Twister: A Space-Warp Operator for the Two-Handed Editing of 3D Shapes

Ignacio Llamas Byungmoon Kim Joshua Gargus Jarek Rossignac Chris D. Shaw

College of Computing
Georgia Institute of Technology

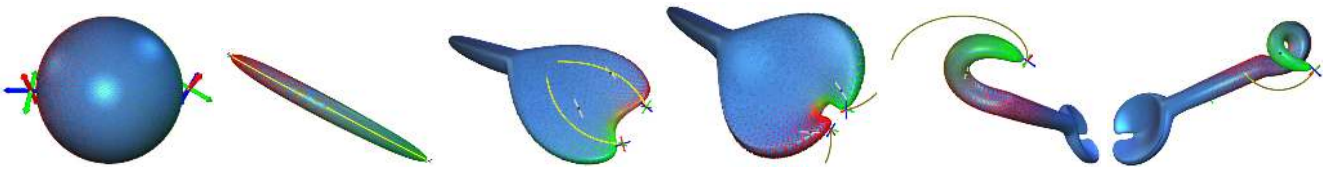

Figure 1: A sphere is deformed into a *spork* in 5 Twister steps. Red an green coloring indicate the area of influence of each hand.

Abstract

A free-form deformation that warps a surface or solid may be specified in terms of one or several point-displacement constraints that must be interpolated by the deformation. The Twister approach introduced here, adds the capability to impose an orientation change, adding three rotational constraints, at each displaced point. Furthermore, it solves for a space warp that simultaneously interpolates two sets of such displacement and orientation constraints. With a 6 DoF magnetic tracker in each hand, the user may grab two points on or near the surface of an object and simultaneously drag them to new locations while rotating the trackers to tilt, bend, or twist the shape near the displaced points. Using a new formalism based on a weighted average of *screw displacements*, Twister computes in realtime a smooth deformation, whose effect decays with distance from the grabbed points, simultaneously interpolating the 12 constraints. It is continuously applied to the shape, providing realtime graphic feedback. The two-hand interface and the resulting deformation are intuitive and hence offer an effective direct manipulation tool for creating or modifying 3D shapes.

CR Categories: 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: free-form deformation, two-handed interaction, displacement and orientation constraints

1 Introduction

Designers' productivity may be enhanced by interactive techniques for shape manipulation, which link gestures to predictable deformation effects and provides realtime visual, and possibly haptic, feedback. Contributions in this field typically strive to define an abstract mapping between gesture and shape modification and an algorithmic description of how parameters of shape modifying operators are to be inferred from measures extracted by tracking a gesture.

In this paper, we advocate a grab and drag operation, but instead of restricting the dragging to a single point, we let the designer use both hands to grab and simultaneously drag and twist two different portions of the space in which the surface is embedded. We compute a 3D space warp that satisfies the position and orientation constraints imposed by both hands. Visual feedback is provided at interactive rates, displaying the immediate effect of the 3D warp on the surface of interest, while the user's hands are still moving and modifying the constraints. We compare our solution for computing the constraint satisfying space warp to other solutions and, based on the deformed surfaces obtained, conclude that it has clear benefits. For example, Figure 1 shows how five two-handed gestures have been used to deform a sphere into a *spork* (combination of spoon and fork).

The method described in this paper focuses on how to produce intuitive deformations of general surfaces from hand gestures. As such, it applies to the creation of 3D models for artistic purposes or for the exploration of crude approximations of shapes. It does not, however, intend to solve issues involved with the precise design of manufactured or functional surfaces.

In the following section we justify our design decisions. Section 3 summarizes related work in the area of non physically based shape deformation. Section 4 presents our preferred solution for the deformation using six and twelve constraints. Section 5 presents alternative solutions and compares them with the solution in section 4. Section 6 provides some implementation details and results.

2 Motivation of Design Choices

In this section we justify our main design choices: use of grab-and-drag shape deforming operations, use of two hands, use of rigid handles, use of orientation constraints, and use of space warps.

Why Grab-and-Drag: Many different techniques can be used to construct 3D shapes. Some interpolate 3D points with implicit surfaces [Turk and O'Brien 2002], others automatically construct surfaces that interpolate 3D curves [Sachs et al. 1991], [Wesche and Seidel 2001], [Grossman et al. 2002], or 2D profiles [Igarashi et al. 1999]. Yet others provide means for the direct drawing of surfaces [Schkolne et al. 2001] or for space painting and carving [Galyean and Hughes 1991]. An alternative to these shape creation techniques is the warping or deformation of existing shapes. Various methods and interaction paradigms have been developed for this purpose. Some let users manipulate the control points of free-form deformation lattices [Sederberg and Parry 1986], [Coquillart 1990], or vertices of multiresolution meshes [Zorin et al. 1997]. Higher-level techniques have been proposed for the direct manipulation of free-form deformation [Hsu et al. 1992] or for the mapping of two-handed spatial and pictographic gestures to axial deformations [Nishino et al. 1998]. The technique described in this paper belongs to this group. It is based on a grab-and-drag shape deforming operator, and thus falls within the paradigm of direct manipulation of shape. It does not limit the user's interaction to control points and does not restrict the operations to be axial deformations.

Why Use Rigid Handles: A human hand offers many degrees of freedom, some of which have been exploited for shape control using gloves [Nishino et al. 1998]. We chose instead to have the user control the position and orientation of a small rigid object in each hand. These two 6 DoF tracked objects represent handles that may each be used to grab a portion of space and to pull and twist it. Our decision was justified by the availability of robust 6 DoF trackers [Polhemus 2002] and by the simplicity of the corresponding intentional semantics.

Why Use Orientation Constraints: The way our fingers and hands manipulate a deformable object imposes constraints on the orientation of its surface after a deformation. Combining multiple position constraints (i.e. 3 triangle vertices), each roughly corresponding to a finger-imposed constraint, would allow the user to control the surface orientation indirectly. However, we have concluded that controlling multiple position constraints individually with the purpose of changing surface orientation may be cumbersome, resulting in undesired undulations of the surface and in numeric instabilities. Thus we support orientation constraints. Previous work by Fowler [1992], for the geometric manipulation of tensor product surfaces, and by Gain [2000], to allow the manipulation of the derivative frame with Directly Manipulated Free-Form Deformation (DMFFD), argue for the usefulness of such orientation control. Figure 2 illustrates orientation constraints showing the deformation of a flat surface with three different constraints.

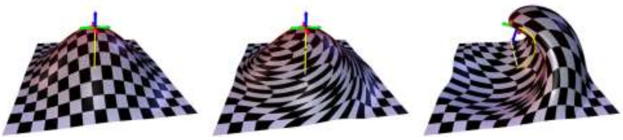

Figure 2: Benefits of orientation constraints (left to right): translation along vector OO', translation and rotation about translation vector OO', translation and screw rotation about an axis perpendicular to the translation vector OO'.

Why Use Two Hands: A previously proposed use of two hands in the context of object manipulation allows the user to control the object's global position and orientation with one hand while the other hand performs some editing operation [Shaw and Green 1997]. This method is used in our system. In addition to such an asymmetric use of the hands, we propose using two hands together to control a shape deforming operation and to provide a natural interface that builds upon our daily experience in manipulating cloth, paper, or plastic with both hands. Figures 1 and 3 show deformations that would be difficult to specify with a single hand.

Why Warp Space: Some shape deformation techniques are physically based ([Terzopoulos and Fleischer 1988], [James and Pai 1999], or [Szeliski and Tonnesen 1993],) providing high fidelity approximations of material properties such as plasticity, elasticity, or flexibility. A physically plausible behavior that is intuitively understood by the designer makes it easier to predict the effect of a gesture and thus to plan a sequence of deformations that lead to a desired shape. Unfortunately, physical realism is too expensive for realtime feedback. Thus, we have opted for a compromise, which offers a simple and intuitive map between hand-gestures and space

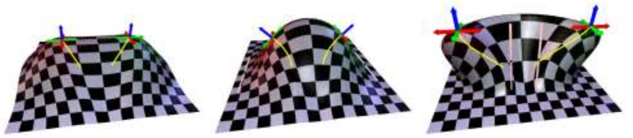

Figure 3: Benefits of two-hand deformation (left to right): translation only, translation and screw rotations, translation and screw rotations bending the surface. The screw motion trajectory is shown as a yellow curve. The screw axis of each screw rotation is shown as a pink cylinder.

warps that is independent of the manipulated surface. The cost of computing the warp parameters is negligible and its effect appears physically plausible and quite predictable.

3 Prior Work

Deformation techniques, including physically based ones, are surveyed in [Gibson and Mirtich 1997] and [Gain 2000]. Space warping and morphing techniques are thoroughly covered in [Gomes et al. 1999]. Physically based deformations are covered in [Metaxas 1996]. In accordance with the focus of this paper, this section summarizes work in non physically based deformations, with special emphasis in space warping techniques.

Several approaches are based on the local deformation of a surface. Parent [1977] used decay functions to diminish the effect of a displacement imposed on a user selected vertex in a mesh as a function of the geodesic distance, approximated by the topological distance on the connectivity graph. Allan et al. [1989] and Bill [1994] developed systems that made use of more elaborate decay functions. Modern software packages, such as Discreet 3ds max 4 and 5 [Discreet 2002], also allow weighted manipulation of vertices with an adjustable decay function.

Zorin et al. [1997] presented a system for multiresolution mesh editing in which vertices at different levels of subdivision can be manipulated with adjustment vectors defined in local frames, with the purpose of preserving details.

Other approaches are based on the idea of space warping, which is independent of the representation ,and thus well suited for editing triangle meshes, voxel volumes, control points of interpolating curve patches or scattered point data. Barr [1984] published global mappings for twisting, bending and tapering space deformations. Chang and Rockwood [1994] introduced a generalized de Casteljau approach to deformations. The *Axial Deformations* of Lazarus et al. [1994] allowed the use of curves with any shape as the axis for a generalized cylinder with variable radii and local frames at key points. *Wires*, by Singh and Fiume [1998], took curve based deformation techniques further, but at a higher computational cost.

Sederberg and Parry [1986] introduced the free-form deformation (FFD), based on lattices of control points and trivariate Bernstein polynomials. Greissmair and Purgathofer [1989] implemented FFD with trivariate B-splines; Coquillart [1990] and MacCracken and Joy [1996] extended FFD to support more general lattices, while Hsu et al. [1992] developed a version of FFD that allows direct manipulation.

Borrel and Bechmann [1991] and later Borrel and Rappoport [1994] developed realtime techniques for computing space warps that simultaneously interpolate several point-displacement constraints. More recently, Milliron et al. [2002] introduced a general framework for geometric warps.

4 The Proposed Twister Solution

By moving a tracker with one hand, the user controls its position and orientation, *i.e.*, the pose of a coordinate system. When the user grabs a point in space by pressing a button, the pose of the tracker is recorded as the starting pose. While the button remains pressed, subsequent displacements and rotations of the tracker update the ending pose. At any moment during the manipulation phase, Twister computes a space warp that takes as input the starting pose C (defined by a local coordinate system with origin O and three orthogonal directions U,V,W) to the ending pose C' (defined by a local coordinate system with origin O' and directions U',V',W'). We assume that $W = U \times V$ and $W' = U' \times V'$. We want a smooth space warp that takes the starting pose to the ending pose exactly and whose effect decays away from the grabbed point. This condition results in three translational and three rotational constraints. To meet these six constraints, we construct a screw motion that moves the starting pose to the ending pose. We apply the full screw at the grabbed point and diminish its magnitude with distance from the grabbed point. When the two trackers are used simultaneously we compute the two screws for each point and blend them.

4.1 Single-hand Twister

The screw motion of a point P describes an helical trajectory. It is fully defined by a starting and ending pose. A screw motion that maps C to C' is completely defined by an axis of unit direction D passing through a point A and by an angle a of rotation around the axis and a translation distance d along the axis [Rossignac and Kim 2001]. A screw motion is minimal when $a \in [0,\pi]$. The parameters of a minimal screw may be computed as follows.

Let $UU' := U' - U$, $VV' := V' - V$, and $WW' := W' - W$. When $|UU'| = |VV'| = |WW'| = 0$, the deformation is a pure translation by vector OO'. Otherwise, only one difference can be null. Assume without loss of generality that UU' is not null. (If UU' is null, rename the vectors.) As illustrated in Figure 4, the direction of the screw axis D, the translation distance d and the angle a are computed by

$$\begin{aligned} &D := UU' \times VV' + VV' \times WW' + WW' \times UU'; \quad D := D/|D| \\ &a := 2\sin^{-1}\left(\frac{|UU'|}{2|D \times U|}\right) \\ &d := D \cdot OO' \end{aligned} \tag{1}$$

The projection of O' on the plane through O with normal D is $O'' := O' - dD$. Let M be the mid-point $(O+O'')/2$ of OO''. From $\tan(a/2) = |OM|/|AM|$ the distance $|AM|$ is computed as $|AM| = \frac{|OO''|}{2\tan(a/2)}$. The direction from M to A is computed by $N := D \times OO''$. Note that since $D \perp OO''$ and $|D| = 1$, $|N| = |OO''|$. The point A is computed as

$$\begin{aligned} A &= M + |AM|\frac{N}{|N|} = M + \frac{|OO''|}{2\tan(a/2)}\frac{D \times OO''}{|OO''|} \\ &= M + \frac{D \times OO''}{2\tan(a/2)} = M + \frac{D \times (OO' - dD)}{2\tan(a/2)} \\ &= M + \frac{D \times OO'}{2\tan(a/2)} \end{aligned} \tag{2}$$

The image of an arbitrary point P under this screw motion is $P' := (AP \cdot U)U' + (AP \cdot V)V' + (AP \cdot W)W' + dD + A$. Consider the family S of interpolating transformations that define the successive poses taken by an object during this screw motion as the time parameter t is varied from 0 to 1. For a given $t, S(t)$ may be produced through the following sequence. Translate by $-A$; Rotate D to Z; Translate by td along Z; Rotate by ta around Z; Rotate Z to D; Translate by A. To animate a screw motion, for each time t, the combined effect of these transformations may be composed into a single 4×4 matrix. If the desired result was the animation of an object along the helical trajectory of the screw motion, the parameter t would be identical for all points in the object. To produce the desired space warp, a different t is computed for each point P on the object (the set of points P depends on the underlying representation and may for instance comprise the vertices of a triangle mesh). We use a function $t(P)$ to control the decay of the effect of the screw with distance from O. By using the decay, $t(P) := \cos^2\left(\frac{|OP|}{r}\frac{\pi}{2}\right)$, we obtain $S(1)$ at O, $S(0)$ outside of a ball of center O and radius r, and a smooth variation in between.

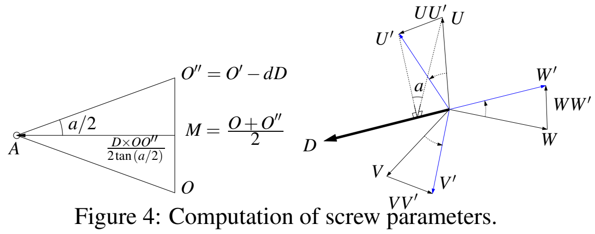

Figure 4: Computation of screw parameters.

4.2 Two-hand Twister

The formulation described above leads to intuitive and effective deformations for a single hand constraint. It may also be used to solve for two simultaneous constraints when the grabbed point O of each hand lies outside of the ball of influence of the other, by simply adding the decayed effect of each screw.

In cases where the ball of influence of one hand contains the grabbed point Q of the other hand (i.e, $|OQ| < r$), naive addition of the displacements produced by the two sets of constraints affecting a common point would not satisfy the constraints. Hence, to ensure the desired constraint satisfaction, we adjust the formula for t as follows. Let $s = |OQ|$ and $p = OP \cdot OQ$. If $p < 0$, we use the same expression for t as before. Otherwise, we replace P by $P + (p/s)((r-s)/s)(OQ/s)$ in the computation of t. This transformation corresponds to a minimal squashing of half of the ball of influence, so that it no longer contains Q. The squashing is formally a scaling by s/r, in the direction OQ, with fixed point O as shown in Figure 5. When needed, we apply such a scaling to the decay functions for both hands. We compute the images of P by the decayed screw motions of both hands and then add the two displacements to P. Note that our choice of the function t guarantees that the sum of the two weights for a point on the line segment between the two grabbed points will be one when $|OQ| < r$; *i.e.*: $\forall P$: $|OP|/r \in [0,1] \Rightarrow t(|OP|/r) + t(1 - |OP|/r) = 1$. This property avoids undesired bumps in the common region.

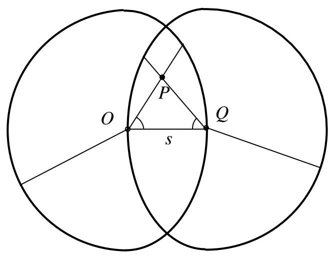

Figure 5: The balls of influence are squashed when they contain the grabbed point of the other ball.

When the radius of influence r is large enough to keep the object inside the ball of influence, and the two grabbed points are placed in appropriate positions, we can obtain deformations which, although not strictly equivalent, are very similar in their effects to some of Barr's global and local deformations [Barr 1984]. For example, we can obtain a *twist*-like deformation by rotating the two trackers in opposite directions about the line that joins them (see Figure 6). A *bend*-like deformation can be obtained by rotating the trackers in opposite direction about an axis perpendicular to the line joining them, and an optionally translating them symmetrically in the plane that has the rotation axis as normal. The main advantage of our technique is that the user can select from a wider range of less *pure* warps through an intuitive two-handed interface, easily combining twisting and bending.

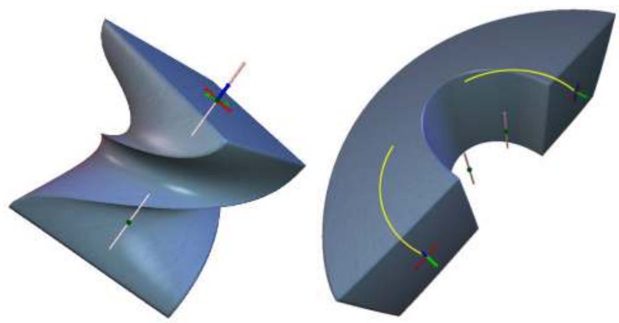

Figure 6: A box twisted (left) and bent (right) by Twister. The axis of each screw deformation is displayed as a small pink cylinder.

5 Alternative Approaches

In this section, we develop two alternative formulations of a deformation that satisfies displacement and orientation constraints and show that they are less effective than the Twister approach introduced in the previous section.

The first alternative is to consider a linear combination of translation and rotation around the constraining point (Figure 7). This simple scheme first satisfies the rotational constraint by applying a rotation about the axis D at point O and then the translational one by applying a translation OO'.

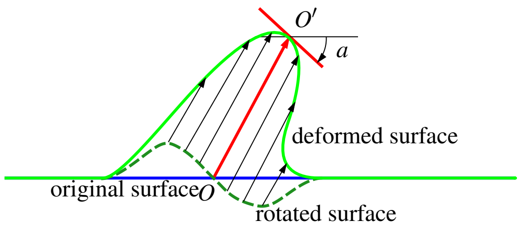

Figure 7: Illustration of the linear combination of translation and rotation.

The same computations used above for the screw motion can be used to compute the rotation axis direction D and angle a. The decay function is the same as the one discussed above. The main difference lies in the choice of the rotation axis, which here goes through O and needs no longer be parallel to the translation vector. The effect of this change on the resulting deformation is significant, especially for large displacements with large angles of rotation.

The second alternative is based on *Scodef* [Borrel and Rappoport 1994], which interpolates multiple translational constraints. Given a set of constraining points $O_1, O_2, \ldots, O_n$ and their warped positions $O'_1, O'_2, \ldots, O'_n$, *Scodef* combines n radial basis functions $\phi(X) \in \mathbb{R}$ to compute a space warp $F(P), P \in \mathbb{R}^3$ that satisfies all of these constraints. The space warp $F(P)$ is in the form of the weighted sum of the bases centered at $O_1, O_2, \ldots, O_n$.

$$F(P) = \sum_{j=1}^{n} \phi_j A_j \ , \quad \phi_j = \phi(P - O_j) \ , \quad A_j \in \mathbb{R}^3 \tag{3}$$

Unknowns are the coefficients A_j that can be solved from the constraining equations.

$$F(O_i) = \sum_{j=1}^{n} \phi_{ij} A_j = O'_i \ , \quad \phi_{ij} = \phi(O'_i - O_j) \tag{4}$$

Even though *Scodef* does not solve the rotational constraints directly, it can achieve rotation by placing three constraint points along the axes of the initial coordinate frames and defining their images in the target frames. Note that the rotation can be arbitrarily close to the commanded rotation by placing these points close enough to the constraining points. In the two-hand case there are two sets of such constraints, therefore we need six constraining points.

Figure 8 shows the result of the deformations produced by the three methods in a 2D planar setting. The original undeformed shape is a straight line. For small rotation angles, all three behave well and produce similar results. However, as the rotation angle increases, self intersections and the sharp cusp that appears near them are least prominent with Twister. We conclude that for deformations combining large displacements and large rotations, our screw-based approach is capable of producing better results than the other two.

In addition to a tendency to produce more self-intersections, the *Scodef*-based method presents another problem, known as *space tearing*, which has its roots in the mathematical blending procedure inherent to it. This problem is reported in [Borrel and Rappoport 1994]. When two constraints are close together compared to the region of influence of the basis function, *i.e.*, in near singular case, they are conflicting with each other. In this case, the resulting magnitudes of the bases are very large. As a result, large portions of the surface are deformed in a direction opposite to that imposed by the displacement constraint.

Although *Scodef* based deformations can accommodate more than two sets of constraints, we do not need to exploit this power in our two hand operation mode. The simplicity of our user interface has enabled us to design a blending function that handles the overlapping area well. We conclude that the combination of the screw motion and the squashing-based blending produces better results than the other two approaches.

6 Implementation and Results

We have implemented Twister in C++ with OpenGL on a dual Pentium 3 866 Mhz with 256 MB of RAM, a NVIDIA Quadro4 900 XGL graphics card and two Polhemus Fastrak trackers with three buttons attached to each. This implementation provides realtime (20 frames per second in average) feedback for meshes with up to 30,000 vertices. In order to preserve smoothness of the deformed surface without imposing an excessive burden on the processing and rendering units, an adaptive surface subdivision must be performed during the deformation [Gain and Dodgson 1999]. To decide whether an edge (A, B) is to be split the distance $|(S(A) + S(B))/2 - S((A + B)/2)|$ is compared against a threshold that can be adjusted to trade speed for smoothness.

By recording the grab-release pairs of constraints for each hand at every deformation step we can reproduce at a later time a sequence of deformations. These deformations can be played back on the same object or on a slightly different or more detailed version of the object. For example, a deformation may be designed

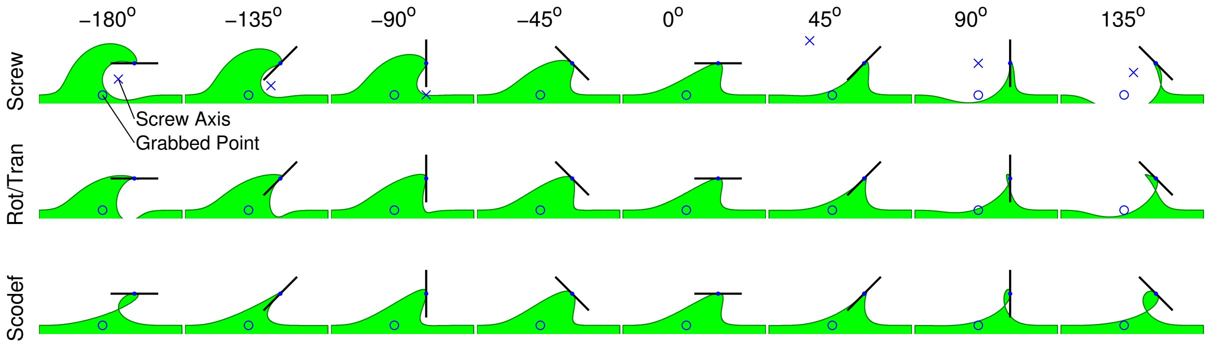

Figure 8: Comparison in various amounts of rotation.

at interactive rates on a simplified imposter and then executed on a full resolution finely tessellated model. The poses that control the animations may be generated procedurally and may be used to illustrate an interactive editing session.

To increase user's productivity, we provide a quick way of changing the radius of the ball of influence by pressing a button on each tracker and moving the trackers closer or further away.

A spherical volume will sometimes select too much. Our users have developed a user interface idiom where they take advantage of the squashing of the volumes that takes place when the distance between the two balls of influence is less than their radii, and then perform the warp with only one hand.

There is a limitation on the maximum angle of rotation that can be reached in a single Twister operation, which is 180 degrees in any direction. The user can concatenate several Twister operations by pressing a button to freeze the model, while maintaining the same selection of vertices and their weights. This allows the user to return the hand to a relaxed position before twisting another 180 degrees. This process can be repeated any number of times.

Our experiments indicate that the Twister operator combined with the two-handed interface is intuitive and effective for creating and editing a large variety of shapes. For example, the series of images in Figure 1 show the 5 Twister operations that were used to transform a sphere into a *spork*. Figure 9 shows three objects produced from a sphere in a few deformation steps.

7 Future Work

In addition to the integration of Twister with a broader set of previously proposed shape editing operation, we plan the following two extensions.

With the recent development of new materials and of high precision manufacturing technologies, one may hope that high-resolution, computer controlled hydraulic mechanisms that sense pressure and manipulate real surfaces will soon become available and will provide the ultimate man-shape interaction paradigm. One such project sponsored by the NSF [Allen et al. 2001] has led to the development of a new haptic interface through which the computer controlled surface can sense and track individual fingers of a human hand [Gargus et al. 2002]. This interface will naturally support multiple displacement constraints. We envision tracking three finger tips per hand and attaching a coordinate system to each triplet. In this way, the user will be able to control the position and orientation of the two coordinate systems with a displacement or rotation of the wrists. We hope that the Twister operator described here will be a useful tool for specifying large deformations of such a new medium.

We are exploring the possibility of using vertex programming facilities available on current graphics hardware to provide better frame rates when interactively editing larger models. We have found two ways of taking advantage of vertex programs. In both cases graphics hardware acceleration is only used for display purposes to provide realtime feedback to the designer while warping. Once complete, the same computations are performed on the CPU; therefore there is no need to read data back from the graphics card. Vertex programs may be used to compute a color that represents the weight that is assigned to each vertex as a function of its distance to the two trackers. In our implementation, this color feedback helps the user decide where to place the trackers. Vertex programs may also compute the warped position for each vertex, as described in section 4. Early experiments with vertex programs have shown significant performance improvements.

8 Acknowledgements

This work was supported by NSF/ITR Grant 0121663, and by a Seed Grant from the GVU Center at Georgia Tech. We also wish to thank Jesse Shieh for his help and Greg Turk for his feedback on the early versions of this manuscript.

References

ALLAN, J. B., WYVILL, B., AND WITTEN, I. 1989. A Methodology for Direct Manipulation of Polygon Meshes. *New Advances in Computer Graphics (Proceedings of CG International '89)*, 451–469.

ALLEN, M., BOOK, W., EBERT-UPHOFF, I., ROSEN, D., AND ROSSIGNAC, J., 2001. Digital Clay for Shape Input and Display: NSF-ITR/PE+SY Award#:0121663, September.

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

BILL, J. R., AND LODHA, S. 1994. Computer Sculpting of Polygonal Models using Virtual Tools. Tech. Rep. UCSC-CRL-94-27, Baskin Center for Computer Engineering and Information Sciences, University of California, Santa Cruz, U.S.A., July.

BORREL, P., AND BECHMANN, D. 1991. Deformation of n-Dimensional Objects. In *SMA '91: Proceedings of the First Symposium on Solid Modeling Foundations and CAD/CAM Applications*, ACM Press, ACM, 351–370.

BORREL, P., AND RAPPOPORT, A. 1994. Simple Constrained Deformations for Geometric Modeling and Interactive Design. *ACM Transactions on Graphics, 13*, 2, 137–155.

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

DISCREET, 2002. Discreet 3ds max 5, http://www.discreet.com/products/3dsmax/.

Figure 9: Some objects generated with Twister from a sphere.

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

GAIN, J. E., AND DODGSON, N. 1999. Adaptive Refinement and Decimation under Free-Form Deformation. In *Eurographics UK '99*, Eurographics.

GAIN, J. E. 2000. *Enhancing Spatial Deformation for Virtual Sculpting*. PhD thesis, St. Johns College, University of Cambridge.

GALYEAN, T. A., AND HUGHES, J. F. 1991. Sculpting: an Interactive Volumetric Modeling Technique. *Computer Graphics (Proceedings of ACM SIGGRAPH 91), 25*, 4, 267–274.

GARGUS, J., KIM, B., LLAMAS, I., ROSSIGNAC, J., AND SHAW, C. 2002. Finger Sculpting with Digital Clay. Tech. Rep. GIT-GVU-02-22, GVU Center and College of Computing, Georgia Institute of Technology.

GIBSON, S. F. F., AND MIRTICH, B. 1997. A Survey of Deformable Modeling in Computer Graphics. Tech. Rep. TR-97-19, Mitsubish Electric Research Laboratoy.

GOMES, J., DARSE, L., COSTA, B., AND VELHO, L. 1999. *Warping and Morphing of Graphical Objects*. Morgan Kaufmann Publishers Inc.

GREISSMAIR, J., AND PURGATHOFER, W. 1989. Deformation of Solids with Trivariate B-splines. In *Eurographics '89*, Eurographics, 137–148.

GROSSMAN, T., BALAKRISHNAN, R., KURTENBACH, G., FITZMAURICE, G., KHAN, A., AND BUXTON, B. 2002. Creating Principal 3D Curves with Digital Tape Drawing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, ACM SIGCHI, 121–128.

HSU, W. M., HUGHES, J. F., AND KAUFMAN, H. 1992. Direct Manipulation of Free-Form Deformations. *Computer Graphics (Proceedings of ACM SIGGRAPH 92), 26*, 2, 177–184.

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: a Sketching Interface for 3D Freeform Design. In *Proceedings of ACM SIGGRAPH 99*, ACM Press, ACM SIGGRAPH, 409–416.

JAMES, D. L., AND PAI, D. K. 1999. ArtDefo - Accurate Real Time Deformable Objects. In *Proceedings of ACM SIGGRAPH 99*, ACM Press, ACM SIGGRAPH, 65–72.

LAZARUS, F., COQUILLART, S., AND JANCENE, P. 1994. Axial Deformations: An Intuitive Deformation Technique. *Computer Aided Design, 26*, 8 (August), 607–613.

MACCRACKEN, R., AND JOY, K. I. 1996. Free-Form Deformation With Lattices of Arbitrary Topology. In *Proceedings of ACM SIGGRAPH 96*, ACM Press, ACM SIGGRAPH, 181–190.

METAXAS, D. N. 1996. *Physics-Based Deformable Models: Applications to Computer Vision, Graphics, and Medical Imaging*. Kluwer Academic Publishers, January.

MILLIRON, T., JENSEN, R. J., BARZEL, R., AND FINKELSTEIN, A. 2002. A Framework for Geometric Warps and Deformations. *ACM Transactions on Graphics, 21*, 1, 20–51.

NISHINO, H., UTSUMIYA, K., AND KORIDA, K. 1998. 3D Object Modeling Using Spatial and Pictographic Gestures. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, ACM Press, ACM, 51–58.

PARENT, R. E. 1977. A System for Sculpting 3-D Data. *Computer Graphics (Proceedings of ACM SIGGRAPH 77)* (July), 138–147.

POLHEMUS, 2002. Polhemus Fastrak, http://www.polhemus.com/ftrakds.htm.

ROSSIGNAC, J. R., AND KIM, J. J. 2001. Computing and Visualizing Pose-Interpolating 3D Motions. *Computer Aided Design 33*, 4, 279–291.

SACHS, E., ROBERTS, A., AND STOOPS, D. 1991. 3-Draw: a Tool for Designing 3D Shapes. *Computer Graphics and Applications, IEEE 19*, 18–26.

SCHKOLNE, S., PRUETT, M., AND SCHROEDER, P. 2001. Surface Drawing: Creating Organic 3D Shapes with the Hand and Tangible Tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, ACM SIGCHI, 261–268.

SEDERBERG, T. W., AND PARRY, S. R. 1986. Free-Form Deformation of Solid Geometric Models. *Computer Graphics (Proceedings of ACM SIGGRAPH 86), 20*, 4, 151–160.

SHAW, C., AND GREEN, M. 1997. THRED: A Two-Handed Design System. *Multimedia Systems, 5*, 2, 126–139.

SINGH, K., AND FIUME, E. 1998. Wires: A Geometric Deformation Technique. In *Proceedings of ACM SIGGRAPH 98*, ACM Press, ACM SIGGRAPH, 405–414.

SZELISKI, R., AND TONNESEN, D. 1993. Surface Modeling with Oriented Particle Systems. In *Proceedings of ACM SIGGRAPH 93*, ACM Press, ACM SIGGRAPH, 185–194.

TERZOPOULOS, D., AND FLEISCHER, K. 1988. Modeling Inelastic Deformation: Viscolelasticity, Plasticity, Fracture. *Computer Graphics (Proceedings of ACM SIGGRAPH 88), 22*, 4, 269–278.

TURK, G., AND O'BRIEN, J. F. 2002. Modelling with Implicit Surfaces that Interpolate. *ACM Transactions on Graphics, 21*, 4, 855–873.

WESCHE, G., AND SEIDEL, H.-P. 2001. FreeDrawer: a Free-Form Sketching System on the Responsive Workbench. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, ACM Press, ACM, 167–174.

YU-KUANG CHANG AND ALYN P. ROCKWOOD. 1994. A Generalized de Casteljau Approach to 3D Free-form Deformation. In *Proceedings of ACM SIGGRAPH 94*, ACM Press, ACM SIGGRAPH, 257–260.

ZORIN, D., SCHROEDER, P., AND SWELDENS, W. 1997. Interactive Multiresolution Mesh Editing. In *Proceedings of ACM SIGGRAPH 97*, ACM Press, ACM SIGGRAPH, 256–268.