CAD Tools for Aesthetic Engineering

Carlo H. Séquin

University of California, Berkeley, sequin@cs.berkeley.edu

ABSTRACT

The role of computers and of computer-aided design tools for the creation geometrical shapes that will be judged primarily by aesthetic considerations is reviewed. Examples are the procedural generation of abstract geometrical sculpture or the shape optimization of constraint curves and surfaces with some global "cost" functional. Different possibilities for such "beauty functionals" are discussed. Moreover, rapid prototyping tools based on layered manufacturing now add a new dimension to the visualization of emerging designs. Finally, true interactivity of the CAD tools allows a more effective exploration of larger parts of the design space and can thereby result in an actual amplification of the creative process.

Keywords: Shape optimization, geometrical sculpture, sculpture generator, rapid prototyping.

1 INTRODUCTION

In this tutorial, we are concerned with computer-aided design tasks in which the final evaluation is mostly based on aesthetic criteria. While most engineers accept the fact that one needs to use computers to design jet engines, computer chips, or large institutional buildings, it is less clear whether computers are also useful in the design of artifacts that are judged mostly by their looks. In a traditional CAD setting, the computer primarily serves as a precise drafting and visualization tool, permitting the designer to view the emerging geometry from different angles and in different projections. A digital representation also makes it possible to carry out some analytical tasks such as determining volume or surface area of a part.

We will show that today the role of the computer goes much further. It actively supports the creation of geometric shapes by procedural means and can even optimize a surface by maximizing some "beauty functional." It further can help to generalize visualization aids for complex parts through the generation of rapid prototypes on layered manufacturing machines. Finally, it may even amplify the creative process itself by allowing the designer to quickly explore a much larger space of design alternatives.

The objects used as examples in this tutorial are mostly abstract geometrical sculptural forms or mathematical visualization models (Fig.1). However, the principles and techniques discussed are readily applicable also to consumer products, or automotive parts and shapes. Creating maximally satisfactory forms for mathematical models or for geometric sculptures poses quite different requirements and constraints for any CAD tool than developing an optimized airplane wing or designing the most powerful computer chip. Real-time interactivity becomes a crucial factor, when a designer's eye is the key evaluation module in the design loop.

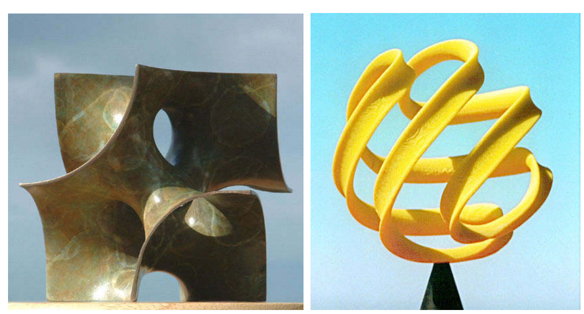

Fig. 1. Geometrical sculptures: (a) *Volution_5*, (b) *Altamont*.

This tutorial overview starts by looking at some generic tasks in curve and surface design, in particular, the efforts for defining a "beauty functional" for procedurally optimizing shapes that are only partially constrained by the designer, as well as for efficient implementations and approximations of such optimization functionals so that they can be used at interactive design speeds. Next, we look a parameterized design paradigm that allows a designer to rapidly explore and compare many alternative versions of a design. Finally, we make the point that a CAD tool that is well matched to the task at hand is much more than just a "drafting assistant" and can indeed become an amplifier for one's creative spark.

2 OPTIMIZATION OF SMOOTH SURFACES

Smooth surfaces play an important role in engineering and are a main application for many industrial CAD tools. Some surfaces are defined almost entirely by their functions; examples are ship hulls and airplane wings. Other surfaces combine a mixture of functional and aesthetic concerns, e.g. car bodies, coffee cups, flower vases… In other cases, aesthetics dominates the designer's concern, for instance in abstract geometric sculpture.

For either situation, 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 the context of this tutorial would mean an optimization with respect to some intrinsic surface quality related to its aesthetic appeal. To be usable in a CAD tool, that quality has to be expressible in a functional or procedural form. Commonly, the characteristics associated with "beautiful" or "fair" surfaces imply smoothness - at least tangent-plane (G^1-) continuity, but often also curvature (G^2-) continuity. If the surface is covered with some textural pattern, then we have to demand more than just geometric continuity and also require parametric continuity, i.e., C^1- or C^2-continuity, respectively. Additional characteristics often cited in the definition of beautiful shapes are symmetry and simplicity. The first implies that symmetrical constraints should result in symmetrical solutions; and the second implies avoidance of unnecessary undulations or ripples.

All these properties are exhibited by *minimal surfaces*, i.e., by the shapes assumed by thin soap membranes spanning some given boundary (as long as the air pressure on both sides is the same). Experimentally, such shapes can be generated by dipping a warped wire loop into a soap solution. The lateral molecular membrane-forces will try to minimize overall surface area and thereby implicitly create a minimal saddle surface in which the mean curvature at every point of the surface assumes the value zero.

$$MinimalSurface \Rightarrow \kappa_1 + \kappa_2 = 0 \tag{1}$$

A generalization of such shapes that extends to closed surfaces can be obtained by minimizing the total bending energy of the surface. In an abstraction and idealization that goes back to Bernoulli, the local bending energy of a thin filament or a thin sheet of stiff material is proportional to the square of the local curvature. The total bending energy of a shape then can be obtained as an arc-length or area integral of curvature squared over the whole shape.

$$MinimumEnergySurface \Rightarrow \int \kappa^2 dA = \min \tag{2}$$

For closed surfaces, it turns out that minimizing bending energy is equivalent to minimizing mean curvature, since the area integral of Gaussian curvature is a topological constant that depends only on the genus of the surface. This energy functional is also known as Willmore energy [5], and the possible minimal-energy shapes for surfaces of different genus are well known [5]. For surfaces of genus 0, the minimal shape is, of course, a sphere, and it has a total bending energy of 4π regardless of its size, since the bending energy functional happens to be scale-invariant. For genus 1, bending energy is minimized in the *Clifford torus* in which the ratio of the two defining radii is equal to $\sqrt{2}$. And for surfaces of higher genus, the energy minimizing shape is the *Lawson surface*. With increasing genus, this surface ever more closely approximates two intersecting spheres with a regular circle of tiny pillars and holes at their intersection line, reminiscent of the central portion in Scherk's second minimal surface [7] wrapped into a toroidal ring. The total Willmore energy for all these surfaces always lies below a value of 8π.

It has been argued [6] that bending energy may not be the best "beauty functional." For most people, a Lawson surface of genus 8 is not clearly the preferred shape with 8 handles or 8 tunnels. Also, if the perfect genus-0 shape is indeed a sphere, shouldn't the "penalty" (energy) function of that shape assume the value 0? Thus we might obtain a better functional to evaluate the fairness of a curve or surface, if we try to minimize the integral over the '*change of curvature*' squared. Moreton has created a first implementation of such a functional by integrating the squares of the derivatives of the principal curvatures in the directions of their respective principal directions [6].

$$Minim.VariationSurface \Rightarrow \int \frac{d\kappa_1}{de_1}^2 + \frac{d\kappa_2}{de_2}^2 dA = \min \tag{3}$$

In surfaces where the principal lines of curvature are exact circles, this *minimum variation* (MV) functional evaluates to zero. Thus all *cyclides* (spheres, cylinders, cones, tori, and even horned tori) are "perfect" surfaces of minimal MVS cost.

The challenge now exists to implement the evaluation of these cost functionals so that surfaces can be optimized at interactive rates. The first system to create *minimum-variation surfaces* (MVS) used biquintic quadrilateral Bezier patches stitched together so as to form the desired shapes [6]. All the degrees of freedom contained in the coordinates of the control points that were not specified by design constraints were then varied with the goal to minimize the overall cost function. The gradient components of all the available degrees of freedom were determined with finite differences, and a conjugate gradient descent method was used to move the system towards a local optimum. The area integral over the change of curvature was evaluated by Gauss-Legendre or by Lobatto quadrature, typically using about 400 sample points per Bézier patch. Penalty functions using Lagrange multipliers were employed in an inner optimization loop to enforce G^1- and G^2-continuity across the seams between adjacent patches. The system was very slow, using many hours for converging on even simple symmetrical shapes; but it produced beautiful results [6].

2.1 Interactive Surface Optimization

Now, a decade later, what are the prospects for evaluating such functionals at the desired, almost instantaneous and truly interactive rate?

First, of course, computer power has increased by one to two orders of magnitude over the last decade, thus bringing us closer to our goal of full interactivity, even without any further innovations.

Second, and most importantly, subdivision surfaces have become mature and popular. They allow us to obtain surfaces with a reasonable degree of built-in continuity by their inherent construction, thus avoiding the very costly inner optimization loops that were used originally to guarantee smoothness at the seams. For instance, Catmull-Clark subdivision surfaces can offer G^1-continuity everywhere and exhibit C^2-continuity almost everywhere, except at extraordinary points where quadrilateral patches join with a valence different from 4.

Third, the inherently hierarchical organization of subdivision surfaces gives us the possibility to optimize the gross shape of the surface at a relatively coarse level, where only a small number of control points have to be adjusted. Then as we gradually refine the surface by increasing the level of subdivision, the number of degrees of freedom grows quadratic; but since the surface is already relatively close to the desired shape, the optimization procedure need not run for many iterations until convergence is achieved.

Fourth, at the research frontier, experiments are now going on to find ways to avoid the expensive numerical integration steps in the inner loop of the optimization. The aim is to find a discretized approximation of the salient surface characteristics, to obtain directly an estimate of the behavior of the cost functional that is good enough to guide the gradient descent optimization in the right direction.

2.2 The New Framework

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 general approach, we can vary the methods for calculating the actual optimization moves, trading off accuracy for speed.

2.3 Approximating the Cost Functional

A first simplification calculates an approximate cost functional directly from the discrete mesh of control points of the subdivision surface, as is done, for instance, in [3]. We have used vertex-based and edge-based 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 in 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, ranging from spheres to more complex surfaces of genus 3, we have compared the shapes (Fig.2a) 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.

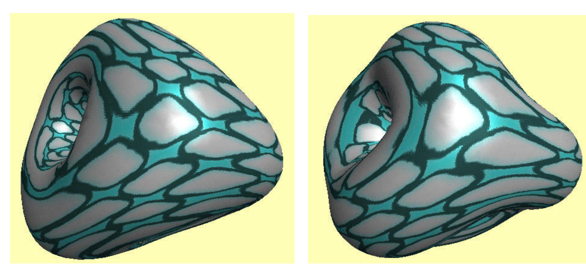

Fig. 2. Genus 3 surfaces: (a) obtained by minimizing a discretized bending energy, and (b) by approximating minimum curvature variation with a direct vertex-move calculation.

2.4 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 [6]. For this purpose, we calculate for each edge in the control mesh a *change in normal curvature* in the direction of the edge, and then aim to move the vertices along their averaged vertex normals so as to reduce this curvature variation. Each vertex obtains a suggested move component from every edge attached to it, and it is then moved proportional to the mean of these components. Figure 2b shows a surface obtained by this direct method; the shape is very close to the shape found in 1992 after many hours of computation [6], but now it can be generated in just a few seconds!

With this speedup resulting from the use of discrete functionals and direct vertex-move calculations, 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 for a quick optimization of the surface. The designer may then compare and contrast the results of two or three different aesthetic functionals and choose the one that is most appropriate for the given application domain.

3 FAIR CURVES ON FAIR SURFACES

A second key CAD problem is the embedding of "beautiful" smooth curves in the optimized surfaces discussed above. Often one needs to draw a fair connecting line between two points on a smooth surface. The most direct such connection is a geodesic line. While it is easy to trace a uni-directional geodesic ray on a smooth surface or on a finely tessellated polyhedral approximation thereof, it is a well-known hard problem to connect two points with the shortest geodesic path on a surface that exhibits many areas of positive and negative mean curvature.

Sometimes the geodesic line segment is too restrictive for design purposes; it offers no degrees of freedom or adjustable parameters to the designer. This limitation is particularly detrimental when multiple lines must radiate from the same point. In this situation a designer would like to have some control over the initial tangent directions of these lines, perhaps to distribute them at equal angles around the point from which they emerge. For this purpose, a good alternative is a line for which its geodesic curvature is either constant or varies linearly as a function of arc length. Such LVC-curves offer the designer two parameters: the values of geodesic curvature at either end of the line segment. These can then be used to set the tangent directions at the two end-points (similar to the controls available in a Bézier curve in the plane). We have developed a scheme to efficiently calculate a good approximation to such LVC-curves on subdivision surfaces.

We will illustrate the use of this technique with an example from mathematical topology concerning a crossing-free embedding of a graph on a surface of a suitably high genus. E.g., K_{12}, the complete (fully connected) graph of 12 nodes, requires a genus-6 surface for an embedding with no crossings, and the 66 edges of this graph will then divide the surface into 44 3-sided regions. To make pleasing-looking, easy-to-understand models of this partitioned surface, we want to make all edges as "fair" as possible, i.e., keep them nice and smooth with no unnecessary undulations. At the same time we would like to have the edges more or less evenly distributed around the nodes where they join. LVC-curves offer just the right amount of control for our purpose.

3.1 Our Approach

The designer starts by constructing a coarse polyhedral model of the needed genus-6 surface as shown in Figure 3a. Choosing oriented tetrahedral symmetry for this surface and exploiting this symmetry to the fullest, the user only has to construct 1/12 of the surface, which can easily be done with 9 quadrilaterals or 18 triangles. The complete surface is then constructed by composing twelve copies of this fundamental domain with suitable rotations. On this surface, the user now places the nodes of the graph and draws piecewise linear connections between them. If the graph also is given the same tetrahedral symmetry, then this work needs be done only on the fundamental domain, i.e. on 1/12th of the surface.

Our algorithm starts from this linear model. The triangle or quad mesh is the basis of a Loop or Catmull-Clark subdivision surface, and the piecewise linear paths between nodes will be converted into LVC-curve segments. The two refinement processes occur in parallel. For each generation of the subdivision process, the piecewise linear paths are modified so as to approximate a curve with linearly varying curvature (LVC).

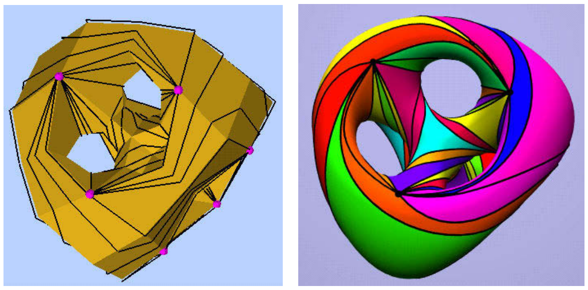

Fig. 3. (a) Initial piecewise linear paths on polyhedral model. (b) Final optimized LVC curves on subdivision surface.

Towards this goal, the vertices where the paths cross over the edges of the control mesh (Fig. 4) are moved with a gradient descent method to approach the desired LVC-behavior. Specifically, each such vertex is moved along the edge on which it lies so as to drive a discretized estimate of geodesic curvature at that point towards the mean of the geodesic curvature values at the two neighboring points on that path. A few dozen iterations of this optimization step are typically sufficient. After this curve optimization process has converged, the surface is subjected to another subdivision step. All linear path segments across all facets in the mesh are then split at the new subdivision edges, and all the path vertices are subjected again to the curve optimization process. This general process loop is repeated until the desired degree of refinement has been reached. The technique works with many popular subdivision schemes.

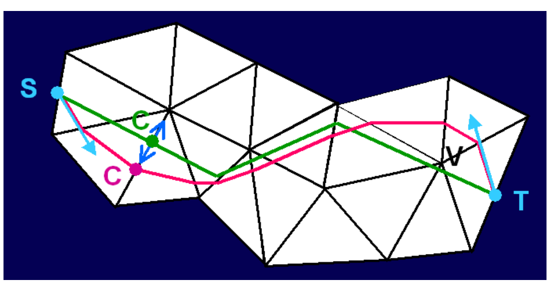

Fig. 4. Optimizing a discretized LVC curve linking S and T; the original path is the one with only three segments.

3.2 Results

The result of this process for the embedding of the K_{12} graph on a genus-6 surface of tetrahedral symmetry is shown in Figure 3b. The LVC curves have been enhanced to black bands to make them more visible, and the nodes of the graph are shown as small hemispheres. The 44 resulting 3-sided facets between the edges have been colored randomly. Thus we are able to provide a crisp visualization model for this difficult graph-embedding problem (Fig. 3b).

4 PARAMETERIZED SHAPE GENERATION

In 1995 I started to collaborate with Brent Collins, a wood sculptor who creates fascinating abstract geometrical shapes [1][2][8]. His work can be grouped into cycles that have a common recognizable constructive logic to them, and which exhibit a timeless beauty that captured my attention immediately when I first saw photographs of his work in *The Visual Mind* [4].

My interaction with Brent Collins was triggered by images of his *Hyperbolic Hexagon* (Fig.5a), which can be understood as a toroidal warp of a six-story segment of the core of Scherk's second minimal surface [7] (Fig.5b). In our very first phone conversation, we discussed the question of what might happen, if one were to take a seven-story segment of such a chain of cross-wise connected saddles and holes, and then bend it into a circular loop. We realized that the chain would have to be given an overall longitudinal twist of 90° before its ends could be joined smoothly. We further envisioned that interesting things might happen in this process: the surface may become single-sided, and its edges could join into a single continuous edge, forming a torus knot.

Since neither of us could visualize exactly what such a construction would look like, we both built little mock-up models from paper and tape (Séquin) or from pipe segments and wire meshing (Collins). In subsequent phone discussions, we expanded the scope of this paradigm. We asked ourselves, what would happen, if we gave the Scherk tower (Fig.5b) a stronger twist of, say, 270°, or of any additional 180° that would allow the ends of the saddle chain to join smoothly. Or, what would a sculpture look like that uses monkey saddles, or even higher-order saddles, rather than the ordinary (biped) saddles of the original *Hyperbolic Hexagon*?

Fig. 5. (a) Collins' *Hyperbolic Hexagon*, (b) 4-story Scherk tower, (c) Collins' *Hyperbolic Heptagon*.

Constructing a realistic maquette of these relatively complex structures, precise enough for aesthetic evaluation, can be a rather labor-intensive process. During the first year of our collaboration, our ideas were coming forth at a rate much greater than what we could possibly realize in physical models. This led me to propose the use of the computer to generate visualizations of the various shapes considered, to judge their aesthetic qualities and to determine which ones might be worthwhile to implement as full-scale physical sculptures. I started to develop a special-purpose computer program that could readily model these toroidal rings of Scherk's saddle chains, as well as all the generalizations that we had touched upon in our discussions. This led to *Sculpture Generator I* which allowed me to create all these shapes interactively in real time by just choosing some parameter values on a set of sliders [9].

In the meantime, Collins had built the *Hyperbolic Heptagon* (Fig.5c), the twisted seven-story ring that we had first discussed on the phone. This two-foot wood sculpture showed us the potential of this paradigm of toroidal loops of saddle chains, and encouraged us to make additional sculptures of potentially much higher complexity. However, such sculptures would require more help from the computer than just the power of previewing the completed shape. Thus I enhanced my program with the capability to print out full-scale templates for the construction of these sculptures. The computer slices the designed geometry at specified intervals, typically 7/8 of an inch, and produces construction drawings for individual pre-cut boards from which the gross shape of the sculpture can then be assembled. Collins still has the freedom to fine-tune the detailed shape and to sand the surface to aesthetic perfection.

This eventually led to our first joint construction, the *Hyperbolic Hexagon II,* which features monkey saddles in place of the original biped saddles. It is possible that Collins could have created this shape on his own without the help of a computer. However, our next joint piece, the *Heptoroid*, a much more complex, twisted toroid, featuring fourth-order saddles (Fig.6a), would definitely not have been feasible without the help of computer-aided template generation.

Fig. 6. (a) Heptoroid, from the collaboration with Brent Collins, and (b) doubly-wound quad Scherk-Collins toroid.

In a further extension of the Scherk-Collins paradigm, it was found, that Scherk's saddle-chain can be wound more than once around the toroidal ring. For a double loop, one needs to choose an odd number of stories, so that they properly interlace on the first and second round. With an appropriate amount of twist and flange-extensions, all self-intersection can be avoided (Fig.6b). With these generalizations of the original paradigm, intricate forms emerged whose relationship to the original *Hyperbolic Hexagon* are no longer self-evident.

4.1 Capturing a Paradigm

In my interaction with Collins, an important new design component is added up front: I have to figure out what it is that I want my sculpture generator program to produce. This means that I first have to see a general underlying structure in a group of similar pieces in Collins' work and extract a common paradigm that can be captured in precise enough terms to be formulated as a computer program. This, by itself, is an intriguing and creative task. Moreover, if the paradigm is captured in a general enough form, it can then be extended to find additional beautiful shapes that have not yet been expressed in Collins' sculptures.

The question arises, whether a commercial CAD tool, such as *AutoCAD*, *SolidWorks*, or *ProEngineer*, would have been adequate to model Collins' sculptures. Indeed, with enough care, spline surface patches and sweeps could be assembled into a geometrical shape that would match one of Collins' creations. But this approach would be lacking the built-in implicit understanding of the constructive logic behind these pieces, which I wanted to generalize and enhance to produce many more sculptures of the same basic type. For that I needed stronger and more convenient procedural capabilities than those that commercial CAD tools had to offer. I chose C, C++, and OpenGL as the programming and graphics environments. The user interface originally relied on Mosaic and later on Tcl/Tk, in which my students had already developed many useful components, such as an interactive perspective viewing utility with stereo capabilities.

Capturing a sculpture as a program forces me to understand its generating paradigm. In return, it offers precise geometry exploiting all inherent symmetries, as well as parametric adjustments of many aspects of the final shape. The latter turns out to be the crux of a powerful sculpture generator. If I build too few adjustable parameters into my program, then its expressibility is too limited to create many interesting sculptures. If there are too many parameters, then it becomes tedious to adjust them all to produce good-looking geometrical forms. Figuring out successful dependencies between the many different parameters in these sculptures and binding them to only a few adjustable sliders is the intriguing and creative challenge.

In practice it turned out that almost every sculpture family that I tackled, required a new program to be written. These programs became my virtual constructivist "sculpting tools." In the last few years, this virtual design environment has become more modular thanks to the SLIDE program library [11] created by Jordan Smith and enhanced with many useful modules for creating freeform surfaces by Jane Yen. Once a new program starts to generate an envisioned group of geometrical shapes, it often will take on a life of its own. In a playful interaction with various sliders that control the different shape parameters, and by occasional program extensions, new shapes are discovered that were not among the originally envisioned geometries. In this process the original paradigm may be extended or even redefined, and the computer thus becomes an active partner in the creative process of discovering and inventing novel aesthetic shapes [10].

4.2 Examples

In the *Family of Twelve Scherk-Collins Trefoils* (Fig.7), the space of parameter combinations is being explored for the range of saddles having from one to four "branches," and for single as well as double loops around the toroidal ring. The concept of a saddle has been extended downwards to also include a single branch ($B=1$), which means that it is just a twisted band. For the case of the doubly wound loop ($W=2$), this band does self-intersect. For the single-branch case, the azimuth parameter has no relevant effect, and thus there are just single instances for $W=1$ and $W=2$. For the cases with 2 and 3 branches, all possible constellations are exhibited, showing both (positive and negative) azimuth values (*An, Ap*) that give front-to-back symmetry for each case. For the fourth-order saddles ($B=4$) the structure becomes rather busy and starts to loose its aesthetic appeal; thus only a single azimuth value is shown for $W=1$ and $W=2$, respectively.

Fig. 7. Hyper-sculpture: *Family of Twelve Scherk-Collins Trefoils* ($B=1$ to 4 from left to right; top: $W=2$; bottom $W=1$).

A graphical interface with individual sliders for each parameter allows the user of *Sculpture Generator I* to explore with ease the space of all Scherk-Collins toroids. For the twelve trefoils in this series (Fig.7), the width and thickness of the flanges was fine-tuned to optimize the aesthetic appeal of each particular trefoil by balancing the relative dimensions of the holes and branches and yielding a pleasing roundness – obviously a rather subjective process. The surface descriptions of the optimized shapes were then transmitted to a Fused Deposition Modeling machine [12] for prototyping. In this process, the geometry of the sculpture is geometrically sliced into thin layers, 0.01 inches thick. These layers are "painted" individually, one on top of another, by a computer-controlled nozzle, which dispenses the white ABS thermoplastic modeling material in a semi-liquid state at 270° centigrade, until the precise three-dimensional shape has been re-created.

Representing geometrical shapes in a procedural form offers several advantages. The designs can easily be optimized with the adjustment of a few parameters. More complex designs can be generated than could be crafted by traditional means. The output can readily be scaled to any size and can be used to produce scale models by layered free-form fabrication. Moreover, interactive play with such parameterized programs extends the horizon of the designer and leads to new conceptual insights; this makes the computer an active part in the creative design process.

4.3 Large-Scale Sculpture

When a sculpture is scaled from a desk model to a large size suitable for a public space, or when the material for its realization changes, the design will often have to be adjusted in subtle ways and cannot just be scaled uniformly. Cross-sectional profiles may have to be thinned or enlarged, flanges may have to be adjusted in thickness, and edges may have to be rounded differently. In this situation it is a big advantage to have a suitably parameterized description of the geometrical form.

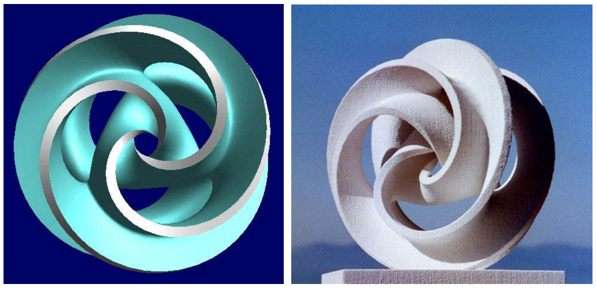

Fig. 8. *Monkey Trefoil*: (a) from Sculpture Generator I, and (b) fine-tuned into a maquette for a 12-foot snow sculpture.

This point was driven home quite clearly in the fall of 2002, when Collins and Séquin were invited on short notice to provide a design for the 13th Annual International Snow-sculpting Championships in Breckenridge, Colorado. First the *Sculpture Generator I* was employed to create a couple of conceptual ideas (Fig.8a) for review by Stan Wagon, the experienced leader of our team. Based on his feedback we could very quickly choose a set of parameters that would balance visual impact, complexity, and the potential for actually being realizable in snow. In a second refinement phase we could then fine-tune the parameters to optimally match the sculpture to the overall dimensions of the snow blocks (10' × 10' × 12 feet tall) that are made available to the competitors. This final CAD description was than used to fabricate a scaled-down maquette on a rapid prototyping machine using a layered manufacturing technique (Fig.8b). The CAD representation also came in handy to make several orthogonal projections (Fig.9a) and cross sectional cuts as blueprints for on-site use during construction.

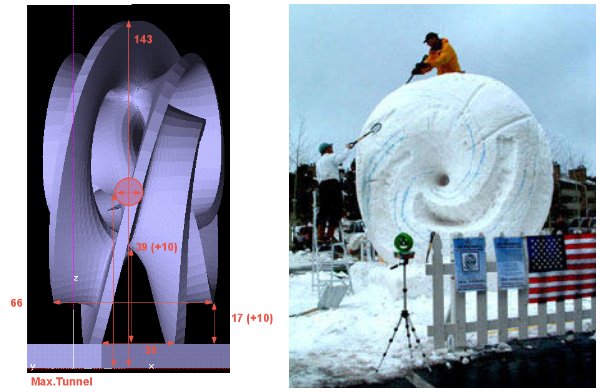

Fig. 9. Construction of snow sculpture: (a) blue-print projection from its side, (b) the definition of the flanges on the basic toroid.

The snow sculpting teams had four days to turn a 20-ton block of snow into some dramatic or whimsical display. To assure the regularity of our desired toroidal shape, we first reduced our block to a 6-foot thick vertical slab. On the faces of this slab we drew concentric circles defining outer perimeter, the major radius, and the small central hole of a perfect torus. We also made a half-circle plywood template with the minor radius of the torus. This template could be "swept" around the outer perimeter to guarantee uniform thickness and roundness of the torus. On this torus surface we could then mark the final visible edges of the spiral flanges (Fig.9b). From there, we proceeded with free-hand sculpting to create the desired shape. It turned out rather nicely (Fig.10) and was awarded the silver medal.

Fig. 10. Snowsculpture: *Whirled White Web*: (a) side, (b) front.

5 TOOLS FOR EARLY CONCEPTUAL DESIGN

The weakest aspect of today's CAD tools is their lack of support for the early, conceptual phase of design. When one starts with a brand new concept, say, '*a bridge in the shape of a Moebius band*,' it is often difficult to enter that first defining shape. 3D sketching tools, as far as they are available, are mostly inadequate. Many artists thus rather use clay, wire, scotch-tape, cardboard, or styro-foam, to make a first conceptual mock-up of a new geometric idea. Effective design ideation involves more than just the eyes and perhaps a (3D?) stylus or other pointing devices. Haptics is a technology that has yet to live up to its potential and to the designers' needs.

So, what is it that we would I like to see in an "ideal" CAD system, useful for the initial design of, say, abstract geometric sculpture or free-form shapes for consumer products? Such a system should combine the best of both the virtual CAD environment and of the real physical world. As virtual elements, any construction parts possess infinite strength, can be glued together easily, and just as easily be disassembled again. Of course, they are not subject to gravity, and thus there is no need for any scaffolding.

On the other hand, some hands-on interaction seems rather desirable. Sweeping a hand through space to define a curve or a profile is often the most natural way to express one's intent. On other occasions, one would like to use a piece of physical material, such as an elastic steel blade, or a piece of heavy velvet cloth to define a shape that is then governed by the intrinsic properties of the chosen material. These physical artifacts would be temporarily be collocated in the virtual context of the emerging design and would there be captured by some vision system or by some fast scanning process. The captured shape is then made available as a new geometric node in the design tree, subject to all the usual manipulations in the virtual design space. Once entered into the system, these shapes could then be assigned new – possibly fictitious – materials properties. Beams might bend like steel wires (MEC); surfaces may stretch like soap films (MES), or they could be subjected to some optimization process under the influence of one of several artificial "beauty" functionals such as a minimum variation functional (MVC, MVS). Good haptic feedback is still a desirable goal, even though it does not look to be close at hand. For many applications dealing with augmented reality, good co-location is also crucial.

6 CONCLUSIONS

Computer-aided design tools can and should be used also for aesthetic shape optimization. Surfaces can now be optimized efficiently using subdivision techniques and tailor-made energy functionals. By using a computer in a real-time, interactive feedback mode, it can become an amplifier for one's creative impulses.

7 ACKNOWLEDGMENTS

I would like to acknowledge the technical contributions by Pushkar Joshi and by Ling Xiao to the prototype CAD tools discussed in this paper. This work was supported by MICRO research grant 03-077, "Collaborative Design Environment," and by the CITRIS Institute, one of the California Institutes for Science and Innovation (CISI).

8 REFERENCES

[1] Collins, B., Evolving an Aesthetic of Surface Economy in Sculpture. *Leonardo*, Vol. 30, No. 2, pp. 85-88, (1997).

[2] Collins, B., Finding an Integral Equation of Design and Mathematics. *Proc. BRIDGES'98*, Winfield KS, pp. 21-28 (1998).

[3] Desbrun, M., Meyer, M., Schroeder, P., and Barr, A., Implicit fairing of arbitrary meshes using diffusion and curvature flow. *Proc. of ACM SIGGRAPH'99*, pp. 317-324, (1999).

[4] Emmer, M., ed., *Visual Mathematics*, special issue, *Leonardo*, Vol 25, No. 3,4, pp. 313-320, (1992); reprinted in M. Emmer, ed., *The Visual Mind*, MIT press (1993).

[5] Hsu, L., Kusner, R., and Sullivan, J., Minimizing the squared mean curvature integral for surfaces in space forms. *Experimental Math* 1, 3, pp. 191-207, (1992).

[6] Moreton, H., and Séquin, C. H., Functional optimization for fair surface design. *Proc. of ACM SIGGRAPH'92*, pp. 167-176. (1992).

[7] Scherk, H. F., Bemerkungen über die kleinste Fläche innerhalb gegebener Grenzen. *J. Reine Angew. Math. (Crelle's Journal)* 13, pp 185-208 (1835).

[8] Séquin, C. H., Virtual Prototyping of Scherk-Collins Saddle Rings. *Leonardo*, Vol. 30, No. 2, pp. 89-96. (1997).

[9] Séquin, C. H., Interactive Generation of Scherk-Collins Sculpture. *Proc. Symp. on Interactive 3D Graphics*, Providence RI, pp. 163-166, (1997).

[10] Séquin, C. H., Computer-Augmented Inspiration. *Proc. ISAMA'99*, San Sebastian, pp. 419-428, (1999).

[11] SLIDE design environment: http://www.cs.berkeley.edu/~ug/slide/docs/slide/spec/

[12] Stratasys Corp., Fused Deposition Modeling machines: http://www.stratasys.com/