Bender: A Virtual Ribbon for Deforming 3D Shapes in Biomedical and Styling Applications

Ignacio Llamas Alexander Powell Jarek Rossignac Chris D. Shaw

College of Computing and GVU Center
Georgia Institute of Technology
Atlanta, GA 30332-0280

ABSTRACT

In contrast to machined mechanical parts, the 3D shapes encountered in biomedical or styling applications contain many tubular parts, protrusions, engravings, embossings, folds, and smooth bends. It is difficult to design and edit such features using the parameterized operations or even free-form deformations available in CAD or animation systems. The Bender tool proposed here complements previous solutions by allowing a designer holding a 6 DoF 3D tracker in each hand to control the position and orientation of the ends of a stretchable virtual ribbon, which is used to grab the shape in its vicinity and to deform it in realtime, as the designer continues to move, bend, and twist the ribbon. To ensure realtime performance and intuitive control of the ribbon, we model its centerline as a circular biarc and perform adaptive refinement of the triangle-mesh approximation of the surface. To produce a natural and predictable warp, we use the initial and final shapes of the ribbon to define a one-parameter family of screw-motions. The deformation of a surface point is computed by finding its locally closest projection, or projections, on the biarc and by applying the corresponding screws, weighted by a function that decays with the distance to the projection. The combination of these solutions leads to an easy-to-use and effective tool for the direct manipulation of organic or stylized shapes.

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Curve, surface, solid and object representations*; I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction Techniques*

Keywords

space-warp, deformation, biarc, 6 DOF tracker, adaptive subdivision

Submission to ACM Symposium on Solid and Physical Modeling 2005

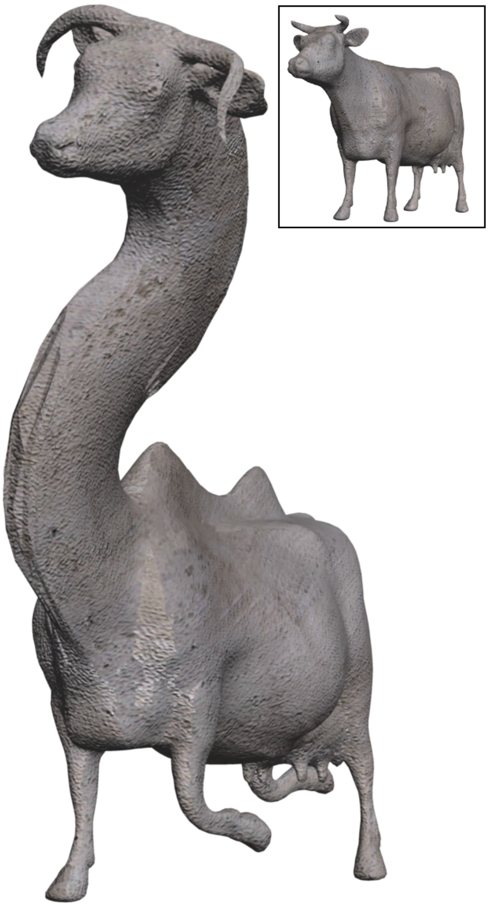

Figure 1: A cow (top) has been deformed in 6 Bender steps: two steps to bend the legs, two more to bend the horns downwards, one to create the humps, and the last one to stretch the neck.

1. INTRODUCTION

The use of two-handed 3D interaction for sketching and editing 3D shapes has the potential to enhance productivity and artistic freedom for designers. We present an interactive surface deformation tool, Bender, which is not meant to replace existing 3D sculpting tools, but complements them by providing unprecedented ease for bending and twisting 3D shapes through direct manipulation.

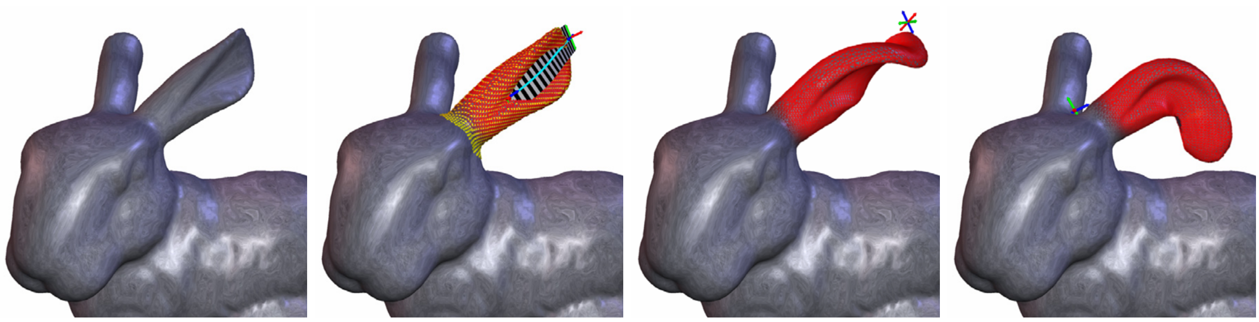

Figure 2: The initial shape of a bunnys ear (left) is stabbed by a user-controlled ribbon (left center). The user can change the shape of the ear interactively by moving the ribbon, bending it (right center), or twisting it around its centerline wire (right).

Like several previously proposed approaches, Bender lets the user control a local space warp, which is applied to the vertices of a triangle mesh representation of the surface being edited (Figure 2). In contrast to shape deformations based on space warps that satisfy point displacement constraints [8] and even position and orientation constraints [17], [33], Bender makes it easy to bend and twist long protrusions, which dominate biomedical and manufactured shapes.

A 3D warp is specified by grabbing a subset of space around a user-controlled virtual ribbon and then changing the shape of the ribbon interactively using a Polhemus [40] tracker in each hand (see Figure 3). The ribbon is defined by a centerline curve (wire) and by the distribution of the twist of the normal plane along the wire. The six degrees of freedom of each 3D tracker control the position of one endpoint of the wire, the direction of the tangent at the endpoint, and the twist of a ribbon around this tangent direction. The concept of the ribbon is used to communicate to the user how this twist is distributed along the wire (Figure 3). A ribbon interpolating these end-conditions is constructed and displayed in realtime as the user moves the 3D trackers.

The 3D warp is defined by the shape of the *initial* ribbon, captured when the user *presses* the tracker's grab button, and by the *final* shape of the ribbon, which is frozen when the user *releases* the grab button. Points in the vicinity of the centerline wire of the initial ribbon follow the displacement and rotation of their locally closest points on the wire. The effect of the warp decays with the distance to the wire, fading to zero when that distance exceeds the radius of influence. Consequently, the Region of Influence (RoI) of the warp is limited to a tubular region around the wire of the initial ribbon. The surface within the grabbed RoI is dragged by motions of the hands, and the resulting deformed surface is displayed in real-time as the user moves, bends, stretches, and twists the wire.

The extent of the Region of Influence and the decay function may be quickly adjusted by the user to support large global deformations or the creation of small details. A decay function with a plateau may be used to ensure the preservation of fine details near the wire.

A space warp that interpolates position, shape, and twist of the initial and the current ribbon is computed and applied in real-time to the surface being edited. This immediate graphic feedback supports the direct manipulation of 3D shapes. When the desired shape is obtained, the user releases the grab button, hence freezing the warp and saving the new shape, which may be further edited by subsequent warps.

Each warp is entirely defined by two pairs of coordinate systems. The initial pair defines an initial ribbon used to grab a portion of the shape when the user presses a button. The final pair is captured when the user releases the button and defines the final ribbon.

In addition to providing a new direct manipulation paradigm, we propose a representation of the ribbon and a mathematical model of the warp that offer specific advantages over previous approaches.

We use a smooth circular biarc for the wire. This choice leads to intuitive control of the wire and to fast computation of the normal projections of any arbitrary point onto the wire. In fact, we prove that at most two such projections exits.

We formulate the warp in terms of a one-parameter family of screw-motions that map local coordinate systems along the initial wire to the corresponding coordinate systems along the current or final wire. This choice leads to natural shape warps that do not surprise the user, allowing the designer to easily estimate the shapes of the initial and final ribbon that will produce the desired shape change.

Furthermore, the combined use of the biarcs for the wire and of the screw-motions family solves the classical tearing problem, which occurs when a vertex of the mesh is closest to one portion of the initial wire and a neighboring vertex is closest to a distant portion.

We demonstrate the ease of use and power of this formulation in an interactive system called Bender. Although we have not used any spatial indexing data structure to optimize performance, Bender provides 3D graphics feedback at more than 10 frames a second when manipulating surfaces with about 70,000 triangles. We use adaptive mesh subdivision to refine the surface in areas where the initial tessellation may produce visible artifacts.

The rest of the paper is organized as follows. In Section 2, we review relevant prior art. In Section 3, we present implementation details and design choices. Section 4 describes some of the biomedical applications of Bender. Finally, we show results and conclude.

2. RELATED WORK

A variety of approaches have been suggested for creating and changing the shape of a surface more than one vertex at a time. The challenge is to find a pleasing, predictable and controllable method that can be applied in real-time. Some approaches construct surfaces that interpolate 2D profiles [28] or 3D curves [49], [22]. Others provide means for the direct drawing of surfaces [43] or for space painting and carving [19].

An alternative to these shape creation techniques is the warping or deformation of existing shapes. Based on a designers natural knowledge of the physical world, we strive to approximate material properties such as elasticity or plasticity. See [36], [20] and [17] for reviews. However, simulated physical realism is generally too expensive for real-time feedback. Furthermore, it unnecessarily limits the scope of the deformations. Hence, our solution does not attempt to mimic the physics of a real ribbon or of real materials being deformed. However, it offers a simple and intuitive map between hand gestures and space warps. The cost of computing the warp parameters is negligible and its effect appears physically plausible and quite predictable. Furthermore, geometric warps are usually preferable to physical material simulation, since they offer a broader set of editing possibilities. For example, the bowl-like shape in Figure 11 would be nearly impossible to achieve using the physical simulation of a real material.

Space warping and morphing techniques are thoroughly reviewed by [21]. We offer below a partial review of the most relevant work in this area.

Barr [4] introduced the general space deformations twist, bend and taper. Chang and Rockwood [11] used a generalized de Casteljau approach to extend Barr's technique. Sederberg and Parry [45] introduced the free-form deformation (FFD), based on lattices of control points and trivariate Bernstein polynomials. Hsu et al. [27] developed a version of FFD that allows direct manipulation, while Coquillart [12] and MacCracken and Joy [35] extended FFD to support more general lattices.

Borrel and Bechmann [7] and Borrel and Rappoport [8] developed real-time techniques for computing space warps that simultaneously interpolate several point displacement constraints. Bill and Lodha [6] and Allan et al. [1] developed systems that displaced a selected vertex and its neighbors by a set of decay functions. Modern software packages, such as Alias®Maya®[1] and Discreet®3ds max®[2], also allow weighted manipulation of vertices with an adjustable decay function. Previous work by Fowler [16], Gain [17] and Llamas et al. [33] support not only point displacement constraints, but also orientation constraints on points. Pauly et al. [38] used a similar deformation model on a system to edit point-sampled geometry. Milliron et al. [37] introduced a general framework for geometric warps in which the use of orientation constraints is also possible. Dachille et al. [29] presented a system that integrated haptics with a dynamic B-spline model, which also allowed the direct manipulation of normal and curvature constraints.

Axial Deformations [32] used piecewise linear curves of any shape as the axis for a generalized cylinder with variable radii and local frames at key points. Wires [47] takes curve based deformation techniques further, but at a higher computational cost. ShapeTape [3] uses B-spline curves to create surfaces and the Wires deformation model to deform shapes using a 3D tracked and instrumented flexible rubber tape.

Other approaches to shape deformation are dependent on the particular underlying representation. Such is the case of Forsey and Bartels Dragon editor [15], which relies on hierarchical refinement of B-spline surfaces. Similarly, Zorin's system for multiresolution mesh editing [50] allows vertices at different levels of subdivision to preserve details by using adjustment vectors defined in local frames. Turk and O'Brien [48] approach shape modeling by constructing an implicit surface from scattered data points and normals. Du and Qin [13] combine PDEs and implicits to achieve interactive manipulation and deformations. Several authors have developed techniques for computing piecewise polynomial surfaces that interpolate points and curves in position and possibly orientation [26], [9], [2]. Pernot et al. [39] use a feature-based approach to free-form surface editing. Celniker and Gossard [10] combine deformable curves and surfaces which try to minimize an energy functional while responding to user controlled constraints and loads.

Twister [33] uses a pair of 3D trackers to grab two points on or near a surface and to warp space with a weighting function that decays with increasing range from the trackers. The work described in the present paper can be viewed as an extension of this approach. It is particularly useful for bending long shapes and for operating on elongated regions of influence. Based on a grab-and-drag shape deforming operator, it allows the direct manipulation of shape. It does not limit the user's interaction to control points and it does not restrict the operations to be axial deformations.

Since designers are naturally capable of operating in 3D space, and since 3D surfaces are to be manipulated, we chose to explore a shape operator that provides a natural control of position and orientation of selected regions of space. We justified this decision on the basis of a well understood interaction style [46], [25] and readily-available hardware [40]. Using two hands allows the user to adopt both asymmetric [24] and symmetric operations with both hands on the surface being edited. Asymmetric operations allow the dominant hand to adjust fine detail while the non-dominant hand sets up context (position and orientation of the model). Symmetric operations allow each hand to edit shapes with its 6 DoF cursor. Offering natural control over six degrees of freedom per hand simplifies the design of complex warps, which will otherwise require a laborious series of 2 DoF or 1

[1] Alias and Maya are registered trademarks of Alias Systems Corp. in the United States and/or other countries.

[2] Discreet and 3ds max are trademarks of Autodesk Canada Inc./Autodesk, Inc. in the USA and/or other countries.

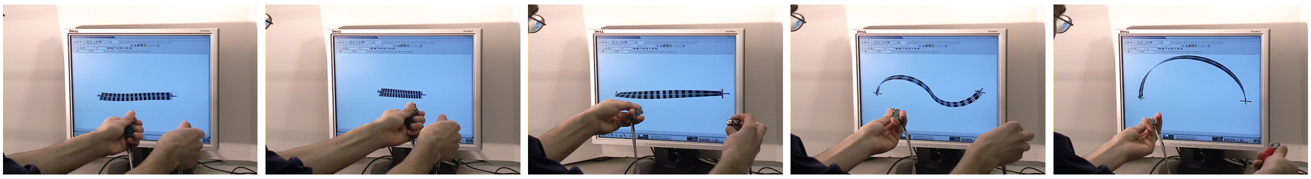

Figure 3: The user manipulates two 6 DOF trackers to control the shape of the ribbon. Note that only the ribbon is shown and no shape is being deformed.

DoF operations if only a mouse is available. Other user interface issues with high degree-of-freedom input devices are further explored in [23].

3. IMPLEMENTATION DETAILS

In this Section, we describe how a central line of the ribbon is computed to interpolate the position and tangent directions at the end-points. Instead of using a cubic parametric curve to solve this Hermite interpolation problem, we use a biarc curve [42] made of two smoothly joined circular arcs. Then, we explain how the additional twist imposed by the two trackers is interpolated along the central line and how it defines a ribbon, and hence defines a coordinate system at each point along the ribbon. We then show how the projection of an arbitrary vertex onto the biarc may be computed efficiently and discuss the key fact that the distance between a point and the biarc may have at most two local minima. Each projection defines a coordinate system on the initial ribbon and the corresponding coordinate system on the final ribbon. To avoid the tearing of space, we compute both projections and use a weighted average of the warps they each define. Then, we describe how a rigid-body screw motion may be computed to interpolate between a starting and an ending coordinate system and also how we combine the effects of two such screw motions. We discuss different weighting functions which, based on the distance between the vertex and its projection on the biarc, determine how much of the rigid body motion will be applied to the vertex. In particular, we discuss the merit of a function with a plateau for preserving the shape of local details. We then discuss techniques that we have developed for ensuring continuity, smoothness, and compactness of the warp along the ribbon. Finally, we provide a brief discussion of the simple strategy we use to perform an adaptive subdivision of the surface where it is required by the extent of the warp.

3.1 Notation

To clarify our notation, consider Figure 4. The wire is a space curve, completely defined by the positions and tangent directions of its two ends (P_0,T_0) and (P_1,T_1). The wire is parameterized by a scalar $s \in [0, 1]$. At every point P_s of the wire, T_s denotes the unit tangent to the wire. We consider the wire to be the centerline of a thin piece of surface that we call a ribbon. At every point P_s of the wire, we have one degree of freedom (which we call twist) for rotating the normal N_s to the ribbons surface around the wires tangent T_s with respect to the local Frenet coordinate system. [3] This twist is designed to provide a smooth field of normal directions as a linear interpolation between the user-controlled twists at the two ends of the wire. The point P_s and the two unit vectors, T_s, and N_s, suffice to define a local coordinate system C_s at P_s that follows the ribbon in position and orientation as s varies from 0 to 1.

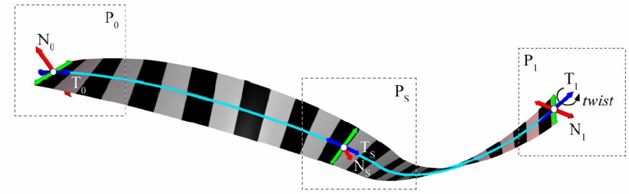

Figure 4: By specifying the six degrees of freedom at each coordinate system at the end of the wire, (P_0, T_0, N_0) and (P_1, T_1, N_1), the user controls the shape of the wire (cyan) and the orientation (twist) of the ribbon around it. A parameter s defines a point PS on the wire and two orthogonal vectors, T_s and N_s.

Hence, for each vertex P of a triangle mesh we compute how the warp affects P. We first compute the projections Q_i of P onto the initial wire. These projections are points on the wire at which the distance to P goes through a local minimum. For all Q_i that are closer to P than a user-prescribed threshold, we compute a displacement vector W_i. The displacement vector W_i is the result of moving P by a fraction f_i of a screw motion M_i. f_i is computed as a function of the distance $||PQ_i||$.

The screw motion M_i is computed as follows. From the position of Q_i along the wire, we compute the corresponding parameter s. Then, we compute the corresponding coordinate systems C_s and C_s on the initial and final wire. M_i is defined as the unique minimal screw motion interpolating between them. We apply a fraction f_i of M_i to P and compute the displacement vector W_i.

3.2 Wire Construction

The wire is defined by the positions and tangent directions of its two ends (P_0, T_0) and (P_1, T_1). We wish to create a smooth 3D curve that interpolates the end conditions and is formed by two circular arcs that are smoothly joined at some point J. The wire is completely defined by computing two scalars, a and b, which define the points $I_0 = P_0 + aT_0$ and $I_1 = P_1 - bT_1$ such that $||I_0 I_1|| = a + b$, as shown in Figure 5.

[3] Although the Frenet frame may be difficult to compute reliably in the general case of 3D curves, in the case of the biarc it is well defined at every point, except at the joint of the two biarcs, where it changes abruptly. Our solution is to create a new frame at each point by twisting the Frenet frame around the tangent to the biarc.

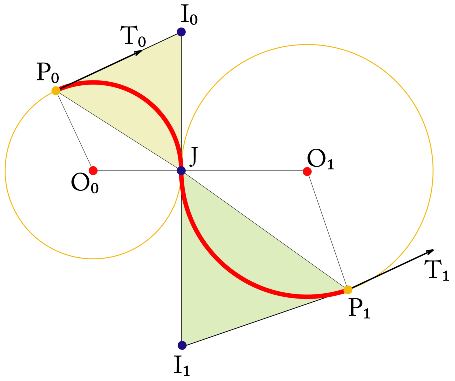

Figure 5: Biarc Nomenclature

Consider the two points $I_0 = P_0 + aT_0$ and $I_1 = P_1 - bT_1$ with a and b chosen so that $||I_0I_1|| = a + b$. Consider the point $J = (bI_0 + aI_1)/(a + b)$. The triangle (P_0, I_0, J) is isosceles and inscribes a first circular arc that starts at P_0 and is tangent to T_0 and that ends in J and is tangent to I_0I_1. Similarly, the triangle (J, I_1, P_1) is isosceles and inscribes a second circular arc that starts at J where it is tangent to I_0I_1 and ends at P_1 with a tangent to T_1. Both arcs meet at J with a common tangent. Although for clarity Figure 5 was drawn in the plane, the construction holds in three dimensions, when the two triangles are not coplanar. To obtain an example of a 3D situation, simply fold the paper along the line I_0I_1.

Following [42], we chose $a = b$. This choice leads to an efficient calculation and provides the wire with a natural and predictable behavior. In fact, in most situations the resulting biarc is very close to a cubic parametric curve with the same end-conditions. The biarc's predictability and ease of control come from the fact that the user needs only to worry about the position of the two endpoints and the tangent direction at these.

To compute the parameter a, we must solve $||(P_0 + aT_0) - (P_1 - aT_1)|| = 2a$, which yields a second degree equation in a: $S \cdot S - 2a(S \cdot T) + a^2(T \cdot T - 4) = 0$, where $S = P_1 - P_0$ and $T = T_0 + T_1$.

In the general case, when $T \cdot T = 4$, we use $a = (sqrt((S \cdot T)^2 + (S \cdot S)(4 - T \cdot T)) - (S \cdot T))/(4 - T \cdot T)$, which produces arcs of less than 180 degrees. In the special case where $T \cdot T = 4$ and $T_1 = T_2$, we use two semi-circles, as discussed in [42].

3.3 Distributing the Twist of the Ribbon along the Biarc

Each arc lies in a plane. The left-hand tracker defines the normal N_0 to the ribbon at P_0. We record the angle a_0 between N_0 and the normal N_0 to the plane of the first arc. Similarly, we record the angle a1 between the normal N_1 to the plane of the second arc and N_1. Let e denote the angle between N_0 and N_1. If we rotate P_0 to follow the first arc, and wished to keep the associated normals N_0 and N_0 in constant orientation with respect to the local Frenet trihedron of the first arc, we would arrive at J with both normals parallel to the plane through J and orthogonal to I_0I_1. Similarly for N_1 and N_1. In this final configuration, the four normals, N_0, N_0, N_1, and N_1 are coplanar and their relative orientations are given by the three angles a_0, a_1, and e. In particular, the angle between N_0 and N_1 is then $a_0 + e + a_1$. Because we wish to obtain a smooth field of normals starting at N_0 and finishing at N_1, we distribute the difference linearly, and twist the local coordinate system C_s along the tangent T_s by an angle equal to $s(a_0 + a_1 + e)$, where the parameter s is equal to 0 at C_0 and equal to 1 at C_1. In practice, for points on the first arc, we twist C_0 by $s(a_0 + a_1 + e)$ and then rotate it around the axis of the first arc.

For points on the second arc, we rotate C_1 by $(1 - s)(a_0 + a_1 + e)$ and then rotate it backward around the axis of the second arc. Figure 6 shows the ribbon for the same wire with and without twisting.

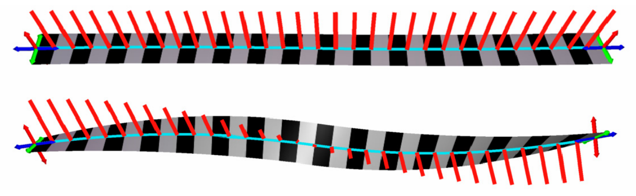

Figure 6: The ribbon (top) is twisted by rotating both trackers around the tangents to the wire at the two ends (bottom).

The choice of the s parameterization affects not only the twist of the ribbon around its wire, but also the correspondence between two wires, and hence the warp. We have explored two parameterizations. The first one is an arc-length parameterization for the whole biarc. The second one uses an arc-length parameterization for each arc, forcing $s = 0.5$ at the junction. A parameterization that maps the first arc of the initial wire onto the first arc of the final wire is slightly simpler to compute. However it tends to produce a non-uniform stretching of space when the arc-length ratios between the first and the second arc differ significantly in each wire. We have therefore opted to use the global arc-length parameterization.

3.4 Projection of Points onto a Biarc

Consider the biarc of the initial ribbon and a point P. As argued above, we want to compute all points Q_i on the biarc where the distance between P and the biarc goes through a local minimum. We will call them the projections of P. We explain in this section how to compute these projections quickly and prove that when P is closer to the wire than the minimum of the biarc radii, at most two such projections exist.

Consider a circle with center O, radius r, and normal N. Let Q be the point on the circle that is closest to P. We compute Q by first computing the normal projection $R = P + (PO \cdot N)N$ of P onto the plane of the arc. Then, Q is obtained by displacing O by r towards R. Hence $Q = O + rOR/||OR||$. If Q lies inside the arc, it is a projection of P. Note that if such a normal projection exists on the arc, then all other points of the arc lie further away from P, including the endpoints of the arc. When Q is not on the arc, we consider the free end of the arc, the end of the biarc, as a candidate projection of P. If, at that free end, the biarc moves away from P, then it is a projection of P, i.e., a local minimum of the distance. Notice that if the other end of the arc were closer to P, it would not be the local minimum for the biarc, since by sliding by an infinitely small amount onto the other arc, it would approach P.

An example where P has a normal projection inside one arc and on the endpoint of the other arc is shown in Figure 7.

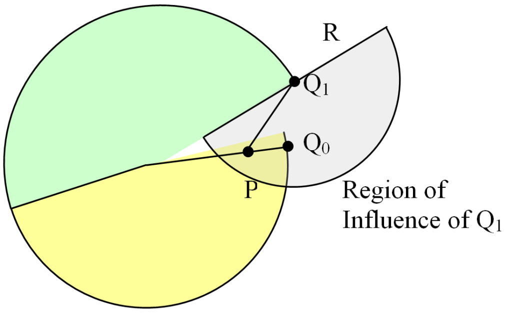

Figure 7: A closest projection Q_0 of P lies inside the first arc. A second closest projection Q_1 lies at the tip of the second arc.

When two projections are returned and both are further away from P than the radius of the region of influence of the warp, P is not affected by the warp. When a single projection is close enough to P, we compute its parameter s on the initial ribbons biarc and use s and $||PQ||$ to compute a warp. In the cases when the projections Q_0 and Q_1 reported for both arcs are within tolerance from P, we compute two warps and blend them. The merit of this solution is discussed below.

3.5 Deforming a Point Using Screw Motion

Given the projection Q of P onto the first arc of the initial wire, we compute its parameter s using the ratio of angles (O_0P_0, O_0Q) and (O_0P_0, O_0J) and the ratio of the arclength of both arcs. A similar approach is used when Q lies on the second arc. We then compute the two coordinates systems, C_s on the initial wire and C'_s on the final wire. They are used as input to compute a fixed point A, an axis direction K, a total rotation angle β, and a total displacement d. These four parameters define a screw motion that transforms C_s into C_s by performing a translation by dK and a rotation around the axis (A, K) of an angle β. The computation of these parameters is inexpensive and easily accessible (for example see [41], [33]). It will not be repeated here.

We also compute the weight $f = F(||PQ||/R)$, where R is the threshold delimiting the radius of influence around the wire. We discuss the nature of the decay function F below. Then we compute the warped version P of P by applying to P a translation by fdK and a rotation by angle $f\beta$ around the axis of the screw that has direction K and passes through A.

3.6 Preventing the Tearing of Space

When there are two projections Q_0 and Q_1 on the arc where the distance to P is locally minimal and when both fall within the Region of Influence of the initial wire, we must take them both into account. Otherwise, a tearing of space may occur. To explain the tearing, suppose that points P and P are the endpoints of an edge of the mesh. Suppose that the s parameter of the closest projection Q of P is very different from the s parameter of the closest projection Q of P. If we were to use the screw associated with a single projection, we would use similar fractions (decay weights) f, for both P and P, but their screws could be very different and may pull them away if, for example, the final wire increases the distance between Q and Q. The edge PP, and hence the incident triangles will be stretched. The corresponding tearing of space is shown in Figure 8.

In such cases, where two projections Q_i fall inside the region of influence, we adjust the corresponding weights f_0 and f_1, as proposed in [33], compute the images P_0 and P_1 of P through both adjusted warps, and add the displacements they each suggest by moving P to $(P_0 + P_1P)$. This eliminates the tearing.

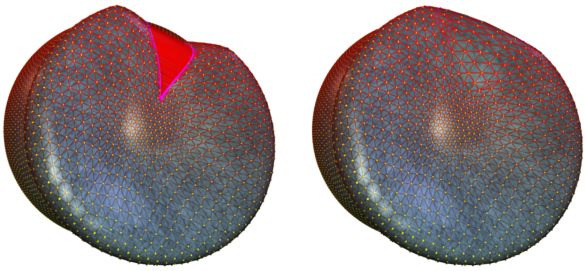

Figure 8: Grabbing a sphere with an initial wire that forms a nearly closed circle and pulling it out and opening the circle produces a tear on the surface (flat region near the top of the left figure). The corrected warp, based on the use of two projections, is shown on the right.

3.7 Choosing Decay Functions

Depending on the type of deformation we want to achieve, different decay functions F may be preferred. Following [30], [32], we let the user switch between a bell-shaped curve and a plateau function (Figure 9), which permits to preserve the shape inside a tube around the wire when the relation between the corresponding portion of the initial and final ribbons is a rigid body transformation. Such relations are maintained when performing warps that achieve rigid bending operation of limbs or tubes.

3.8 Maintaining Continuity

In this subsection, we discuss a modification to the screw computation, which was necessary to ensure the continuity of the warp through space.

The screw motion interpolation used in [33] always generates screws of minimal angle, which is always less than 180 degrees. Consider two points P_s and P_s traveling simultaneously on the initial and final wire. Assume that they move towards each other, then go through a singular situation where their velocities are parallel, and finally diverge.

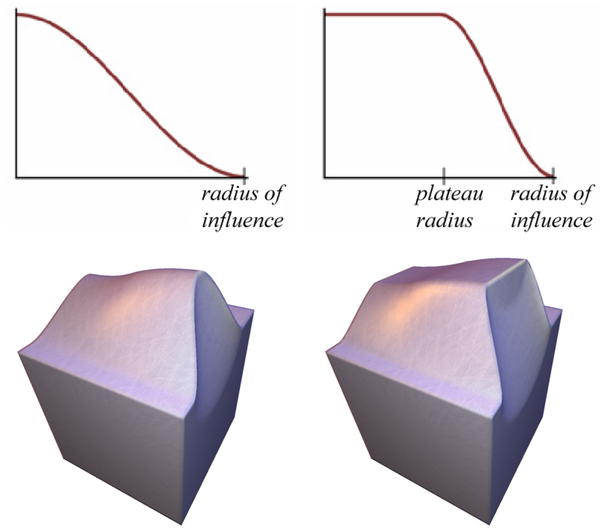

Figure 9: We show the results produced by using the two decay functions. The top shows a profile view of the decay functions. The initial and final wires were identical in both cases.

As we pass through the singular situation, the orientation of the screw axis K is reversed. The displacement values and the direction of rotation are also reversed. This flip produces a discontinuity in the pencil of helix trajectories taken by points of the initial wire as they are warped (Figure 10). We detect these situations using the sign of the dot product of consecutive K vectors. To prevent the discontinuity, we simply revert the flip. This correction results in rotation angles β that may temporarily exceed 180 degrees. We compute the angle as before, and simply replace it by $(\beta 2\pi)$. The K axis is reversed and the distance d negated. We do this change at each singular point.

When no correction is needed, we use the natural direction of K given by the original construction in [33]. When one or more corrections are needed, the user may press a button to toggle between the two possibilities, the one defined at $s = 0$ by the original construction of K and the one where all the K directions are reversed.

Note however, that neither the flip of K nor the blending of screws associated with two projections will solve the problem of space inversion that is inherent to all wire-based warps and may occur when the radius of the region of influence is larger than the minimum radius of curvature of the wire. Modeling the wire as a biarc makes it trivial to detect these situations because the radius of curvature is known for each arc. Thus, we have considered reducing the radius of the region of influence automatically to avoid such inversions. However, because undesirable space inversions are easy to detect visually and avoid with direct manipulation, we have opted not to perform the automatic adjustment to avoid surprising the user with the occasional incorrect choice.

3.9 Adaptive Subdivision

When the mesh is stretched by a warp, the density of its tessellation may no longer be sufficient to produce a smooth warped surface (Figure 11). We use a simple and very efficient technique for adaptively subdividing the surface wherever appropriate. After each warp, when the user freezes the shape, the system starts an adaptive subdivision process and replaces the warped surface with a smoother one. Note that our subdivision simply splits some triangles into 2, 3, or 4 smaller triangles without changing the initial shape. Contrary to subdivision procedures that smoothen the shape, in our implementation, the new vertices are positioned exactly in the middle of the old edges and the old vertices are not adjusted. Tucking in of the old vertices as a Loop subdivision would do [34], or bulging out the edges as a Butterfly subdivision would do [14] is unnecessary, if the initial shape was sufficiently smooth. Hence, we do not have to respect restrictions on the subdivision levels between neighboring triangles.

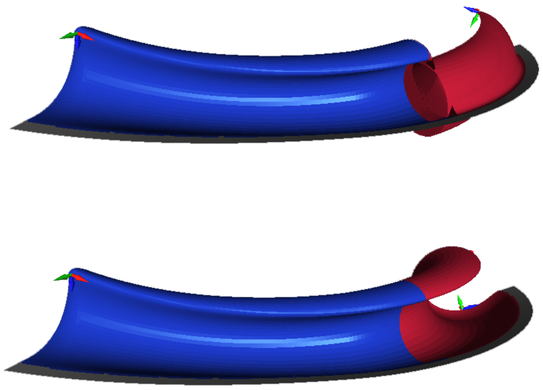

Figure 10: The surface (top) that interpolates the two wires is swept by the helix trajectory followed by a wire point P_s when it is moved by the corresponding screw C_s to its destination. The undesired bulges are removed (bottom) by preventing sudden flips of the screw axis direction K and by permitting for the screw motion to have an angle of more than 180 degrees.

Let the term initial mesh denote the mesh before the current warp, which deforms it into a final mesh. Note that the initial mesh may have been produced by a series of previous warps and subdivisions. Each edge of the initial mesh is tested and marked if subdivision is required. Then, each marked edge is split at its mid-point and each triangle with m marked edges is subdivided into $m + 1$ triangles, using a standard split. This simple approach guarantees preservation of connectivity and does not introduce T-junctions. To test whether an edge should be marked, we compute the distance between the mid-point of its warped vertices and the warped midpoint of its vertices. If that distance exceeds a threshold, we mark the edge. The process is repeated until no more edges need to split or until the user starts a new warp.

This simple approach works well in practice and is very fast. However, it does not guarantee detection of all cases where subdivision is needed. For example, a local stretch occurring inside a triangle that does not affect the edge midpoints could remain undetected. The subdivision may also lead to overly long triangles in some areas of the model, so a criterion to deal with these could be added. The adaptive subdivision method of [31] deals with this, leading to an isotropic tessellation. By not dealing with these triangles we obtain an anisotropic tessellation, which may be preferred in many cases.

Finally, a simplification procedure might be desirable in some instances, where the user desires to eliminate excessive sampling previously introduced. Such procedures could be called upon request, targeting specific regions of the mesh selected by the user, or instead, could be run after a deformation, to coarsen areas that have been flattened. This idea is integral to Gains adaptive refinement and decimation approach [18].

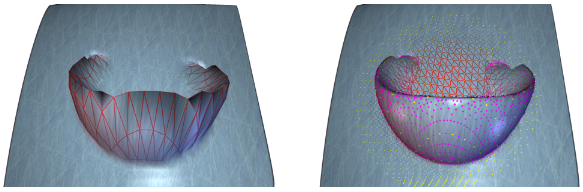

Figure 11: A surface has been warped using its original triangulation (left). A smoother surface is produced through an adaptive subdivision (center, right).

4. BIOMEDICAL APPLICATIONS

Besides the traditional use of space warping techniques for artistic purposes, Bender has applications in the area of Biomedical Engineering. One such application is the planning of cardiovascular surgery in patients with congenital heart defects. These patients need to undergo a series of complex surgical procedures during the first few years of life. These procedures aim to change the blood flow distribution by bending the arteries and/or moving their junction.

Optimizing the blood vessel design through trial and error on the patient would be devastating. Hence, fluid simulation is used on modified 3D models of the patient's arteries. Designing candidate shapes in conventional CAD systems is tedious. Bender makes it possible for practitioners to directly interact with the shape of the vessels and to bend and adjust them as desired. Figure 12 shows how Bender can be used in this context to effectively modify a 3D model of a vessel junction, extracted from an MRI scan [4].

5. CONCLUDING REMARKS

By combining a biarc wire with the concept of a twisted ribbon around it and with a screw-based motion that interpolates corresponding portions of the initial and final ribbons, we have created a new formulation of a space warp that is completely defined by four coordinate systems.

We have developed a 3D user interface for the direct manipulation of these coordinate systems through the use of two Polhemus trackers. We show that the approach makes it easy to extrude, bend, twist, or warp a variety of shapes.

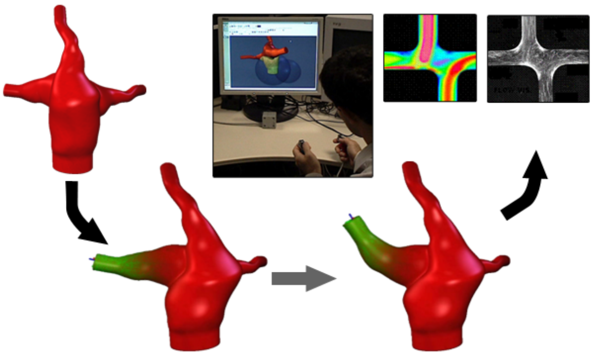

Figure 12: Bender is used in surgery planning to edit a 3D model of a junction of vessels. The model was extracted from an MRI scan. The results are used to evaluate the dynamics of the blood flow using Computational Fluid Dynamics (CFD).

Before opting for this approach, we have explored other formulations for the wire and for the warp. For instance, using a cubic curve [44] or a helix as a wire results in a more expensive computation of the vertex projections on the wire and could potentially generate a larger number of projections. Also, our choice ensures that there are at most two locally closest projections Q_0 and Q_1 of any point P onto a biarc. Based on this observation, we are able to develop a simple technique for avoiding the tearing of space that happens when two neighboring surface points P and P have locally closest projections that are distant along the wire. Milliron et. al [37] proposed extending the Wires approach [47] with a blending approach in which the entire curve has some effect on each deformed point, however the method is approximating instead of interpolating, so user-defined constraints are not satisfied.

We have also explored using transformations that are not formulated as a parameterized family of screw motions for the interpolation between a coordinate system along the initial wire and its counterpart on the final wire. In particular, we have explored the use of a biarc-driven warp. We have concluded that the combination presented in this paper is a good compromise between computational cost and flexibility, producing natural warps, avoiding undesired bulges, and yielding a very fast implementation.

We also thought that Bender could be extended by providing the ability to snap the ribbon to an arbitrary surface. However, when the surface is not smooth, the initial ribbon will not have the simplicity of a biarc, hence this improvement will increase the cost of computing the closest projection, annihilate the guarantee of having at most two such projections, and make the warp less regular.

We have also considered supporting a decay function based on geodesic distance, rather than on Euclidean distance, because it seems better suited in some situations, as shown by

[4] Model courtesy of Dr. Ajit Yoganathan

Bendels and Klein [5]. The mathematical specification of such a formulation is delicate when the initial wire does not lie on the surface, but is a user-controlled 3D curve. Projecting it on a surface may lead to complex and unanticipated artifacts. Furthermore, the family of the coordinate systems along the projected curve may be highly irregular, yielding unanticipated irregularities in the warp.

The design choices we made provide an intuitive and predictable deformation, even when the changes in the shape and twist of the initial and final ribbons are significant. The solution we present is the result of extensive research and of a careful evaluation of the tradeoffs involved. The resultant deformation model permits a real-time direct manipulation, even for shapes of significant complexity. For example, our current, unoptimized implementation produces 10 frames per second with models of about 70,000 triangles on a Pentium 4 2.6 Ghz, with 1 GB of RAM and an NVIDIA Quadro FX 3000 graphics accelerator.

Figures 13 and 14 provide examples of the shape deformations that may be trivially achieved by a single Bender warp. Note that achieving them may require extensive operations with previously proposed tools, including Twister.

6. REFERENCES

[1] J. B. Allan, B. Wyvill, and I. Witten. A Methodology for Direct Manipulation of Polygon Meshes. *New Advances in Computer Graphics (Proceedings of CG International '89)*, pages 451–469, 1989.

[2] C. Bajaj, J. Chen, and X. u. G. Modeling with Cubic A-patches. *ACM Transactions on Graphics*, 14(2):103–133, 1995.

[3] R. Balakrishnan, G. Fitzmaurice, G. Kurtenbach, and W. Buxton. Digital Tape Drawing. In *Proceedings of the 12th annual ACM Symposium on User Interface Software and Technology*, pages 161–169. ACM Press, 1999.

[4] A. H. Barr. Global and Local Deformations of Solid Primitives. *Computer Graphics (Proceedings of ACM SIGGRAPH 84)*, 18(3):21–30, 1984.

[5] G. H. Bendels and R. Klein. Mesh Forging: Editing of 3D-Meshes Using Implicitly Defined Occluders. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 207–217. Eurographics Association, 2003.

[6] J. R. Bill and S. Lodha. Computer Sculpting of Polygonal Models using Virtual Tools. Technical report, Baskin Center for Computer Engineering and Information Sciences, University of California, 1994.

[7] P. Borrel and D. Bechmann. Deformation of n-Dimensional Objects. In *SMA '91: Proceedings of the First Symposium on Solid Modeling Foundations and CAD/CAM Applications*, pages 351–370. ACM Press ACM, 1991.

[8] P. Borrel and A. Rappoport. Simple Constrained Deformations for Geometric Modeling and Interactive Design. *ACM Transactions on Graphics*, 13(2):137–155, 1994.

[9] J. C. Carr, W. R. Fright, and R. K. Beatson. Surface Interpolation with Radial Basis Functions for Medical Imaging. *IEEE Transactions on Medical Imaging*, 16:96–107, 1997.

[10] G. Celniker and D. Gossard. Deformable curve and surface finite-elements for free-form shape design. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 257–266. ACM Press, 1991.

[11] Y. K. Chang and A. P. Rockwood. A Generalized de Casteljau Approach to 3D Free-form Deformation. In *Proceedings of ACM SIGGRAPH 94*, pages 257–260. ACM Press ACM SIGGRAPH, 1994.

[12] S. Coquillart. Extended Free-Form Deformation: A Sculpting Tool for 3D Geometric Modeling. *Computer Graphics (Proceedings of ACM SIGGRAPH 90)*, 24(4):187–196, 1990.

[13] H. Du. Interactive shape design using volumetric implicit pdes. In *ACM Symposium on Solid Modeling and Applications*, pages 235–246. ACM Press, 2003.

[14] N. Dyn, D. Levin, and J. A. Gregory. A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control. *ACM Transactions on Graphics (TOG)*, 9(2), 1990.

[15] D. R. Forsey and R. H. Bartels. Hierarchical B-spline refinement. In *Computer Graphics (Proceedings of ACM SIGGRAPH 88)*, pages 205–212. ACM Press ACM SIGGRAPH, 1988.

[16] B. Fowler. Geometric Manipulation of Tensor Product Surfaces. In *Proceedings of the 1992 Symposium on Interactive 3D graphics*, pages 101–108. ACM Press ACM SIGGRAPH, 1992.

[17] J. E. Gain. *Enhancing Spatial Deformation for Virtual Sculpting*. PhD thesis, St. John's College, University of Cambridge, 2000.

[18] J. E. Gain and N. Dodgson. Adaptive Refinement and Decimation under Free-Form Deformation. In *Eurographics UK '99*. Eurographics, 1999.

[19] T. A. Galyean and J. F. Hughes. Sculpting: an Interactive Volumetric Modeling Technique. *Computer Graphics (Proceedings of ACM SIGGRAPH 91)*, 25(4):267–274, 1991.

[20] S. F. F. Gibson and B. Mirtich. A Survey of Deformable Modeling in Computer Graphics. Technical report, Mitsubish Electric Research Laboratoy, 1997.

[21] J. Gomes, L. Darse, B. Costa, and L. Velho. *Warping and Morphing of Graphical Objects*. Morgan Kaufmann Publishers Inc., 1999.

[22] T. Grossman, R. Balakrishnan, G. Kurtenbach, G. Fitzmaurice, A. Khan, and B. Buxton. Creating Principal 3D Curves with Digital Tape Drawing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 121–128. ACM Press, ACM SIGCHI, 2002.

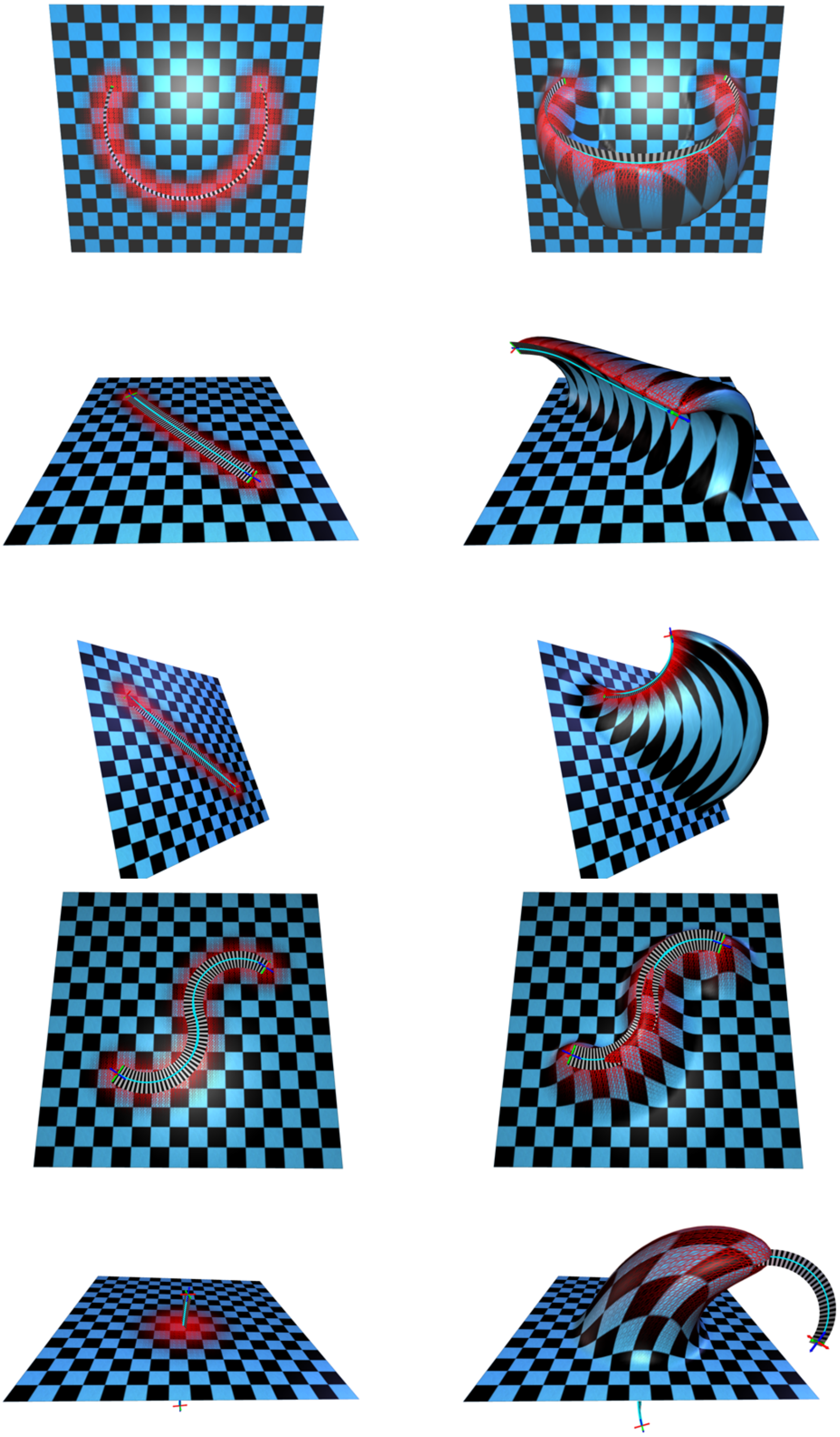

Figure 13: Bender deformations make easy the creation of a wide range of shape features. Five examples showing the placement of the Bender ribbon on a plane and the results after the deformation: a bowl, a wave, a fin, an embossed 'S' and a horn.

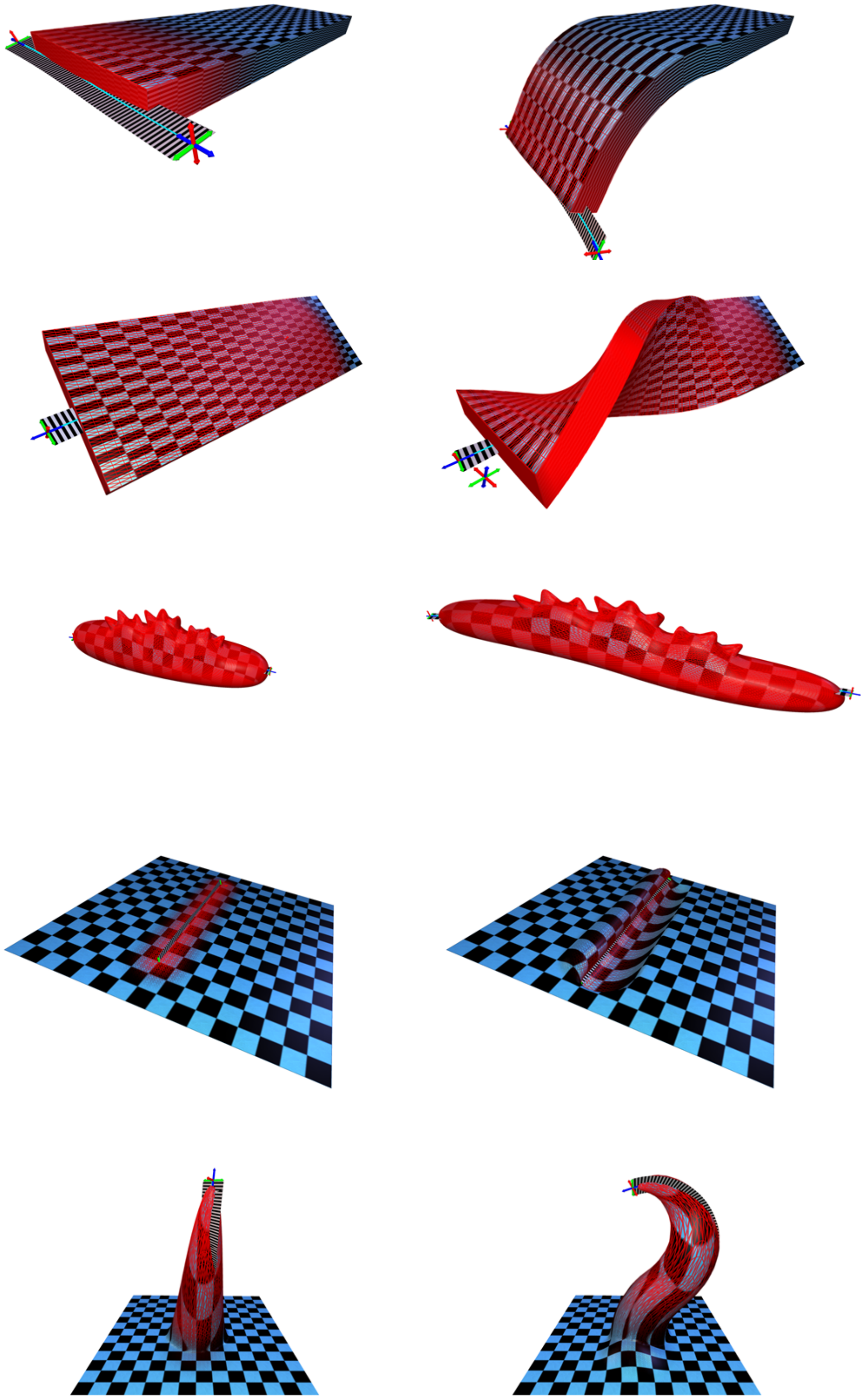

Figure 14: Bender is also useful to deform existing shape features. Five examples showing the placement of the Bender ribbon on the original shape and the results after the deformation: bending, twisting, stretching, creating a curl and making a hook.

[23] T. Grossman, R. Balakrishnan, and K. Singh. An interface for creating and manipulating curves using a high degree-of-freedom curve input device. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press, ACM SIGCHI, 2003.

[24] Y. Guiard. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *The Journal of Motor Behavior*, 19(4):486–517, 1987.

[25] K. Hinckley, R. Pausch, J. C. Goble, and N. F. Kassel. Passive Real-World Interface Props for Neurosurgical Visualization. In B. Adelson, S. Dumais, and J. Olson, editors, *Human Factors in Computing Systems CHI'94 Conference Proceedings*, pages 452–458. ACM SIGCHI, 1994.

[26] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface Reconstruction from Unorganized Points. *Computer Graphics (Proceedings of ACM SIGGRAPH 92)*, 26(2):71–78, 1992.

[27] W. M. Hsu, J. F. Hughes, and H. Kaufman. Direct Manipulation of Free-Form Deformations. *Computer Graphics (Proceedings of ACM SIGGRAPH 92)*, 26(2):177–184, 1992.

[28] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a Sketching Interface for 3D Freeform Design. In *Proceedings of ACM SIGGRAPH 99*, pages 409–416. ACM Press, ACM SIGGRAPH, 1999.

[29] F. D. IX, H. Qin, A. Kaufman, and J. El-Sana. Haptic sculpting of dynamic surfaces. In *ACM Symposium on Interactive 3D Graphics*, pages 103–110. ACM Press ACM, 1999.

[30] X. Jin, Y. Li, and Q. Peng. General Constrained Deformations based on Generalized Metaballs. *Computers and Graphics*, 24(2):219–231, 2000.

[31] L. Kobbelt, T. Bareuther, and H.-P. Seidel. Multiresolution Shape Deformations for Meshes with Dynamic Vertex Connectivity. *Computer Graphics Forum*, 19(3), 2000.

[32] F. Lazarus, S. Coquillart, and P. Jancene. Axial Deformations: An Intuitive Deformation Technique. *Computer Aided Design*, 26(8):607–613, 1994.

[33] I. Llamas, B. Kim, J. Gargus, J. Rossignac, and C. D. Shaw. Twister: a Space-Warp Operator for the Two-Handed Editing of 3D Shapes. *ACM Transactions on Graphics*, 22(3):663–668, 2003.

[34] C. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, 1987.

[35] R. MacCracken and K. I. Joy. Free-Form Deformation With Lattices of Arbitrary Topology. In *Proceedings of ACM SIGGRAPH 96*, pages 181–190. ACM Press, ACM SIGGRAPH, 1996.

[36] D. N. Metaxas. *Physics-Based Deformable Models: Applications to Computer Vision, Graphics, and Medical Imaging*. Kluwer Academic Publishers, january edition, 1996.

[37] T. Milliron, R. J. Jensen, R. Barzel, and A. Finkelstein. A Framework for Geometric Warps and Deformations. *ACM Transactions on Graphics*, 21(1):20–51, 2002.

[38] M. Pauly, Keiser, L. P. Kobbelt, and M. Gross. Shape modeling with point-sampled geometry. *ACM Transactions on Graphics*, 22(3), 2003.

[39] J.-P. Pernot, B. Falcidieno, F. Giannini, S. Guillet, and J.-C. L&#233;on. Modelling free-form surfaces using a feature-based approach. In *ACM Symposium on Solid Modeling and Applications*, pages 270–273. ACM Press, 2003.

[40] Polhemus. Polhemus Fastrak, http://www.polhemus.com/fastrak.htm. 2004.

[41] J. Rossignac and J. J. Kim. Computing and Visualizing Pose-Interpolating 3D Motions. *Computer Aided Design*, 33(4):279–291, 2001.

[42] J. Rossignac and A. Requicha. Piecewise-Circular Curves for Geometric Modeling. *IBM Journal of Research and Development*, 13:296–313, 1987.

[43] S. Schkolne, M. Pruett, and P. Schröder. Surface Drawing: Creating Organic 3D Shapes with the Hand and Tangible Tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 261–268. ACM Press, ACM SIGCHI, 2001.

[44] P. J. Schneider. *Graphics Gems I*, chapter Solving the nearest-point-on-curve problem, pages 607–611. Morgan Kaufmann, 1st edition, June 1990.

[45] T. W. Sederberg and S. R. Parry. Free-Form Deformation of Solid Geometric Models. *Computer Graphics (Proceedings of ACM SIGGRAPH 86)*, 20(4):151–160, 1986.

[46] C. Shaw and M. Green. THRED: A Two-Handed Design System. *Multimedia Systems*, 5(2):126–139, 1997.

[47] K. Singh and E. Fiume. Wires: A Geometric Deformation Technique. In *Proceedings of ACM SIGGRAPH 98*, pages 405–414. ACM Press, ACM SIGGRAPH, 1998.

[48] G. Turk and J. F. O'Brien. Modelling with Implicit Surfaces that Interpolate. *ACM Transactions on Graphics*, 21(4):855–873, 2002.

[49] G. Wesche and H. P. Seidel. FreeDrawer: a Free-Form Sketching System on the Responsive Workbench. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 167–174. ACM Press, 2001.

[50] D. Zorin, P. Schröder, and W. Sweldens. Interactive Multiresolution Mesh Editing. In *Proceedings of ACM SIGGRAPH 97*, pages 256–268. ACM Press, ACM SIGGRAPH, 1997.