An Interface for Virtual 3D Sculpting via Physical Proxy

Jia Sheng, Ravin Balakrishnan, Karan Singh
Department of Computer Science
University of Toronto
www.dgp.toronto.edu

ABSTRACT

We explore the design space of using direct finger input in conjunction with a deformable physical prop for the creation and manipulation of conceptual 3D geometric models. The user sculpts virtual models by manipulating the space on, into, and around the physical prop, in an extension of the types of manipulations one would perform on traditional modeling media such as clay or foam. The prop acts as a proxy to the virtual model and hence as a frame of reference to the user's fingers. A prototype implementation uses camera-based motion tracking technology to track passive markers on the fingers and prop. The interface supports a variety of clay-like sculpting operations including deforming, smoothing, pasting, and extruding. All operations are performed using the unconstrained fingers, with command input enabled by a small set of finger gestures coupled with on-screen widgets.

Categories and Subject Descriptors

H.5.2 [Information Interfaces & Presentation]: Interaction styles.

General Terms

Design, Human Factors.

Keywords

Direct finger input, gestural interaction, clay modeling, conceptual 3D modeling, high degree-of-freedom input.

1. INTRODUCTION

Current digital 3D modeling software enables the creation and manipulation of very sophisticated geometric objects. However, the user interfaces for these 3D modelers can be quite complex and ill-suited to the creative process. Artists and designers who can transform an idea into a rough prototype in a very short time using traditional physical media such as clay, often have to expend significantly more effort to create the digital equivalent. This is partly due to the interface requiring users to understand and work with the low-level attributes of the digital model's underlying mathematical representation. While this additional complexity may be acceptable when a precise final model is required, it is quite undesirable when the artist just wants to create "quick and dirty" concept models. Therefore, it is unsurprising that traditional media and methods continue to be used in industry in the early concept design stages of a project. With the increasing quality of 3D range scanners and point-cloud data processing software, it is even relatively easy to convert the physical models into digital form when required. However, despite their many advantages, physical media simply do not have the highly desirable features of the digital form such as easy replication, construction history, undo and redo, easy spatial scaling, precision, and the ability to access interior parts of the model. Thus, it would be desirable if the physical process could be replaced with a digital one right from the start, but only if the resultant loss of fidelity that exist in current digital interfaces can be mitigated. Researchers have made significant inroads into this problem with a variety of innovative interfaces [1, 5, 12, 15, 21, 25, 29, 31, 35, 39, 43] that allow creation of rough models. However, these typically require the use of input devices and/or abstractions that introduce a level of indirection compared to direct manipulation of physical media with the human hand.

In this paper, we present an interface for "clay-like" conceptual 3D modeling that tracks passive markers on the user's fingers in conjunction with a deformable physical prop to allow for very direct creation and manipulation of virtual model (Figure 1). In a style akin to working with physical deformable media, the user sculpts the virtual model by moving his/her fingers on, into, and around the physical prop which serves as a proxy to the virtual model and thus as a frame of reference to the user's fingers. Techniques commonly used in clay modeling, such as smoothing, deforming, pasting, and extruding are implemented and extended to leverage the capabilities of the virtual realm. Command input is enabled via a small set of finger gestures in conjunction with on-screen widgets, allowing all interaction to be achieved without additional input devices. Our goal is not to create a 3D modeler per se, but rather to explore the design space of this new style of direct finger and prop based interaction that more closely approximates the fluid manipulations that are possible with physical media, while retaining the many benefits of the digital form. Our interface is intended to support the quick creation of rough models, but not for precision modeling. We focus on the interaction issues, in particular on using the physical prop as a bridge between the physical and virtual worlds, and not on the geometric representations. We envisage our ideas can be used to drive a variety of different virtual clay formulations [10, 11, 31].

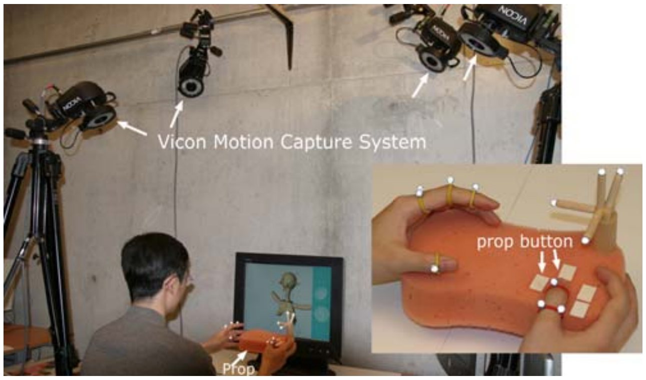

Figure 1. System setup. (inset): Fingers & prop with markers.

GRAPHITE 2006, November 29, 2006, Kuala Lumpur, Malaysia.
Copyright 2006 ACM 1-58113-000-0/00/0004…$5.00.

2. RELATED WORK

The SKETCH system [43] enables rapid sketching of geometric 3D scenes and Teddy [21] models organic shapes by drawing 2D silhouettes. Chateau [20] extends the sketching paradigm by providing suggestions to the user as to possible next steps. Wang and Kaufman [40] combine 2D sketches with image processing, suggestions, and user input for 3D modeling. These all attempt to leverage artists' skills in drawing sketches of models, rather than directly creating the models in a manner similar to using clay.

VR researchers have explored direct manipulation for 3D modeling. HoloSketch [8] uses a 3D mouse/wand, Nishino et al. [29] use two-handed gestures and Schkolne et al. [36] use repeated marking to construct 3D objects. Interaction in these systems are performed in free-space, without physical aids such as our prop, to guide the hands.

Computer vision techniques have been used to achieve gesture based manipulation. Pavlovic et al. [30] provide a good survey on the visual interpretation of hand gestures. Research in this area relevant to our work focuses on the relatively difficult task of using vision techniques to track freehand gestures. For example, Ringel et al. [33] integrate vision tracking of hand poses with a touch sensitive large display to enable direct hands-on interaction that goes beyond single point input. Freeman and Weissman [14] describe a system for controlling a television using freehand gestures, while Segen and Kumar [38] use vision tracking of a set of hand gestures for spatial interaction. Gloves that detect hand movement can also be used for gestures. Our use of small passive markers is a refinement of the glove approach and perhaps less intrusive than wearing gloves.

Physical tools have been explored in many interfaces. The tools are either passive, i.e. with no electronics inside and the gestures are inferred [6]; or the tool itself contains some sensors that enable it to get spatial and temporal parameters directly. Grossman et al. [16] describe using a digital spline to manipulate curves directly, while the Twister system [25] uses 6-dof trackers in each hand to deform 3D shapes. Fitzmaurice et al. [13] use physical objects as handles to virtual 2D widgets, while Ishii and colleagues (e.g. [22, 32]) have extended this concept to a research agenda in tangible interfaces. Hinckley et al. [19] use physical props to aid surgical planning. The prop in our prototype serves a similar function, albeit to support different interaction. Anderson et al. [1] describe a hybrid approach to 3D modeling combining tangible interfaces, traditional media and 3D scanners to help bridge the gap between physical and virtual realms. We do not create models by scanning physical media but propose using a deformable physical prop as a proxy to sculpt virtual models.

Grossman et al. [17] explore multi-finger gestures to select and transform objects within a 3D volumetric display. While they use the same motion capture setup, their interactions focus on controlling rigid objects in a 3D display space. Our work focuses on a new paradigm for conceptual 3D modeling using deformable physical proxies on a regular 2D display, resulting in a distinct vocabulary of gestures and finger manipulation metaphors.

Rossignac et al. [34] describe ideas for merging physical and digital media, where their "clay" sheet has the ability to alter its shape and to exert a resistive force upon the user's hands. Wang and Kaufman [41] computationally simulate the familiar process of clay sculpting for creating roughcasts. Ferley et al. [12] describe quick shape modeling using a sculpting metaphor with a virtual tool. Ix et al. [23] demonstrate sculpting of surfaces using a haptic device, while McDonnell et al. [26] describe a suite of haptic tools for realtime virtual sculpting using a clay metaphor. Dewaele & Cani [10, 11] describe compelling mathematical representations for simulating virtual clay and indicate that their work is a "first step towards a very challenging long-term goal: enabling an artist to use his hands for modeling virtual clay as he would a real material. Of course, an appropriate interface is needed to capture hands and fingers motion" [11]. Our work is a step towards such an interface.

Geometry processing algorithms, such as Laplacian editing[28], that facilitate high-level object deformation is the subject of active research in graphics. These approaches focus on making specific mathematical representations more usable. We instead take a user-centric approach that avoids any user knowledge of the underlying geometric representation. Our designs are not restricted by the relative merits of geometric representations and are amenable to benefiting from innovations in geometry.

In summary, our literature review reveals significant effort in geometry processing, new interfaces for virtual sculpting, and attempts to utilize more direct finger and hand input in various interface tasks. There has been little research, however, on the focus of our work: using direct finger/hand input with deformable physical props for 3D virtual sculpting. We are, however, inspired by tangible user interface research in our use of a physical prop as a proxy to the virtual world and a reference frame for the fingers.

3. SYSTEM DESCRIPTION

3.1 Intended Workflow

Our goal is to create an interface for quick sculpting of rough virtual models of the sort shown in Figure 2, using the indicated operations in a fluid manner that approximates and extends the possibilities afforded by real clay.

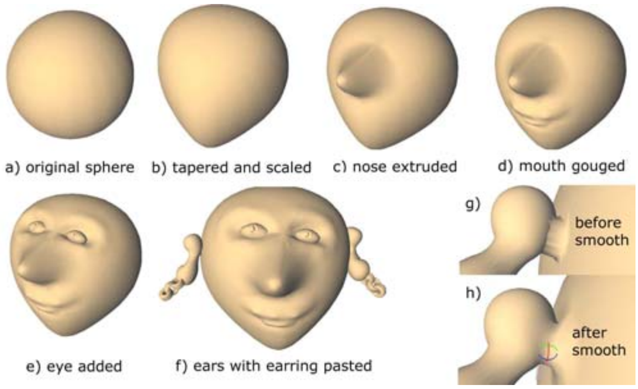

Figure 2. Model sculpted with techniques in our system

3.2 Tracking System

A Vicon camera-based motion tracking system (www.vicon.com) provides real-time 3D positions of passive markers placed on the user's fingers and a physical prop. Our prototype uses just four cameras for tracking, as illustrated in Figure 1. We note that while current tracking technology require markers for robust tracking, it is probable that advances in computer vision will obviate the need for markers in a few years. Using markers only marginally detracts from the overall usability of our prototype, but allows us to explore advanced freehand interaction techniques of the future with current technology.

3.3 Multi-Point Finger Input

All interactions in our system are accomplished by tracking the user's fingers and a physical prop. The thumb and index finger of the user's dominant hand are the primary input channel for manipulative operations and commands, drawing from observations by Molteni [27]: “Besides the palm of the hand, most of the time you will be using only two fingers when working with clay: the thumb and the index finger.” We believe that this choice should enable the user to find the interaction process similar to the existing physical experience and thus be able to focus primarily on their creative activity. The non-dominant hand is used to hold and move the prop, and its thumb is tracked for additional command input and parameter adjustment. Tracking both hands enable the exploration of two-handed operations, as commonly observed in real-world clay modeling. Although we could have easily tracked more fingers on both hands using the Vicon system, we believe that it could complicate the interaction vocabulary. At this stage of our research we feel it prudent to constrain the design space to only three fingers on two hands.

The index finger and the thumb are independent but also highly correlated. Thus, they can each be assigned different independent simple tasks such as pointing and tapping, or can work together to accomplish more complex tasks such as pinching and twisting. The input bandwidth is not only increased as compared to the single point 2D or 3D trackers that have been used in other interesting prototype interfaces for conceptual modeling [1, 5, 12, 15, 21, 25, 29, 31, 35, 39, 43], but the inherent skeletal and kinesthetic relationship between the thumb and index finger can be exploited to create complex yet facile interactions. We also explore the use of whole finger in addition to fingertip interaction.

3.4 Virtual Model Representation

We use polygon meshes as the underlying representation of the virtual models in our system. The mesh is automatically adaptively subdivided in real-time to provide adequate resolution during sculpting operations. Mesh manipulation is implemented using free form deformations (FFD) where the displacement of mesh vertices is computed according to a set of control points [37]. Other more sophisticated representations could be used instead [10, 11, 31]. We deliberately chose to use a relatively simple representation and efficient linear time (in the number of mesh vertices) algorithms to demonstrate our interaction ideas.

3.5 Physical Prop as Proxy to Virtual Model

A sponge prop acts as a physical proxy to the virtual models being created (Figure 1 inset), bridging the gap between the physical and virtual worlds. Using the prop as a passive frame of reference for the manipulating fingers, users can specify spatial relationships, locate regions of interest, and carry out manipulations on the virtual model in a fairly direct manner thus reducing the need for the multitude of complex virtual widgets that are common in status-quo applications. As Hinckley et al. [19] discuss, the use of physical props as proxies can exploit users’ existing skills, provide direct action-task correspondence, facilitate bimanual interaction, and provide tactile and kinaesthetic feedback to the user. The visceral kinaesthetic and tactile feedback obtained by directly grasping the prop is likely as important to clay modelers as it is to the surgeons in Hinckley et al.’s study. We choose to use a sponge as the material for the prop because it provides good tactile feedback and enables the user to have an instant idea of the extent of their manipulations, particularly when deforming the virtual model. Also, the sponge prop returns to its original shape after each manipulation. Thus, each manipulation is independent of one another in the physical realm, enabling virtual operations like undo and redo to be applied without concern for the state of the prop. It is important to note, however, that the physical properties of sponge and clay differ in several ways, including in elasticity and plasticity. This might cause false expectations or difficulty in understanding the interactions. However, we believe that the gradually incremental interaction style and appropriate visual feedback used throughout our system can minimize this negative effect, if any.

3.5.1 PropWidget

In the standard WIMP interface, the cursor serves as an abstract representation of mouse movements, acting as an intermediary between the physical mouse and the virtual world. Building upon foundational work by Conner et al. [7] and others [16], we have designed a similar intermediary called the *PropWidget* that enables abstract mappings between the physical prop and virtual model to be easily supported. Displayed as a wireframe cuboid, the PropWidget moves in accordance with the movement of the physical prop, and is used to manipulate virtual objects, much like a cursor is manipulated by a mouse and used to manipulate GUI widgets. Scaling the PropWidget’s size up/down is accomplished by pressing the respective two right buttons on the prop. Changing the PropWidget’s scale implicitly alters the control-display gain between the fingers/prop and the virtual object, enabling manipulation at different granularities in a continuous range from coarse to fine as desired. Pressing the left prop button toggles a lock on the PropWidget’s position, allowing for more stable manipulations such as fine sculpting when required. Our prop and corresponding PropWidget have a regular cubic shape that does not change over time except when being manipulated, while the virtual model can take on any shape. Thus, apart from the most trivial geometric models, there typically isn’t a simple 1-1 shape correspondence between the virtual model and the prop or PropWidget. To effectively use the PropWidget to manipulate models, we have developed two mapping techniques.

3.5.2 Global Nonlinear Mapping

Here, the PropWidget is mapped to the portion of the virtual model’s mesh encompassed by the PropWidget in a nonlinear way. First, the original correspondence between the PropWidget and mesh is recorded. By keeping track of subsequent deformations of the mesh and/or the PropWidget, we maintain a relative correspondence between the two. For example, consider the case where the mesh is a cube, just like the PropWidget. The mapping from a point p on the PropWidget to the point p' on the mesh is easily computed as the relative difference between the two. We note this mapping as $p' = F(p)$. After the mesh is deformed, every point p' becomes $p'' = G(p') = G(F(p))$ where G is the deformation function applied. If G,F are reversible, then $p = F^{-1} (G^{-1}(p''))$, as illustrated in Figure 3a. This nonlinear global mapping is a parameterization of the virtual model’s mesh, where the original coordinate is the parameter of each element and the mapping is done by computing the parameters. This allows for easy location of corresponding virtual elements after deformation. For example, if moving the fingers from points p to g on the prop originally maps to p' and g' respectively on the mesh, it will be mapped to p'' and g'' on the deformed mesh. This is useful for various global deformation techniques to be discussed shortly.

3.5.3 *Local Relative Mapping*

Although the global nonlinear mapping provides some flexibility in dealing with deformed meshes, it is not sufficient when the mesh undergoes several arbitrary manipulations or when the user wishes to manipulate just a small part of the mesh. Our local relative mapping technique solves this by using the PropWidget as a frame of reference for the fingers which act upon portions of the mesh that are within reach of the PropWidget's zone of influence. The global coordinates of the user's fingers in the physical world, GP_{finger}, as captured by the Vicon system are converted to local coordinates relative to the physical prop, denoted by LP_{finger}. The finger's global coordinates in virtual space, GV_{finger}, are computed as $GV_{finger} = GV_{PropWidget} * LP_{finger}$ where $GV_{PropWidget}$ are the local coordinates of the PropWidget as controlled by the physical prop. Thus, this mapping is compatible with established bimanual interaction where the non-dominant hand provides the frame of reference for dominant hand actions [18].

We investigated several algorithms for mapping the virtual finger coordinates GV_{finger} to desired points on the mesh. The first two approaches – nearest distance and wrapping – were unacceptable in certain conditions (Figure 3). The most successful, which we use throughout our prototype, was a variant on a ray cursor (Figure 3). A major advantage of this approach and our chosen eyepoint is that it is akin to the user sitting in the virtual world, looking down at the fingers and manipulating the object. Although the virtual eyepoint is predefined and fixed in our current prototype, we can imagine extending it to following the user's head movements if head tracking was implemented.

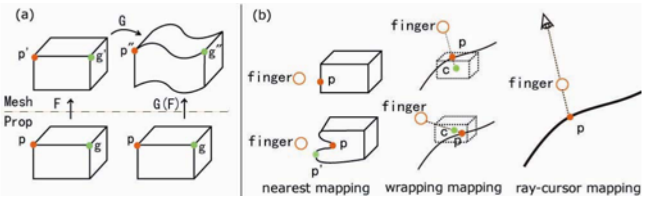

Figure 3. PropWidget mappings. (a) Global nonlinear mapping. On the left, point *p* on the PropWidget is mapped to *p'* on the virtual mesh by function *F*. On the right, after the virtual mesh was deformed by function *G, p'* changed to *p"*, and *p* would be mapped to *p"* by function *G(F).* (b) Local relative mapping. Nearest distance mapping: in upper figure, virtual finger is mapped to point *P* as intended, but in the lower figure the mesh topology results in an undesirable mapping. Wrapping mapping: finger is mapped to intersection point between mesh and line connecting GV_{finger} to centre *C* of PropWidget, analogous to wrapping the PropWidget around the mesh like plastic wrap. This works in some cases (upper figure) but fails when *C* is outside the mesh (lower figure). Ray cursor mapping computes the target position as the intersection between the mesh and the line connecting GV_{finger} and a predefined virtual eyepoint.

3.6 Inferred Actions and Gesture Set

The lack of any electronics on the hand or the prop implies that neither can actively communicate much information about its state to the computer. Rather, state information is indirectly inferred based on global finger positions and movements as well as local positioning relative to the prop. One could conceivably perform a huge set of gestural actions even with just a few fingers. However, unless users can easily understand and form a suitable mental model of the gestures, the interaction will become unwieldy. To address this issue, we limit the number of possible gestures to a small set (Figure 4), and provide appropriate visual feedback to aid comprehension. All gestures are designed to conform to the natural movement ranges of the thumb and index finger, thus avoiding awkward postures and movements. When more complex interface behavior is required than is afforded by this set of gestures, we use visual interface widgets operable by these gestures. We ensure consistency in the types of actions assigned to particular gestures, thus minimizing the amount of learning required of the user. We strive for a minimally modal interface, where the user can grab the prop and deform it directly, much like one would with clay. We also sense the relative pressure exerted (measured as displacement) by the fingers on the prop to enable a smooth transition along a continuum of different but related sculpting operations. Four button icons are also attached to the prop, activated by the non-dominant hand's thumb. The left button starts and completes operations, the middle one changes discrete parameters, and the right two enable up/down adjustment of continuous parameters. These are passive buttons, with actions inferred by recognizing the location and gesture of the thumb on the button. On-prop buttons enable modulating commands to be performed while the prop is being manipulated, and avoid the pausing problem inherent in freehand gestural interaction. A further three passive buttons operated by the dominant hand are placed on the tabletop where the prop is used.

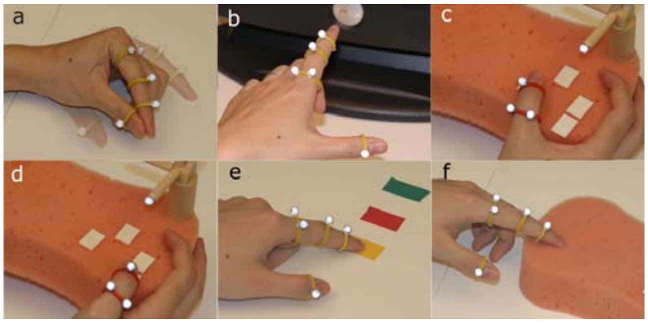

Figure 4. Gesture set. (a) clapping performed by briefly touching and releasing the tips of the dominant hand index finger and thumb; (b) pointing performed by opening the dominant hand index finger and thumb wide and pointing the index finger toward the screen; (c, d) clicking passive on-prop buttons by moving the non-dominant hand thumb onto the button; (c) clicking left button; (d) clicking lower right button; (e) clicking on-table buttons with the tip of the dominant hand index finger; (f) touching and/or pressing on the prop surface.

3.7 Basic Interactions

3.7.1 *Interface Layout and Command Input*

The system's basic interface layout consists of a storage panel on the right side that displays available basic mesh primitives and partially completed models, and a work area on the left where users manipulate virtual models (Figure 5). Models can be moved between the two areas via a simple finger point and click action. The most frequently used commands in our system are invoked directly using either simple gestures or with the tabletop and on-prop buttons. Since we deliberately kept the gesture set small (Figure 4) to maximize usability, less frequent commands are invoked from a marking menu (Figure 5) using a clapping gesture and a mark in the appropriate direction.

3.7.2 Translation, Rotation, Scaling

When global nonlinear mapping of PropWidget to virtual model is used, translation and rotation of the virtual mesh is achieved by appropriate movements of the prop in physical 3D space while pressing the prop's middle button. Scaling up/down is achieved by the non-dominant thumb pressing on the prop's two right buttons respectively. The buttons also act as clutches to allow for relative translation, rotation, and scaling. In addition to controlling scale factor, we use the right button for adjustment of several other continuous parameters in the system depending on context, thus maintaining consistency of actions throughout.

When local relative mapping is used, the PropWidget is not directly connected to the virtual mesh and as such cannot directly translate, rotate, and scale it. Instead, we use a arcball technique to achive 3D mesh rotation by 2D movements of the dominant hand's index finger on the tabletop, much like spinning a lazy-susan or a potter's wheel. This allows for smooth sequencing of events, such as rotating the mesh with the dominant hand while simultaneously positioning the PropWidget and then manipulating the mesh with the non-dominant hand.

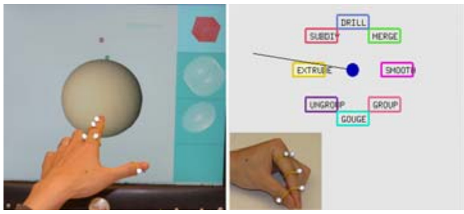

Figure 5. (a) Interface layout. Meshes moved between storage pane at right and working area by finger pointing and button clicking. (b) marking menu invoked by clapping gesture (inset)

3.8 Topology Invariant Deformations

This section focuses on two styles of interaction techniques for deforming the mesh while maintaining its topology. The first, global nonlinear deformations, is characterized by using the prop and PropWidget as the selector and deformation tool; the second, local deformations, uses the fingers as the selector and deformation tool while the PropWidget acts as the frame of reference to the fingers. In both styles, the physical prop plays a central role by bridging and mapping the physical world to the virtual one. Note that the techniques described here are those that we have implemented, but other more specific types of deformation techniques could easily be added using the same proxy based interaction framework.

3.8.1 Global Nonlinear Deformations

The system connects the relevant portion of the mesh to the PropWidget using the global nonlinear mapping discussed earlier. An operation zone that is slightly larger than the bounds of the physical prop is defined to delineate the working area for the fingers. When outside this operation zone, the dominant hand's thumb and index finger are displayed in the virtual space as two small wireframe spheres, colored red for the index finger and green for the thumb. Within the operation zone, the spheres are rendered solid to indicate the zone change, in addition to the physical kinesthetic feedback the user gets from simply being on the prop. As discussed earlier, we chose to use a deformable sponge prop because it provides good tactile feedback to direct grasping actions and allows the user to get an instant sense of the extent of their manipulations. The prop inherently affords actions such as grabbing, squeezing, twisting and bending, which we take advantage of in deformation techniques. We use the free-form deformation algorithm described by Sederberg and Parry [37] where the mesh vertices are displaced based on the deformed shape of a surrounding 3D grid of points. The middle button loops through the primary axes of the prop as the axis of manipulation. Aligning one of the primary prop axes with a desired deformation axis allows arbitrary axis deformation. Axes can also be specified as described later for object drilling.

Tapering and Flaring

By moving the thumb and index finger in opposite directions along an axis (Figure 6), the mesh can be tapered and flared. Moving them towards one another squeezes the physical prop and consequently tapers the virtual mesh. Moving them away from one another does not affect the physical prop, but results in flaring the mesh. We implement taper/flare by changing the distances of vertices to the axis proportional to the displacement of the fingers.

Twisting

Twisting is achieved by moving the thumb and index fingers around an axis, just like screwing/unscrewing the lid of a jar. We set the rotation angle of the mesh vertices to be proportional to their distance along the twist axis [4].

Bending

Bending is achieved by grabbing and moving a portion of the prop along the desired direction (Figure 6). In the current prototype we support bending along a centerline parallel to the y or z axes. The algorithms for bending and inverse transformation are as described by Barr [4], and consist of rotation and translation of the bent region.

Stretching

By grabbing and moving a portion of the prop along the x-axis, the mesh can be stretched. We implement stretching by changing the the inter-vertex distance along the stretch axis in proportion to the displacement of the fingers.

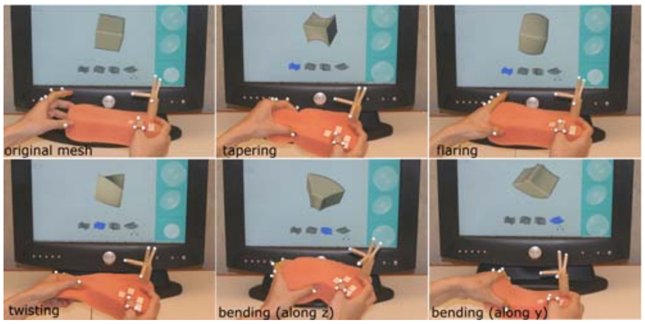

Figure 6. Global nonlinear deformations.

Combining the Deformations

After grabbing the prop, the user can taper, flare, twist, bend and stretch the mesh without explicitly selecting a specific transformation function. The intended transformation is recognized heuristically after an axis is selected. For example, with the x-axis selected, change of distance between the tips of the index finger and thumb indicates tapering/flaring, rotation of the grabbing hand along the x- axis indicates twisting, rotation along the y or z axis indicates bending along the respective axis, while movement along the x-axis implies stretching. As a feedback aid to the user, four icons are shown below the mesh to indicate the four possible transformations, with the active one highlighted. Pressing the left prop button with the non-dominant thumb terminates the deformation. By using the left prop button to lock/unlock the PropWidget and repositioning the prop, successive relative deformation actions can be accomplished.

We note that traditional 3D modeling interfaces also provide similar deformation functionality by typically requiring the deformations to be applied one at a time with an explicit mode/tool switch between actions. Our approach allows for seamless application of multiple deformations in a smooth sequence. Such interactions are not only more efficient, but may also facilitate more creative manipulations of the virtual objects in a manner resembling working with real clay. Note that while our techniques retain a strong analogy to real clay manipulations, some of them, such as flaring and stretching, go beyond what is possible in the physical world. This is one example where the digital realm enhances the interaction vocabulary and sophistication of the resulting geometric model.

3.8.2 Local Deformations

We support several local deformation techniques, including smoothing, gouging, and extrusion. Our techniques all draw inspiration from traditional clay modeling, but in many cases extend the physical world techniques to leverage the additional capabilities of the virtual form. All our techniques work by using the local relative mapping between PropWidget and mesh. The desired region of the mesh can be quickly located by positioning the PropWidget appropriately using the non-dominant hand, while the dominant hand's index finger and/or thumb are used to apply the desired deformation operation.

Fingertip Smoothing

In real clay modeling, artists often refine the surface texture or seams of their models by rubbing their fingers back and forth along the surface till the desired smoothness is achieved. This allows for gradual refinement, modulated by the pressure of the fingers on the surface as well as the number of back-and-forth rubbing actions. We support a similar local smoothing operation (Figure 7). After selecting "smooth" from the marking menu, the tip of the dominant hand's index finger is represented virtually as a wireframe ball. We use a wireframe representation in order to reduce occlusion of the underlying mesh being smoothed. Once the desired portion of the mesh is located by positioning the PropWidget, the dominant hand's index finger is moved back and forth along the surface of the prop to smooth the relevant mesh surface. The wireframe ball representing the fingertip is mapped to the mesh using the ray cursor algorithm (Figure 3). When the finger leaves the operation zone around the PropWidget, smoothing terminates; reentering the zone resumes the smoothing operation. The user can rotate the mesh by touching and moving the index finger on the table at any time. A clapping gesture returns the system to the initial state.

This smoothing interaction has several nice properties: the local relative mapping makes it easy for the user to locate the desired area; the tactile feedback from directly touching the prop and real time update of the virtual mesh provide the user with a sense of the effects of their manipulations; and two handed interaction is leveraged with the non-dominant hand holding the prop as a frame of reference within which the dominant hand operates [18].

Our implementation uses conventional Laplacian smoothing, developed from a two-dimensional heat equation. A displacement vector $D(P)$ is defined [24] by:

$$D(P) = \frac{1}{n} \sum_{i \in N_1(P)} Q_i - P$$

Where P is a mesh vertex, and $N_1(P) = \{Q_0, Q_1, \ldots, Q_{n-1}\}$ is the 1-ring of mesh vertices neighbouring on P. Then the mesh vertex is updated with $P_{\text{new}} = P_{\text{old}} + \lambda D(P)$ where $D(P)$ is the displacement vector and λ is a step-size parameter. More sophisticated operators such as Mean Curvature Flow [9] or median filters [42] can be used as well.

Whole Finger Smoothing

Artists working with clay use not only fingertips, but also their whole fingers and sometimes even the whole hand to achieve uniform effects over a large area. Our prototype does not support whole hand manipulations due to the lack of markers on multiple fingers, but the whole index finger can be used for smoothing (Figure 7). When the finger is parallel to the prop surface, we enable smoothing over a large area and around corners and edges with fluid coordinated finger movements. We seamlessly transition between whole finger and fingertip smoothing based on the finger's angle relative to the prop's surface.

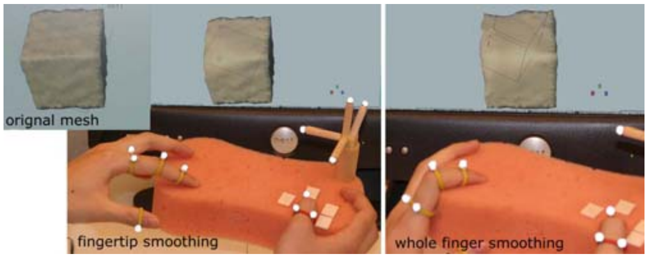

Figure 7. Smoothing.

Gouging

Another commonly used technique in clay modeling is when the artist uses fingers or other tools to sculpt desired indentations into clay. In our system, the thumb and index finger of the dominant hand can both be used for gouging operations. Once "gouge" is selected from the marking menu, two spheres are displayed on the screen, a red one representing the tip of the index finger and a green one for the thumb. In addition, a widget displays the gouging weight and direction (Figure 8). This widget takes the form of a Gaussian distribution curve, which peaks in the centre and diminishes smoothly on both sides. The height of the curve indicates the gouging weight, the longer the heavier; while the distance between the two endpoints of the curve indicates the gouge diameter. If the curve is below the horizontal axis (i.e., negative), gouging operations will make dents into the mesh while a positive curve results in bumps being formed on the mesh. The two right prop buttons are used to move the curve's central point up/down which in turns reduces or increases the distance between the curve's endpoints respectively. Thus, gouging weight and diameter are coupled. This adjustment of gouging parameters can be done in real time while the gouging. The amount of pressure applied (i.e., displacement) by the finger on the prop surface controls the amount of gouging/bumping. Analogous to real clay modeling, if the finger lightly scratches the prop surface, a sandpaper style smoothing effect occurs, while harder presses really gouge or bump the mesh. Using both the thumb and index simultaneously also allows for pinching of the mesh, which cannot easily be achieved with a single point of input. The whole finger can also be used for gouging over a larger area. The implementation is as follows. Given a vertex P on the mesh, we compute the displacement amount $D(P) = \lambda G(Dist(P,C)$ where C is the projected position of the virtual finger on the mesh surface, $Dist(P,C)$ is the Euclidean distance between P and C, and G is a Gaussian displacement function based on the input distance:

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/(2\sigma^2)}$$

where λ is a weight factor: $\lambda > 0$ when the curve widget is positive, $\lambda < 0$ when negative. The displacement direction is along the vertex normal, which is the average of the neighbouring faces. Pressure plays an important role in this process: the harder one presses on the prop, the bigger $|\lambda|$ and the more the object surface will be pushed in (or bumped out). As seen in Figure 8 we support approximate local volume preservation as well as elastic and plastic material properties in our system. We note that precisely simulating real clay's physical properties [10] could further improve the system's realism.

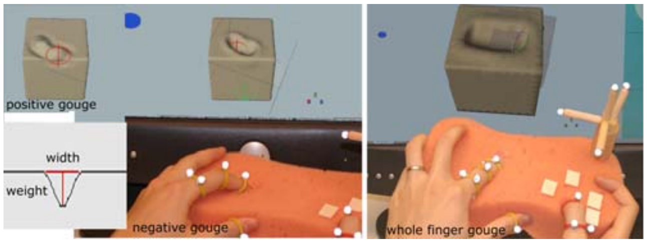

Figure 8. Gouging. Feedback widget is inset.

Extrusion

The thumb and index finger working together can form a powerful compound tool for pinching and pulling clay. The distance between the thumb and index finger tips control the diameter of a selection sphere that defines the region of the model to be extruded, while the finger positions control the locality of extrusion. By dragging the selection sphere away from the surface of the mesh, the mesh is extruded along the direction of movement. We provide two types of extrusion (Figure 9). The first is constrained extrusion which occurs along a straight line from the selected region on the mesh to tips of the moving fingers. The diameter of the free end of the extrusion can be rotated and scaled. When close to a major axis, the extrusion snaps to it. The second is unconstrained, where the fingers sweep out a extrusion path controlling translation, rotation, and scaling in one continuous motion. We iteratively add new mesh segments as needed to maintain resolution. Subdivision of the mesh is used to smooth boundaries between the mesh and the extrusion. Sweepers [3] and swirling sweepers [2] may be a better constant volume representation; however, our interaction style would still apply.

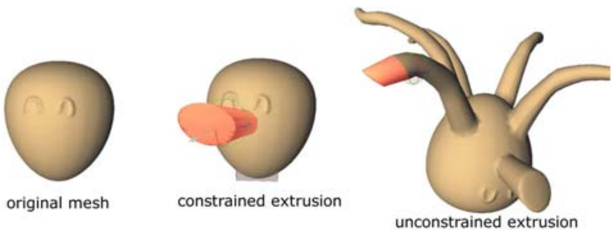

Figure 9. Extrusion.

3.9 Topology Varying Deformations

The topology invariant deformations described in the previous section provide much of the functionality and corresponding aesthetic quality that is inherent in clay modeling. However, interesting 3D models also often involve deformations that change their topology. We support four topology varying deformations: drilling, pasting, grouping, and cutting (Figure 10).

Drilling

Once "drill" is selected from the marking menu, the index finger acts as the "drill bit" and controls its position and orientation. The drill bit can be scaled in real time using the two right prop buttons. Touching the prop with the index finger results in a corresponding hole being cut in the mesh.

Grouping & Pasting

To create complex models, multiple meshes can be loaded, scaled, positioned, grouped and ungrouped as desired. While grouping maintains relative spatial relationships between the component meshes, pasting weaves the component meshes into a single whole. One can think of grouping as sticking two pieces of metal with a drop of glue, while pasting is more like welding them with the welding material filling in around the seams.

Cutting

Here we explore the use of physical tools in addition to the fingers. A physical knife tracked by the Vicon system is used to define, position, and orient a cutting plane relative to the prop and PropWidget. Touching the prop with the knife cuts the mesh along the defined cutting plane.

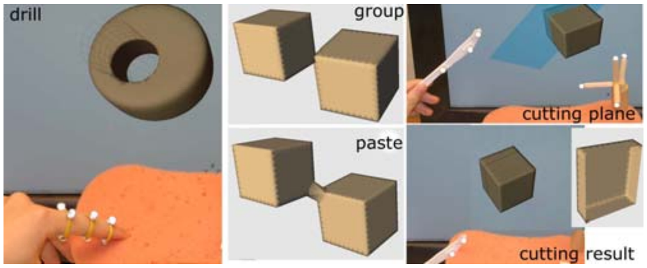

Figure 10. Topology varying deformations

4. USER FEEDBACK and CONCLUSIONS

Our work has explored a range of interaction techniques for virtual 3D sculpting using direct finger manipulation with a physical prop that acts as a proxy to the virtual model, and a reference for the fingers. We had a professional sculptor and digital modeler use our system, in addition to our own experience, with the following major observations:

Expressivity: The abstraction of the underlying mesh from the interface allows users to work freely. The modeless transition from smoothing to gouging and the combination of various deformations was found to be appealing. The relative mapping of the proxy made it attractive for quickly making incremental changes to existing models.

Learning curve: The gestures and interactions were easily understood and used, particularly when the set of such actions was reasonably small. The modeling metaphor resounded with the sculptor who had tried unsuccessfully to use CAD modelers.

Fatigue: Our system potentially suffers from the same issues of user fatigue as other input devices and when modeling with real clay. We found that users often rest their elbows on the table to alleviate fatigue. There is also nothing preventing the user from resting the prop on the table for the local modeling operations.

Given that designers and artists are continually seeking increasingly facile ways to bridge the gap between physical and digital media in order to effectively turn their mental concepts into reality, it is critical that we continue to explore alternative input and interaction modalities that exploit inherent human capabilities. The work presented here is one step towards more specialized interfaces that move beyond the current reliance on standard techniques that do permit many tasks to be accomplished but not necessarily in the most optimal or pleasing manner

5. VIDEO

A video demonstrating this system can be downloaded from www.dgp.toronto.edu/~ravin/videos/graphite2006_proxy.mov

6. REFERENCES

1. Anderson, D., Yedidia, J., Frankel, J., Marks, J., Agarwala, A., Beardsley, P., Hodgins, J., Leigh, D., Ryall, K., & Sullivan, E. (2000). Tangible interaction + graphical interpretation: a new approach to 3D modeling. *SIGGRAPH*. p. 393-402.
2. Angelidis, A., Cani, M.-P., Wyvill, G., & King, S. (2004). Swirling-Sweepers: Constant-volume modeling. *Pacific Graphics*. p. 10-15.
3. Angelidis, A., Wyvill, G., & Cani, M.-P. (2004). Sweepers: Swept user-defined tools for modeling by deformation. *Shape Modeling International*. p. 63-73.
4. Barr, A. (1984). Global and local deformations of solid primitives. *Computer Graphics*, *18(3)*. p. 21-30.
5. Bourguignon, D., Chaine, R., Cani, M.-P., & Drettakis, G. (2004). Relief: A modeling by drawing tool. *Eurographics Sketch-based Interfaces and Modeling*. p. 151-160.
6. Cao, X. & Balakrishnan, R. (2003). VisionWand: interaction techniques for large displays using a passive wand tracked in 3D. *UIST*. p. 173-182.
7. Conner, B., Snibbe, S.S., Herndon, K.P., Robbins, D., Zeleznik, R., and Dam, A.v. (1992). Three dimensional widgets. *Computer Graphics*, *22(4)*. p. 121-129.
8. Deering, M. (1995). HoloSketch: a virtual reality sketching/animation tool. *ACM TOCHI, 2(3)*. p. 220-238.
9. Desbrun, M., Meyer, M., Schroder, P., & Barr, A. (1999). Implicit fairing of irregular meshes using diffusion and curvature flow. *SIGGRAPH*. p. 317-324.
10. Dewaele, G. & Cani, M.-P. (2004). Interactive global and local deformations for virtual clay. *Graphical Models*,*66*.p.352-269.
11. Dewaele, G. & Cani, M.-P. (2004). Virtual clay for direct hand manipulation. *Eurographics (Short Papers)*.
12. Ferley, E., Cani, M., & Gascuel, J. (2000). Practical volumetric scuplting. *The Visual Computer*, *16*. p. 469-480.
13. Fitzmaurice, G., Ishii, H., & Buxton, W. (1995). Bricks: Laying the foundations for graspable user interfaces. *CHI*. p. 442-449.
14. Freeman, W. & Weissman, C. (1995). Television control by hand gestures. *International Workshop on Automatic Face and Gesture Recognition*. p. 179-183.
15. Galyean, T. & Hughes, J. (1991). Sculpting: An interactive volume modeling technique. *SIGGRAPH*. p. 267-274.
16. Grossman, T., Balakrishnan, R., & Singh, K. (2003). An interface for creating and manipulating curves using a high degree-of-freedom input device. *CHI*. p. 185-192.
17. Grossman, T., Wigdor, D., & Balakrishnan, R. (2004). Multi finger gestural interaction with 3D volumetric displays. *UIST*. p. 61-70.
18. Guiard, Y. (1987). Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, *19(4)*. p. 486-517.
19. Hinckley, K., Pausch, R., Goble, J., & Kassell, N. (1994). Passive real-world interface props for neurosurgical visualization. *CHI*. p. 452-458.
20. Igarashi, T. & Hughes, J. (2001). A suggestive interface for 3D drawing. *UIST*. p. 173-181.
21. Igarashi, T., Matsuoka, S., & Tanaka, H. (1999). Teddy: a sketching interface for 3D freeform design. *SIGGRAPH*. p. 409-416.
22. Ishii, H. & Ullmer, B. (1997). Tangible bits: towards seamless interfaces between people, bits and atoms. *CHI*. p. 234-241.
23. Ix, F., El-Sana, J., Qin, H., & Kaufman, A. (1999). Haptic sculpting of dynamic surfaces. *ACM Symposium on Interactive 3D Graphics*. p. 103-110.
24. Kobbelt, L., Campagna, S., Vorsatz, J., & Seidel, H. (1998). Interactive multiresolution modeling on arbitrary meshes. *SIGGRAPH*. p. 105-114.
25. Llamas, I., Kim, B., Gargus, J., Rossignac, J., & Shaw, C.D. (2003). Twister: a space-warp operator for the two-handed editing of 3D shapes. *SIGGRAPH*. p. 663-668.
26. McDonnell, K., Qin, H., & Wlodarczyk, R. (2001). Virtual clay: A real-time sculpting system with haptic toolkits. *ACM Symposium on Interactive 3D Graphics*. p. 179-190.
27. Molteni, M. (1992). The clay modeling handbook: Learning from the masters. NY: Clarkson Potter Publishers.
28. Nealen, A., Sorkine, O., Alexa, M., & Cohen-Or, D. (2005). A sketch-based interface for detail-preserving mesh editing. *ACM Transactions on Graphics*, *24(3)*. p. 1142-1147.
29. Nishino, H., Utsumiya, K., & Korida, K. (1998). 3D object modeling using spatial and pictographic gestures. *ACM Symp. on Virtual Reality Software and Technology*. p. 51-58.
30. Pavlovic, V., Sharma, R., & Huang, T. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans PAMI, 19(7)*. p. 677-695.
31. Perry, R. & Frisken, S. (2001). Kizamu: a system for sculpting digital characters. *SIGGRAPH*. p. 47-56.
32. Piper, B., Ratti, C., & Ishii, H. (2002). Illuminating clay: A 3D tangible interface for landscape analysis. *CHI*. p. 355-362.
33. Ringel, M., Berg, H., Jin, Y., & Winograd, T. (2001). Barehands: implement-free interaction with a wall-mounted display. *CHI Extended Abstracts*. p. 367-368.
34. Rossignac, J., Allen, M., Book, W., Glezer, A., Ebert-Uphoff, I., Shaw, C., Rosen, D., Askins, S., Bai, J., Paul Bosscher, Gargus, J., Kim, B., Llamas, I., Nguyen, A., Yuan, G., & Zhu, H. (2003). Finger sculpting with digital clay: 3D shape input and output through a computer-controlled real surface. *Shape Modeling International*. p. 229.
35. Sachs, E., Roberts, A., & Stoops, D. (1991). 3-draw: A tool for designing 3D shapes. *IEEE Computer Graphics and Applications*, *11(6)*. p. 18-26.
36. Schkolne, S., Pruett, M., & Schroeder, P. (2001). Surface drawing: Creating organic 3D shapes with the hand and tangible tools. *CHI*. p. 261-268.
37. Sederberg, T. & Parry, S. (1986). Free-form deformation of solid geometric models. *SIGGRAPH*. p. 151-160.
38. Segen, J. & Kumar, S. (1998). Gesture VR: Vision-based 3D hand interface for spatial interaction. *Multimedia*. p. 455-464.
39. Shaw, C. & Green, M. (1994). Two handed polygonal surface design. *UIST*. p. 205-212.
40. Tsang, S., Balakrishnan, R., Singh, K., & Ranjan, A. (2004). A suggestive interface for image guided 3D sketching. *CHI*. p. 591-598.
41. Wang, S. & Kaufman, A. (1995). Volume Sculpting. *ACM Symposium on Interactive 3D Graphics*. p. 151-156.
42. Yagou, H., Ohtake, Y., & Belyaev, A. (2002). Mesh smoothing via mean and median filtering applied to face normals. *IEEE Geometric Modeling and Processing Theory and Applications Conference*. p. 124-131.
43. Zeleznik, R.C., Herndon, K., & Hughes, J. (1996). SKETCH: An interface for sketching 3D scenes. *SIGGRAPH*. p. 163-170.