*Computer-Aided Design and Applications*
© 2009 CAD Solutions, LLC
http://www.cadanda.com

# Ribbed Surfaces for Art, Architecture, and Visualization

James F. Hamlin[1] and Carlo H. Séquin[2]

[1]University of California, jfhamlin@berkeley.edu
[2] University of California, sequin@cs.berkeley.edu

## ABSTRACT

Sequences of parameterized Hermite curves following with their endpoints along two guide rails are used to create “transparent” surfaces and tubular sculptures. This parameterized set-up allows modeling a wide variety of shapes in a natural way by just changing a few parameters. Potential applications range from mathematical visualization models to architecture.

**Keywords:** ribbed surfaces, ruled surfaces, tubular sculptures, generalized Hermite sweeps.
**DOI:** 10.3722/cadaps.2009.xxx-yyy

## 1. INTRODUCTION

The concept of “ribbed surfaces” emerged from an attempt to reverse-engineer some of the tubular sculptures created by Charles Perry [5]. As a specific example, Solstice [6], located in downtown Tampa, Florida (Fig.1a), has a thick supporting “rail” in the shape of a (3,2) torus knot. The three segments of this dominant curve, which pass through any radial section of the embedding torus, are connected by curved “ribs” that roughly form a hyperbolic triangle. This triangle rotates through 240 degrees as it sweeps once around the major loop of the torus. Once this construction principle is understood (Fig.1b), it seems natural to try to change some of the key parameters so as to accommodate other types of $(p,q)$ torus knots, and create different ribbed surfaces between different segments of the dominant rail forming that knot. As an example, Figure 1c shows a similar sculpture resulting from a rail in the form of a (2,3) torus knot. This surface was obtained by just changing the $(p,q)$ parameters that define the torus knot in the procedural description of the first sculpture.

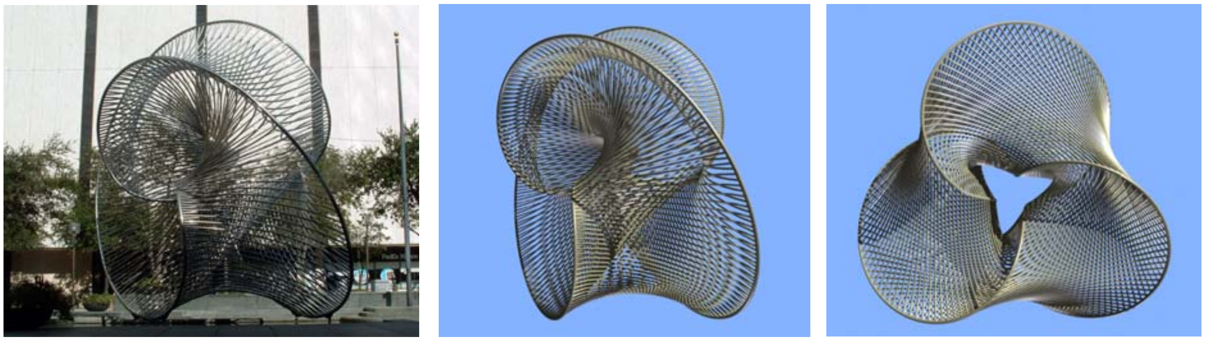


Fig. 1: (a) Solstice by C. Perry; (b) emulation: ribbed (3,2) torus knot; (c) ribbed (2,3) torus knot

The study of variations of this and other tubular sculptures revealed the versatility of this way of generating shapes with curved "surfaces" on a large scale, where shaping an actual surface by forging steel plates or by casting bronze might be prohibitively expensive. Bending and assembling narrow ribbons or thin tubular elements is a much more flexible and affordable approach.

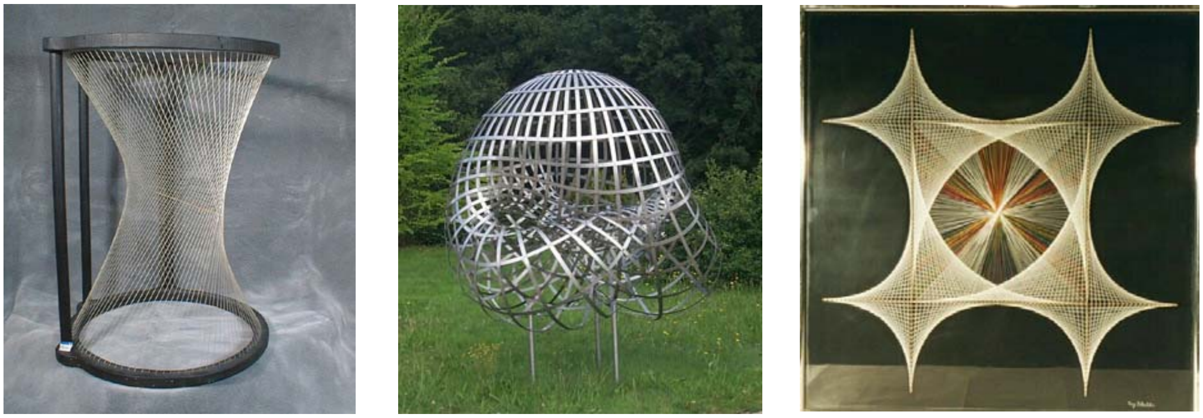


Fig. 2: Models with transparent ruled or ribbed surfaces: (a) hyperboloid; (b) Boy's surface; (c) string art by Ray Schechter [8].

Also, this type of "ribbed representation" can provide surfaces with adjustable "transparency." This is of particular importance when providing visualizations for complex surfaces with high depth complexity, where it is desirable to accommodate a look to the inside or to the backside of the structure. Mathematicians have modeled ruled surfaces with a "ribbed" approach in the form of "string-art" models to depict surfaces such as the hyperboloid depicted in Figure 2a. A sculpture depicting Boy's surface, a finite immersed model of the projective plane, uses metal bands to create a surface of some transparency that allows one to look inside and see the internal lines of self-intersection and the triple intersection point (Fig.2b). And many string-art sculptures make use of the partial transparency that they can achieve for enhanced aesthetic results (Fig.2c). For large public sculptures, a transparent look and feel is often preferred to a massive, solid shape that would cast stark shadows. Tubular construction of ribbed surfaces can provide this airy realization, and it often also reduces the overall weight and construction costs.

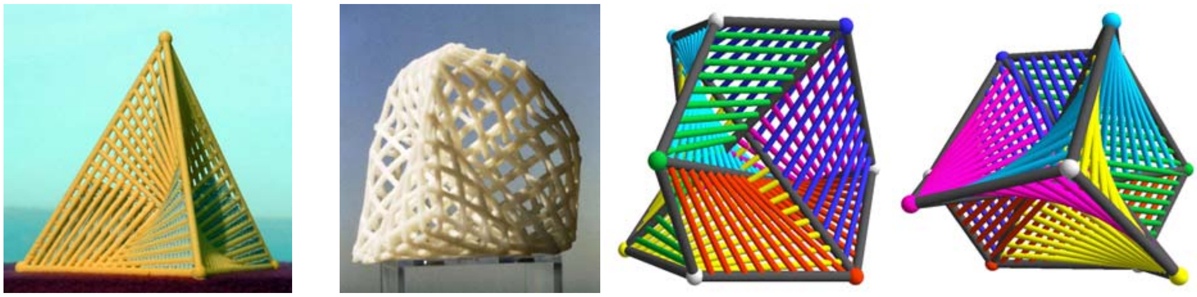


Fig. 3: Mathematical visualization models: (a) hemi-cube; (b) Steiner's Roman surface; (c) (d) two virtual views of a hemi-dodecahedron.

As an example of the added visualization power that ribbed surfaces can provide because of their adjustable "transparency," Figure 3 shows models of abstract polytopes that exhibit intersecting faces when modeled in 3-dimensional space. Figure 3a depicts a physical model of a hemi-cube comprising 4 vertices, 6 edges, and 4 non-planar mutually intersecting quadrilateral faces. Figure 3b is a ribbed model of Steiner's Roman surface, the simplest and most symmetrical object of a single-sided, nonorientable surface of genus 1. Both these models were made on a Fused Deposition Modeling machine. Figure 3c and 3d depict two views of a computer model of a hemi-dodecahedron consisting of six pentagonal faces, 15 edges, and 10 vertices. These polytopes form the building blocks for the construction of some intriguing higher-dimensional polychora of very high symmetry [2] [3] [9]. They have been generated with an early precursor program in our family of ribbed surface generator programs. They mostly just use straight ribs and thus depict models that are mostly bounded by intersecting, piecewise ruled surfaces.

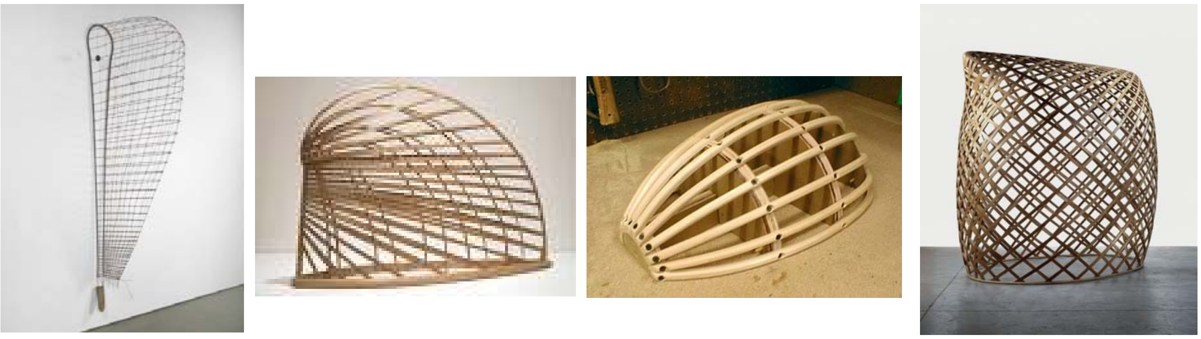


Fig. 4: Ribbed surfaces in the arts by Martin Puryear [4].

Expressibility greatly improves, if we do not just use straight line segments to create ruled surfaces, but can obtain more intricate forms by controlling the shape of the ribs. This will then permit us to make quite intricate mathematical models and intriguing art objects. Figure 4 shows sculptures by Martin Puryear [4] that could be described most appropriately with ribbed surfaces.

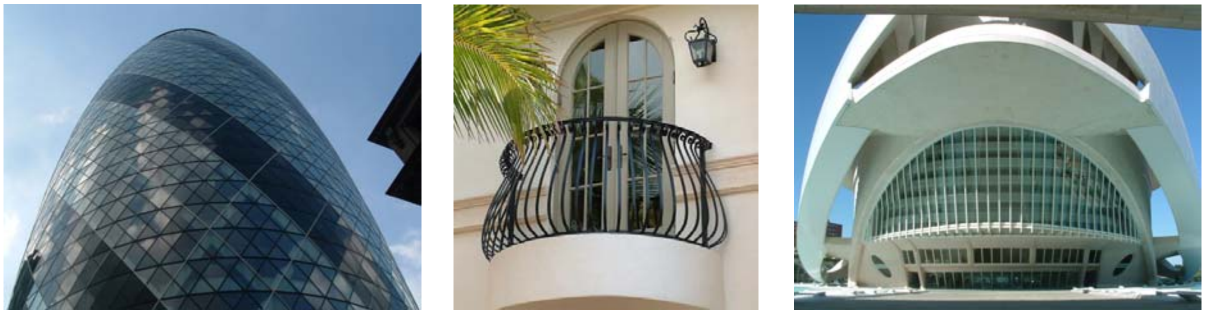


Fig. 5: Use of ribbed surfaces in architecture

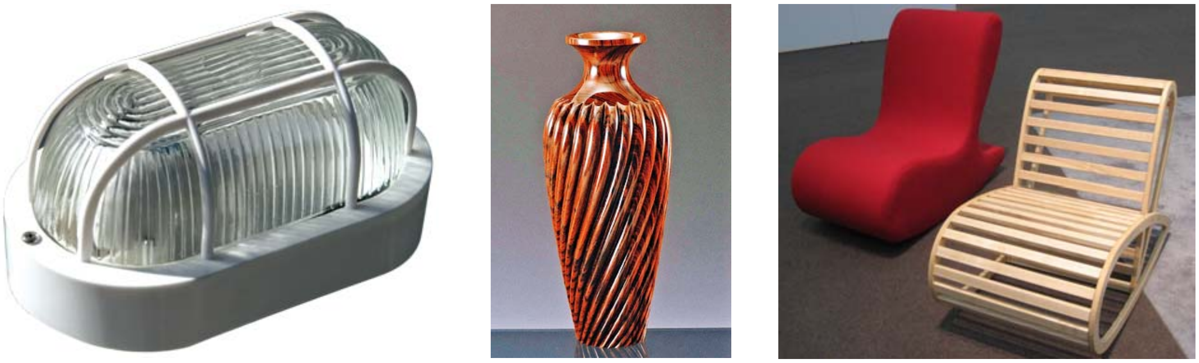


Fig. 6: Further application areas for ribbed surfaces: (a) utility lights; (b) vases; (c) furniture.

There are several other domains where a parameterized ribbed surface may be the natural choice for modeling the objects to be designed. The examples that readily come to mind range from architectural facades and balconies (Fig.5), through consumer products (Fig.6), to ship building (Fig.7). They include utility lights (Fig.6a), artistic vases (Fig.6b), and furniture designs (Fig.6c).

Fig. 7: Ship building is another potential application area for ribbed surfaces.

## 2. BASIC SET-UP OF RIBBED SURFACES

Ribbed surfaces are a versatile generalization of the well-known ruled surfaces. In both cases the generating principle is to slide some secondary curve, forming the “ribs,” along two primary curves, which we call the “guide rails.” In the case of the ruled surfaces, the ribs are straight line segments; in our case we allow the ribs to be more general curves, which have their end points located on two different segments of the rail curve(s). Our prototype implementations include circular arcs and Hermite splines. In each case a small set of parameters controls the behavior of the ribs as their end points progress along the rails.

### 2.1 Parameterization

In order to create a ribbed surface, we need to define one or two parameterized guide rail curves and the rail cross-sections that will be swept along the rail curve(s). For the ribs we have to specify: the total number of ribs; their cross-section information; two parameter intervals $[sb0, sb1]$ and $[se0, se1]$, for the beginning (b) and end (e) point locations on the appropriate guide rail(s); an application-dependent set of geometric end-point parameter functions that define the shape of the individual ribs as the sweep parameter $s$ is running through the interval from $s0$ to $s1$. These parameters control turning angle and embedding plane for circular arcs, or they specify the end tangents of cubic Hermite rib curves in various ways. They may be enhanced in the future to also control the curvature values at the rib ends in the context of quintic Hermite splines. The specified number of ribs is then generated and uniformly positioned over the domain of $s$. Some extra higher-level control functions make it easy for the user to create sets of rib end points that coincide or, alternatively, are evenly interleaved when multiple sets of ribs are ending on the same rail. Different shape parameters are made available depending on the application; these issues are discussed below.

Consider as a first examples the case of a single guide rail (Fig.8a), parameterized from $s$=0 to $s$=1, starting from an arbitrary starting point. The beginnings of the ribs would then be placed uniformly over the interval $s$: [0, 0.5], and the end-points over the interval $s$: [0.5, 1.0].

As a second example, we assume that we have two separate guide rails, Gb and Ge, for the rib end-points (Fig.8b). The beginnings might be spread over the interval sb: [0, 0.5] on rail Gb and the end-points over the interval se: [0, 1.0] on Ge.

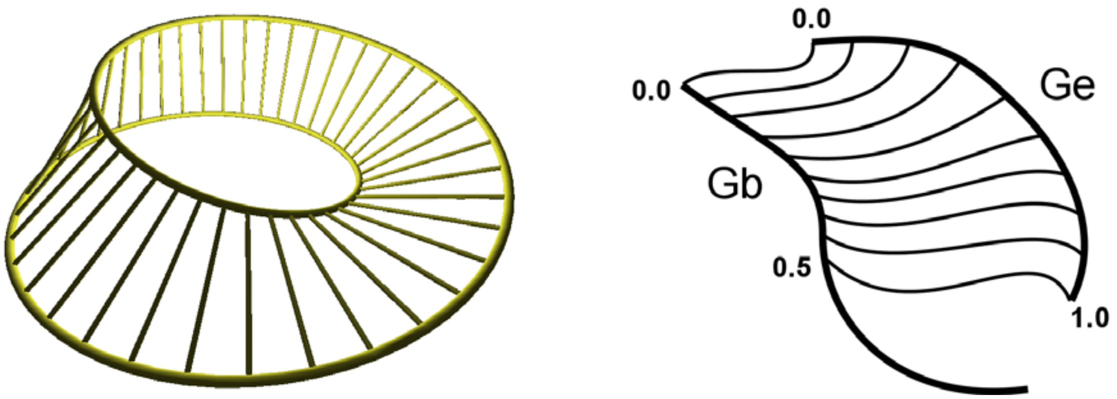


Fig. 8: Rib end-point placement: (a) on single guide rail, (b) on two separate guide rails.

### 2.2 Rib End Conditions

We have employed several alternative coordinate systems to specify the end conditions of the ribs, and thus to define the shape of the ribs. When the ribs are represented as cubic Hermite curves, there are the following constraints: two endpoint positions and two tangent vectors. The endpoints are simply points along the guide rail curves, evaluated at the proper parameter values of the sweep parameter for that rib. The end tangents might be oriented in any direction, and different coordinate systems have different advantages for specifying the tangent directions, as will be discussed below. The orientation angles of the end tangents in these rib-local coordinate systems may vary as a function of the sweep parameter *s*. In the simplest case, the rib end-condition parameters are specified for the start and end of the sweep and are linearly interpolated for all ribs in between. More complicated functions can easily be introduced.

The most general coordinate system would simply use the Frenet frame of the rail curve(s) as the coordinate system to define the rib end tangent vectors (Fig. 9a). A first parameter $\tau$ specifies the angle between the rail tangent and the rib tangent. A second parameter $\rho$ specifies the rotation angle around the rail tangent, starting from the osculating plane; thus the special parameter combination of $\tau$=90 and $\rho$=180 would result in a rib that takes off perpendicularly to the rail curve in the outward direction of its bend. This coordinate system is most convenient when we are primarily concerned with the structuring of the rib ends around the rail curves. Hermite rib curves have an additional ("velocity") parameter to specify the length of the tangent vector (derivative) at each end.

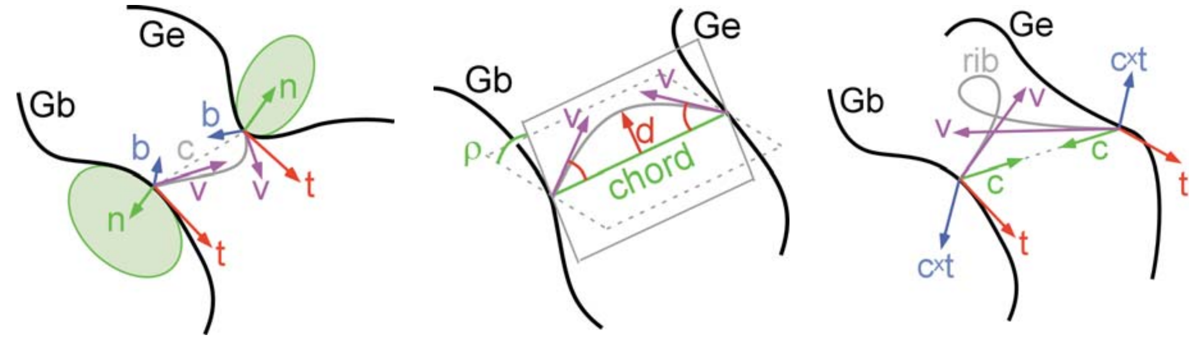


Fig. 9: Coordinate systems to specify rib end conditions: (a) based on Frenet frame of rail curve, (b) special reference frame for planar ribs, (c) oblique coordinate systems.

Often we are more concerned with the overall shape of the ribs themselves than with their end conditions. In this case we specify the geometry of the rib endings in a coordinate system that is more intimately tied to the bulk of the rib. The chord that connects the two end points of the rib sliding along the rails forms the dominant axis of such a coordinate system. A plane that passes through this dominant axis is selected by a single angle parameter $\rho$. The plane defining the zero angle depends on the application. In many cases, it is desirable to keep the ribs planar and symmetric; in this case all that needs to be specified is the orientation angle $\rho$ of the rib plane around the dominant coordinate axis and the tangent angles $\tau$ that the rib ends form with the chord (Fig.9b). Alternatively, the amount of bending of the rib in the given plane could be specified as an offset distance $d$ of the rib mid-point from the chord mid-point.

As a third alternative we have used a mixture of these two paradigms, resulting in end-point coordinate frames in which the three axes may not be mutually orthogonal (Fig.9c). For each end point, the first axis is the vector along the chord from that point to the opposite rib endpoint. The second axis is the tangent of the guide rail curve. The third axis is simply the cross product of the first and second axes. Some default functions based on the length of the chord and on the angle between the guide rail tangent and the chord may be used for setting the rib tangent direction and velocity; such default functions may cover a large set of practical applications.

The proper parameterization of the rib end conditions is ultimately application-specific and may reflect both aesthetic and pragmatic requirements. For example, if planar ribs approximating elliptical arcs are the preferred solution, the second coordinate system described above might be used, perhaps with a further abstraction to specify the rib shape in terms of elliptic eccentricity.

### 2.3 Rib Optimization

Hermite curves are mathematically smooth in their parameter domain, but they are not necessarily geometrically smooth; cubic Hermite curves might still have loops, cusps, or folds. When the intention is to realize a ribbed surface by bending physical materials, geometric smoothness is desirable. To this end one can replace Hermite curves with optimized geometric Hermite curves or composite optimized geometric Hermite curves that adjust the magnitudes of the curve end tangents so as to minimize overall strain energy [11]. The initial endpoint tangents could be specified as detailed above, and then sent to an optimization procedure to construct the energetically optimized and geometrically smoothed curves.

### 2.4 Comparison to Other Sweeps

Ribbed surfaces bear some similarity to another kind of “swept” surface, often called “lofted surfaces” or “skinned surfaces.” These latter surfaces are specified by a sequence of two or more cross-section curves, called “profile” curves, and the surface is obtained by interpolating between each adjacent pair of profiles. The Maya 3D modeling editor [1] generalizes this concept to a construct called “Birail” curves. Profile curves are defined with two fix-points each, which can slide along two separate “rail” curves, while the various profiles are properly scaled and oriented so that each of the two fix-point remains on one of the rail curves. The surface is again obtained by interpolating point by point between adjacent pairs of profiles, which are specified in with reference to the two guide rails. “Birail” sweep surfaces require the explicit input of a discrete number of profile curves whose shapes are then interpolated to generate intermediate ribs, and eventually the whole surface.

In contrast, our general ribbed surfaces permit the procedural generation of unique rib curves for each value of the sweep parameter. Interpolation is not done in a low-level, geometrical manner between adjacent curves, but rather in the parameter space that defines the ribs. This adds a level of abstraction and extra degrees of freedom for the generated surfaces. It also reduces the amount of data that has to be specified explicitly for a given surface. Consider the specification of a ribbed surface with $N$ ribs where the ribs are specified by $M$ control points. The Birail specification would require the transmission of O($M$*$N$) data values, whereas the ribbed surface would require only O($M$) input parameters, if the ribs can be specified satisfactorily in terms of some constant number of interpolated parameters to the rib generator function.

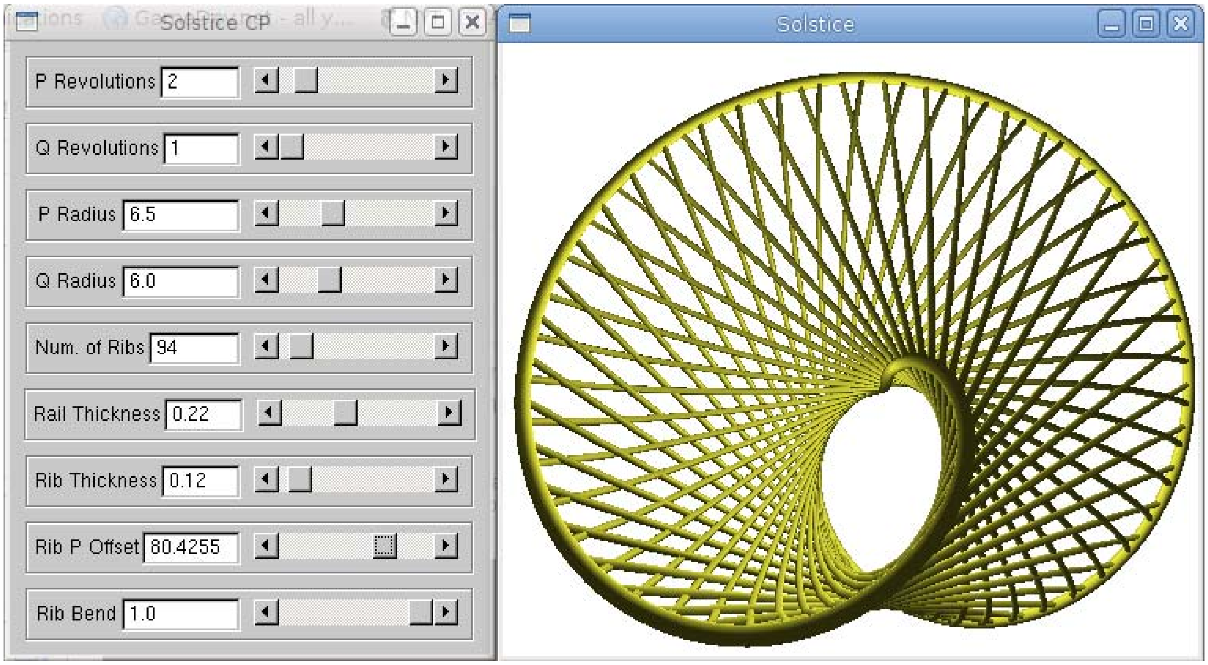
Moreover, there is, of course, the obvious difference in that most traditional sweeps are continuous 2D surfaces, while ribbed surfaces are composed of sequences of distinct “ribs” that suggest such a surface. As has been discussed above, this feature is what makes ribbed surfaces amenable to some novel uses in certain applications.

## 3. DEMONSTRATION PROGRAMS

In the following we describe in some detail a few demonstration programs that have somewhat richer user interfaces and thereby allow the user to model a much wider range of shapes without having to touch the source code of the ribbed-surface generator environment.

### 3.1 Perry’s *Solstice* Sculpture

As a first example, we will discuss a sculpture generator program based on Perry's *Solstice* sculpture (Fig.1). The program is implemented in C++, using OpenGL for rendering and GLUI for creating the graphical user interface. In the current implementation, when any parameter is changed, the entire surface is regenerated. Despite this, performance remains interactive.



P Revolutions 2
Q Revolutions 1
P Radius 6.5
Q Radius 6.0
Num. of Ribs 94
Rail Thickness 0.22
Rib Thickness 0.12
Rib P Offset 80.4255
Rib Bend 1.0

Fig. 10: Solstice program: Screen shot of user interface.

All geometry generated will always have a guide rail that is a (*P*,*Q*) torus knot embedded in a torus that is specified by its defining major and minor radii. Thus the first four parameters that the user can control (Fig.10), "P" and "Q", "P Radius" and "Q Radius", specify the (*P*, *Q*) torus knot as well as the dominant radius of the torus (P Radius) and the thickness of the toroidal arm (Q Radius). *P* and *Q*, of course, must be mutually prime, lest the guide rail decompose into multiple, intertwined torus knots. A parameter "Rail Radius" controls the thickness of the circle swept along the rail curve to produce the rail geometry. In a similarly straight-forward manner the user can set the “Number of Ribs” and the “Rib Radius.” The starting points of all these ribs will now be distributed uniformly along the whole length of the torus knot; their endpoints are always 1/*P* ahead on the same torus knot rail in parameter space. A particularly interesting parameter is “Rib Offset;” it shifts the endpoint of the rib forward or backwards by an additional amount. For Rib Offset equal to zero, the ribs will always form 1/*P* P-gons that lie in the cross-sectional slicing planes of the toroidal arm; they may be simple regular P-gons or star-P-gons, depending on the torsion of the torus knot. If Rib Offset is changed by a small amount, then these P-gons break open and become helical pathways. Some internal program logic guarantees that the Rib Offset always snaps to a value that guarantees that the end points of the set of ribs that land on the same rail in the same neighborhood either match exactly or are offset by exactly ½ of the rib spacing to obtain a regular interleaving.

In this program we restrict ourselves to symmetrical, planar ribs and thus use a local coordinate system that is centered around the mid-point of the chord, using a default plane for the ribs that passes through the special radial vector in the torus arm that points toward the chord mid-point. Within this plane we now control the bending of the rib by adjusting the angle between the chord and the rib endpoint tangent with the parameter "Rib Bend." Figure 11 shows a more complex (5,4) torus knot sculpture with fairly dramatic bending of the ribs.

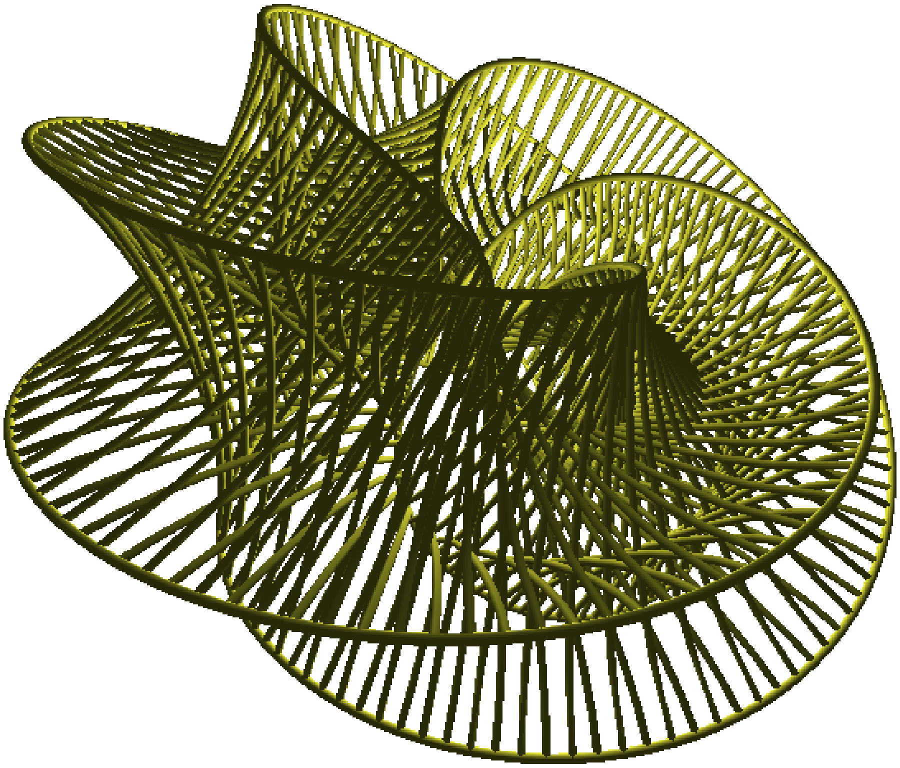


Fig. 11: Solstice program: Self-intersection Moebius prism based on (5,4) torus knot.

### 3.2 Perry's *Early Mace* Sculpture

We have also used the ribbed surface paradigm to emulate another one of Perry's sculptures, called *Early Mace*, installed in Atlanta, Georgia (Fig.12a). Here the guide rails are pairs of simple semicircles with two widely different radii. The ribs are almost quarter circles and are always connecting one of the large hemi-circles with one of the small ones. In our emulation program, implemented with a TCL module in the Berkeley SLIDE environment [10], the rib end conditions are specified using the more general coordinate system in terms of the "velocity" vector, guide rail tangent, and their cross product (Fig.12b). As a simple but intriguing variation we show what happens if we reverse the bulge of all the ribs so that they form an overall convex, nearly spherical shape with two semicircular slots (Fig.12c).

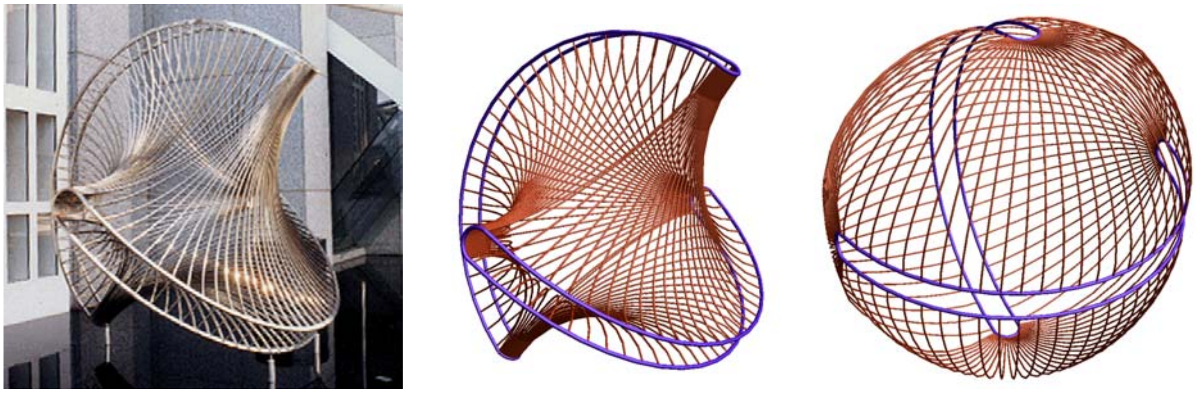


Fig. 12: Early Mace by C. Perry: (a) original sculpture; (b) our emulation; (c) a convex variation.

### 3.3 Ribbed Helical Arches

A third demonstration application generates hemi-elliptical arches traced out by a ribbed double helix. The two guide rail curves form this double helix. The radius of the two helices at the endpoints and the radius at the center may be adjusted independently. The double helix is curved to follow a hemi-elliptical path, which is specified in terms of the parameters of interest for the overall arch structure: the width of its base and its height. The rib end conditions are specified, as in the *Early Mace* emulation of section 3.2, using the more general coordinate system of "velocity" vector, guide rail tangent, and their cross product. The ribs take off in the direction of this cross product, and these end conditions are specified for each endpoint of the arch to cause them to take off upwards at both ends. The values are then interpolated for each rib in-between. Figure 13 shows one of the sculptures output by this program, which was rendered in Pixie, an open-source RenderMan renderer.

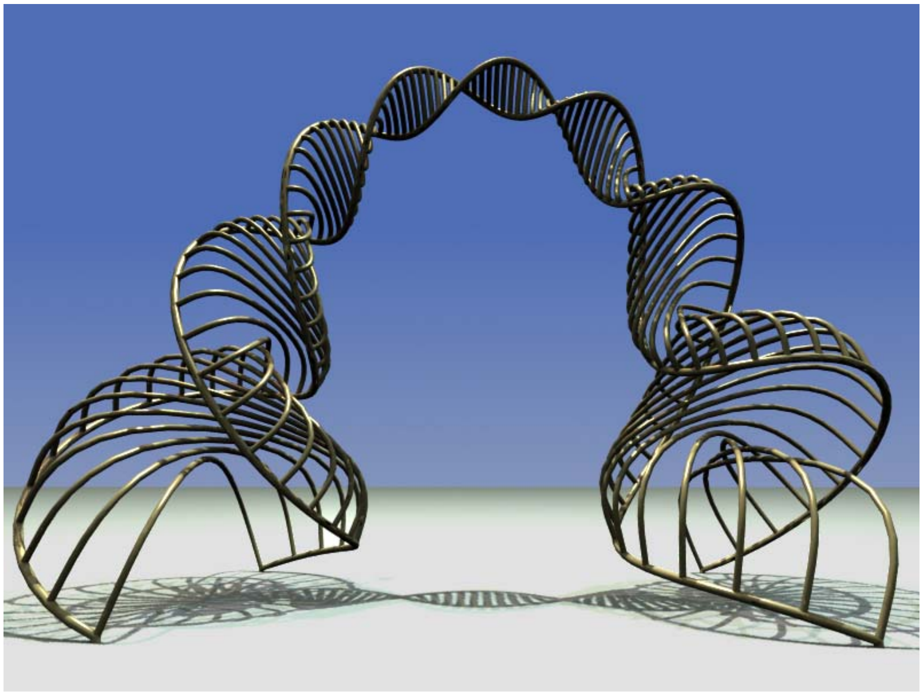


Fig. 13: A ribbed double helix forming an elliptical arch.

## 4 SUMMARY AND CONCLUSION

The notion of "ribbed surfaces" has been introduced as a generalization of the well-known ruled surfaces. Rather than just sweeping straight line segments between two generator rails, we may use, for instance, Hermite curves for the ribs, for which the end conditions are specified by parameterized functions of the global parameter *s* that controls the iterative placement of all the ribs and their sweep-progress along the generator rails. One of our prototype implementation uses cubic Hermite curves in which we can control the first-degree derivative parameters at the end points of the ribs. Future extensions could include quintic Hermite curves, which would allow us to control the curvature at these points, or the use of optimized geometric Hermite curves, as described above.

In design applications such as those described in Section 3 or shown in Figures 2 and 3, it would be quite useful, if the system could detect mutual intersections between ribs. This is particularly important if the structure is to be realized from actual physical tubular elements. Mutual intersections between any pair of geometrical elements could indeed be detected with well know but somewhat computationally expensive collision detection modules.

For many applications, included those presented here, finding such potential intersections is not a dominant concern. In the case of Perry's sculptures, the ribbed surfaces do not actually intersect one another. Thus a simple visual inspection is sufficient to make sure that a set of rib parameters has been chosen that keeps the various parts of the ribbed surface from bulging into one another.

On the other hand, in some of the mathematical models, where the key goal is to present intersecting surfaces with this ribbed implementation style, detecting intersections is often the easy part of the problem. The much more difficult challenge is to find a set of rib parameters that avoids *all* intersections between any two ribs, without breaking the regularity of the rib spacing along the surface. Complicated search and optimization techniques might have to be applied to look through the whole combinatorial space of parameter value combinations to find an acceptable solution. The models displayed in Figure 3 were fabricated on rapid prototyping machines base on layered manufacturing technology. Fortunately, these machines have no problem building mutually intersecting tubular structures. To obtain truly intersection-free model, such as the one depicted in Figure 3b, requires a higher-level understanding of the structure being designed and a careful planning of the positions of individual ribs before the shape of the resulting surface is specified in detail.

## REFERENCES

[1] Autodesk: *Learning Autodesk Maya, The Modeling & Animation Handbook,* Autodesk Maya Press, (2008).

[2] Coxeter, H.S.M.: *Ten toroids and fifty-seven hemi-dodecahedra*, Geometriae Dedicata, 13, (1982), pp 87-99.

[3] Coxeter, H.S.M.: *A Symmetrical Arrangement of Eleven Hemi-Icosahedra*, Annals of Discrete Mathematics 20 (1984), pp 103-114.

[4] Elderfield, J.; Reede, E.; Powell, R.; and Auping, M.: *Martin Puryear*, Museum of Modern Art, New York, 2007.

[5] Perry, C. O.: *Charles O. Perry*, deCesare Design Associates, Darien, CN, 1990.

[6] Perry, C. O.: *Ribbed Sculpture*, personal website, 2009, http://www.charlesperry.com/Ribbed.html

[7] Pixie, open-source RenderMan renderer, 2009, http://www.renderpixie.com/

[8] Schechter, R.: *Geometric String Art*, website 2009, http://www.trocadero.com/condonkay/catalog/Fine_Art:Mixed_Media:Relief.html

[9] Séquin, C. H.; Lanier, J.: *Hyper-Seeing the Regular Hendeca-choron*, Proc. ISAMA, Texas A&M, May 17-21, 2007, pp 159-166.

[10] Smith, J.: *SLIDE*, program website, 2003. http://www.cs.berkeley.edu/~ug/slide/docs/slide/spec/

[11] Yong, J; Cheng, F.: *Geometric Hermite curves with minimum strain energy*, Computer Aided Geometric Design 21, 3 (2004), pp 281-301.