_{ijk})$.

If we let d^{Δ}_{ijk} be $d(t_{0,i}, t_{1,j}, t_{2,k})$, then the *normalized B-splines* are defined as $N^{\Delta}_{ijk}(u) = d^{\Delta}_{ijk} M(u|V^{\Delta}_{ijk})$. Each $N^{\Delta}_{ijk}(u)$ is a DMS basis function, and normalization ensures that they sum to one.

A degree n DMS spline surface F over triangulation T is then defined as

$$F(u) = \sum_{\Delta \in T} \sum_{i+j+k=n} c^{\Delta}_{ijk} N^{\Delta}_{ijk}(u), \quad c^{\Delta}_{ijk} \in \mathbb{R}^3. \quad (4)$$

3 Triangulating the Blend or Hole Region

We begin with the assumption that the surfaces adjacent to the region to be filled and/or blended are already expressed in triangular B-spline form, with corresponding domain triangulations. This does not restrict the class of surfaces to be blended, since every piecewise polynomial can be represented as a triangular B-spline. The problem then is to automatically extend these triangulations to the as yet untriangulated region. The newly triangulated region should contain triangles that match the scale of the existing triangles. If the new triangles are of the same scale as the older triangles, then the corresponding basis functions will exhibit the same degree of influence on the shape of the final surface. Furthermore, the newly generated triangles must not be overly elongated, since keeping the new triangles "fat" prevents numerical problems.

In order to properly extend the neighbouring regions, the newly triangulated region should share edges, vertices and knot clouds with the given regions along its boundary. We then rely on the mathematical properties of triangular B-splines to ensure maximal parametric continuity between the old surfaces and the new blending/filling surface.

In both blending and hole filling problems, the parametric region to be triangulated must be identified, most appropriately as a boundary polygon. This is the natural form of boundary specification for hole filling problems. A blending problem might, however, specify only some segments of this boundary (Figure 1). In this case, the procedure must complete the boundary before triangulation can continue.

In order to complete the boundary, we join consecutive boundary pieces with a new edge. The resulting polygon defines the parametric boundary of the blend surface. If the length l of the new edge is longer than the average length l_{avg} of the two edges that it joins, then the new edge is split into $\frac{l}{l_{avg}}$ shorter edges, by introducing $\frac{l}{l_{avg}} - 1$ new vertices uniformly into the new edge. This keeps the lengths of newly introduced edges consistent with those already given.

Once the boundary polygon is complete, an extended triangulation could be formed that only uses vertices from that polygon. Such a triangulation is more than likely to contain elongated triangles (Figure 1). Moreover, when individual triangles span the entire gap, there are very few degrees of freedom available for "fairing" the blending/filling surface. Therefore we should introduce new vertices as appropriate into the parametric region to be triangulated.

3.1 Selection of Vertices

Forming triangulations of given boundary polygons is common practice in finite-element analysis, where the physical properties of an object are analyzed by dividing the object into a number of small elements [Cav74, JS86, BWS$^+$87]. We will use the technique outlined in [WH94] to build our triangulation. This technique incrementally adds new vertices while maintaining a Delaunay triangulation [For94]. The Delaunay triangulation maximizes the minimum internal angle of all triangles and thus discourages elongated triangles.

Once the domain polygons encompassing the region to be filled or blended (from now on the *region*) are determined, a simple triangulation is formed using only the vertices of the boundary. A robust method is given in [SRK94]. This initial triangulation is likely to contain triangles that are quite elongated. To obtain a better triangulation we must introduce vertices into the interior of the region and retriangulate.

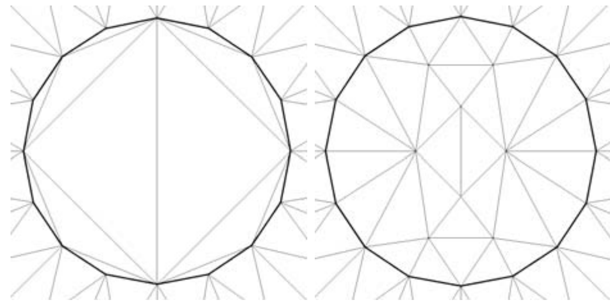

Figure 2: A domain hole is triangulated. On the left, an initial triangulation using only boundary vertices is formed. On the right, the final triangulation.

An appropriate approach is given in [WH94], which incrementally chooses new vertices within the region, and then reforms the Delaunay triangulation. The key idea of this approach is to assign a *scale* to each vertex, which represents the minimum acceptable distance between a vertex and its nearest neighbouring vertices. Whenever a new vertex is considered, a value for this distance is estimated and used to decide whether or not to use the vertex in the triangulation.

The initial vertices of the region are assigned scales that are the average of the lengths of the two boundary edges to which they belong. During each iteration, the centroids of the currently existing triangles are examined and possibly added to a list of candidate vertices. At the end of an iteration step, these candidate vertices are introduced into the triangulation, and the candidate list is cleared. The entire process ends when no more candidates can be introduced (see Figure 2).

The centroid of triangle Δ is accepted as a candidate if it passes the following checks: First, the centroid is assigned a scale that is the average of the scales of the vertices of Δ. The distance from the centroid to any of these vertices must be less than this scale, or else the centroid is rejected. If the distance from the centroid to already accepted candidates is also greater than this scale, the centroid is added as a candidate to the list. At the end of the iteration step, accepted candidates are added to the triangulation, and the Delaunay property is reestablished.

```
do {
    find candidates for insertion
    for each existing triangle Δ
        compute the centroid c_i and scale(c_i)
        for each corner r of Δ,
            verify dist(c_i, r) ≥ α scale(c_i) and
                dist(c_i, r) ≥ α scale(r)
        if verified, add c_i to the list of candidates

    check that new vertices lie far enough apart
    for each candidate c_i
        for each accepted candidate c_j
            if dist(c_i, c_j) ≥ β scale(c_i) and
              dist(c_i, c_j) ≥ β scale(c_j)
                accept candidate c_i

    put the triangulation back into shape
    rebuild triangulation to include accepted candidates
    reestablish the triangulation Delaunay property
} until no more accepted candidates
```

Figure 3: Pseudo-code of the Weatherill/Hasṣan algorithm for refining a triangulation based on vertex density criterion.

The above checks can also be modified by multiplying the distance scale by a variable parameter. In the first check, a factor α can be used, which affects the density of the triangulation created, while in the second check, a factor β can be used to vary the regularity of the triangulation (see [WH94]). Pseudo-code for the algorithm is found in Figure 3. As a final step, the quality of the new triangulation can be improved, using other techniques (see for example, [Cav74, BWS$^+$87]), in order to promote symmetry or other properties that may be desired.

Once the region has been triangulated, the newly inserted vertices may need to be moved in order to prevent them from lying collinearly with knots from preexisting knot clouds. Those vertices which are collinear have their position perturbed in order to ensure that the resulting surface will exhibit maximal parametric continuity between preexisting surfaces and the new surface.

3.2 Defining Knot Clouds

Once the triangulation of the region is complete, the new vertices must be assigned knot clouds. The assignment of knot clouds defines the DMS basis functions for the blending or filling surface. The already existing knots define, in pairs, lines in the parameter space which must be avoided in order to maintain maximal parametric continuity. Knots are placed successively in order to avoid those "forbidden lines," in a manner similar to that given in [AGNS91, page 81].

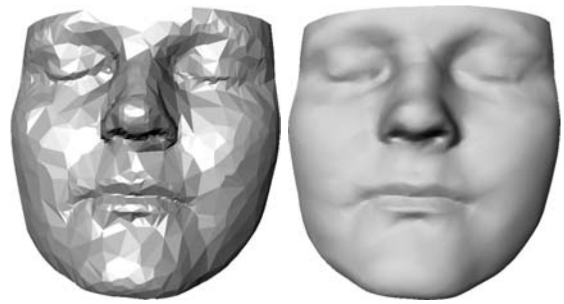

Figure 4: Smoothing a polygonal data set using a fairing functional. Left, the polygonal data. Right, the smoothed surface.

3.3 World Space Relationships

The above triangulation is formed in the parameter space of the spline surface. We assume that the user has assigned appropriate parametric triangulations to the original surfaces before blending[2]. For example, if the surfaces being blended are far apart in world space, then placing their corresponding triangulations close together in parameter space will generate long thin patches in the final blend. Likewise, if the triangular patches of the surfaces to be blended are of similar size in world space, then the user should try to ensure that their triangulations contain triangles of comparable parametric size. In short, the user should take into account the world space relationships of the given surfaces when formulating the blending or filling problem.

4 Fairing by Minimizing a Functional

Fairing is the process of altering a surface to make it smoother. Surface fairing usually proceeds by defining some *fairness functional* for a given set of surfaces, then finding the surface F that minimizes the functional with respect to that set. Numerous functionals have been proposed and used successfully for fairing [MS92, WW92, Gre94a, Gre94b]. Figure 4 shows a polygonal dataset of a face smoothed using

[2]Since DMS surfaces are affinely invariant, affine transformations of the parameter space do not alter the final surface.

a fairing function[3]. If the chosen functional is quadratic then we can reformulate the minimization problem as a linear system, which can then be solved using matrix techniques. Furthermore, a unique minimum is guaranteed to exist if the bilinear form of the functional is positive definite for the given set of basis functions.

The simplest functional that can be used to minimize surface curvature is the *linearized thin plate energy* functional $L(F(x,y))$ [Gre94a, Gre94b, CG91, HKD93, PS95], defined as

$$L(F) = \int_{\Omega} ||F_{xx}||^2 + 2\,||F_{xy}||^2 + ||F_{yy}||^2 \, dxdy. \quad (5)$$

The action of the functional is restricted to the region Ω, which localizes the corresponding smoothing effect. We use the technique given in [PS95] to minimize this functional for quadratic DMS splines. Since we do not extend this technique here, we refer the reader to [PS95] for further details.

The linearized thin plate energy is a good approximation to the true energy only if the given parameterization is close to isometric. In blending applications this is often the case. In more general situations, the linearized thin plate energy has to be replaced by a quadratic functional based on the *Laplace-Beltrami* operator [Gre94b]. Due to the quadratic nature of this functional, its minimum can again be found by solving a linear system, as above.

5 Summary and Examples

We now summarize the steps involved in automatically finding a blending or fill surface. First, the parametric boundary of the blending or filling region is established, and any additional boundary vertices are created as necessary. An extended triangulation is then formed that shares boundary edges with the adjacent regions. Next, new vertices are created within the region and its Delaunay triangulation is formed. Knot clouds are then assigned to each new vertex, defining the DMS basis functions for the blending/filling surface. Finally, the minimization problem is formed for the new basis functions, giving a linear system that is solved using standard techniques from linear algebra. The solution yields the values of the basis function coefficients, completing the defintion of the blending or filling surface.

The following four examples show this technique in practice. The first example fills a circular hole in a truncated cone. In the second example, we construct blends and fills for a five-sided region. Two pipes are blended to form a "tee" joint in the third example. The final example creates a blend between three in-coming pipes and the bottom of a flat basin. In figures where Gaussian curvature is plotted, the colour green indicates zero curvature, blue increasingly positive curvature, and red increasingly negative curvature[4].

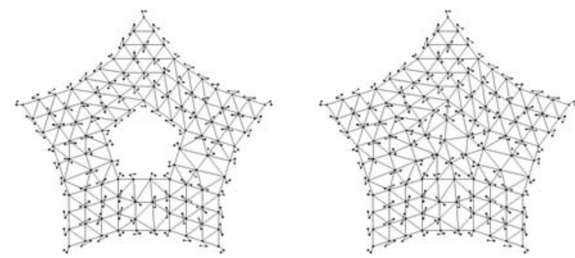

Figure 5: Blending and filling a five-sided hole. Left, the original triangulation extended with three blending triangulations. Right, the completed triangulation of the five-sided hole with knot clouds.

5.1 Truncated Cone

The first example smoothly fills a roughly circular region of a truncated cone. Figure 9, left, shows the cone surface in yellow, and the filling surface in blue-grey. The boundary polygon given to the algorithm consists of the edges and vertices corresponding to the inner rim of the truncated cone. Figure 9, right, shows the Gaussian curvature of the surface. The curvature plot indicates positive curvature over the capped area, and zero curvature over the cone surface.

5.2 Five-Sided Hole

The second example shows the creation of three blend and one filling surface in order to smoothly join three different planar regions, where the final hole to be filled is five-sided. The filling surface must exhibit more complicated geometry in order to provide a smooth fill.

Figure 10, left, shows the five-sided hole, with three blending surfaces (in green) connecting the different original planar regions, and the five-sided hole filled with the blue surface. The Gaussian curvature

[3] Dataset courtesy LNT, Universität Erlangen-Nürnberg.

[4] Colour figures appear on the final page of this document.

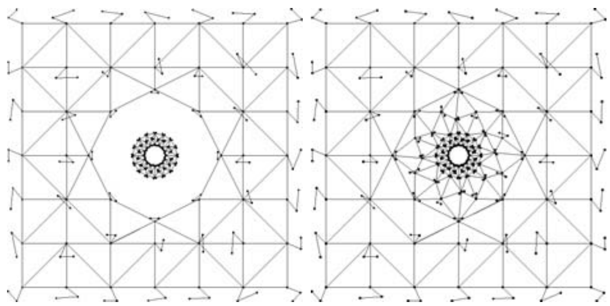

Figure 6: Tee joint triangulation: Left, the original triangulation with a non-convex fill region. Right, the final triangulation.

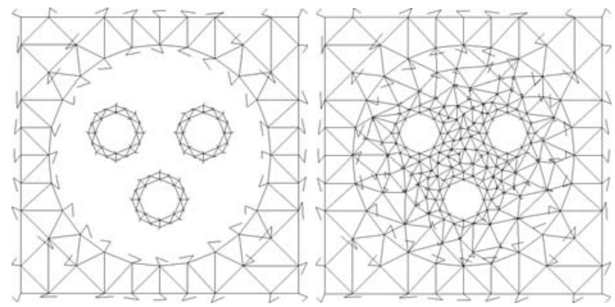

Figure 7: Pipe junction: The original triangulation and the extended triangulation produced by the algorithm.

of the final surface is shown in Figure 10, right. Here we see that curvature does indeed vary over the filling surface, with regions of negative curvature near the centre, and flatter regions near the upper surface. Figure 5 shows the triangulation of the original planar surfaces augmented first with the extended triangulations of the blend surfaces, then with the extended triangulation of the filling surface.

5.3 Tee Joint

In this example, a vertical tube is connected to a hole in a horizontal tube to form a single surface in the form of a tee joint. Figure 8, left, shows the two tubes. The horizontal pipe contains an opening that will be filled by the blending surface. Figure 8, middle, shows the two tubes with their blending surface in blue, and Figure 8, right, plots the Gaussian curvature over the blend surface.

Figure 6 shows the opening triangulation along with the extended triangulation computed by the algorithm. In contrast to the previous examples, the parametric region underlying the blended surface is not convex.

5.4 Pipe Junction

The final example illustrates triangulation and blending over a more complicated non-convex region. Here, three pipes come together to join to the flat bottom of a basin. Figure 11, left, shows the view with each of the three pipes approaching the basin. Figure 11, right, shows the results of the blending operation. In the parameter domain, Figure 7 shows the starting triangulation and the extended triangulation computed by the algorithm. As in the previous example, the parametric region being blended is not convex.

6 Conclusions

We have demonstrated an automatic method for forming blending surfaces and filling polygonal holes using DMS splines. The method starts by extending the surface definitions to the region to be blended or filled and then establishes control points for the extended surface by minimizing a functional of the blending or filling surface that represents the amount of surface curvature. The procedure works for both convex and non-convex domains.

A number of problems remain for further study. Methods should be developed to assist the user in taking world space issues into account when assigning the parametric triangulations. The triangulation algorithm should be adjusted to take into consideration the locations of preexisting knot clouds when new vertices are introduced. Finally, the use of more sophisticated fairing functions, such as the Laplace-Beltrami functional, should be addressed.

Acknowledgements

Thanks to the anonymous referees for their valuable input.

References

[AGNS91] S. Auerbach, R. H. J. Gmelig Meyling, M. Neamtu, and H. Schaeben. Approximation and geometric modeling with simplex B-splines associated with irregular triangles. *Computer Aided Geometric Design*, 8:67–87, 1991.

[BWS+87] P. L. Baehmann, S. L. Wittchen, M. S. Shephard, K. R. Grice, and M. A. Yerry. Robust, geometrically based, automatic two-dimensional mesh generation. *International Journal for Numerical Methods in Engineering*, 24:1043–1078, 1987.

[Cav74] J. C. Cavendish. Automatic triangulation of arbitrary planar domains for the finite element method. *International Journal for Numerical Methods in Engineering*, 8:679–696, 1974.

[CG91] G. Celniker and D. Gossard. Deformable curve and surface finite-elements for free-form shape design. In *Proceedings of SIGGRAPH '91*, pages 257–265. ACM SIGGRAPH, acm Press, 1991.

[DMS92] W. Dahmen, C. A. Micchelli, and H.-P. Seidel. Blossoming begets B-spline bases built better by B-patches. *Mathematics of Computation*, 59(199):97–115, July 1992.

[For94] S. Fortune. Voronoi diagrams and Delaunay triangulations. In D.-Z. Du and F.K. Hwang, editors, *Computing in Euclidean Geometry*, pages 193–233. World Scientific Publishing, 1994.

[Gre94a] G. Greiner. Surface construction based on variational principles. In P. J. Laurent, A. Le Méhauté, and L. L. Schumaker, editors, *Wavelets, Images and Surface Fitting*. A K Peters, 1994.

[Gre94b] G. Greiner. Variational design and fairing of spline surfaces. In *Proceedings of EUROGRAPHICS '94*, pages C.143–C.154. Eurographics Association, Blackwell, 1994.

[HKD93] M. Halstead, M. Kass, and T. DeRose. Efficient, fair interpolation using Catmull-Clark surfaces. In *Proceedings of SIGGRAPH '93*, pages 35–44. ACM SIGGRAPH, acm Press, 1993.

[JS86] B. Joe and R. B. Simpson. Triangular meshes for regions of complicated shape. *International Journal for Numerical Methods in Engineering*, 23:751–778, 1986.

[Mic79] C. A. Micchelli. On a numerically efficient method for computing multivariate B-splines. In W. Schempp and K. Zeller, editors, *Multivariate Approximation Theory*. Birkhäuser, Basel, 1979.

[MS92] H. P. Moreton and C. H. Séquin. Functional optimization for fair surface design. In *Proceedings of SIGGRAPH '92*, pages 167–176. ACM SIGGRAPH, acm Press, 1992.

[PS95] R. Pfeifle and H.-P. Seidel. Fitting triangular B-splines to functional scattered data. In *Proceedings of Graphics Interface '95*, pages 80–88. Canadian Human-Computer Communications Society, Morgan Kaufmann, 1995.

[Sei91] H.-P. Seidel. Polar forms and triangular B-Spline surfaces. In *Blossoming: The New Polar-Form Approach to Spline Curves and Surfaces SIGGRAPH '91 Course Notes # 26*. ACM SIGGRAPH, 1991.

[SRK94] G. Subramanian, V. V. S. Raveendra, and M. Gopalakrishna Kamath. Robust boundary triangulation and Delaunay triangulation of arbitrary planar domains. *International Journal for Numerical Methods in Engineering*, 37:1779–1789, 1994.

[WH94] N. P. Weatherill and O. Hassan. Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering*, 37:2005–2039, 1994.

[WW92] W. Welch and A. Witkin. Varational surface modeling. In *Proceedings of SIGGRAPH '92*, pages 157–166. ACM SIGGRAPH, acm Press, 1992.

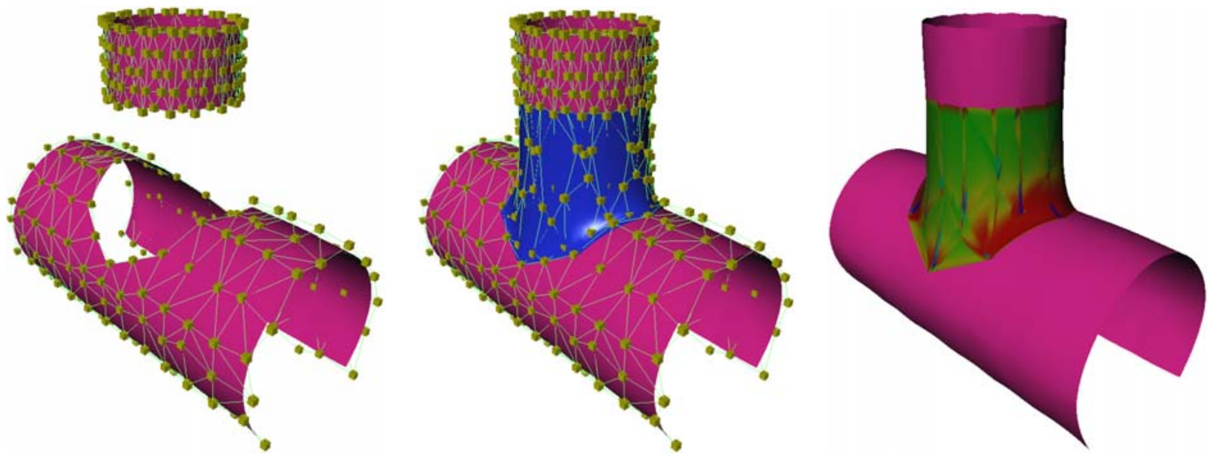

Figure 8: Forming a tee joint between two tubes. Left, the original surfaces. The horizontal tube is open to allow for the connection of the blend surface. The surfaces' control points appear as small yellow cubes. Middle, the completed blend (in blue). Right, Gaussian curvature on the blend surface.

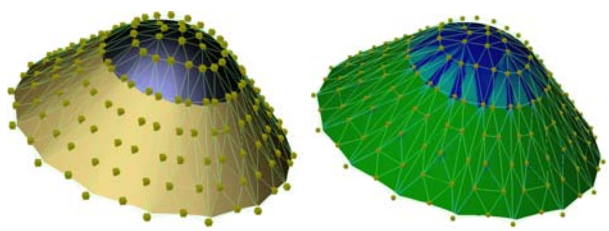

Figure 9: Capping a truncated cone using the linearized thin-plate energy functional. Left, a cone-shape surface (yellow) is smoothly capped with a filling surface (blue-grey). Right, plot of Gaussian curvature of the capped cone.

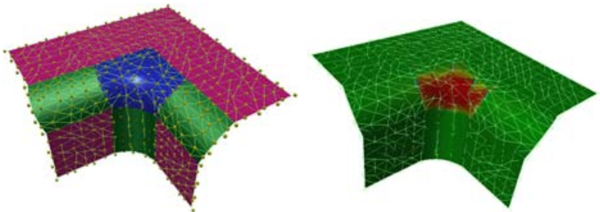

Figure 10: Left, three planar regions (pink) have been successively joined using three blend surfaces (green) and one fill (blue). Right, plot of Gaussian curvature of the surface. The filled five-sided hole exhibits predominantly negative curvature.

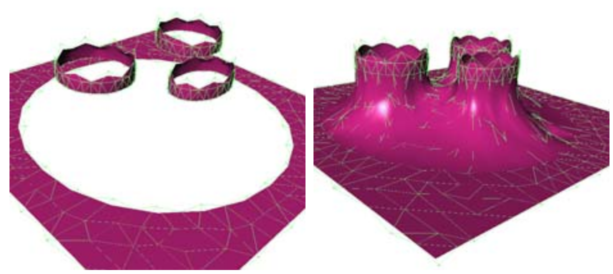

Figure 11: Three pipes meeting at a junction. Left, the tubes and the basin bottom. Right, the blended surface. This examples illustrates that very complicated fills can be computed using this approach.