Cutting, Deforming and Painting of 3D meshes in a Two Handed Viso-haptic VR System

Adam Faeth[1], Michael Oren[2], Jonathan Sheller, Sean Godinez, Chris Harding[3]

Iowa State University

Abstract

We describe M4, the multi-modal mesh manipulation system, which aims to provide a more intuitive desktop interface for freeform manipulation of 3D meshes. The system combines interactive 3D graphics with haptic force feedback and provide several virtual tools for the manipulation of 3D objects represented by irregular triangle meshes. The current functionality includes mesh painting with pressure dependent brush size and paint preview, mesh cutting via drawing a poly-line on the model and two types of mesh deformations. We use two phantoms, either in a co-located haptic/3D-stereo setup or as a fish tank VR setup with a 3D flat panel. In our system, the second hand assists the manipulation of the object, either by "holding" the mesh or by affecting the manipulation directly. While the connection of 3D artists and designers to such a direct interaction system may be obvious, we are also investigating its potential benefits for landscape architects and other users of spatial geoscience data. Feedback from an upcoming user study will evaluate the benefits of this system and its tools for these different user groups.

CR Categories and Subject Descriptors: H.5.2 Haptic I/O; I.3.7 Three-Dimensional Graphics and Realism – Virtual Reality; I.3.4 Graphics Utilities – Graphic Editors; I.3.6 Methodology and Techniques - Interaction techniques

Additional Keywords: Haptics, H3D, 3D graphics, X3D, shaders, digital shapes, surface mesh, geometric modeling, deformation, cutting

1 Introduction

The creation and manipulation of 3D digital shapes (3D models) is at the heart of many 3D modeling applications, commercial (Maya, 3D Studio Max, Cinema 4D) and open source (Blender). While these applications support many sophisticated methods for creating and manipulating 3D models, performing freeform manipulations of the 3D models, such as painting, sculpting or cutting, they can be tedious using the typical mouse/keyboard setup. For example, the user may first navigate (rotate/translate) to a certain area of the model, perform a desired operation (deformation), change the view to check the result due to inadequate visual feedback during the operation, and possibly undo and repeat it because the result was not satisfactory. In other words, the traditional mouse/keyboard interactions tend to force the user to go through many inherently 2D mechanisms when trying to make a 3D computer model correspond to a mental 3D model.While expert 3D modelers may become proficient in this form of 3D-2D-3D translation, direct 3D interactions that are based on our real world experiences may be superior to translate from a 3D model in the user's mind to a 3D computer model. In everyday life, we often hold a tool in one hand and use a second hand that holds and repositions the object during manipulation, e.g., painting Easter eggs with a brush or carving a piece of wood with a knife. Following this tool-object metaphor, we have created the core of a small-scale, desktop VR system called M4, short for Multi-Modal Mesh Manipulation system (fig. 1) for the manipulation (or editing) of 3D shapes made from non-volumetric triangle meshes.

In the M4 system 3D stereo vision and force feedback provide the necessary perceptional cues about the 3D object, about the virtual tool and about the state of the manipulation currently being performed. Unlike surgical simulations, which aim to be a realistic reproduction of a specific case of reality, our system uses 3D graphics and haptic force feedback to create potentially novel, but not necessarily realistic, ways of interactions that could increase the user's ability to perform certain operations more intuitively and more efficiently. Currently, our efforts focus on exploring and evaluating the potential benefits for 3D artists and designers because their requirements are well established. Landscape architecture and structural geology are other domains that today routinely deal with visualizing 3D meshes, such as Digital Elevation Models (DEMs). However, existing applications to change the shape of these meshes for planning purposes are again solely based on mouse and keyboard and M4's tools can be adapted to evaluate their use in these domains as well. For example the paint tool could be used to select certain parts of a DEM and to then deform only those areas, or the cutting tool could be used to adjust fault planes within 3D subsurface models of geologic strata.

2 Related Work

There have been many efforts related to designing interfaces that integrate some form of force feedback (haptics) into the manipulation of 3D models – mostly in the context of art & design, CAD/CAM manufacturing, and surgical training.

Foskey et al. [1] and Gregory et al. [2] deal with painting and deformation (but not cutting) of 3D models made from arbitrary polygonal meshes. Baxter et al. [14] simulate haptic brushes for painting, and Johnson et al. [3] deal with painting textures on trimmed NURBS. Kim et. al [4] explore the editing of discrete, volumetric implicit surface representations; Dachille et. al [5] deal with deformations of B-Splines surfaces. Freeform [6,7], the only commercially developed system, and Cani et al. [8] use a virtual clay approach to deform volumetric representations of objects. Bendels et. al. [9] use a dual-hand combination of a phantom and hand gestures on a mirrored display. Keefe et al. [10] use haptics to support free handed 3D drawing via drag. The use of two hands for interactions and its benefits have been

[1] email: adfaeth@iastate.edu

[2] email: moren@iastate.edu

[3] email: charding@iastate.edu

IEEE Virtual Reality 2008
8-12 March, Reno, Nevada, USA
978-1-4244-1971-5/08/$25.00 ©2008 IEEE

investigated extensively e.g. Boeck et al. [16] and Hinckley et al. [17]. Anecdotal evidence [2,6] and user evaluations [10,16,17] suggest a substantial advantage for interactions via 3D haptic force feedback compared to the traditional 2D mouse/keyboard systems. In the area of surgical training, many efforts have centered around real-time interactions with 3D representations (tissue, organs), for cutting with a haptic virtual scalpel see e.g., [11,12].

3 METHODS

3.1 Hardware & software used

For haptic force feedback we use two SensAble Phantoms, which accept 6 degrees-of-freedom (DOF) input from the user via a stylus end effector and provide a 3 DOF output to the user via a point at the tip of the stylus (fig.2, right, showing a Desktop Phantom). We use these phantoms in two configurations: A) in a co-located active stereo setup, which, when viewed via the mirror (fig. 2 left) provides a 1-to-1 overlap of the haptic workspace and the 3D stereo image and B) as a fishtank VR setup with a passive stereo flat panel (Arisawa P240W) with the phantoms positioned to the left and right in front of the display (fig. 1). Both setups implement an egocentric view for the user's interaction with objects and advocate a tool-object metaphor that is set in a proprioceptive frame of reference [16]. A comparison of the effectiveness of these two setups for manipulating 3D meshes is part of the upcoming user study.

M4 is built with the open source H3D API created by SenseGraphics AB. H3D is a haptic extension of the X3D scene-graph API which renders a scene graphically and haptically – a scene's objects have graphical properties (e.g., color) and haptic properties (e.g., friction). H3D's force feedback is based on the haptic proxy model. Using C++, we extended several H3D nodes (such as the indexed triangle set class) to implement functionality for grabbing, mesh cutting, mesh deforming and mesh painting. Routes between the scene graph nodes and Python scripts, that can read/write to the nodes, allowed us to create several haptic tools and their complex, event-driven interactions with the scene graph's objects, including the combined interaction of two different tools with the mesh. A 3D, haptic graphical user interface, allows the user to select a tool for each hand and to fine-tune its functionality. This menu is circular to minimize the need for physical movement during interaction with it; it can be summoned to appear close to the current stylus tip and can be hidden again if not needed for the moment.

3.2 Tool-object metaphor for two handed interaction

Using the tool-object metaphor, the dominant hand (DH, simulated via one phantom) operates a manipulation tool on the 3D mesh, which is held by the non-dominant hand (NDH, simulated by the second phantom). This holding of the mesh is simulated by selecting the "grab tool" for the NDH, bringing the tool tip into contact with the mesh and holding down the button on the physical stylus. Any rotation and translation of the NDH is now applied to the mesh, be it to simply inspect the mesh or to adjust its position and orientation while the DH manipulates the mesh, in which case both of the hands receive force feedback via its phantom. It is also possible to assign a mass to the mesh and to add effects like weight and inertia. Although these effects add realism, adding a dampening or viscosity effect is actually more supportive of the fine motor manipulation tasks used here, just as a fluid-head tripod supports smooth panning of a video camera.

Two handed interaction is not limited to one hand holding the object and the other hand manipulating it. The NDH may also be assigned a tool for *dual-tool manipulation* of the object. There are several interesting dual-tool interactions possible with the M4 system. The use of two deformers allows the mesh to be stretched apart or folded. A deformer operated with a cutter supports an interaction similar to tearing, but with more direct control over the line of the tear. The painting tool, held in the NDH, may also be used to change the mesh's material properties (soft – hard) while the DH deforms these parts. These are just a sample of the potential dual-tool interactions, and part of the upcoming user study will evaluate the use of such two handed interactions. A directional light source is attached to each stylus, which use per-pixel lighting and greatly enhances the user's sense for surface details.

3.3 Mesh cutting via 3D path draping

Initially we based our cutting approach on a technique from the tissue cutting domain [18], which simulates a scalpel that immediately cuts the mesh at the point of contact. After some early feedback from users, M4's cutting operation was re-designed to first plan the a draped path on the mesh's surface, by planting a series of nodes with the phantom - planting a new node creates a new segment of the draped path. Most of the actual changes in the mesh's topology, i.e., the creation of new triangles, vertices and edges along this path occur only after the user is satisfied with this path (fig. 3).

This two-stage approach allows us to experiment with several novel approaches, such as the preview feature shown in fig. 4., this visualizes the drape-line between the last node and the current tip position via a semi-transparent rectangle that visually intersects the mesh and that follows the orientation of the stylus. This preview allows the user to rotate the stylus around the point on the mesh that is touched with the tip and immediately see the change in the drape path. Simply turning the stylus towards the viewer in fig. 4 will move the drape path towards the bottom of the hill – once the desired path is found, it can be integrated into the mesh.

A typical approach for the path planning stage is to hold the mesh with the NDH and to touch the mesh with the cutting tool held in the DH. Touching the mesh with the DH tool renders the surface as solid to the touch and allows the user to detect small scale features embedded in the mesh (such as a groove) which can help to pinpoint the right spot for the next node. Pressing the stylus button (with the DH) while in contact with the mesh plants a node at the point of contact and creates a new path segment. The rotation and translation afforded by holding the mesh in the NDH allows the user to seamlessly change visual context while the DH plans the path, e.g., to investigate potential target areas.

While this technique is useful to quickly drape long, straight line segments on the mesh, a variation allows the user to drag the cutting tool more slowly over the surface and have it plant the nodes of the path automatically small distances apart. Although this drag-draw does not provide the preview plane, it allows the user to capture small details of the mesh with a drape line by using the continuous force feedback and the visual feedback from changing the mesh's position and orientation with the NDH. Using both modes together, it is possible to combine small, detailed parts with large, straight parts, both modes allow the user to reposition or undo the previously planted nodes.

Besides using the grab tool to hold the mesh, the NDH can also switch to a deform tool, allowing it to alter the mesh geometry directly while the DH drapes the path over this part. While this ability is novel and potentially very powerful it currently requires a good deal of manual dexterity – results from the user study are needed to provide a context for the interaction.

Once the user is satisfied with the drape path, two forms of topological change can be performed by the user - either creating an incision, along the path (fig. 3) that can be widened later (fig. 5) or creating a hole directly by implicitly connecting the first and last node and thus deleting the outline of a polygon. This change in topology is very fast, even for large meshes, because the drape phase performs part of the integration of the line into the mesh before the user decides to make the cut.

3.4 Chainmail deformation tool

H3D offers a rudimentary deformer (called Gaussian) that displaces the vertices of a mesh that lie within a radius of the stylus tip. The deformer fits the displaced vertices to a bell shaped surface that resembles a Gaussian surface with the apex at the contact point of the stylus. We enhanced the deformer by slaving the tip to the deforming mesh geometry with a very strong haptic attraction effect. This allows us to not only deform the mesh via push, but also via pull and to move the tip laterally during the deformation, which works very well for creating linear features such as troughs and crests. However, since the Gaussian deformer moves vertices without regard to the topology of the mesh, we adapted a generalized version of the Chainmail deformation algorithm [12, 13] (fig. 5).

The Chainmail deformation is specifically designed to operate at interactive rates by loosely approximating a cloth-like material. Touching the mesh and holding down the stylus button activates the Chainmail deformer tool; this selects the mesh vertex closest to the initial point of contact and attaches it directly to the tip. Moving the tip will move this seed vertex, which will then possibly move its neighbors; if a vertex does move, each of its neighbors may also move (each vertex is only moved once). The decision to move a vertex is based on the accumulated movements of its neighbors and by set material properties (stretch, compression, and shear) stored for each vertex. The deformation floods outwards from the seed vertex and stops dependent both on the displacement of the seed vertex from its original position, and on the mesh's material parameters. For example, the stretch value lets a vertex move away from its neighbor without the neighbor following, high stretch values will therefore mimic a rubbery surface; another combination of the parameters inhibits the vertex's movement and, in effect, stiffens the mesh. This deformation calculation is performed for each frame, until the stylus button is released. The deformation is independent of the object movement and orientation, so using the NDH to move or reorient the mesh while the Chainmail tool deforms does not move the stylus of the DH. This allows the user to change the viewing angle while deforming and, e.g., judge if the current deformation is too high by looking at it from a different side and adjust the deformation accordingly.

During the deformation, a force vector is calculated from the number of displaced vertices, modified by the mesh's material parameters and sent to the phantom in the DH. This force model increases the resistance felt by the user as more and more of the mesh is displaced. Each vertex carries its own set of parameters,, this allows us to experiment with several interesting possibilities, such as loading the mesh's material parameters in the form of material textures, in which the vertex's material parameter corresponds to the value of its texel. This allows some parts of the mesh to be rigid and other parts to be extremely flexible to deformation. Furthermore, the paint tool has the ability to paint into a material texture, and this causes a change in the material properties for vertices mapped to the altered texels. The NDH can use the paint tool to affect the material properties of an area while deforming with the DH, effectively melting rigid areas or stiffening flexible areas of the mesh during deformation. Again, this requires manual dexterity and its effectiveness needs to be investigated closer as part of the upcoming user study.

3.5 Paint tool

H3D allows the use of specific shader nodes that connect other parts of the scenegraph (for example the current position of the haptic interface point) with GLSL shader programs. We use shaders, which are integrated into H3D scenegraph nodes to implement per-pixel lighting (phong fragment shader) for a directional light source and for the paint tool. This shader works by rendering to an off-screen "paint" texture to accumulate the paint in a separate paint texture. This texture can refer to a conventional, visual, texture but also to a texture containing the mesh's material properties. For the paint tool (fig. 6), each fragment's color accumulates based on its proximity to the position of the tip - fragments that are closer to the brush's tip receive more paint per render pass than those further away, which leads to feathered (blended) edges. The brush size, modeled as the radius of a sphere around the tip, changes with the force the user applies with the phantom into the mesh. The upcoming user study will also investigate the preferred function to map force to brush size; we intend to test linear, exponential, and logarithmic mappings between force and brush size. The painting tool offers a preview function, also implemented using a shader, that works when the mesh is touched but the stylus button is not pressed. The preview indicates which part of the mesh would be filled if the user were to press the button.

4 Conclusions and future work

The M4 system represents an initial framework for a dual-handed mesh manipulation system within a personal scale stereo VR setting. The combination of dual-handed force feedback and stereo-vision provides us with many interesting and potentially novel ways of interacting with triangle meshes. We highlighted many of these interactions in this sketch but many possibilities are left for future investigations. An important next step is to provide the ability to fill-in holes and incisions made by cutting and deforming and to upgrade the Chainmail deformer to operate with a set of seed points, either from a poly-line or from a "painted" area on the mesh, instead of just one seed point.

Our initial experience with the current system suggests that it would be functionally far superior to other freeform-type manipulations currently offered in conventional 3D modeling system. A formal usability study is planned, which will provide more specific answers about the potential advantages on the aforementioned two-handed manipulations. Although the system is geared toward freeform manipulation of 3D shapes in and Art & Design context, we see potential for its adaptation for specific parts of the geosciences which also deal with the deformation of meshes, such as the "clay models" of landscapes described in [19].

References

[1] M. Foskey, Otaduy, M. A., and Lin, M. C., ArtNova: Touch-enabled 3D model design. Proc. of IEEE Virtual Reality Conference. 2002.

[2] A. Gregory, S. Ehmann, and M. C. Lin., "inTouch: Interactive multiresolution modeling and 3D painting with a haptic interface". Proceedings of IEEE VR Conference. 2000.

[3] D. Johnson, T. V. Thompson II, M. Kaplan, D. Nelson, and E. Cohen, Painting textures with a haptic interface. Proc. of IEEE Virtual Reality Conference. 1999.

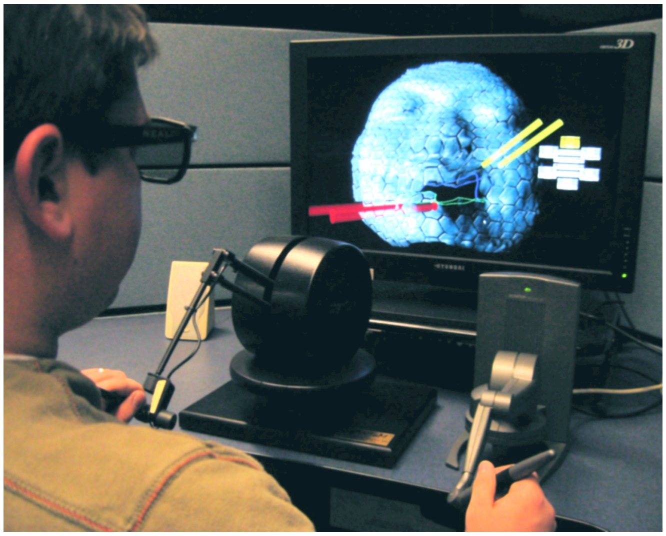

Fig. 1: The M4 system is used to directly manipulate a textured, irregular triangle mesh with two haptic devices.

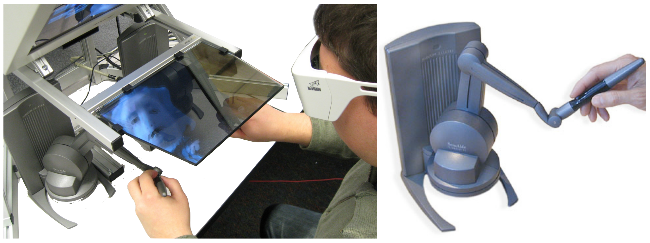

Fig.2: Phantom Force Feedback device (Desktop model), right, and co-located display combining a two phantom, haptic workspace and stereo vision by using a mirror (left).

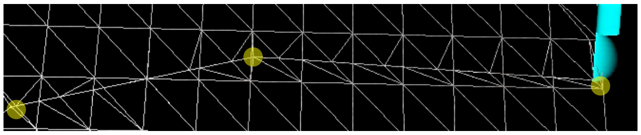

Fig. 3: Changes in mesh topology - the three yellow nodes were created touching the mesh and are connected by path segments.

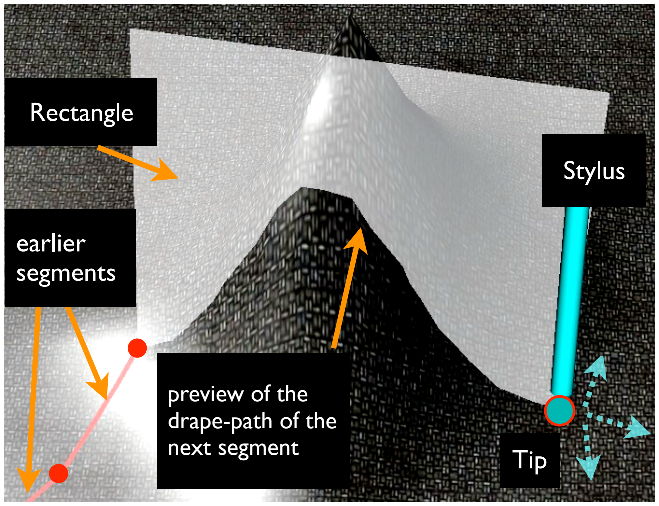

Fig. 4: Touching the mesh with the phantom tip gives a preview of the drape line of the next poly-line segment, rotating the stylus around the tip will change the plane's orientation.

[4] Laehyun Kim, Gaurav S. Sukhatme, Mathieu Desbrun. 2003. Haptic Editing of Decoration and Material Properties, Proceedings of HAPTICS'03. 2003

[5] F. Dachille IX, H. Qin and A. E. Kaufman. 2001. A novel haptics-based interface and sculpting system for physicsbased geometric design. Computer-Aided Design, Vol. 33, No. 5, pp 403-420

[6] B. Sener, Pedgley, O.F., Wormald, P.W. and Campbell, R.I., "Incorporating the Freeform Haptic Modelling System into New Product Development", Proceedings of the EuroHaptics conference, 2003

[7] Al Dean, SensAble Freeform Modelling, Product Development and Manufacturing Solutions, Jan. 2006.

[8] M. Cani, and A. Angelidis, Towards virtual clay. In ACM SIGGRAPH 2006 Courses, SIGGRAPH '06. ACM Press, 67-83. 2006

[9] G. H. Bendels,, Kahlesz, F., and Klein, R. Towards the next generation of 3D content creation. In Proceedings of the Working Conference on Advanced Visual interfaces, ACM Press, New York, 2004.

[10] Daniel Keefe, Robert Zeleznik, David Laidlaw, "Drawing on Air: Input Techniques for Controlled 3D Line Illustration," IEEE Visualization and Computer Graphics, vol. 13, no. 5, pp. 1067-1081, Sept/Oct, 2007

[11] Yi-Je Lim, Wei Jin, Suvranu De, 2007, On Some Recent Advances in Multimodal Surgery Simulation: A Hybrid Approach to Surgical Cutting and the Use of Video Images for Enhanced Realism, Presence, December 2007, Vol. 16, No. 6, Pages 563-583

[12] Huynh Quang Huy Viet, Takahiro Kamada and Hiromi T. Tanaka . 2006. An Adaptive 3D Surface Mesh Cutting Operation, Lecture Notes in Computer Science, Volume 4069/2006, p. 366-37

[13] S. Gibson, 1997. 3D ChainMail: A Fast Algorithm for Deforming Volumetric Objects. Proc. Symp. Interactive 3D Graphics, pp. 149-154, 1997.

[14] Y.Li and K. Brodlie. Soft Object Modelling with Generalised ChainMail - Extending the Boundaries of Web-based Graphics. Computer Graphics Forum, Vol 22, Number 4, December 2003

[15] Bill Baxter, Vincent Scheib, Ming C. Lin and Dinesh Manocha. DAB: Interactive Haptic Painting with 3D Virtual Brushes, proc. ACM SIGGRAPH 2001. pp. 461-468

[16] Boeck, Joan De, Raymaekers, Chris and Coninx, Karin (2006): Exploiting Proprioception to Improve Haptic Interaction in a Virtual Environment. Presence,15 (6) pp. 627-636

[17] Hinckley, K., Pausch, R., Proffitt, D., and Kassell, N. F. 1998. Two-handed virtual manipulation. ACM Trans. HCI. vol. 5/3 , 260-302

[18] Bruyns, C. and S. Senger. 2001. Interactive cutting of 3-D surface meshes. Computer & Graphics, 25(4):635--642, 2001.

[19] Mitasova, H., Mitas, L., Ratti, C., Ishii, H., Alonso, J., and Harmon, R. S. 2006. Real-Time Landscape Model Interaction Using a Tangible Geospatial Modeling Environment. IEEE Computer Graphics & Applications,. 26, 4 (Jul. 2006), 55-63

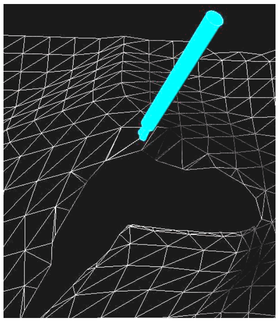

Fig. 5: Widening a polyline cut (incision) with the chainmail deformer tool.

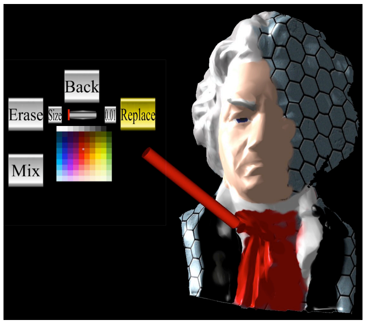

Fig. 6: The paint tool with the free floating, circular, touch-enabled user interface.