Wires: A Geometric Deformation Technique

Karan Singh
Eugene Fiume*
Alias|wavefront

Abstract

Finding effective interactive deformation techniques for complex geometric objects continues to be a challenging problem in modeling and animation. We present an approach that is inspired by armatures used by sculptors, in which *wire* curves give definition to an object and shape its deformable features. We also introduce *domain curves* that define the domain of deformation about an object. A wire together with a collection of domain curves provide a new basis for an implicit modeling primitive. Wires directly reflect object geometry, and as such they provide a coarse geometric representation of an object that can be created through sketching. Furthermore, the aggregate deformation from several wires is easy to define. We show that a single wire is an appealing direct manipulation deformation technique; we demonstrate that the combination of wires and domain curves provide a new way to outline the shape of an implicit volume in space; and we describe techniques for the aggregation of deformations resulting from multiple wires, domain curves and their interaction with each other and other deformation techniques. The power of our approach is illustrated using applications of animating figures with flexible articulations, modeling wrinkled surfaces and stitching geometry together.
Keywords: deformations, implicit models, interactive graphics, animation.

1 Introduction

The modeling and animation of deformable objects is an active area of research [1, 2, 7, 8, 10, 12, 13, 15]. Free-form deformations (FFDs) [13] and their variants [6, 7, 9, 10], for example, are popular and provide a high level of geometric control over the deformation. These approaches involve the definition and deformation of a lattice of control points. An object embedded within the lattice is then deformed by defining a mapping from the lattice to the object. The user thus deals with a level of detail dictated by the density of the control lattice. While very useful for coarse-scale deformations of an object, the technique can be difficult to use for finer-scale deformations, where a very dense and customized control lattice shape [7, 10] is usually required. Arbitrarily shaped lattices can be cumbersome to construct and it is often easier to deform the underlying geometry directly than to manipulate a dense control lattice.

Axial deformations provide a more compact representation in which a one-dimensional primitive, such as a line segment or curve, is used to define an implicit global deformation [12]. Our approach, called *wire* deformations, is related to axial deformations, although we have a different motivation and formulation. Our main point of departure is our desire to bring geometric and deformation modeling closer together by using a collection of wires as both a coarse-scale representation of the object surface, and a directly manipulated deformation primitive that highlights and tracks the salient deformable features of the object. As can be seen in Figure 1, projections of the wire curves provide a sketch-like representation of the object, which is how many artists prefer doing design.

Wire deformations may be likened to a constructive sculpting approach in which the wires of an armature provide definition to the object and control its deformable features. As in scuplture, the wire curves themselves give a coarse approximation to the shape of the object being modeled. A wire deformation is independent of the complexity of the underlying object model while easily allowing finer-scale deformations to be performed as either object or deformation complexity increases. In fact, an animator can interact with a deformable model, namely the wires, without ever having to deal directly with the object representation itself. Wires can control varying geometric representations of the same object and can be reused on different objects with similar deformable features.

There are two stages in the wire deformation process. In the first, which is typically computed once, an object is bound to a set of wires. In the second, any manipulation of a wire effects a deformation of the object. Implicit function based techniques are used to implement wire deformations. The deformation algorithm is conceptually simple and efficient. Through several examples, we shall illustrate the expressiveness of wires for feature based object design and animation, including facial animation.

Section 2 presents the wire deformation algorithm. Section 3 introduces the use of domain curves to refine the regions affected by a wire. Section 4 describes the techniques used to provide user control over aggregate wire (or other) deformations. Section 5 demonstrates the power of wires for the modeling and animation of wrinkled surfaces, flexible articulated structures and stitched surfaces. Section 6 concludes with discussion of our results.

2 Wire Definition and Algorithm

A *wire* is a curve whose manipulation deforms the surface of an associated object near the curve. We define a wire as a tuple $\langle W, R, s, r, f\rangle$, where W and R are free-form parametric curves, s is a scalar that controls radial scaling around the curve, and r is a scalar value defining a radius of influence around the curve; the scalar function $f : \mathbb{R}^+ \rightarrow [0, 1]$ is often referred to in implicit function related literature as a *density function* [15]. Normally, f is at least C^1 and monotonically decreasing with $f(0) = 1$, $f(x) = 0$ for $x \geq 1$ and $f'(0) = 0$, $f'(1) = 0$. [1]

*Alias|wavefront, 210 King St. E., Toronto, Canada M5A 1J7. ksingh@aw.sgi.com, elf@aw.sgi.com

[1] Wire deformations on a surface preserve continuity up to the degree of continuity of the function f. As an example, we use a C^1 function $f(x) = (x^2 - 1)^2, x \in [0, 1]$, in our implementation.

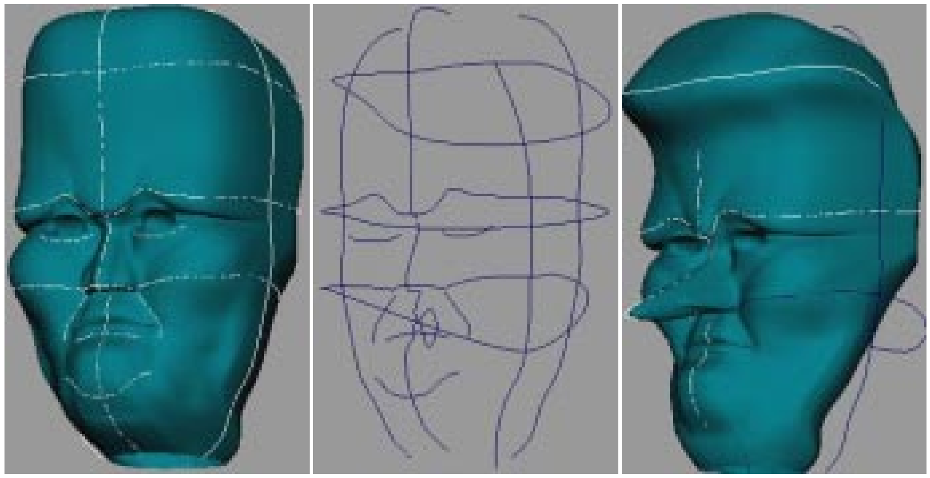


Figure 1: Wires: A geometric deformation technique.

The parameters f and r can be used to define a volume about a curve bounded by an offset surface at a distance r from the curve [4]. Together with a scale factor s, and given r and f, a wire is defined by specifying a curve W and a congruent copy of the curve, R. We refer to W as the *wire curve* and R as the *reference curve* for a wire. When an object is bound to the wire, the domain of influence of the wire is demarkated by the offset surface of radius r defined around the reference curve R. The influence for points of the object within this offset volume are calculated using the density function f. Subsequent manipulation of W results in a change between W and R, which is used along with s to define the deformation. The actual deformation applied to a point is modulated by its influence calculated when the object was bound to the wire.

Let $C(u)$ be a space curve, parametrized without loss of generality by $u \in [0, 1]$. For any point $P \in \mathbb{R}^3$, let $p_C \in [0, 1]$ be the parameter value that minimizes the Euclidean distance between point P and curve $C(u)$. If there is more than one minimum, we arbitrarily define p_C to be the parameter with the smallest value.[2] For any point P and curve C, we define the function $F(P, C)$ as

$$F(P, C) \;=\; f\left(\frac{||P - C(p_C)||}{r}\right).$$

From the properties of f it is clear that $F(P, C)$ varies from zero for $||P - C(p_C)|| \geq r$ (points on and outside the offset volume defined by C and r), to $F(P, C) = 1$ when $||P - C(p_C)|| = 0$ (P lies on C). $F(P, C)$ defines the influence that a curve C has on a point P. This is the usual function definition for implicitly defined offset shapes [4], and it will be used below in defining the semantics of the deformation.

As with any deformation, a wire deformation is a pointwise function mapping $\mathbb{R}^3$ onto $\mathbb{R}^3$. For each object O, let $\mathbf{P}_O$ be the point-based representation to which the wire deformations will be applied. Typically $\mathbf{P}_O$ contains all points necessary to construct or approximate an object's surface. $\mathbf{P}_O$ could thus be a set of control vertices for freeform surfaces, a set of vertices in a polymesh, or an unstructured set of points in space.

When an object O is bound to a wire $\langle W, R, s, r, f\rangle$, the parameters p_R and $F(P, R)$ are computed for every point $P \in \mathbf{P}_O$. Only points on the object within the offset volume of radius r from the curve R will be deformed (i.e., points P with $F(P, R) > 0$). Figure 3(a) shows how one wire with a larger r affects a larger region of the object, the other deformation parameters being identical.

[2] In most cases, this is an effective choice, but it can be overly simple in geometrically delicate situations, which we discuss in Section 6.

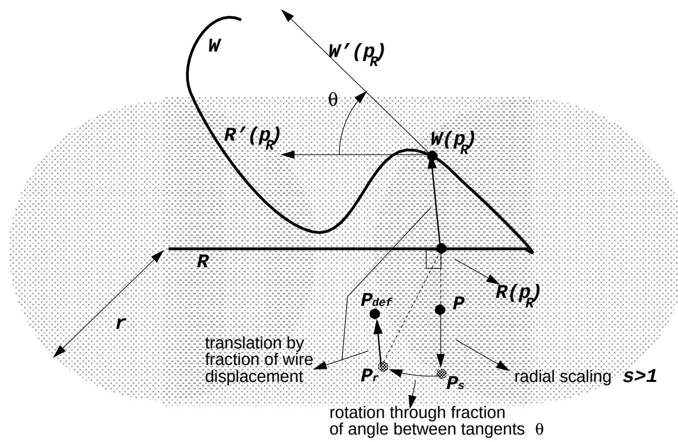
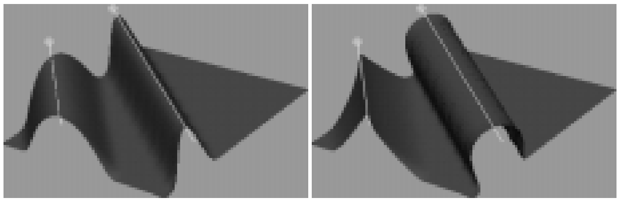


Figure 2: Deformation of a point P to P_{def} by a wire W.

The deformation at a point P is related to the deviation of the closest point on the reference curve $R(p_R)$ from a corresponding point on the wire curve W. We use a direct correspondence between curves R and W based on parameter value, but correspondences such as an arc-length parametrization can instead be used. The computation thus far defines the region of the object to be deformed.

When W is manipulated, the object is deformed for every point P of the undeformed object for which $F(P, R) > 0$ (see Figure 2):

1. Uniformly scale P about point $R(p_R)$ resulting in point P_s. Specifically, $P_s = P + (P - R(p_R)) \cdot (1 + (s-1) \cdot F(P, R))$.
2. Let $C'(u)$ be the tangent vector to curve C at u, and let θ be the angle between $W'(p_R)$ and $R'(p_R)$. Rotate P_s by the modulated angle $\theta \cdot F(P, R)$, around the axis $W'(p_R) \times R'(p_R)$, about point $R(p_R)$. This provides a screw-like deformation, resulting in point P_r (see Figure 2). Rotational transformations such as a twist along the wire can be easily specified as in Section 3, and composed with the rotation specified here.
3. Finally add the translation $(W(p_R) - R(p_R)) \cdot F(P, R)$ to the result of the rotation P_r. The resulting deformed point P_{def} is thus $P_{def} = P_r + (W(p_R) - R(p_R)) \cdot F(P, R)$.



(a) Varying r (b) Varying s

Figure 3: Varying r and s on wires.

Observe the following properties of our formulation.

- Objects are not deformed upon initial creation of wire and reference curves: R, being a copy of W, coincides with it, so no rotation or translation is applied. For a default scale parameter of $s = 1$, no deformation is applied to the object.
- Points on the object outside the offset volume of radius r from the reference curve (points P with $F(P, R) = 0$) are not deformed regardless of the value of s. This is because $F(P, R)$ attenuates each step of the deformation.
- Points on the object that are on the reference curve, when the object is bound to the wire, track the wire curve precisely. For a point P on the undeformed object that coincides with a point on the reference curve, $R(p_R)$ is identical to P and thus $F(P, R) = f(0) = 1$. The scale and rotation have no effect as they are applied about point $R(p_R)$ itself. P thus moves to $P + (W(p_R) - R(p_R))$ or the point $W(p_R)$ on W.
- The deformation of points on the object between those on the reference curve and those outside its realm of influence is smooth and intuitive. The factor $F(P, R)$ controls the attenuation of the deformation, varying from precise tracking for points on the reference curve to no deformation at or beyond the offset volume boundary. The properties of the function f dictate the behavior of $F(P, R)$ and the smoothness properties of the deformation.
- For $s = 1$, the cross-section of the deformed object surface in a plane perpendicular to the wire curve at a point closely resembles the profile of f (see Figure 3(a)). Manipulating f provides intuitive control over the shape of the deformed object surface and directly controls the degree of continuity preserved by the deformed surface. Figure 3(b) also shows how reducing s on one wire and increasing it on the other provides sucking or bulging control over the deformation.

Axial deformations [12] also use the notion of a reference curve R and closest point computation p_R for a point P. The axial deformation technique relates two Frenet frames attached at $W(p_R)$ on the deformed curve and $R(p_R)$ on the reference curve. The deformation imparted to point P is a portion of the transformation from the reference curve's Frenet frame to the Frenet frame on the deformed curve. The proportion is based on an interpolation of the closest distance of P to the reference curve $||P - R(p_R)||$ between two cut-off radii R_{in} and R_{out}.

While axial deformations and the deformation of a single wire share some similarities, a wire has several differences. First, the separation of the scale, rotation and translational components of the wire deformation provides a user with more selective control over the resulting deformation than the integrated transformation of a Frenet frame. Second, Frenet frames are harder to control and have orientation problems when the curvature of a curve vanishes. Third, simple non-linear transformations can be incorporated seamlessly into the deformation algorithm at the appropriate point. For example, as seen in Figure 4, an interpolated twist around the wire can be implemented by rotating the point around the axis along the reference-curve $R'(p_R)$ by a specified angle as part of the rotational step of the deformation algorithm. Fourth, using an implicit function to control the spatial influence of the wire on the deformed objects makes the technique accessible to more general implicit surface animation techniques. The extensions in Section 3 will show how implicit functions can be overlaid by a user to determine what parts of the deformed objects are affected and by how much.

Figure 5 shows the effect of the various deformation parameters. A cylindrical object with an associated wire is depicted in Figure 5(a). Figure 5(b) shows the deformation to the surface as a result of moving a control point on the wire curve. A more global deformation to the entire object as result of a large increase to r is illustrated in Figure 5(c). Another control point is moved in Figure 5(d). When r is large, the entire object tracks the wire. Figure 5(e) depicts the effect of reducing the scale factor s on the configuration in Figure 5(d). Figure 5(f) further illustrates how the three stages of deformation can be tuned individually by attenuating the rotational aspect and inducing a shear on the configuration in Figure 5(d).

3 Controlling Wire Parameters

Our technique was designed with usability and direct manipulation in mind. We are thus interested in ways of giving finer user control over the deformation parameters. Allowing a specified portion such as a subset of control vertices on an object to be deformed affords some degree of control. However, continuity properties may be compromised in parts of the object surface defined by control vertices that are selectively deformed. This is shown in Figures 6(a,b). Usually one would expect a smoother dropoff based on the region selected, such as that shown in Figure 6(c).

3.1 Locators

One solution involves using *locators* along a wire curve to specify the values of parameters along the wire. An animator can position locators along curves as needed to control locally not only the radius of influence r but any attribute related to wire deformation. We calculate the attribute being localized at a parameter value p as an interpolation between the attribute values specified at the two locators that bracket p. Two wire locators are used to model the cone-spherical shape of an Adam's apple in Figure 7(a) by varying r. Local control over the amplitude of deformation causes the transformation from an "**l**" in Figure 7(b) to an "**i**" in Figure 7(c). Locators can also be used to incorporate non-linear transformations such as a twist (see Figure 4), where they are used to control the twist angle along the wire.

As mentioned in Section 2, the implicit function F can be combined with other functions. In particular, we can get directional control by modulating F with an implicit function for an angular dropoff around an axis perpendicular to the wire. Both the directional axis and dropoff angle can be interpolated by locators.

3.2 Domain Curves

Locators provide radially symmetric local control along and around a wire curve. Anisotropic directional control is provided by domain curves as illustrated in Figure 6(c). *Domain curves* along with an associated wire's reference curve define an implicit primitive function over a finite volume. This provides incremental, direct control over what parts of the object are deformed (using domain curves) and by how much they are deformed (using wire curves). We shall deal here with a single domain curve for a given wire. The use of multiple domain curves will be discussed in Section 4.

As illustrated in Figure 6(c), a domain curve demarkates a region of the object surface to be deformed, and along with the reference

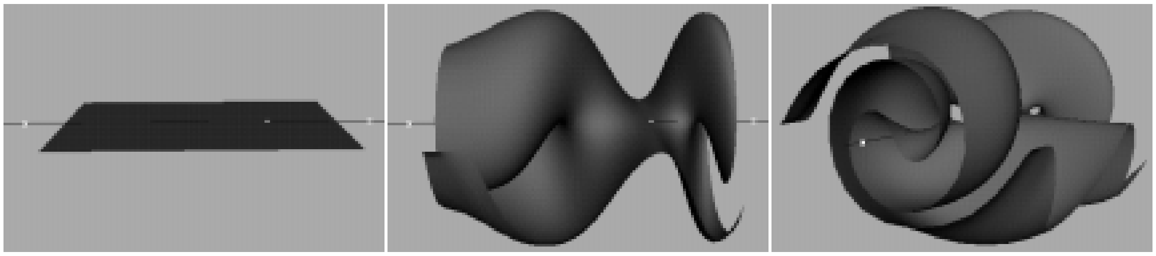
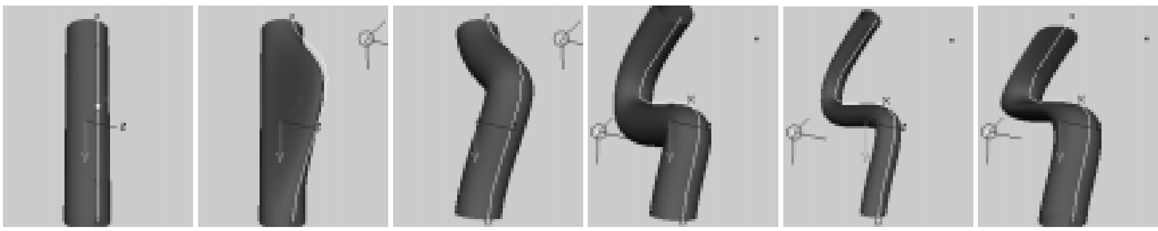
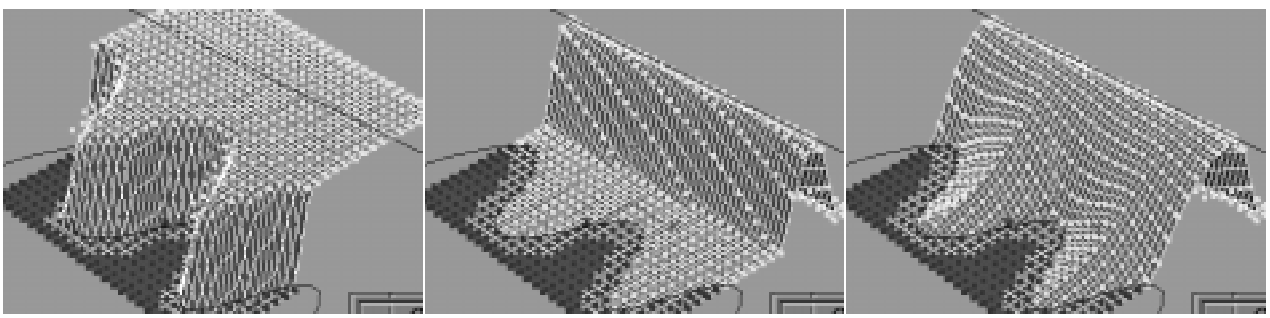


Figure 4: Interpolated twist around a wire.



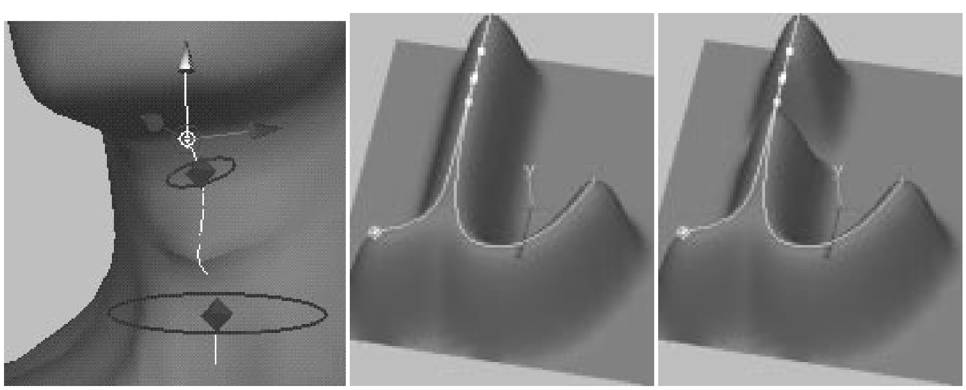
(a) no deformation (b) deform with small r (c) increase r (d) more deformation (e) reduce s (f) attenuate rotation

Figure 5: More variations of r and s on wires.



(a) sharp vertical decay (b) sharp horizontal decay (c) smooth decay

Figure 6: Region of influence of a wire.



(a) r (b) deformation amplitude (c) deformation amplitude

Figure 7: Varying deformation parameters along a wire.

curve it acts as an anchor for the deformation. More generally, we defined the domain curve to be a free-form curve rather than a closed curve on the object surface. Such a domain curve does not unambiguously determine which control points on the object surface will be deformed. Most animators, however, have a very good idea of how a given domain curve will affect the region of the object to be deformed, based on the spatial relationship between the reference curve, domain curve and the object surface. In our implementation we use distance and angle computations between points on the object surface, the domain curve and reference curve to determine if and by how much the point will be influenced.

In Figure 6(c), we chose the domain curve to have a *one-sided* influence region affected by the wire. The other side is affected by the conventional dropoff radius r. Our formulation of one-sided domain curves is as follows.

We first determine if the domain curve D will be used to define the function f at a point P. Let $cosangle = (D(p_D) - R(p_R)) \cdot (P - R(p_R))$. The domain curve will define the function if $cosangle > 0$. This heuristic attempts to select points P that are thought to lie on the same side of R as D (even though the concept of *side* is not well-defined mathematically). As can be seen in the Figure 8, this notion of same side tends to be captured by an acute angle subtended at $R(p_R)$, for the triangle with vertices at P, $D(p_D)$ and $R(p_R)$. With this edge condition,

$$F(P, R) \;=\; f\left(\frac{||P - R(p_R)||}{||R(p_R) - D(p_D)||}\right).$$

For points considered to be outside the domain defined by the domain curve, the conventional dropoff radius calculation can be applied. This formulation is likely to lead to a discontinuity in the neighbourhood of points where $cosangle = 0$. The discontinuity may be removed by specifying a $\delta \in (0, 1)$, so that for $cosangle \in [0, \delta]$,

$$F(P, R) \;=\; f\left(\frac{||P - R(p_R)||}{Interp(cosangle)}\right),$$

where the function $Interp$ gives a smoothly interpolated value from r to $||R(p_R) - D(p_D)||$ as $cosangle$ varies from 0 to δ.

Figure 9 uses a domain curve under the eye to limit the influence of the wire to the figure's cheek. Domain curves can easily be used to control other spatially variable parameters.

4 Multiple wires

Recall that our approach is driven by the interaction of multiple wires that together provide an overall definition of the object's shape (cf. Figure 1). We appeal to a sculptor's armature metaphor to give the expected behavior of a deformation in regions where more than one wire has an effect. In an armature, an overall shape deformation can be seen as a smoothed union of the deformations caused by each wire. This behavior is evident in the **X** pulled out of a plane by two wires in Figure 10(a). The results are distinct from the traditional superposition of the deformations due to each wire as in Figure 10(b). This behavior is analogous to that discussed in [4] distinguishing implicit function based convolution and distance surfaces. We further require that subdividing a wire curve into two curves does not affect the deformation applied to the object (such as an unwanted bulge where the two curves abut).

The problem of unwanted aggregate blobs is circumvented in [7, 10] by making deformations due to multiple deformers incremental. While Coquillart's technique for FFDs can be readily applied to wires[7], it would defeat our main purpose of getting interesting aggregate behavior from many interacting wires.

Our solution is as follows. Let the i^{th} wire curve deforming an object be $\langle W_i, R_i, s_i, r_i, f_i\rangle$. Let us suppose the deformation of a point P on an object induced by wire i results in P_{def_i} (as defined in Section 2). Let $\Delta P_i = P_{def_i} - P$. The deformed point P_{def} as influenced by all wires is defined as the following blend:

$$P_{def} \;=\; P + \frac{\sum_{i=1}^{n} \Delta P_i \cdot ||\Delta P_i||^m}{\sum_{i=1}^{n} ||\Delta P_i||^m}.$$

The resulting behavior varies with m from a simple average of the ΔP_i when $m = 0$, converging to $\max\{\Delta P_i\}$ for large m (see Figure 11). When m is negative, it is technically possible to have a singular denominator. But if we reformulate this expression as

$$P_{def} \;=\; P + \frac{\sum_{i=1}^{n} \Delta P_i \cdot \prod_{j\neq i} ||\Delta P_j||^{|m|}}{\sum_{i=1}^{n} \prod_{j\neq i} ||\Delta P_j||^{|m|}},$$

we note that the singularity is removable. In practice, it is preferable to use the original formulation even for negative m and simply omit those ΔP's that are zero. Observe that as m gets increasingly negative, the displacement approaches $\min\{\Delta P_i\}$. Indeed, each wire i could have its own exponent m_i, giving finer control over its contribution to the result in regions of interaction.

It is easy to verify that the above formulation has several desirable properties for typical values of $m \geq 1$:

1. In a region where only one wire is relevant, the result is precisely the deformation of that wire.
2. When several wires produce the same deformation, the result is the deformation induced by any one of those wires.
3. In general, the result is an algebraic combination of the individual wire deformations, with a bias (controlled by m) toward the deformations of larger magnitude.

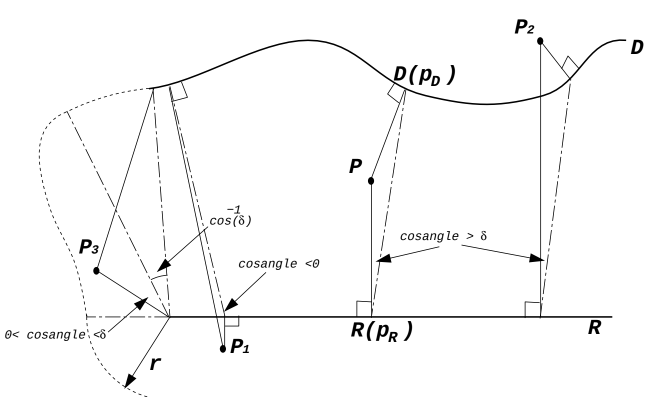


Figure 8: Implicit function defined using a domain curve.

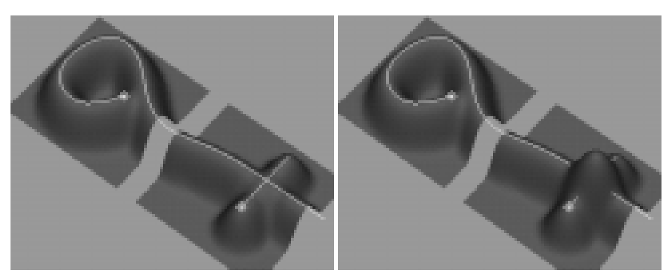


Figure 10: Multiple wires on multiple surfaces with differing deformations. Left, (a): integrated deformation that avoids blobby superposition. Right, (b): the traditional additive deformation.

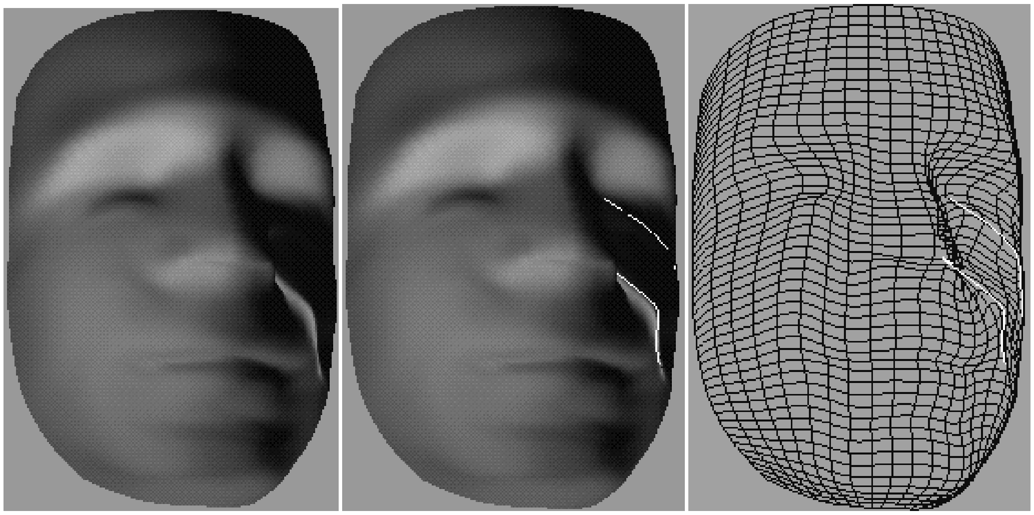


Figure 9: Using domain curves to animate a facial crease.

Many augmentations of our formulation are possible. For example, we can blend the above deformation with an aggregation of wire deformations given by $P_{def} = P + \sum_{i=1}^{n} \Delta P_i$. We can also attach different exponents to each domain curve, allowing us to introduce domain curves that refine an implicit volume in an additive or subtractive fashion controlled by the sign of each exponent.

Another useful variation is to introduce a local influence of a wire at a point on an object's surface relative to other wires. In the formulation above, only wires that directly deform a point are of consequence. In Figure 12(b), the central straight wire lifts a large portion of the surface when it is translated upward. Because the outer curve did not move, it did not influence the surface. In Figure 12(c), however, it acts as an anchor, exercising a local influence on the surface that is independent of the deformation it imparts (in this case none), but depends on the proximity of points in space to the curve. We use $F(P, R_i)$ as a measure of proximity or local influence for the wire. The formulation used for this behavior is

$$P_{def} \ = \ P + \frac{\sum_{i=1}^{n} \Delta P_i \cdot F(P, R_i)^k}{\sum_{i=1}^{n} F(P, R_i)^k}.$$

The factor k has a similar effect that m had earlier. A parameter $localize$ combines this deformation with the others defined earlier.

A similar effect can be seen in Figure 13, where wires simulate the behavior of an FFD lattice. A wire curve is generated along each lattice line. Large dropoff radii ensure that planarity is preserved on the deformed cube when the right face of the lattice is translated outward, as can be seen in Figures 13(b,c). The difference in behavior with and without the localized influence computation is evident from the more global deformation in Figure 13(c) over 13(b). The formulations we have described are equally applicable to other deformation techniques and can be used to combine the results of different deformation approaches.

5 Applications

We shall illustrate the versatility of wires with three examples that exercise different aspects of wire deformations. We show how wires may be used to control wrinkle formation and propagation on a surface. Such *surface oriented* deformations are localized to increase surface detail. We apply wires to stitching and tearing geometry, which again is a surface oriented deformation. Lastly, we describe a *volume oriented* deformation, in which a flexible skeletal curve is generated from a traditional joint hierarchy and is used to bind articulated geometry as a wire. Figures 5(b) and (c) distinguish between surface and volume oriented deformations.

5.1 Wrinkles

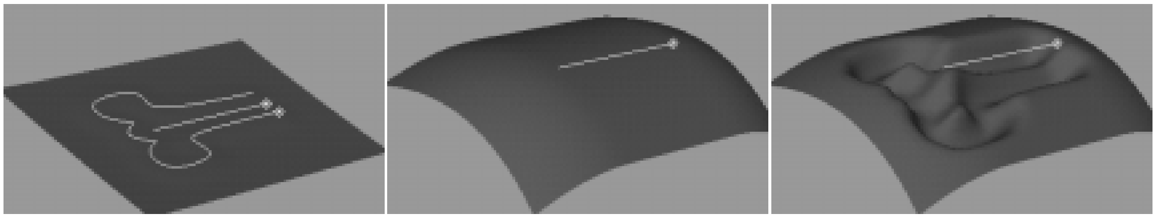
Wrinkles and creases can greatly enhance the realism of animated deformable objects. Cloth animation has become an important area of computer animation, especially related to human figure animation [11]. We show here how wires are effective in animating the crease lines along which wrinkles propogate. Wrinkle creases are either drawn as curves on the object surface by the animator or automatically generated in a set of predefined patterns.

Typical properties such as wrinkle thickness, intensity and stiffness of the material are easily captured by the various wire deformation parameters. The extent of wrinkle propagation can also be controlled. Figure 14(a) shows two wire curves as magnified wrinkles. Figure 14(b) shows the wrinkles propagating along the object surface. While one wrinkle is pulled along, remaining anchored, the other travels along the surface. The travelling wrinkle in Figure 14(b) is a result of pulling the reference curve R along the object surface with the wire curve W.

Figure 15 shows wrinkles that are procedurally generated by specifying parameters such as the number of crease lines, thickness, intensity, stiffness, and resistance to propagation. The approach is geometric and fast; it allows the animator to intuitively control over many salient visual features of wrinkle formation and propagation. Figure 16 illustrates this with a curtain animated using wires. A dynamic simulation of the wire curves results in a bead-curtain like animation. The wires then deform the object surface.

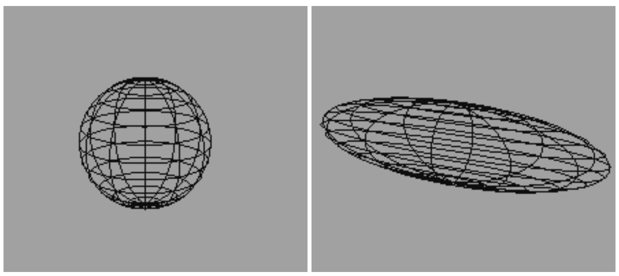
5.2 Stitching Object Surfaces

A wire-based geometry stitcher is a two step process. The first is the creation of wire curves along two edges of the geometry to be stitched. The wire curves are then blended pairwise to common seams. The object surfaces track the common seam, resulting in a stitch. We reparametrize the matching edges to a common domain before defining the stitch. Figure 17 depicts the stitching of one edge onto the other. Figure 18 demonstrates the levels of control available as various parameters are changed. These parameters give control not only over the stitch but also over the tearing of the stitch through the use and animation of locators along the seam (see Figure 18(f)). Figure 19 shows the results of a four-way stitch.

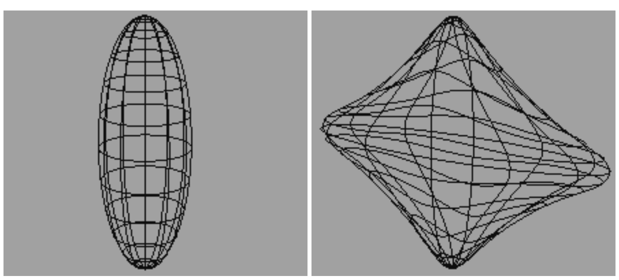


(a) no deformation (b) $localize = 0$ (c) $localize = 1$

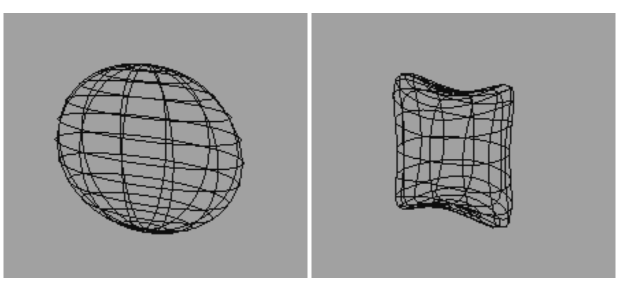
Figure 12: Localized influence of wires.



(a) no deformation (b) deformation 1



(c) deformation 2 (d) $m = 5$



(e) $m = 0$ (f) $m = -2$

Figure 11: Integration of two squash-stretch deformations using wires and different values of m.

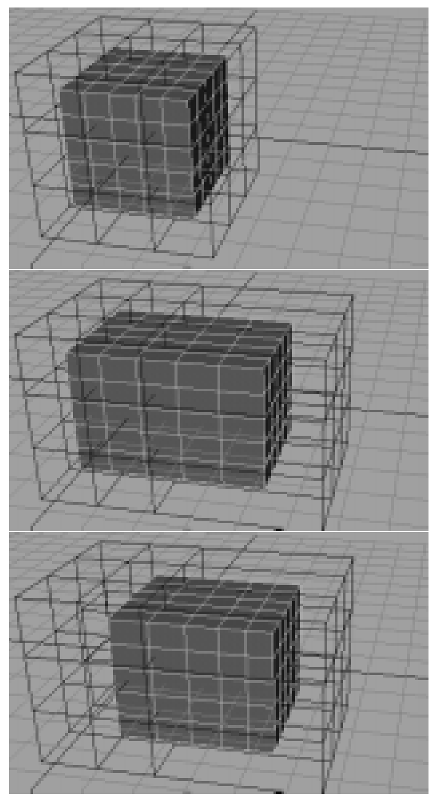
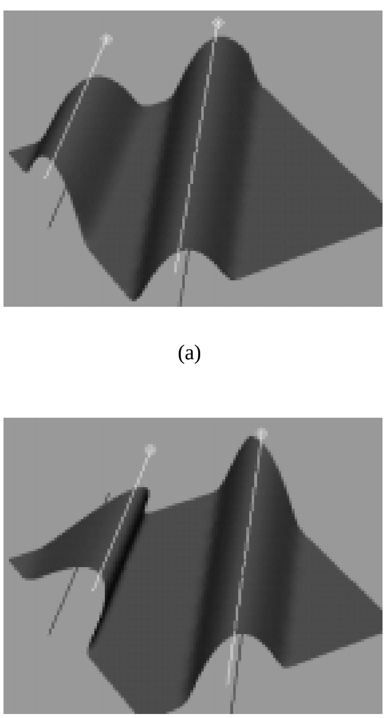


Figure 13: Wires simulating a free-form deformation lattice. Above, (a): no deformation. Middle, (b): deformed with $localize = 1$. Below, (c): deformed with $localize = 0$.

Figure 16: Curtain animation.



(a)

(b)

Figure 14: Wrinkle propagation using reference curves.

There are two shortcomings of the above approach. The first is that since each object is deformed independently, high orders of surface continuity across the stitch cannot be guaranteed. The control afforded by wire parameters r,s and f in particular, alleviates this to an extent. Second, seams are currently stitched pair-wise, thus imposing a stitching order, which can be restrictive.

5.3 Kinematics for flexible skeletons

Inverse kinematics on joint chains driving attached object geometry is popular for articulated-figure animation. Most IK solvers, especially efficient single chain solvers, have a problem with segments that scale non-uniformly during animation. This is essential if, for example, we wish to model a character with partially elastic bones. We replace a joint chain with a curve passing through it, so the control polygon of the curve acts like an articulated rigid body. We also introduce a rubberband like behavior by transforming the control points of the curve proportionally along the joint chain based on the motion of the end effector. The result is a semi-elastic skeletal curve. The curve then deforms the object geometry associated with the joint hierarchy as a wire. We use a large dropoff radius r so we can assume that every point on the geometry will track the curve equally and precisely (see Figure 5(b)). This in itself takes care of smoothing the regions around joints that often require special techniques to solve. Further, the arc-length of the wire curve is used to modulate the wire scale factor s, providing visually realistic volume preservation of the geometry on elastic deformations. Figure 20 shows the deformation to an arm as the kinematic solution is varied from perfectly rigid to perfectly elastic.

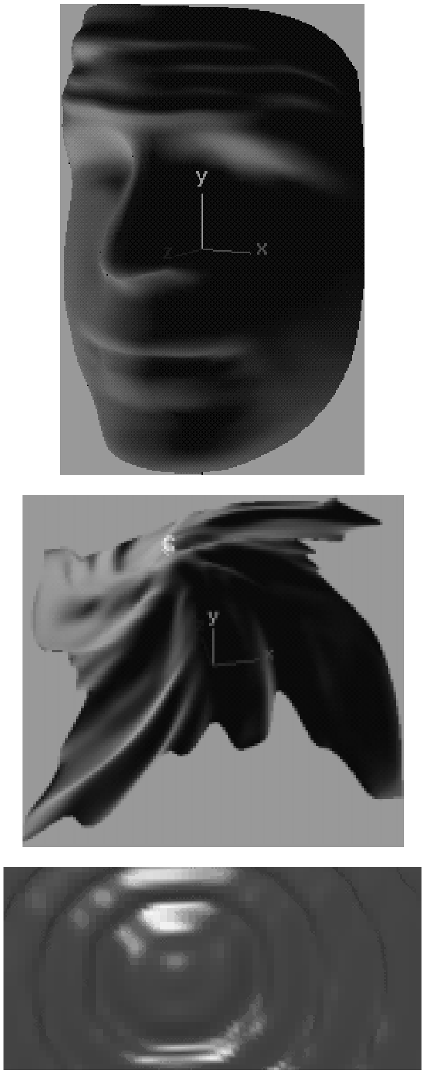


Figure 15: Procedural wrinkles. Top, (a): Tangential. Middle, (b): Radial. Bottom, (c): Ripple.

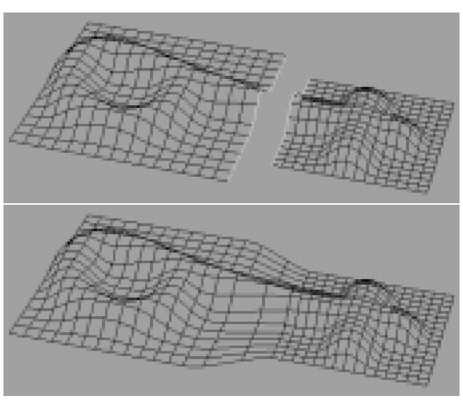
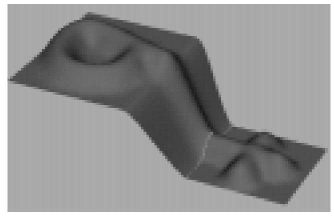
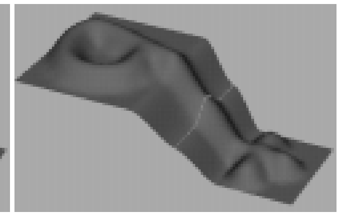


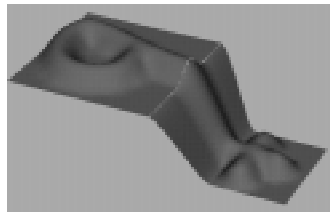
Figure 17: Simple stitch on two surfaces.



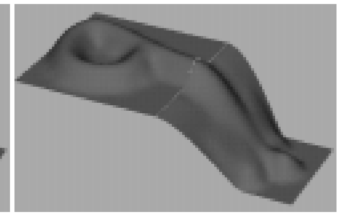
(a) blend weight=0



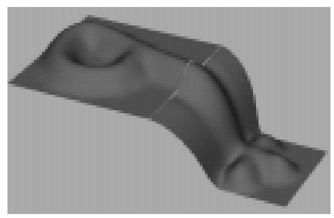
(b) blend weight=0.5



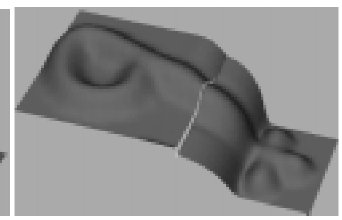
(c) blend weight=1



(d) Varying r



(e) Varying s



(f) Tear propagation

Figure 18: Control over stitch parameters.

6 Discussion

This paper has presented an effective geometric deformation technique, employing space curves and implicit functions that cleanly aggregate to deform an object. Our system has been completely implemented as a module in Alias|wavefront's *Maya* production modeling, animation and rendering graphics product. The slowest part of the algorithm is the closest-point on curve [14] calculation p_R for points P of the object geometry. Fortunately, this can be precomputed for each point P and must be recalculated only if the reference curve R is changed. In such cases, many values can be preprocessed, reducing the online wire deformation algorithm to a few vector operations per control vertex of the object geometry. Multiple wire interactions are accumulated incrementally in one pass.

Wire deformations work very well alone or in combination with existing techniques. FFDs, for example, are well suited for volume-oriented deformations. Arbitrarily shaped lattices can be cumbersome to construct for finer surface-oriented deformations. FFD lattices also usually have far more control points than wire curves for deformations of similar complexity. Wire curves can help by providing higher level control for lattice points to make complex FFD lattices more tractable (see Figure 21). Conversely, wires can emulate FFD lattices (see Figure 13).

Wires allow one to localize the complexity of a deformation on an object, and they provide a caricature of the object being modeled. The coupling of deformation and geometry is a significant advantage of wires. The technique also makes it easy to work in a

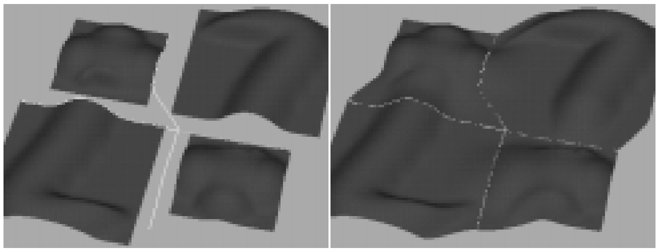


Figure 19: Four-Way stitch. Above, (a): the individual patches. Below, (b): the stitched patches.

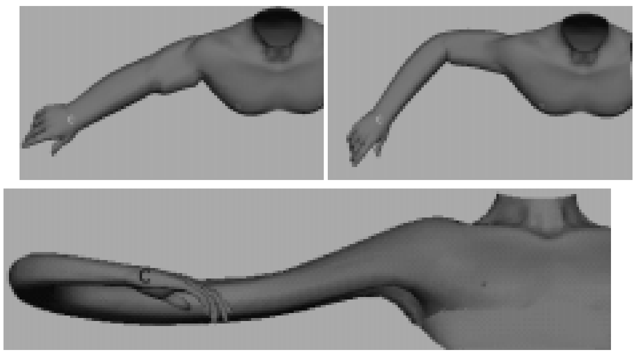


Figure 20: Inverse Kinematics for highly flexible skeletons.

multi-scale fashion. At the highest level a user may simply create a few wire curves, associate them with an object and move them around to verify that the object's surface properly tracks the motion of the curves. The region of the surface to be influenced can then be refined by adding domain curves and locators to the wires; finer-scale deformations can be added with more wires.

A point of comparison to our approach is "curve on surface" manipulation techniques that are found in some CAGD systems. There, least-squares techniques are used to isolate the control vertices relevant to a curve placed on or near a surface so that motion of the curve displaces the control points, which in turn changes the surface. Wires in most ways are a superior interaction technique because they are easier for a user to control, they are efficiently computed, and they apply to more general object representations. For surface patches with a low density of control points, changing a surface by deforming control points may not be as precise as a least squares solution, and it can suffer from aliasing artifacts.

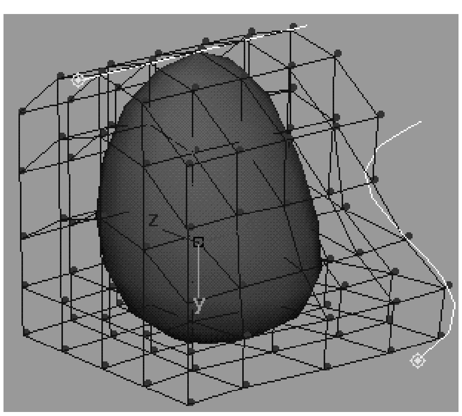


Figure 21: Deformation of a lattice by a wire.

In our implementation, some of our geometric algorithms could be made more efficient. Both finding the closest-point on curve and finding the region of influence of domain curves are good candidates for reworking. In our formulation in Section 2, we noted that there may be several closest points on a curve to a point in space. In cases of wire curves with of high curvature, the policy of picking the closest point with the smallest parameter value can cause singularities in the deformation. Such cases can be handled heuristically by breaking a wire curve into multiple wire curves in regions of high curvature.[3] While subdivision is rarely necessary, it is worth improving our policy to see if extreme cases can be handled automatically.

Acknowledgements

We thank Tom Sederberg and Kris Klimaszewski for their editorial help in the final stages of the paper, as well as the anonymous reviewers. Jeff Bell, Lincoln Holme and the animator geeks of Alias|wavefront were invaluable for their technical and creative help with this project. Finally, our congratulations and gratitude to the wonderful technical staff at Alias|wavefront who made *Maya* a reality. It is a remarkable accomplishment.

References

[1] A. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1:1–20, 1981.

[2] T. Beier and S. Neely. Feature based image metamorphosis. *Computer Graphics*, 26(2):35–42, 1992.

[3] J. Bloomenthal and B. Wyvill. Interactive techniques for implicit modeling. *Computer Graphics*, 24(4):109–116, 1990.

[4] J. Bloomenthal and K. Shoemake. Convolution surfaces. *Computer Graphics*, 25(4):251–256, 1991.

[5] J. Chadwick, D. Haumann and R. Parent. Layered construction for deformable animated characters. *Computer Graphics*, 23(3):234–243, 1989.

[6] Y.K. Chang and A.P. Rockwood. A generalized de Casteljau approach to 3D free-form deformation. *Computer Graphics*, 28(4):257–260, 1994.

[7] S. Coquillart. Extended free-form deformations: A sculpting tool for 3D geometric modeling. *Computer Graphics*, 24(4):187–196, 1990.

[8] M.-P. Gascuel. An implicit formulation for precise contact modeling between flexible solids. *Proc. of SIGGRAPH*, pages 313–320, 1993.

[9] W. Hsu, J. Hughes and H. Kaufman. Direct manipulation of free-form deformations. *Computer Graphics*, 26(2):177–184, 1992.

[10] R. MacCracken and K. Joy. Free-form deformations with lattices of arbitrary topology. *Computer Graphics*, 181–189, 1996.

[11] N. Magnenat-Thalmann and Y. Yang. Techniques for cloth animation. *SIGGRAPH Course Notes C20*, 151–163, 1991.

[12] F. Lazarus, S. Coquillart, and P. Jancene. Axial deformations: an intuitive deformation technique. *Computer-Aided Design*, 26(8):607-613, August 1994.

[13] T. Sederberg and S. Parry. Free-form deformation of solid geometric models. *Computer Graphics*, 20:151–160, 1986.

[14] P. Schneider. Solving the Nearest-Point-on-Curve Problem. *Graphics Gems*, Academic Press, vol.1:607–612, 1990.

[15] G. Wyvill, C. McPheeters and B. Wyvill. Animating soft objects. *Visual Computer*, 2:235–242, 1986.

[3] Recall from Section 4 that our formulation ensures that abutting wires do not introduce seaming or bulging artifacts.