Variational Design with Parametric Meshes of Arbitrary Topology

Leif Kobbelt

University of Erlangen

Abstract: Many mathematical problems in geometric modeling are merely due to the difficulties of handling piecewise polynomial parameterizations of surfaces (e.g., smooth connection of patches, evaluation of geometric fairness measures). Dealing with polygonal meshes is mathematically much easier although infinitesimal smoothness can no longer be achieved. However, transferring the notion of fairness to the discrete setting of triangle meshes allows to develop very efficient algorithms for many specific tasks within the design process of high quality surfaces. The use of discrete meshes instead of continuous spline surfaces is tolerable in all applications where (on an intermediate stage) explicit parameterizations are not necessary. We explain the basic technique of *discrete fairing* and give a survey of possible applications of this approach.

1 Introduction

Piecewise polynomial spline surfaces have been the standard representation for free form surfaces in all areas of CAD/CAM over the last decades (and still are). However, although B-splines are optimal with respect to certain desirable properties (differentiability, approximation order, locality, ...), there are several tasks that cannot be performed easily when surface parameterizations are based on piecewise polynomials. Such tasks include the construction of globally smooth closed surfaces and the shape optimization by minimizing intrinsically geometric fairness functionals [5, 12].

Whenever it comes to involved numerical computations on free form surfaces — for instance in finite element analysis of shells — the geometry is usually sampled at discrete locations and converted into a piecewise linear approximation, i.e., into a polygonal mesh.

Between these two opposite poles, i.e., the *continuous* representation of geometric shapes by spline patches and the *discrete* representation by polygonal meshes, there is a compromise emerging from the theory of *subdivision surfaces* [4]. Those surfaces are defined by a *base mesh* roughly describing its shape, and a *refinement rule* that allows one to split the edges and faces in order to obtain a finer and smoother version of the mesh.

Subdivision schemes started as a generalization of *knot insertion* for uniform B-splines [11]. Consider a control mesh $[\mathbf{c}_{i,j}]$ and the knot vectors $[u_i] = [i\,h_u]$ and $[v_i] = [i\,h_v]$ defining a tensor product B-spline surface $\mathcal{S}$. The same surface can be given with respect to the refined knot vectors $[\hat{u}_i] = [i\,h_u/2]$ and $[\hat{v}_i] = [i\,h_v/2]$ by computing the corresponding control vertices $[\hat{\mathbf{c}}_{i,j}]$, each $\hat{\mathbf{c}}_{i,j}$ being a simple linear combination of original vertices $\mathbf{c}_{i,j}$. It is well known that the iterative repetition of this process generates a sequence of meshes C_m which converges to the spline surface $\mathcal{S}$ itself.

The generic subdivision paradigm generalizes this concept by allowing arbitrary rules for the computation of the new control vertices $\hat{\mathbf{c}}_{i,j}$ from the given $\mathbf{c}_{i,j}$. The generalization also includes that we are no longer restricted to tensor product meshes but can use rules that are adapted to the different topological special cases in meshes with arbitrary connectivity. As a consequence, we can use any (manifold) mesh for the base mesh and generate smooth surfaces by iterative refinement.

The major challenge is to find appropriate rules that guarantee the convergence of the meshes C_m generated during the subdivision process to a smooth limit surface $\mathcal{S} = C_\infty$. Besides the classical stationary schemes that exploit the piecewise regular structure of iteratively refined meshes [1, 3, 9], there are more complex geometric schemes [15, 8] that combine the subdivision paradigm with the concept of optimal design by energy minimization (*fairing*).

The technical and practical advantages provided by the representation of surfaces in the form of polygonal meshes stem from the fact that we do not have to worry about infinitesimal inter-patch smoothness and the refinement rules do not have to rely on the existence of a globally consistent parameterization of the surface. In contrast to this, spline based approaches have to introduce complicated non-linear geometric continuity conditions to achieve the flexibility to model closed surfaces of arbitrary shape. This is due to the topologically rather rigid structure of patches with triangular or quadrilateral parameter domain and fixed polynomial degree of cross boundary derivatives. The non-linearity of such conditions makes efficient optimization difficult if not practically impossible. On discrete meshes however, we can derive *local* interpolants according to local parameterizations (*charts*) which gives the freedom to adapt the parameterization individually to the local geometry and topology.

In the following we will shortly describe the concept of *discrete fairing* which is an efficient way to characterize and compute dense point sets on high quality surfaces that observe prescribed interpolation or approximation constraints. We then show how this approach can be exploited in several relevant fields within the area of free form surface modeling.

The overall objective behind all the applications will be the attempt to avoid, bypass, or at least delay the mathematically involved generation of spline CAD-models whenever it is appropriate. Especially in the early design stages it is usually not necessary to have an explicit parameterization of a surface. The focus on polygonal mesh representations might help to free the creative designer from being confined by mathematical restrictions. In later stages the conversion into a spline model can be based on more reliable information about the intended shape. Moreover, since technical engineers are used to perform numerical simulations on polygonal approximations of the true model anyway, we also might find short-cuts that allow to speed up the turn-around cycles in the design process, e.g., we could alter the shape of a mechanical part by modifying the FE-mesh directly without converting back and forth between different CAD-models.

2 Fairing triangular meshes

The observation that in many applications the global fairness of a surface is much more important than infinitesimal smoothness motivates the *discrete fairing* approach [10]. Instead of requiring G^1 or G^2 continuity, we simply approximate a surface by a plain triangular C^0– mesh. On such a mesh we can think of the (discrete) curvature being located at the

vertices. The term *fairing* in this context means to minimize these local contributions to the total (discrete) curvature and to equalize their distribution across the mesh.
We approximate local curvatures at every vertex $\mathbf{p}$ by divided differences with respect to a locally isometric parameterization $\mu_{\mathbf{p}}$. This parameterization can be found by estimating a tangent plane $T_{\mathbf{p}}$ (or the normal vector $\mathbf{n}_{\mathbf{p}}$) at $\mathbf{p}$ and projecting the neighboring vertices $\mathbf{p}_i$ into that plane. The projected points yield the parameter values (u_i, v_i) if represented with respect to an orthonormal basis $\{\mathbf{e}_u, \mathbf{e}_v\}$ spanning the tangent plane

$$\mathbf{p}_i - \mathbf{p} \;=\; u_i\,\mathbf{e}_u + v_i\,\mathbf{e}_v + d_i\,\mathbf{n}_{\mathbf{p}}.$$

Another possibility is to assign parameter values according to the lengths and the angles between adjacent edges (*discrete exponential map*) [15, 10].
To obtain reliable curvature information at $\mathbf{p}$, i.e., second order partial derivatives with respect to the locally isometric parameterization $\mu_{\mathbf{p}}$, we solve the normal equation of the Vandermonde system

$$V^T V \,[\, \tfrac{1}{2} f_{uu},\, f_{uv},\, \tfrac{1}{2} f_{vv} \,]^T \;=\; V^T \,[d_i]_i$$

with $V = [\,u_i^2, u_i v_i, v_i^2\,]_i$ by which we get the best approximating quadratic polynomial in the least squares sense. The rows of the inverse matrix $(V^T V)^{-1} V^T =: [\alpha_{i,j}]$ by which the Taylor coefficients f_* of this polynomial are computed from the data $[d_i]_i$, contain the coefficients of the corresponding divided difference operators Γ_*.
Computing a weighted sum of the squared divided differences is equivalent to the discrete sampling of the corresponding continuous fairness functional. Consider for example

$$\int_{\mathcal{S}} \kappa_1^2 + \kappa_2^2 \; d\mathcal{S}$$

which is approximated by

$$\sum_{\mathbf{p}_i} \omega_i \,\Big(\|\Gamma_{uu}(\mathbf{p}_j - \mathbf{p}_i)\|^2 + 2\,\|\Gamma_{uv}(\mathbf{p}_j - \mathbf{p}_i)\|^2 + \|\Gamma_{vv}(\mathbf{p}_j - \mathbf{p}_i)\|^2 \Big). \qquad (1)$$

Notice that the value of (1) is independent of the particular choices $\{\mathbf{e}_u, \mathbf{e}_v\}$ for each vertex due to the rotational invariance of the functional. The discrete fairing approach can be understood as a generalization of the traditional finite difference method to parametric meshes where divided difference operators are defined with respect to locally varying parameterizations. In order to make the weighted sum (1) of local curvature values a valid quadrature formula, the weights ω_i have to reflect the local area element which can be approximated by observing the relative sizes of the parameter triangles in the local charts $\mu_{\mathbf{p}} : \mathbf{p}_i - \mathbf{p} \mapsto (u_i, v_i)$.
Since the objective functional (1) is made up of a sum over squared local linear combinations of vertices (in fact, of vertices being direct neighbors of one central vertex), the minimum is characterized by the solution of a global but sparse linear system. The rows of this system are the partial derivatives of (1) with respect to the movable vertices $\mathbf{p}_i$. Efficient algorithms are known for the solution of such systems [6].

3 Applications to free form surface design

When generating fair surfaces from scratch we usually prescribe a set of interpolation and approximation constraints and fix the remaining degrees of freedom by minimizing an energy functional. In the context of discrete fairing the constraints are given by an initial triangular mesh whose vertices are to be approximated by a fair surface being topologically equivalent. The necessary degrees of freedom for the optimization are obtained by uniformly subdividing the mesh and thus introducing new *movable* vertices.

The discrete fairing algorithm requires the definition of a local parameterization $\mu_{\mathbf{p}}$ for each vertex $\mathbf{p}$ including the newly inserted ones. However, projection into an estimated tangent plane does not work here, because the final positions of the new vertices are obviously not known a priori. In [10] it has been pointed out that in order to ensure solvability and stability of the resulting linear system, it is appropriate to define the local parameterizations (local metrics) for the new vertices by *blending* the metrics of nearby vertices from the original mesh. Hence, we only have to estimate the local charts covering the original vertices to set-up the linear system which characterizes the optimal surface. This can be done prior to actually computing a solution and we omit an additional optimization loop over the parameterization.

When solving the sparse linear system by iterative methods we observe rather slow convergence. This is due to the low-pass filter characteristics of the iteration steps in a Gauß-Seidel or Jacobi scheme. However since the mesh on which the optimization is performed came out of a uniform refinement of the given mesh (*subdivision connectivity*) we can easily find nested grids which allow the application of highly efficient multi-grid schemes [6].

Moreover, in our special situation we can generate sufficiently smooth starting configurations by midpoint insertion which allows us to neglect the pre-smoothing phase and to reduce the V-cycle of the multi-grid scheme to the alternation of binary subdivision and iterative smoothing. The resulting algorithm has linear complexity in the number of generated triangles.

The advantage of this discrete approach compared to the classical fair surface generation based on spline surfaces is that we do not have to approximate a geometric functional that uses true curvatures by one which replaces those by second order partial derivatives with respect to the fixed parameterization of the patches. Since we can use a custom tailored parameterization for each point evaluation of the second order derivatives, we can choose this parameterization to be isometric — giving us access to the true geometric functional. Figure 1 shows an example of a surface generated this way. The implementation can be done very efficiently. The shown surface consists of about 50K triangles and has been generated on a SGI R10000 (195MHz) within 10 seconds. The scheme is capable of generating an arbitrarily dense set of points on the surface of minimal energy. It is worth to point out that the scheme works completely automatic: no manual adaption of any parameters is necessary, yet the scheme produces good surfaces for a wide range of input data.

4 Applications to interactive modeling

For subdivision schemes we can use any triangular mesh as a control mesh roughly describing the shape of an object to be modeled. The flexibility of the schemes with respect to the connectivity of the underlying mesh allows very intuitive modifications of the mesh. The

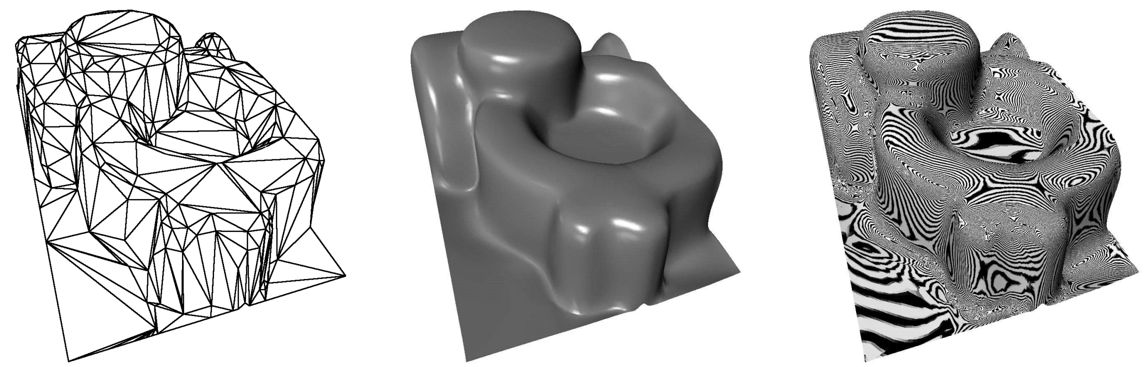

Fig. 1 A fair surface generated by the discrete fairing scheme. The flexibility of the algorithm allows to interpolate rather complex data by high quality surfaces. The process is completely automatic and it took about 10 sec to compute the refined mesh with 50K triangles. On the right you see the reflection lines on the final surface.

designer can move the control vertices just like for Bezier-patches but she is no longer tied to the common restrictions on the connectivity which is merely a consequence of the use of tensor product spline bases.

When modeling an object by Bezier-patches, the control vertices are the handles to influence the shape and the de Casteljau algorithm associates the control mesh with a smooth surface patch. In our more general setting, the designer can work on an *arbitrary* triangle mesh and the connection to a smooth surface is provided by the discrete fairing algorithm. The advantages are that control vertices are interpolated which is a more intuitive interaction metaphor and the topology of the control structure can adapt to the shape of the object.

Figure 2 shows the model of a mannequin head. A rather coarse triangular mesh allows already to define the global shape of the head (left). If we add more control vertices in the areas where more detail is needed, i.e., around the eyes, the mouth and the ears, we can construct the complex surface at the far right. Notice how the discrete fairing scheme does not generate any artifacts in regions where the level of detail changes.

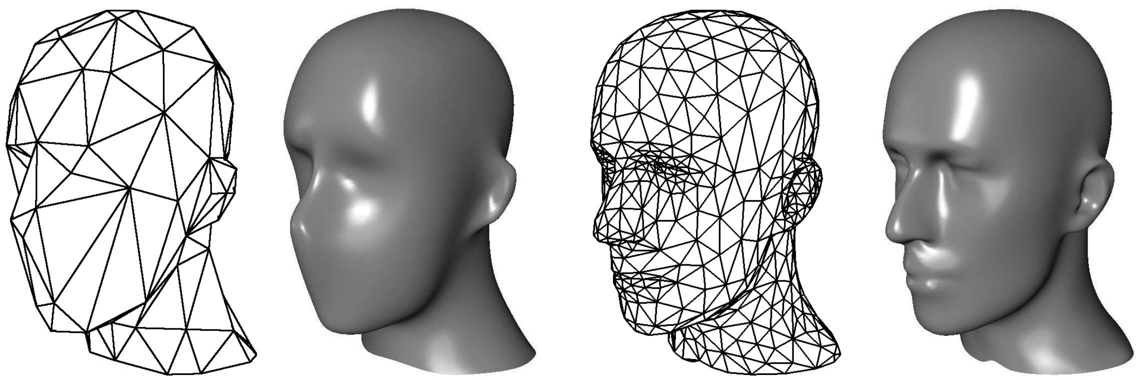

Fig. 2 Control meshes with arbitrary connectivity allow to adapt the control structure to the geometry of the model. Notice that the influence of one control vertex in a tensor product mesh is always rectangular which makes it difficult to model shapes with non-rectangular features.

5 Applications to mesh smoothing

In the last sections we saw how the discrete fairing approach can be used to generate fair surfaces that interpolate the vertices of a given triangular mesh. A related problem is to smooth out high frequency noise from a given *detailed* mesh without further refinement. Consider a triangulated surface emerging for example from 3D laser scanning or iso-surface extraction out of CT volume data. Due to measurement errors, those surfaces usually show oscillations that do not stem from the original geometry.
Constructing the above mentioned local parameterizations, we are able to quantify the noise by evaluating the local curvature. Shifting the vertices while observing a maximum tolerance can reduce the total curvature and hence smooth out the surface. From a signal processing point of view, we can interpret the iterative solving steps for the global sparse system as the application of recursive digital low-pass filters [13]. Hence it is obvious that the process will reduce the high frequency noise while maintaining the low frequency shape of the object.
Figure 3 shows an iso-surface extracted from a CT scan of an engine block. The noise is due to inexact measurement and unstabilities in the extraction algorithm. The smoothed surface remains within a tolerance which is of the same order of magnitude as the diagonal of one voxel in the CT data.

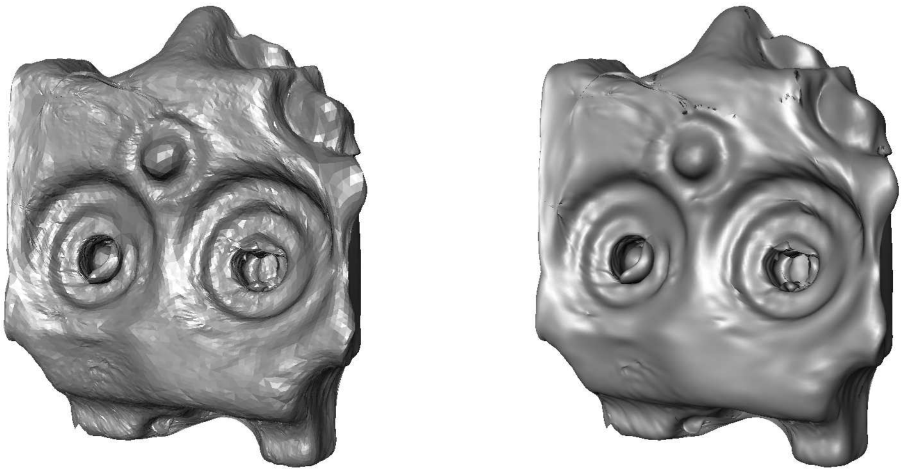

Fig. 3 An iso-surface extracted from a CT scan of an engine block. On the left, one can clearly see the noise artifacts due to measurement and rounding errors. The right object was smoothed by minimizing the discrete fairing energy. Constraints on the positional delocation were imposed.

6 Applications to surface interrogation

Deriving curvature information on a discrete mesh is not only useful for fair interpolation or post-processing of measured data. It can also be used to visualize artifacts on a surface by plotting the color coded discrete curvature directly on the mesh. Given for example the output of the numerical simulation of a physical process: since deformation has occurred during the simulation, this output typically consists merely of a discrete mesh and no continuous surface description is available.

Using classical techniques from differential geometry would require to fit an interpolating spline surface to the data and then visualize the surface quality by curvature plots. The availability of samples of second order partial derivatives with respect to locally isometric parameterizations at every vertex enables us to show this information directly without the need for a continuous surface.
Figure 4 shows a mesh which came out of the FE-simulation of a loaded cylindrical shell. The shell is rigidly supported at the boundaries and pushed down by a force in normal direction at the center. The deformation induced by this load is rather small and cannot be detected by looking, e.g., at the reflection lines. The discrete mean curvature plot however clearly reveals the deformation. Notice that histogram equalization has been used to optimize the color contrast of the plot.

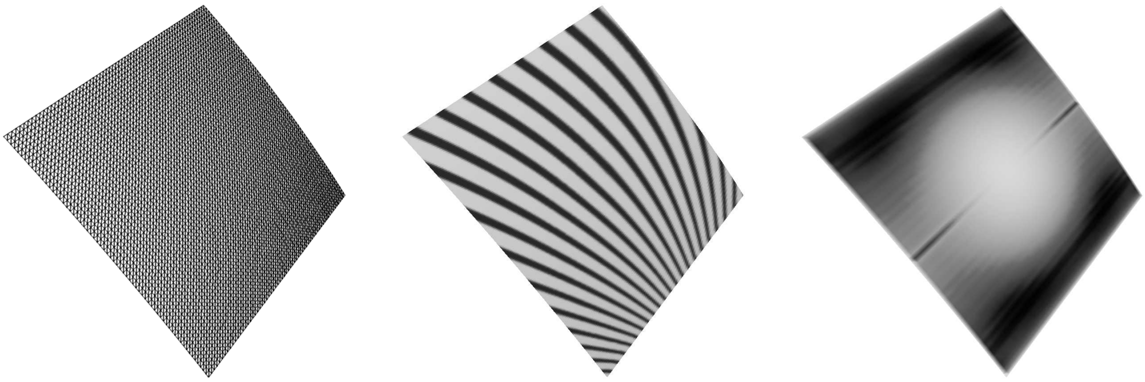

Fig. 4 Visualizing the discrete curvature on a finite element mesh allows to detect artifacts without interpolating the data by a continuous surface.

7 Applications to hole filling and blending

Another area where the discrete fairing approach can help is the filling of undefined regions in a CAD model or in a measured data set. Of course, all these problems can be solved by fairing schemes based on spline surfaces as well. However, the discrete fairing approach allows one to split the overall (quite involved) task into simple steps: we always start by constructing a triangle mesh defining the global topology. This is easy because no G^1 or higher boundary conditions have to be satisfied. Then we can apply the discrete fairing algorithm to generate a sufficiently dense point set on the objective surface. This part includes the refinement and energy minimization but it is almost completely automatic and does not have to be adapted to the particular application. In a last step we fit polynomial patches to the refined data. Here we can restrict ourselves to pure fitting since the fairing part has already been taken care of during the generation of the dense data. In other words, the discrete fairing has recovered enough information about an optimal surface such that staying as close as possible to the generated points (in a least squares sense) is expected to lead to high quality surfaces. To demonstrate this methodology we give two simple examples.
First, consider the point data in Figure 5. The very sparsely scattered points in the middle region make the task of interpolation rather difficult since the least squares matrix for

a locally supported B-spline basis might become singular. To avoid this, fairing terms would have to be included into the objective functional. This however brings back all the problems mentioned earlier concerning the possibly poor quality of parameter dependent energy functionals and the prohibitive complexity of non-linear optimization.
Alternatively, we can connect the points to build a spatial triangulation. Uniform subdivision plus discrete fairing recovers the missing information under the assumption that the original surface was sufficiently fair. The un-equal distribution of the measured data points and the strong distortion in the initial triangulation do not cause severe instabilities since we can define individual parameterizations for every vertex. These allow one to take the local geometry into account.

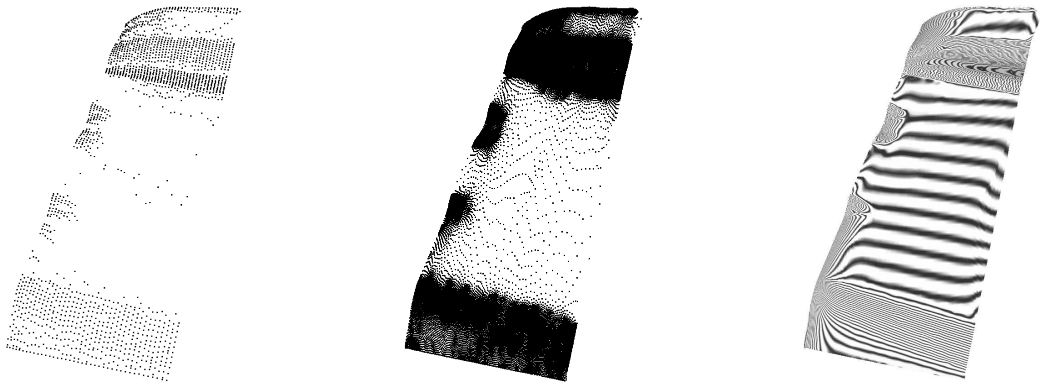

Fig. 5 The original data on the left is very sparse in the middle region of the object. Triangulating the points in space and discretely fairing the iteratively refined mesh recovers more information which makes least squares approximation much easier. One the right, reflection lines on the resulting surface are shown.

Another standard problem in CAD is the *blending* or *filleting* between surfaces. Consider the simple configuration in Figure 6 where several plane faces (dark grey) are to be connected smoothly. We first close the gap by a simple coarse triangular mesh. Such a mesh can easily be constructed for any reasonable configuration with much less effort than constructing a piecewise polynomial representation. The boundary of this initial mesh is obtained by sampling the surfaces to be joined.
We then refine the mesh and, again, apply the discrete fairing machinery. The smoothness of the connection to the predefined parts of the geometry is guaranteed by letting the blend surface mesh overlap with the given faces by one row of triangles (all necessary information is obtained by sampling the given surfaces). The vertices of the triangles belonging to the original geometry are not allowed to move but since they participate in the global fairness functional they enforce a smooth connection. In fact this technique allows to define Hermite-type boundary conditions.

8 Conclusion

In this paper we gave a survey of currently implemented applications of the discrete fairing algorithm. This general technique can be used in all areas of CAD/CAM where an approximation of the actual surface by a reasonably fine triangular mesh is a sufficient

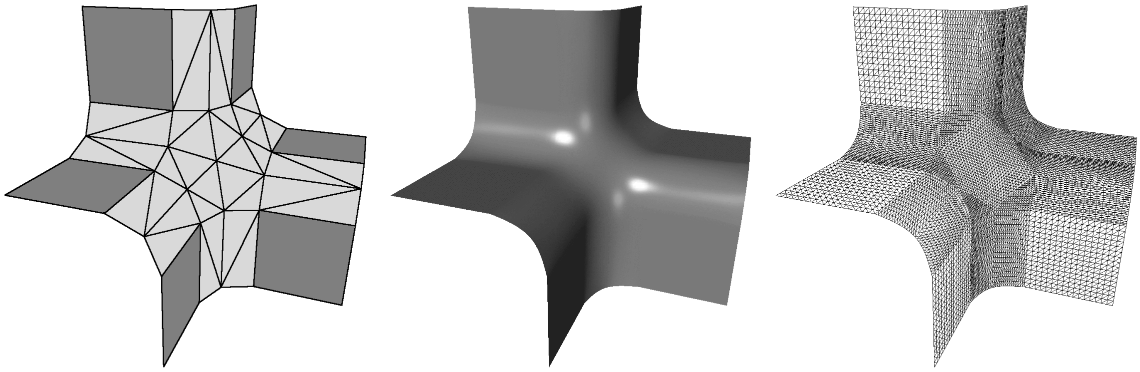

Fig. 6 Creating a “monkey saddle“ blend surface to join six planes. Any blend surface can be generated by closing the gap with a triangular mesh first and then applying discrete fairing.

representation. If compatibility to standard CAD formats matters, a spline fitting post-process can always conclude the discrete surface generation or modification. This fitting step can rely on more information about the intended shape than were available in the original setting since a *dense* set of points has been generated.
As we showed in the previous sections, mesh smoothing and hole filling can be done on the discrete structure *before* switching to a continuous representation. Hence, the bottom line of this approach is to do most of the work in the discrete setting such that the mathematically more involved algorithms to generate piecewise polynomial surfaces can be applied to enhanced input data with most common artifacts removed.
We do not claim that splines could ever be completely replaced by polygonal meshes but in our opinion we can spare a considerable amount of effort if we use spline models only where it is really necessary and stick to meshes whenever it is possible. There seems to be a huge potential of applications where meshes do the job if we find efficient algorithms.
The major key to cope with the genuine complexity of highly detailed triangle meshes is the introduction of a hierarchical structure. Hierarchies could emerge from classical multiresolution techniques like subdivision schemes but could also be a by-product of mesh simplification algorithms.
An interesting issue for future research is to find efficient and numerically stable methods to enforce convexity preservation in the fairing scheme. At least local convexity can easily be maintained by introducing non-linear constraints at the vertices.
Prospective work also has to address the investigation of explicit and reliable techniques to exploit the discrete curvature information for the detection of feature lines in the geometry in order to split a given mesh into geometrically coherent segments. Further, we can try to identify regions of a mesh where the value of the curvature is approximately constant — those regions correspond to special geometries like spheres, cylinders or planes. This will be the topic of a forthcoming paper.

9 Acknowledgments

I would like to thank Swen Campagna, Torsten Hesse, Christoph Lürig, and Jens Vorsatz for supporting the generation of the figures and Tanja Höchstetter for her patience.

References

[1] E. Catmull, J. Clark, *Recursively generated B-spline surfaces on arbitrary topological meshes*, CAD 10 (1978), pp. 350–355

[2] Celniker G. and D. Gossard, *Deformable curve and surface finite elements for free-form shape design*, ACM Computer Graphics **25** (1991), 257–265.

[3] D. Doo / M. Sabin, *Behaviour of Recursive Division Surfaces Near Extraordinary Points*, CAD 10 (1978), pp. 356–360

[4] N. Dyn, *Subdivision Schemes in Computer Aided Geometric Design*, Adv. Num. Anal. II, Wavelets, Subdivisions and Radial Functions, W.A. Light ed., Oxford Univ. Press, 1991, pp. 36–104.

[5] Greiner G., *Variational design and fairing of spline surfaces*, Computer Graphics Forum **13** (1994), 143–154.

[6] Hackbusch W., *Multi-Grid Methods and Applications*, Springer Verlag 1985, Berlin.

[7] Hagen H. and G. Schulze, *Automatic smoothing with geometric surface patches*, CAGD **4** (1987), 231–235.

[8] Kobbelt L., *A variational approach to subdivision*, CAGD **13** (1996), 743–761.

[9] Kobbelt L., *Interpolatory subdivision on open quadrilateral nets with arbitrary topology*, Comp. Graph. Forum **15** (1996), 409–420.

[10] Kobbelt L., *Discrete fairing*, Proceedings of the Seventh IMA Conference on the Mathematics of Surfaces, 1996.

[11] J. Lane / R. Riesenfeld, *A Theoretical Development for the Computer Generation and Display of Piecewise Polynomial Surfaces*, IEEE Trans. on Pattern Anal. and Mach. Int., 2 (1980), pp. 35–46

[12] Moreton H. and C. Séquin, *Functional optimization for fair surface design*, ACM Computer Graphics **26** (1992), 167–176.

[13] Taubin G., *A signal processing approach to fair surface design*, ACM Computer Graphics **29** (1995), 351–358

[14] Welch W. and A. Witkin, *Variational surface modeling*, ACM Computer Graphics **26** (1992), 157–166

[15] Welch W. and A. Witkin, *Free-form shape design using triangulated surfaces*, ACM Computer Graphics **28** (1994), 247–256