Functionally Optimized Subdivision Surfaces

Pushkar Joshi,* Carlo H. Séquin, and Ryo Takahashi
CS Division, U.C. Berkeley

1 Introduction

It can be argued that an ideal surface design system should allow a designer to specify all the boundary conditions and constraints and then provide the "best" surface under these circumstances. "Best" in this context may mean an optimization with respect to some intrinsic surface quality expressible in a functional or procedural form. For instance, the designer may want to minimize surface area (i.e. bending energy), variation of curvature, or some other aesthetically motivated functional. Systems that optimize such functionals have been demonstrated in the past, but in most cases, the optimization algorithm was too complex and too slow to provide the desired, almost instantaneous, and truly interactive surface optimization.

As our basic framework, we use subdivision surfaces to represent the shapes to be optimized. Using finite differences based on incremental movements of the control vertices, a gradient vector for the chosen cost/energy functional is obtained and then used to evolve the surface iteratively towards a local cost minimum. After obtaining the minimum energy surface for a given mesh resolution, the mesh is subdivided to produce new vertices and therefore *new parameters* for optimization. In this framework of subdivision followed by geometric optimization, we can vary the methods for calculating the actual optimization moves, trading off accuracy for speed.

2 Exact Evaluation

As a baseline to compare the various methods we use exact evaluation of the subdivision surface, sampling the limit surface to obtain its geometric properties. Using differential geometry and numerical integration by Gauss-Legendre quadrature, we can compute a cost functional such as the bending energy with high accuracy. Using this energy computation in the above framework, we have obtained robust results that agree with the theoretically known energy minima for some highly symmetrical smooth surfaces, such as spheres, tori, or the known energy minimizers of higher genus [Hsu et al. 1992]. Since numerical integration and gradient calculations are computationally expensive, this method may take a few hours for surfaces like those depicted in Figure 1. However, it serves as an excellent benchmark for evaluating the following more approximate methods.

3 Discrete Approximation of Cost Functional

The first simplification calculates an approximate cost functional directly from the discrete mesh of control points of the subdivision surface as done, for instance, in [Desbrun et al. 1999]. We have used vertex-based and edge-based MES functionals that express the surface energy as a summation over the local energy at all vertices or edges, using polynomial expressions of vertex coordinates and/or dihedral angles along the edges. These discretized functionals are adequate to guide the gradient descent process into the same direction as a more exact functional evaluation would, but do so at significantly reduced cost and thus with higher speed. For various test cases of surfaces, ranging from spheres to more complex surfaces of genus 3, we have compared the shapes (Fig.1a) obtained with the discretized functional in mere minutes to the previously calculated "benchmark" shapes, and we found the results to be in very good geometric agreement.

* e-mail: ppj@eecs.berkeley.edu

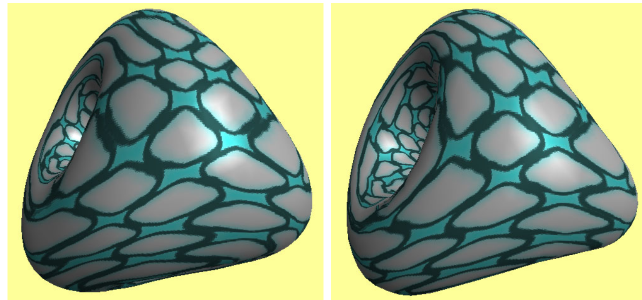

Figure 1. Genus 3 surfaces obtained by minimizing discretized bending energy (MES) {a}, and by aiming for minimum variation of curvature (MVS) with direct vertex-move calculations {b}.

3 Direct Vertex Move Calculations

The second simplification avoids the gradient calculation based on finite differences. Instead we calculate directly the moves for the control vertices that will optimize the surface in the desired direction. In particular, we have developed a vertex-move procedure that aims to minimize the *variation* of curvature as attempted by [Moreton and Séquin 1992]. For this purpose, we calculate, for each edge in the control mesh, a change in the turning angle in the direction of the edge and then aim to swivel the edge about a point on it so as to reduce this turning variation. Each vertex obtains a suggested move component from every edge attached to it, and is then moved proportional to the mean of these components. Figure 1b shows a surface obtained by this direct method; the shape is very close to the MVS shape found by [Moreton and Séquin 1992] after many hours of computation, but now it can be generated interactively in just a few seconds!

4 Conclusions

We are able to gain significant speedup by using discrete functionals and direct vertex-move calculations without compromising the final shapes obtained. Thus we can envision a CAD system in the not-too-distant future, where the designer specifies boundary conditions and constraints, and then picks one of several cost functionals to quickly optimize the surface with. The designer may then compare and contrast the results of two or three different aesthetic functionals and chose the one that seems most appropriate for the given application domain.

References

DESBRUN, M., MEYER, M., SCHROEDER, P., AND BARR, A. 1999. Implicit fairing of arbitrary meshes using diffusion and curvature flow. Proc. of ACM SIGGRAPH 1999.

HSU, L., KUSNER, R., AND SULLIVAN, J. 1992. Minimizing the squared mean curvature integral for surfaces in space forms. Experimental Math 1, 3, 191–207.

MORETON, H., AND SÉQUIN, C. 1992. Functional optimization for fair surface design. Proc. of ACM SIGGRAPH 1992, 167–176.