

2009

# Expressive cutting, deforming, and painting of three-dimensional digital shapes through asymmetric bimanual haptic manipulation

Adam Joseph Faeth  
*Iowa State University*

Follow this and additional works at: <https://lib.dr.iastate.edu/etd>

 Part of the [Electrical and Computer Engineering Commons](#)

## Recommended Citation

Faeth, Adam Joseph, "Expressive cutting, deforming, and painting of three-dimensional digital shapes through asymmetric bimanual haptic manipulation" (2009). *Graduate Theses and Dissertations*. 10570.  
<https://lib.dr.iastate.edu/etd/10570>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact [digirep@iastate.edu](mailto:digirep@iastate.edu).

**Expressive cutting, deforming, and painting of three-dimensional digital shapes  
through asymmetric bimanual haptic manipulation**

by

**Adam Joseph Faeth**

A thesis submitted to the graduate faculty  
in partial fulfillment of the requirements for the degree of  
MASTER OF SCIENCE

Major: Human Computer Interaction

Program of Study Committee:  
Chris Harding, Major Professor  
James Oliver  
Stephen Gilbert

Iowa State University

Ames, Iowa

2009

Copyright © Adam Joseph Faeth, 2009. All rights reserved.

## TABLE OF CONTENTS

LIST OF FIGURES	iv
LIST OF TABLES	vi
ACKNOWLEDGEMENTS	vii
ABSTRACT	ix
CHAPTER 1. INTRODUCTION	1
1. Thesis Organization	3
2. Digital shape manipulation background	3
2.1. Geometric representation of digital surfaces	3
2.2. Mesh Cutting	4
2.3. ChainMail deformation	7
2.4. Virtual clay	9
3. Bimanual interaction	10
4. Contribution	13
CHAPTER 2. CUTTING, DEFORMING, AND PAINTING OF 3D MESHES IN A TWO HANDED VISO-HAPTIC VR SYSTEM	16
Abstract	16
1. Introduction	17
2. Related Work	19
3. Methods	20
3.1. Hardware & software used	20
3.2. Tool-object metaphor for two handed interaction	22
3.3. Mesh cutting via 3D path draping	23
3.4. ChainMail deformation tool	26
3.5. Paint tool	28
4. Conclusions and future work	29
CHAPTER 3. COMBINING 3-D GEOVISUALIZATION WITH FORCE FEEDBACK DRIVEN USER INTERACTION	31
Abstract	31
1. Introduction	32
1.1. Motivation	32
1.2. Related work	34
1.3. Haptic (touch-based) interaction	35
1.4. Combining 3-D graphics and haptics	38
2. Methods	39
2.1. Software and hardware used	39
2.2. GIS Data preparation	43
3. Examples of 3-D interactions with geospatial data	46
3.1. Digitizing line features on a 3-D surface	47
3.2. Touch pressure based spatial selection	49
3.3. Spatial selection via slope based flooding	51
3.4. Interactive deforming of the terrain surface	52

4. Conclusions and future work	55
5. Acknowledgements	57
<b>CHAPTER 4. SUPPORTING INTERACTIVE HAPTIC SHAPING OF 3D GEOLOGIC SURFACES WITH DEFORMATION PROPERTY PAINTING</b>	<b>58</b>
Abstract	58
1. Introduction	59
2. Related work	61
2.1. Haptics interaction with 3D models	62
2.2. Bimanual interaction	63
2.3. ChainMail deformation	63
3. Deformation property painting and deforming of 3D meshes	64
3.1. Visualizing the deformation property values	66
3.2. Bimanual interaction	68
3.3. Example of using deformation painting in a geoscience context	68
4. ChainMail algorithm for inhomogeneous mesh deformation	71
4.1. Details of Generalized ChainMail	71
4.2. Malleability gradient propagation	73
4.3. Priority queue for deformation candidates	74
5. Results	76
6. Conclusions and future work	77
7. Acknowledgements	78
<b>CHAPTER 5. CONCLUSIONS AND FUTURE WORK</b>	<b>79</b>
<b>REFERENCES</b>	<b>82</b>

## LIST OF FIGURES

Figure 1. A force-feedback haptic device.....	2
Figure 2. Cutting by dragging a tool along a surface (Bruyns & Senger, 2001).....	5
Figure 3 (A-D). Methods of defining a cut path on a mesh (Bruyns, et al., 2002).....	6
Figure 4. The M4 system is used to directly manipulate a textured, irregular triangle mesh with two haptic devices .....	18
Figure 5. Phantom Force Feedback device (Desktop model), right, and co-located display combining a two Phantom, haptic workspace and stereo vision using a mirror (left)....	21
Figure 6. Changes in mesh topology – the three yellow nodes were created touching the mesh and are connected by path segments .....	23
Figure 7. 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 .....	24
Figure 8. Widening a polyline cut (incision) with the ChainMail deformer tool .....	26
Figure 9. The paint tool with the free-floating, circular, touch-enabled user interface .....	29
Figure 10. Using the Phantom haptic device, the user is able to feel the terrain's 3-D shape while draping a line feature directly onto a DTM .....	32
Figure 11. Touching a DTM via a point haptic force feedback device -- a fast succession of forces generated at the tip conveys the terrain's geometry to the stylus held by the user.....	36
Figure 12. Low-end point force feedback devices: Phantom Omni (left) and Novint's Falcon (right).....	37
Figure 13. Using a collocated setup where active stereo graphics are displayed onto a mirror, beneath which the Phantom is operated.....	39
Figure 14. Example of a touch-enhanced X3D file .....	41
Figure 15. M4-Geo's touchable User Interface panel .....	43
Figure 16. Using the stylus to orient a rectangle that shows the predicted intersection of a new segment with the terrain .....	48
Figure 17. The result of digitizing (draping) a polyline on the mesh .....	49
Figure 18. Manual spatial selection of parts of the terrain surface using the interactive paint tool .....	50
Figure 19. Adjusting the size of the paint tool's brush by varying the force applied to a location on the terrain mesh.....	50
Figure 20. Effects of using the flooding selection tool with increasing amounts of force ....	52
Figure 21. Real time terrain deformation .....	53
Figure 22. Constrained terrain deformation by first painting parts of the terrains to become harder (black).....	55
Figure 23. Vertical cut through a geologic model of rock layers with potential variations of a slope angle .....	60
Figure 24. Example of horizontal rock boundaries (horizons) cut by vertical fault planes... ..	60
Figure 25. Phantom haptic force feedback device .....	62
Figure 26. Example linear transfer functions from malleability to deformation property values .....	67

Figure 27 (A-E). Applying deformation parameters to a subsurface mesh, and interactive deformation.....	70
Figure 28. A ChainMail displacement that propagates from the malleable grey nodes to the rigid black nodes fails to preserve the fixed displacement between the black nodes.....	71
Figure 29. Valid region of a point before and after a ChainMail deformation displaces its sponsor.....	72
Figure 30. Comparison of candidate data structures for ChainMail deformation.....	75

## LIST OF TABLES

Table 1. Human manual activities (Guiard, 1987) .....	11
Table 2. Comparison of timings in (ms) for deformation candidate structures.....	77

## ACKNOWLEDGEMENTS

This research is the product of a thousand conversations with my advisors, colleagues, friends, and family. It would not have been possible without the support of so many individuals who helped me to shape and develop the project. I would like to express my gratitude to my committee, Dr. Chris Harding, Dr. James Oliver, and Dr. Stephen Gilbert. Dr. Oliver found a place for me in the Virtual Reality Applications Center and started me on a course in Human Computer Interaction. Dr. Gilbert taught me to look for the theoretical framework in my research, and was a rich source for synthesizing research areas. Dr. Harding introduced me to the field of haptics, secured funding for the project through the National Science Foundation, and provided a playground of devices to for my experimentation. I would especially like to thank my advisor, Dr. Harding, for guiding me through moments of frustration and his active participation in my graduate education.

The development of the M4 project would not have been possible without the efforts of Michael Oren, Jonathan Sheller, and Sean Godinez. Thank you for catching my bugs and your friendship outside of the project.

I am grateful for the friendship of so many of the students and staff of the Human Computer Interaction program and the Virtual Reality Applications Center. I would also like to thank Pamela Shill, Karen Koppenhaver, Jean Bessman, and Lynette Sherer for their tireless efforts to make opportunities possible for me.

I would also like to recognize Jeremy Bennett and Michael Carter for providing advice on build environment issues, tricky bugs, and graphics card nuance. Michael Carter



also deserves my gratitude for mentoring me as I transitioned from Journalism to Computer Engineering after graduating.

My wife Erin provided me with unwavering support, even as I reenrolled in a completely different discipline and subsequently enrolled in graduate school a few years later. Her quiet confidence in me, and her enthusiastic tenacity helped me to grow even when I was unsure which direction that growth would take.

I would also like to thank my parents, Loren and Arleen Faeth, who have always encouraged me in any pursuit. My parents helped to make many so opportunities possible for me, I am grateful for the experiences we shared together.

## ABSTRACT

Practitioners of the geosciences, design, and engineering disciplines communicate complex ideas about shape by manipulating three-dimensional digital objects to match their conceptual model. However, the two-dimensional control interfaces, common in software applications, create a disconnect to three-dimensional manipulations. This research examines cutting, deforming, and painting manipulations for expressive three-dimensional interaction. It presents a cutting algorithm specialized for planning cuts on a triangle mesh, the extension of a deformation algorithm for inhomogeneous meshes, and the definition of inhomogeneous meshes by painting into a deformation property map. This thesis explores two-handed interactions with haptic force-feedback where each hand can fulfill an asymmetric bimanual role. These digital shape manipulations demonstrate a step toward the creation of expressive three-dimensional interactions.

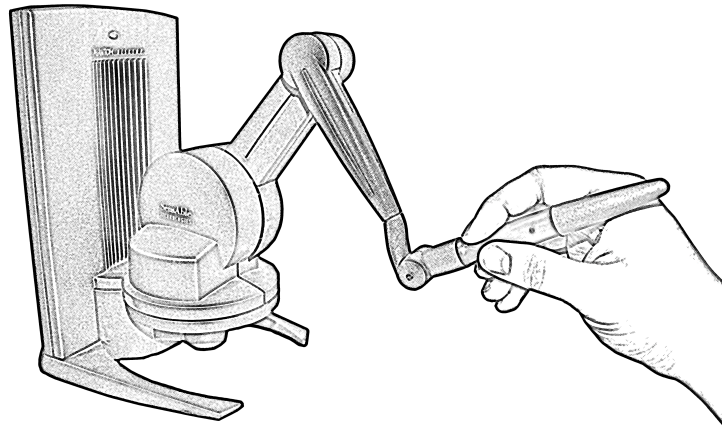
## CHAPTER 1. INTRODUCTION

While humans have a long history of carving, cutting, and shaping the materials of their surroundings, the relatively recent use of computers to perform these manipulations has reduced these tasks to two-dimensional interactions. Digital shape creation provides greater precision, faster replication, and wider dissemination compared to traditional manipulations of real-world objects. Computer-Assisted Design packages such as Pro/E (PTC) and 3D modeling software such as Maya (Autodesk) are designed for digital shape manipulation with a mouse and keyboard. These two-dimensional interfaces are a reduction of three-dimensional interaction and not well suited for expressing three-dimensional manipulations.

Digital shape manipulation is necessary for more than just the creation of new objects. Geoscientists rely on a number of different data gathering methods to interpret the makeup and shape of rock formations beneath the surface of the earth, each with a degree of inaccuracy. The geoscientist often has to combine several sources of data covering the same rock formations to build a better representation. Digital shape manipulation allows the geoscientist to directly manipulate the shape of rock formations to express their mental model in three-dimensional digital form.

The mouse and keyboard are currently the prevalent input devices for interacting with a computer. While no single device is likely to replace either the mouse or the keyboard, a number of new input devices are emerging for specialized applications. An example of a device for three-dimensional input is the haptic force-feedback device, shown in Figure 1. A haptic force-feedback device pushes back against the user's movements in a three-dimensional workspace to create the illusion of the user touching virtual shapes. The haptic

device allows the user to feel the three-dimensional contours of the shape, and provides a more direct input for three-dimensional interaction.



**Figure 1. A force-feedback haptic device**

A limitation of mouse and keyboard interaction for three-dimensional shape manipulation is that it emphasizes one-handed interaction. The hand controlling the mouse performs most manipulation tasks, while the hand on the keyboard typically issues separate commands to the application. This is a reduction of most two-handed tasks where one hand will continuously reestablish a comfortable working position for the finer manipulations of the other hand. For example, when dealing cards, one hand will hold the deck in an opportune position for the other hand to take cards off the top. In contrast, the hand on the keyboard issues shortcut commands to handle separate tasks in most applications instead of contributing a role to the task performed by the hand on the mouse.

The goal of this research is to increase the expressiveness and power of direct manipulation of three-dimensional digital surfaces via bimanual three-dimensional interactions. The challenge in bimanual interaction is how to allow the user to effectively use both hands cooperatively. This research explores the use of two haptic force-feedback devices to perform cutting, deformation, and painting manipulations. Each of these

manipulations adds another challenge. The manipulation for planned cuts must separate large areas of the mesh at interactive rates when the user is satisfied with the planned cut instead of incrementally separating small paths on the mesh. A challenge to creating expressive deformations is how to specify regions that should respond more or less freely to a deformation. This thesis details the development of a mesh cutting operation for planning cuts on a mesh, a deformer specialized for inhomogeneous mesh deformation, and a painting manipulation that also allows the definition of an inhomogeneous mesh.

## **1. Thesis Organization**

The remainder of this chapter provides an introduction to digital shape representation, manipulation operations on digital shapes, and bimanual interaction. Chapter 2 details the creation of the software package for the manipulation of digital surfaces. Chapter 3 details three-dimensional manipulations of digital surfaces for tasks useful in the geosciences. Chapter 4 explains a manipulation interaction that leverages bimanual interaction to provide greater control of the effect of deformation on specific areas of a digital surface. Chapter 5 draws conclusions about the software and provides direction for future analysis of the interaction.

## **2. Digital shape manipulation background**

### **2.1. Geometric representation of digital surfaces**

Computers represent complex digital shapes as clouds of points, sets of discrete polygons, or continuous curves based on mathematical equations. One common method for representing complex digital shapes is with a triangle mesh that breaks a complex shape into

a surface composed of triangular faces. This not only allows modern graphics hardware to leverage optimizations on the underlying triangles, it also allows algorithms to easily locate the neighboring triangles on an otherwise complex shape to simplify manipulations of that shape.

This thesis will examine several methods of direct cutting, deforming, and painting manipulations of three-dimensional triangle meshes. Direct manipulations are reversible, incremental, and provide immediate feedback to the user (Shneiderman, 1983). Mesh cutting separates neighboring regions of the mesh by creating incisions, holes, or a completely distinct new surface. Mesh deformation changes the shape of the mesh by displacing one or more contact points on the mesh and propagating the effect of that displacement through the surrounding regions of the mesh. Mesh painting applies color or texture directly onto the surface of the mesh, and can also apply abstract properties directly to the surface for deformation property painting. This thesis provides greater detail on mesh cutting and mesh deformation because later chapters extend these manipulations.

## **2.2. Mesh Cutting**

One example of a manipulation operation on a mesh is the cutting operation. Cutting separates the neighboring regions of a mesh along a defined path. Common applications of cutting operations include surgical simulation, clothing design, and CAD/CAM manufacturing (Bruyns, et al., 2002). In the geosciences, cutting is useful for separating a single surface that incorrectly aggregates two separate rock surfaces across the boundary of a fault.

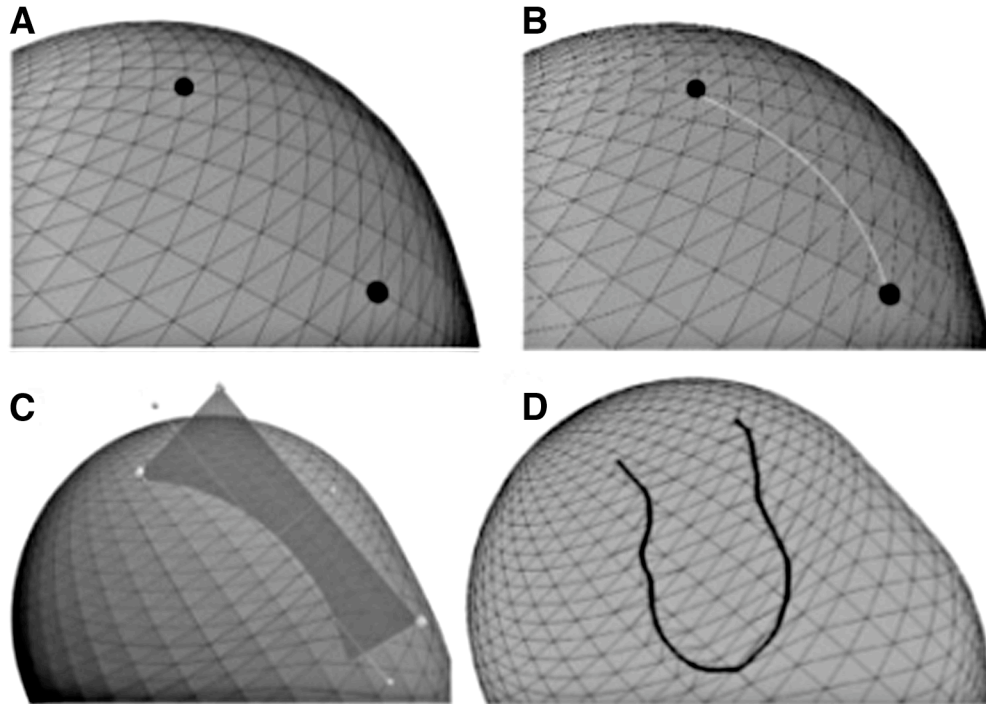
There are two major steps in the cutting operation. The first step defines the path of the cut on the mesh. The second step then updates the primitives of the mesh to reflect the path of the cut. Cutting algorithms approach these steps differently, and those differences can be used to classify cutting algorithms. In addition to classifying cutting algorithms by how they perform path definition and mesh updating, Bruyns et al. also define three additional classification dimensions (2002).



**Figure 2. Cutting by dragging a tool along a surface (Bruyns & Senger, 2001)**

Cutting algorithms approach the definition of the cut path by either connecting control points on the mesh, placing a three-dimensional template through the mesh, or moving a virtual tool over the surface of the mesh. Seed points can be connected by geodesic or Euclidian shortest paths, shown in Figure 3B, or with the definition of a cutting plane between the two points. Cutting algorithms that define the path of a cut with a template create the cut path by finding the intersection of the three-dimensional object with the mesh, shown in Figure 3C. Tracing on the mesh to produce the path can be similar to placing seed

points, but typically involves detecting the places where the traced path crosses a mesh primitive, rather than creating a continuous line between points on the path.



**Figure 3 (A-D). Methods of defining a cut path on a mesh (Bruyns, et al., 2002)**

One important classification of mesh cutting algorithms is how they update the mesh to reflect the path of the cut. Mesh cutting algorithms fall into two categories: deletion cutters and re-meshing cutters. Deletion cutters either remove whole faces intersected by the path of the cut, or separate neighboring faces on opposite sides of the cut path. This approach leads to a quick execution time, but tends to dispel the illusion of the mesh representing a complex shape by exposing ragged triangles at the cut edges. Some deletion cutters subdivide faces close to the path of the cut to decrease the ragged appearance of these edges (Viet, Kamada, & Tanaka, 2006). Re-meshing cutters embed the path of the cut in the mesh either by moving edges to align with the path of the cut or subdividing faces to follow the path of the cut. The re-meshing cutter produces precise cuts even in coarse meshes because the cut



follows the actual intersection with the mesh instead of the path of the nearest existing elements (Bruyns & Senger, 2001). However, many existing interactive cutting algorithms are intended for making short incremental cuts, so the challenge is to create a cutting algorithm that can plan cuts across a large mesh and later embed the planned cut at interactive rates.

### **2.3. ChainMail deformation**

In a software system that supports mesh cutting, the deformation manipulation must also preserve the updated mesh created by the cut. Researchers have developed a number of deformation algorithms with different tradeoffs, such as mass-spring models, finite element analysis, radial basis functions, free form deformation, and ChainMail (Dräger, 2005).

ChainMail is well suited to interactive freeform deformation because it approximates some material properties while providing the user with immediate feedback about the shape of the deformation.

ChainMail deformation can model elastic and plastic behavior. An elastic deformation will return to the original shape when the deformation stimulus is no longer present. A plastic deformation will remain in its deformed shape after the deformation stimulus is no longer present. Other deformation algorithms may model both behaviors (Teschner, Heidelberger, Muller, & Gross, 2004). However, this research only models plastic deformation behavior because the goal is to manipulate a digital model to match a conceptual model.

Gibson developed ChainMail for interactive deformation of 2D and 3D objects consisting of hundreds of thousands of nodes on a regular grid (Gibson, 1997). ChainMail

approximates rigid, elastic, and plastic materials with three deformation property values: stretch, compression, and shear. While ChainMail only provides an approximation of physical properties, it has several performance advantages. ChainMail only moves each node once during a deformation cycle, and it only analyzes the nodes that must move as a result of the displacement of the contact points. These performance advantages make it possible for the user to see the result of the ChainMail deformation in each graphical frame, and interactively shorten the displacement of the contact points to reduce the severity of a deformation.

Further development on ChainMail diverged into two separate paths. The Enhanced ChainMail algorithm adapted ChainMail to operate on inhomogenous meshes (Schill, Gibson, Bender, & Manner, 1998). A homogeneous mesh has consistent deformation property values over the entire mesh, while an inhomogeneous mesh may contain nodes with varied deformation property values. Inhomogeneous mesh deformation can produce interesting results where parts of the shape are frozen or thawed. However, Enhanced ChainMail was intended to operate on quadrilateral meshes.

The Generalized ChainMail algorithm later removed the requirement that the mesh be quadrilateral, adapting ChainMail to work on arbitrary meshes in 3D (Li & Brodlie, 2003). Specifically, this extension enables the ChainMail deformation of triangular meshes commonly used in computer graphics. However, Generalized ChainMail omitted any consideration for inhomogeneous mesh deformation, which is necessary for expressing regions that should react more or less severely to deformation.

## 2.4. Virtual clay

Research into virtual clay is a field that primarily explores the use of deformation manipulations to achieve similar goals of manipulating digital shapes and simulating the expressiveness of working with real clay. Some systems also use a haptic force-feedback device for interacting with the virtual clay. Commercial systems, such as FreeForm and ClayTools have also been developed to simulate the manipulation of virtual clay with a haptic force-feedback device (SensAble).

The inTouch system allows the user to paint and manipulate 3D virtual objects with a simple heuristic deformation (Gregory, Ehmann, & Lin, 2000). ArtNova later expanded on inTouch to improve haptic feedback and scene navigation and introduced a texture-painting tool that allows the user to paint an object with image textures. Users of ArtNova commented that the force-feedback provided by a haptic device helped them to maintain contact with the surface of the object when performing detail painting of the object (Foskey, Otaduy, & Lin, 2002).

Another approach to real-time virtual clay modeling uses a volumetric representation to simplify accurate deformation calculations (Dewaele & Cani, 2003). This virtual clay model uses a deformation algorithm with a striking resemblance to the ChainMail algorithm, with the deformation propagating clay to neighboring cells until the cells absorb the deformation (Dewaele & Cani, 2003). Cani and Angelidis discuss the advantages and disadvantages of two other approaches: volumetric implicit surfaces and a geometric model framework (2006). Recently, Pihuit et al. explored the use of a foam ball augmented with force sensors and a Phantom force-feedback device for pinching, stretching, and deforming the volumetric representation of virtual clay (2008).

McDonnell et al. presented another model of virtual clay based on subdivision solids (2001). This approach allows users to manipulate the points on a control lattice to alter the geometry of the shape. One of the interesting tools included in this system was a stiffness painting tool, which allowed the user to paint stiffness values onto the nodes of the control lattice, which would then stiffen or soften those control lattice nodes in subsequent deformations (McDonnell, et al., 2001).

### 3. Bimanual interaction

A limitation of most conventional methods for manipulating three-dimensional digital shapes is that they primarily rely on unimanual manipulations. One hand performs the manipulation with the mouse, while the other hand often rests on the keyboard or performs a completely different task. The hand on keyboard often has no way to assist with the manipulation task. This is different from the way humans typically manipulate non-digital shapes with two hands.

The majority of human manipulation tasks are performed with asymmetrical bimanual actions (Guiard, 1987). Actions performed with one or two hands can be categorized in a hierarchy, shown in Table 1, according to whether the action is unimanual or bimanual, symmetric or asymmetric, and whether the actions of each hand are in-phase or out-of-phase. Unimanual tasks are performed completely with one hand, while bimanual tasks are performed with two hands contributing actions to the task. Symmetric tasks assign the same role to both hands, while asymmetric tasks assign different, but complementary roles to each hand. For in-phase actions, the movement of both hands occurs simultaneously, while the movement will occur independently during out-of-phase tasks.

**Table 1. Human manual activities (Guiard, 1987)**

Unimanual	Bimanual		
	Asymmetric	Symmetric	
		In phase	Out of phase
Dart throwing Brushing one's teeth	Dealing cards Playing a string instrument	Rope skipping Weightlifting	Rope climbing

The Kinematic Chain model predicts that in asymmetric bimanual tasks, one hand will perform the macrometric actions, and the other hand will perform the micrometric actions (Guiard, 1987). For example, when writing with a pen and paper, the hand with the macrometric role positions the paper in an opportune location for the hand with the micrometric role to control the fine movements of the pen. The choice of which hand fulfills the macrometric role and which fulfills the micrometric role manifests as the individual's lateral preference. Guiard makes a distinction between lateral preference and manual preference, and argues that lateral preference subsumes manual preference because unimanual actions can be represented by a bimanual task with the second hand performing an empty task. (Guiard, 1987)

There have been several experiments to test the application of the Kinematic Chain model for 3D manipulation tasks. In an asymmetric task that asked users to fit a tool into a target object, users performed the task faster and made fewer errors when they held the target object with their nondominant hand and the tool with their dominant hand, compared to the reversed situation (Hinckley, Pausch, Proffitt, Patten, & Kassell, 1997). Another experiment asked users to align two digital objects using tracked tangible objects, and found that the movements of the dominant hand were made relative to the frame of reference established by

the nondominant hand (Hinckley, Pausch, Proffitt, & Kassell, 1998). Hinckley et al. further conclude that employing two hands for a task will not always result in a reduction in completion time, since the Kinematic Chain predicts that a hierarchy exists between the actions of the hands. However, because users are able to accurately determine the position of their hands relative to each other, bimanual interface design might lead to the creation of interfaces which rely less on visual attention (Hinckley, et al., 1998). De Boeck et al. demonstrated that users made fewer errors when they could grab a digital menu with a tracked hand and select items on it with a haptic force-feedback device than when touching a floating menu with the haptic device alone (2006).

Another experiment compared the use of a Phantom force-feedback device to a Phantom with a fixed space mouse, and a Phantom with a trackball to control rotation for CAD part manipulation tasks (Fiorentino, Uva, & Dellisanti Fabiano, 2008). Both bimanual setups reduced completion times for the tasks. In another asymmetric task that asked users to manipulate the control points of a 2D curve to match a predefined curve, researchers found that two-handed manipulation demonstrated an advantage in completion time over one-hand manipulation as the difficulty of the task increased (Owen, Kurtenbach, Fitzmaurice, Baudel, & Buxton, 2005).

Researchers have also developed other 3D applications to leverage asymmetrical bimanual interaction. MHaptic was designed to allow users to use two haptic force-feedback gloves to reposition objects in a virtual environment, though it does not detail the roles each hand may perform (Ott, De Perrot, Thalmann, & Vexo, 2007). THRED allows users to manipulate the vertices of a hierarchical quadrilateral surface with one hand while manipulating the virtual environment with another hand (Shaw & Green, 1997). SFA is a

similar system for visualizing volumetric data which also uses a magnetic tracker in one hand to frame the scene and control the environment while another hand with a magnetic tracker performs selection within the visualization (Ebert & Shaw, 2001). Grossman et al. describe a different bimanual interaction where the dominant hand controls the environment and the nondominant hand provides an anchor point for a subsequent drawing by the dominant hand (2001). Kron et al. outline an approach for using two Phantom force-feedback devices to control two arms of a bomb-disposal robot through telepresence (2004). However, the use of two haptic force-feedback devices for virtual manipulation tasks remains unexplored.

#### 4. Contribution

This thesis presents an asymmetric bimanual interaction for performing cutting, painting and deforming manipulations of digital surfaces. A Phantom force-feedback device controlled by the user's nondominant hand rotates and positions a 3D surface with six degrees of freedom to provide a comfortable workspace for the tool controlled by the dominant hand. A second Phantom controlled by the dominant hand may then perform micrometric manipulations of the three-dimensional surface using tools to cut, deform, or paint the digital object. The assignment of macrometric and micrometric actions may be assigned to the user's hands based on their individual lateral preference.

This research also presents a cutting algorithm intended to allow planning of cuts on a mesh and the ability to change the cut path before embedding the cut into the mesh. Based on the categorization presented in Section 2.3, this cutting algorithm is a re-meshing cutter that uses a template for path determination. Other cutting algorithms for surgical simulation are intended to immediately cut the mesh to represent the travel of the scalpel since the previous

calculation of the cut. In contrast, the cutting algorithm presented in this research allows the user to plan long cuts across the surface of the mesh, undo segments of the path, and even move segments of the cut path before embedding the cut in the mesh. To maintain interactive rates when the user subsequently embeds the cut in the mesh, the cutting algorithm pre-calculates the elements of the mesh that will need to be re-meshed, and uses that information to speed up the subdividing and splitting operations.

ChainMail deformation is useful in this system because it preserves the topology changes created by cutting operations on the mesh. This thesis presents an extension of Generalized ChainMail that allows deformation of inhomogeneous triangular meshes. This extension unifies the ability to operate on triangular meshes provided by Generalized ChainMail and the ability to deform inhomogeneous meshes provided by Enhanced ChainMail. However, the algorithm presented in Chapter 4 implicitly enforces a malleability gradient rather than calculating which of the neighbors to move each time it moves a node.

This research further presents a painting metaphor for defining inhomogeneous mesh properties. In addition to painting into a color texture, users may also paint a malleability value into a deformation property map to specify local deformation property values. Instead of painting with deformation property values of stretch, shear, and compression, the system uses a malleability value and uses transfer functions to map malleability values to deformation property values. Since the malleability value only has one dimension, it can be visualized through a temperature metaphor, with stiffer regions taking on the color of cold iron, and malleable regions being represented by the color of white-hot iron.

The combination of deformation property painting and inhomogeneous mesh deformation presented in this thesis leads to a complete interaction where a user can define



local regions of a shape that will remain rigid, and other regions of the shape to bend into place. This provides precise control over the effect of a deformation on a surface. This combined interaction is one step towards the goal of this research, to enable more expressive manipulation of three-dimensional surfaces through three-dimensional interaction.

## CHAPTER 2. CUTTING, DEFORMING, AND PAINTING OF 3D MESHES IN A TWO HANDED VISO-HAPTIC VR SYSTEM

Modified from a paper published in

*Proceedings of IEEE Virtual Reality (IEEE VR '08)*, March 2008, Reno, Nevada, USA.

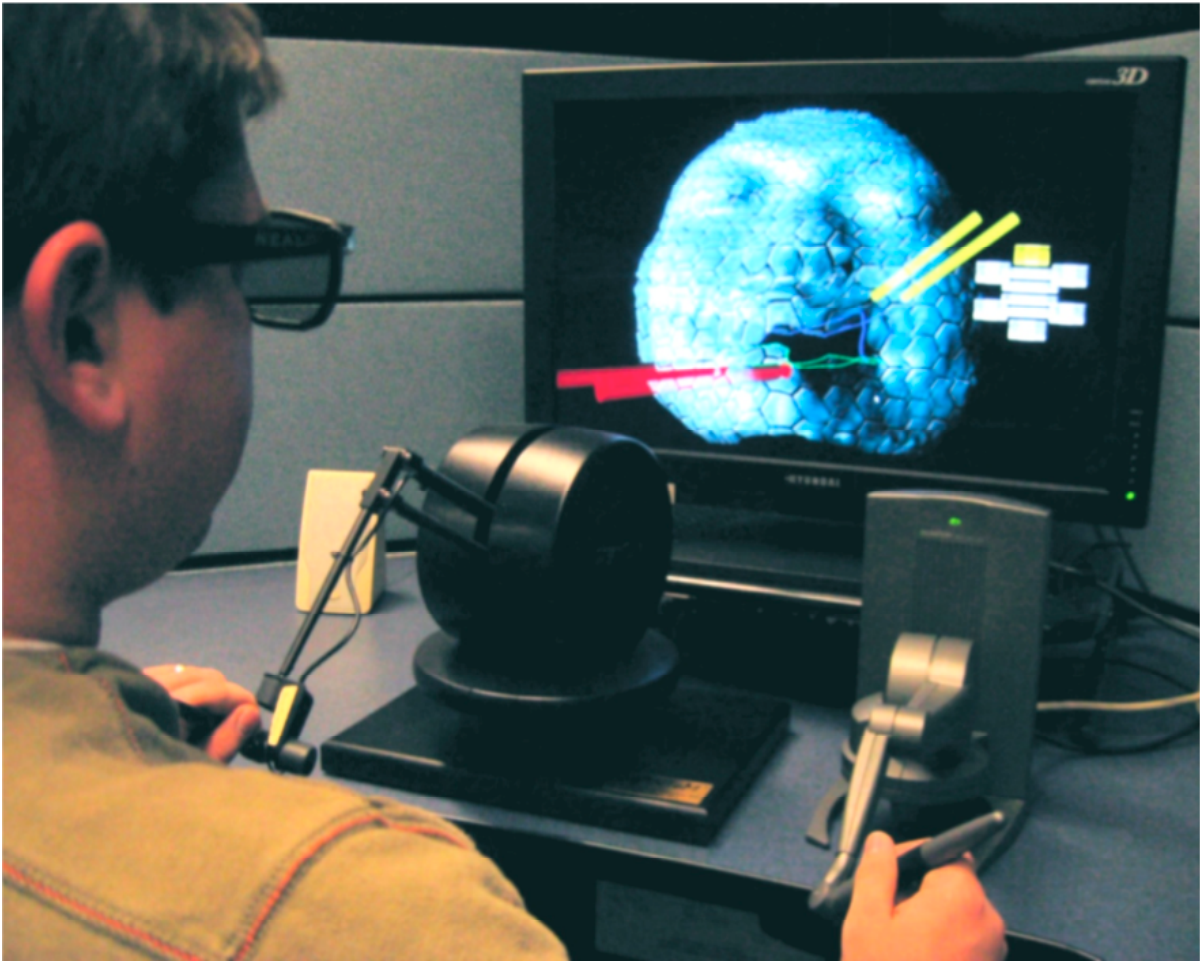
Adam Faeth, Michael Oren, Jonathan Sheller, Sean Godinez, Chris Harding

### 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.

## 1. Introduction

The creation and manipulation of 3D digital shapes (3D models) is at the heart of many 3D modeling applications, commercial applications such as Maya (Autodesk), 3D Studio Max (Autodesk), Cinema 4D (Maxon), and open source applications such as Blender (The Blender Foundation). 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 (Figure 4) for the manipulation (or editing) of 3D shapes made from non-volumetric triangle meshes.



**Figure 4. The M4 system is used to directly manipulate a textured, irregular triangle mesh with two haptic devices**

In the M4 system 3D stereo vision and force feedback provide the necessary perceptual 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.

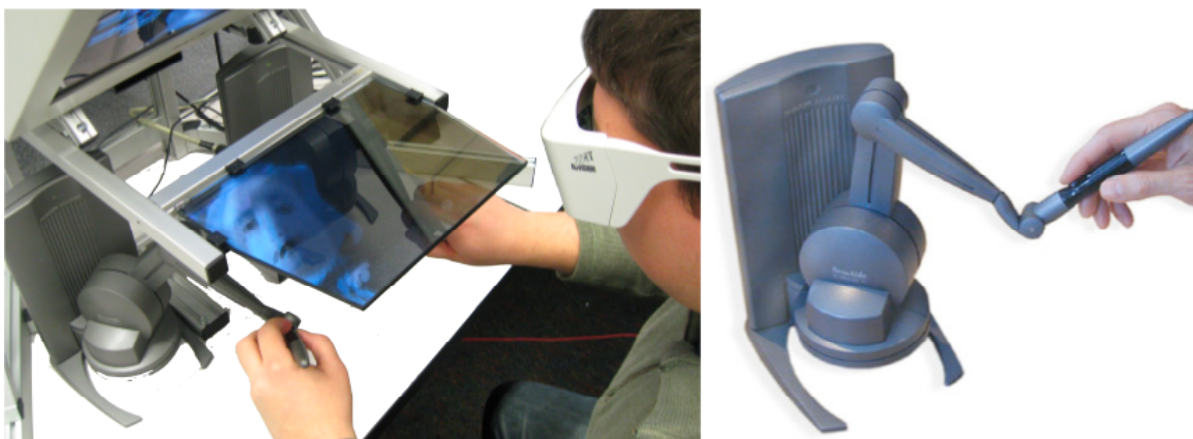
Foskey et al. (2002) and Gregory et al. (2000) deal with painting and deformation (but not cutting) of 3D models made from arbitrary polygonal meshes. Baxter et al. simulate haptic brushes for painting (2001), and Johnson et al. deal with painting textures on trimmed NURBS (1999). Kim et al. explore the editing of discrete, volumetric implicit surface representations (2003); Dachille et al. deal with deformations of B-Splines surfaces (2001). Freeform, the only commercially developed system (Dean, 2006; Sener, Pedgley, Wormald, & Campbell, 2003), and Cani et al. use a virtual clay approach to deform volumetric representations of objects (2006). Bendels et al. use a dual-hand combination of a Phantom and hand gestures on a mirrored display (2004). Keefe et al. use haptics to support freehanded 3D drawing via drag (2007). The use of two hands for interactions and its benefits have been investigated extensively e.g. De Boeck et al. (2006) and Hinckley et al.

(1998). Anecdotal evidence (Gregory, et al., 2000; Sener, et al., 2003) and user evaluations (De Boeck, et al., 2006; Hinckley, et al., 1998; Keefe, et al., 2007) 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., (Lim, Jin, & De, 2007; Viet, et al., 2006).

### 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 (Figure 5, 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 (Figure 5, left) provides a 1-to-1 overlap of the haptic workspace and the 3D stereo image and B) as a fish tank VR setup with a passive stereo flat panel (Arisawa P240W) with the Phantoms positioned to the left and right in front of the display (Figure 4). 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 (Boeck, Raymaekers, & Coninx, 2006). A comparison of the effectiveness of these two setups for manipulating 3D meshes is part of the upcoming user study.



**Figure 5. Phantom Force Feedback device (Desktop model), right, and co-located display combining a two Phantom, haptic workspace and stereo vision using a mirror (left)**

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

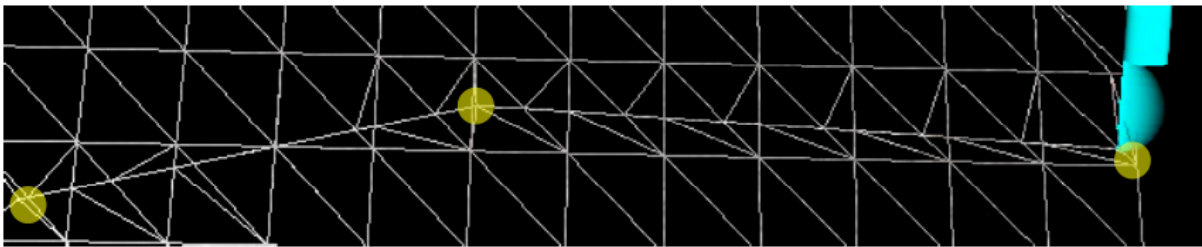
Using the tool-object metaphor, the dominant hand (DH, simulated via one Phantom) operates a manipulation tool on the 3D mesh, which is held by the non-dominant hand (NDH, simulated by the second Phantom). This holding of the mesh is simulated by selecting the "grab tool" for the NDH, bringing the tool 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.



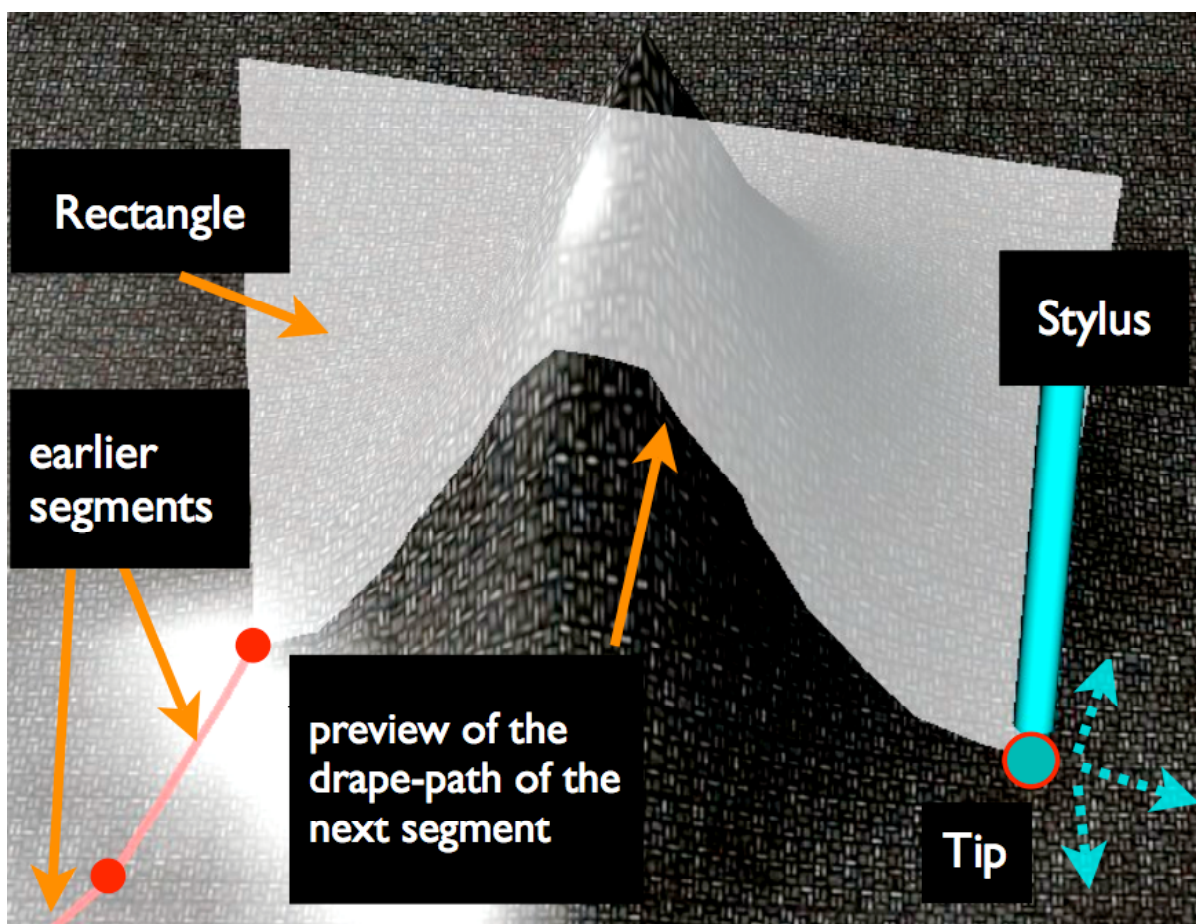
### 3.3. Mesh cutting via 3D path draping

Initially we based our cutting approach on a technique from the tissue cutting domain (Bruyns & Senger, 2001), 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 (Figure 6).



**Figure 6. Changes in mesh topology – the three yellow nodes were created touching the mesh and are connected by path segments**

This two-stage approach allows us to experiment with several novel approaches, such as the preview feature shown in Figure 7, 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 Figure 7 will move the drape path towards the bottom of the hill -- once the desired path is found, it can be integrated into the mesh.



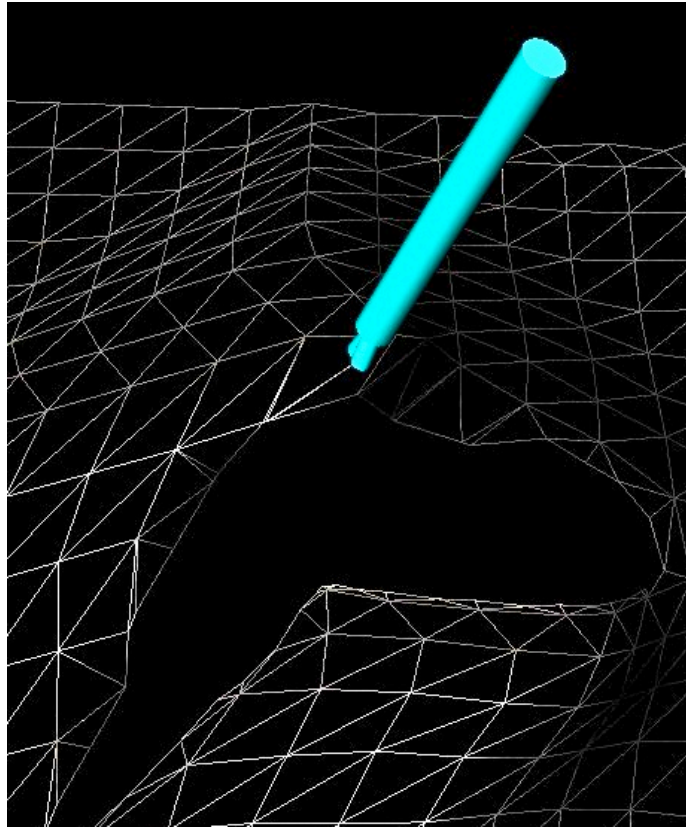
**Figure 7. Touching the mesh with the Phantom tip gives a preview of the drape line of the next poly-line segment, rotating the stylus around the tip will change the plane's orientation**

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 (Figure 6) that can be widened later (Figure 8) 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.



**Figure 8. Widening a polyline cut (incision) with the ChainMail deformer tool**

### **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 (Figure 8) (Gibson, 1997; Li & Brodlie, 2003).

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 (Figure 9), 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.



Figure 9. The paint tool with the free-floating, circular, touch-enabled user interface

#### 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 stereovision 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 (Mitasova, et al., 2006).



## CHAPTER 3. COMBINING 3-D GEOVISUALIZATION WITH FORCE FEEDBACK DRIVEN USER INTERACTION

Modified from a paper published in

*Proceedings of the 16th annual ACM Advances in Geographic Information Systems (ACM GIS)*, November 2008, Irvine, California, USA.

Adam Faeth, Michael Oren, Chris Harding

### Abstract

We describe a prototype software system for investigating novel human-computer interaction techniques for 3-D geospatial data. This system, M4-Geo (Multi-Modal Mesh Manipulation of Geospatial data), aims to provide a more intuitive interface for directly manipulating 3-D surface data, such as digital terrain models (DTM). The M4-Geo system takes place within a 3-D environment and uses a Phantom haptic force feedback device to enhance 3-D computer graphics with touch-based interactions. The Phantom uses a 3-D force feedback stylus, which acts as a virtual "finger tip" that allows the user to feel the shape (morphology) of the terrain's surface in great detail. In addition, it acts as a touch sensitive tool for different GIS tasks, such as digitizing (draping) of lines and polygons directly onto a 3-D surface and directly deforming surfaces (by pushing or pulling the stylus in or out). The user may adjust the properties of the surface deformation (e.g., soft or hard) locally by painting it with a special "material color."

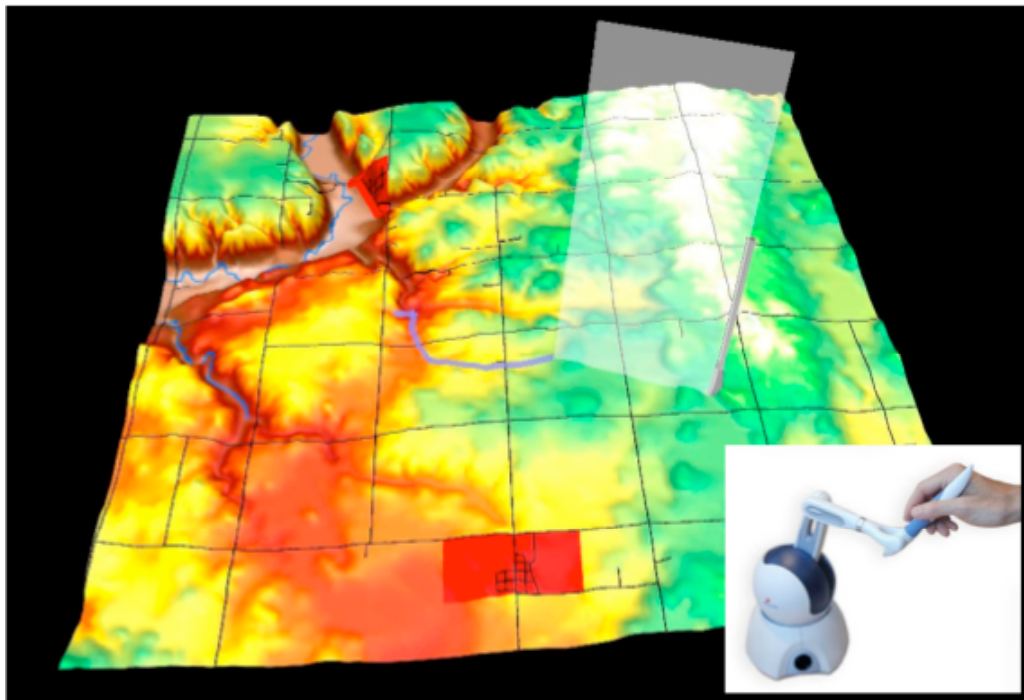
The overlap of visual and force representation of 3-D data aides hand-eye coordination for these tasks and helps the user to perceive the 3-D spatial data in a more

holistic, multi-sensory way. The use of such a 3-D force feedback device for direct interaction may thus provide more intuitive and efficient alternatives to the mouse and keyboards driven interactions common today, in particular in areas related to digital landscape design, surface hydrology and geotechnical engineering.

## 1. Introduction

### 1.1. Motivation

Geovisualization (Kraak & MacEachren, 2005; MacEachren & Kraak, 2001) may be described as the intersection of many different fields, including Geographic Information Systems (GIS), 3-D scientific visualization, computer graphics, human-computer interaction and virtual reality.



**Figure 10.** Using the Phantom haptic device, the user is able to feel the terrain's 3-D shape while draping a line feature directly onto a DTM

The long-term aim of this research is to fundamentally improve the way GIS users are able to interact with 3-D geospatial data by combining 3-D computer graphics with the emerging field of haptics. The use of a stylus as physical interface for user interaction allows us to explore ways of extending the traditional "pencil & paper map" interface into 3-D and to create a touchable 3-D version of the 2-D paper map. The ability to "trace" the underlying terrain morphology in 3-D is a valuable sensory addition during many types of geoscientific tasks, including the precise placement of points, lines or polygons (Figure 10), or the selection of parts of the terrain via painting them with a certain color. Beyond the haptovisual perception of static geometry, where the user perceives multiple channels of information (output) about a 3-D model but does not change it, the system also supports dynamic, touch-based interactions, such as the real-time terrain deformation, in which the 3-D model changes according to the user's input and the changes can be seen and felt immediately.

Given the ongoing trend from 2-D GIS to (visual) 3-D geovisualization and the spreading use of the Phantom in other, non-geoscience applications areas, such as 3-D scientific visualization of high-dimensional data or for medical simulation, the specific research questions for our work are: What geoscientific tasks can be fundamentally improved by adding haptics? What technology gives programmers the ability to create visual-haptic 3-D interfaces at a high level and supports several types of common haptic devices? The M4-Geo, multi-sensory (multimodal) interface presented here, provides several examples of interacting with 3-D geospatial data via both vision and touch. M4-Geo is based on the open source H3D API and offers a view ahead to new types of fundamentally different interactions that would become possible in a next-generation, touch-enabled 3-D GIS system.

## 1.2. Related work

The research presented here relates to geographic visualization (Nöllenburg, 2007), especially the 3-D visualization of geospatial data (Döllner, 2005; Hay, 2003; Wood, Kirschenbauer, Döllner, Adriano, & Bodum, 2005) and to geoscientific virtual environments (Fröhlich, Barrass, Zehner, Plate, & Göbel, 1999), to human-centered visualization techniques (Fikkert, D'Ambros, Bierz, & Jankun-Kelly, 2007; Haan, Koutek, & Post, 2002), to usability issues (Slocum, et al., 2001) and to 3-D user interface design (Bowman, Kruijff, LaViola, & Poupyrev, 2001).

Earlier work in investigating terrain meshes (in this case from bathymetry data) via touch was presented in (Harding, Kakadiaris, Casey, & Loftin, 2002). This system used early versions of several visualization techniques that M4-Geo has implemented via shader technology. Kowalik presented work on integrating surface modeling via force feedback into a geologic 3-D (subsurface) modeling system (2001). Although no longer developed, this work inspired many of the interaction techniques explored in M4-Geo. Aspects of interactive editing large terrain data were discussed in (Atlan & Garland, 2006); usability experiments for scientific visualization in immersive virtual environments performed by van Dam, Laidlaw, and Simpson included interaction with 3-D Mars terrain data via a PDA (2002). Krum et al. presented a speech and gesture driven whole Earth 3D terrain model (2002).

Our work is also related to Tangible User Interfaces (Mitasova, et al., 2006; Ratti, Wang, Ishii, Piper, & Frenchman, 2004). These systems deal with a scientific subject matter very similar to M4-Geo's and also address the user's need for more intuitive methods of shaping terrain; however, the physical interface is a shapeable model of "illuminated clay." Our work is also part of the larger field of haptic visualization (Roberts & Franklin, 2005;

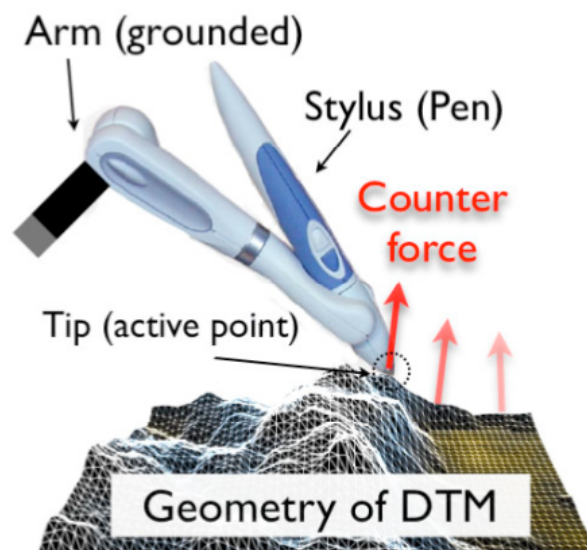
Roberts & Paneels, 2007), which is beginning to gain importance. Haptic rendering of 3-D scientific data, either in support of already visually rendered data or without a visual counterpart, aims to enhance the users holistic perception and comprehension of complex high-dimensional data, such as computational fluid dynamics (CFD) data (Ikits, Brederson, Hansen, & Johnson, 2003; Lawrence, Pao, Lee, & Novoselov, 2004; Lundin, Cooper, Persson, Evestedt, & Ynnerman, 2007; Qi, 2006).

### **1.3. Haptic (touch-based) interaction**

The human haptic system incorporates two types of sensory touch information: skin-pressure-based (tactile) feedback and joint/muscle-based force-feedback (kinesthetic). Computer haptics refers to methods for creating haptic impression to a human user via a specialized piece of hardware (haptic interface or display). Much like a 3-D computer graphics system conveys information about virtual (non-existing) 3-D objects to the user via a display, a computer haptics system generates computer-controlled forces to convey the shape, friction, etc. of virtual 3-D objects. To feel the 3-D object, the user needs to be in direct contact with the haptic interface device; the device is, in turn, controlled by haptic rendering algorithms (software). This connection creates a closed loop that continuously exchanges force signals and position signals between the user and the virtual 3-D objects.

For technical reasons the most commonly used type of haptic interface devices used today are point haptic devices, which use only a single point of contact to provide active, kinesthetic feedback but provide no tactile (skin-pressure-based) feedback to the user. In the case of the SensAble Phantom models this single point sits at the end of a small, grounded arm and acts as the tip for a stylus (or pen). The movement of the tip within the haptic

workspace is limited to typically 2 to 4 inches on each side. The process of haptically rendering the geometry of a 3-D object composed of triangles, for example a cube, first requires a test to detect if the user guided tip is currently inside the object, if not, no force is generated and the tip's path is not impeded. However if the Phantom's arm detects (tracks) the tip at a position that would be inside the object, a point that lies directly on the surface, the so-called proxy point is calculated and the Phantom's motors generate a counter force so that the tip is pushed towards the proxy. The maximum force generated by a Phantom is only 3-7 Newton, depending on the type of haptic device. Despite the use of discrete "jolts" of force the user's hand is given the impression of feeling the continuous surface of an object that is placed within the haptic 3-D workspace (Figure 11), because this detect-push-cycle happens at very fast rates, typically ~1000 times per second. Salisbury et al. presents an introduction to haptic rendering and provides further details (Salisbury, Conti, & Barbagli, 2004).



**Figure 11. Touching a DTM via a point haptic force feedback device -- a fast succession of forces generated at the tip conveys the terrain's geometry to the stylus held by the user**

Several haptic properties can be attached to the geometry of such a 3-D object, similar to defining the color of an object for visual rendering. Such haptic properties include stiffness (the softness of an object), static and dynamic friction, attraction effects, and a gray scale image of "micro displacements," similar to bump map textures in visual rendering.

Besides triangle-based objects, poly-lines and points can be rendered via magnetic attraction forces that become active within a certain range and attract the stylus towards the (closest) point. Beyond the rendering of geometrically bounded objects, generalized force field effects can also be used to craft more ephemeral expressions, for example, 2-D or 3-D vector field can be rendered by setting the tip's force to the direction and magnitude of data at tip's current location and can facilitate the exploration of higher-dimensional data (Ikits, et al., 2003; Lawrence, et al., 2004; Lundin, et al., 2007; Qi, 2006). The haptic effects can be mixed together and added to the haptic rendering of 3-D geometric objects. For example, the vectors of water flowing downhill over a terrain surface could be translated into force vectors and added to the rendering of the surface.



**Figure 12. Low-end point force feedback devices: Phantom Omni (left) and Novint's Falcon (right)**

Until recently the cost of haptic force feedback devices was relatively high: even the Phantom Omni, a relatively low end device shown in Figure 12 (left) costs around \$2000, larger and more powerful devices cost substantially more. The Novint Falcon (Figure 12,

right) costs only \$200 but has a relatively small workspace. It uses a 4-button "knob" instead of a pen for the user to move around and the software would need to compensate for this.

#### **1.4. Combining 3-D graphics and haptics**

The power of 3-D interactive graphics for 3-D geovisualization is well established. The M4-Geo system uses the Phantom force feedback device as an addition to a 3-D visual environment, using either monoscopic or stereoscopic display technology. This results in what could be called a form of multi-sensory (vision + touch) virtual environment in which the Phantom's stylus provides a virtual tool that is held in the user's dominant hand. In addition, the user's non-dominant hand could be represented by a second Phantom, opening up many new ways for interacting with 3-D objects, some of which we are currently investigating.

As the user moves the Phantom's tip and rotates the stylus around the tip, both are displayed in 3-D relative to the other 3-D objects, for example a 3-D terrain model. M4-Geo uses this setup to provide the user with a very intuitive form of egocentric navigation system, where the terrain model is treated as an object that sits in front of the user, and which the user can move and rotate (grasp) while the user's head position remains static. When the user touches the surface and holds down the stylus button, the surface is momentarily attached to the stylus and the user can move and rotate the surface, dropping and re-grasping it as needed. If a second Phantom is available for a second hand, it becomes possible to grasp the surface at two points and to scale it up or down by pulling the two points apart or bringing them closer together (similar to the zoom functionality implemented in multi touch devices such as the iPhone).



Because of the size and mechanical nature of the Phantom's arm, it is advantageous to use stereo graphics and a mirrored display to give the user the impression that the physical stylus of the device is co-located with the virtual stylus and that the visual workspace and the haptic workspace thus overlap spatially (Figure 13).



**Figure 13. Using a collocated setup where active stereo graphics are displayed onto a mirror, beneath which the Phantom is operated**

## 2. Methods

### 2.1. Software and hardware used

The M4-Geo system uses the open source H3D API created by SenseGraphics AB; its current version is 2.0. H3D is an implementation of the X3D scenegraph API (the successor to VRML) extended for haptic rendering and for haptic interaction via one or more point haptic force feedback devices. H3D supports the SensAble line of Phantoms (Omni, Desktop,

Premium), ForceDimension's Delta and Omega devices and Novint's Falcon device. H3D renders a 3-D scene graphically and haptically; the scene's objects have graphical properties such as color, and haptic properties, such as friction. Much like other scenegraph APIs, such as OpenSG, OpenScenegraph, and Performer, H3D is a high level API that takes care of many complex rendering details and internally converts high-level concepts into calls to lower level haptics APIs (the equivalent to OpenGL for 3D graphics). For example, H3D automatically provides the haptic thread needed for the concurrent haptic rendering and collision detection at rates of ~1000 frames/second while the scenegraph (and the graphics) are updated at ~30-60 frames per second. H3D typically uses a point-based haptic proxy model, a sphere-based proxy model is offered in the latest version (2.0).

H3D provides configuration files to adjust the haptic workspaces' size and position, the display setup, and other parameters to accommodate the different devices and displays. This flexibility allows us to run the M4-Geo software in many different settings. We have used it with a standard mono display (even on a laptop), with an active stereo display (using a NVidia Quadro card and a CRT monitor) and with a passive stereo display (Arisawa P240W LCP display). We have used it with a single Phantom (Omni, Desktop or Premium model) or a Novint Falcon attached. We have also explored the use of two Phantom devices together where one Phantom acts as dominant hand for operating a virtual tool and the second Phantom can simultaneously translate and rotate the object as if it were an object held in the users other hand.

H3D provides a basic viewer (H3DViewer) for viewing and touching the content of a X3D file. X3D retains many elements of the older VRML2 format but uses the XML-style syntax. For example, the text file shown in Figure 14 creates a sphere in the center of the

haptic workspace with visual and haptic properties. M4-Geo uses this X3D file format to initially load 3-D objects, such as triangle meshes, into scenegraph nodes.

```
<X3D profile="Immersive">
  <Scene>
    <Shape>
      <Appearance>
        <Material diffuseColor="0 0.5 1"/>
        <ImageTexture url="earth-topo.jpg"/>
        <MagneticSurface snapDistance="0.1"
          dynamicFriction="0.6" staticFriction="0.2"/>
      </Appearance>
      <Sphere radius="0.5"/>
    </Shape>
  </Scene>
</X3D>
```

**Figure 14. Example of a touch-enhanced X3D file**

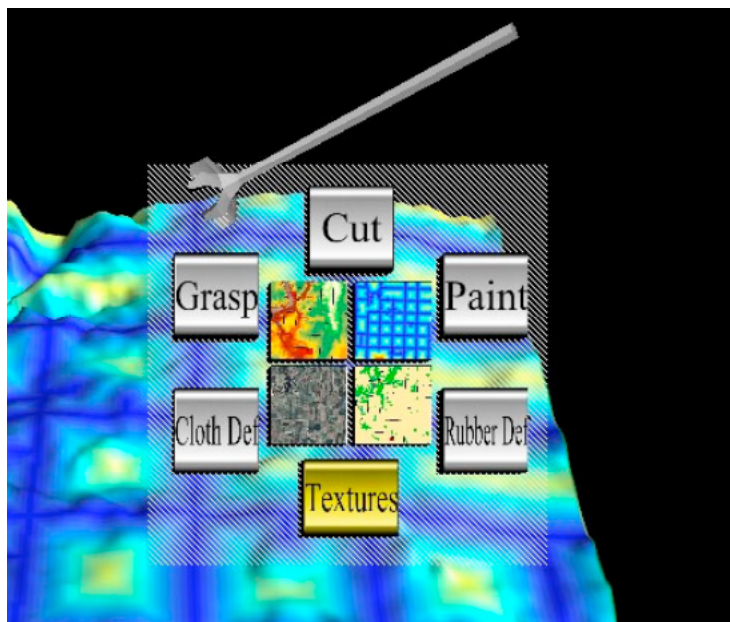
One of the most powerful aspects of H3D is its ability to use python scripts as part of the scene graph. Python scripts can read or change the attributes of the scenegraph nodes through predefined fields; they are key to implementing complex, event driven user interactions between the haptic device and the scenegraph objects. For example a python script could map the force (pressure) the user applies to a sphere to the visual transparency of the sphere and making it fade away as more and more force is applied.

For more complex or time critical functionality H3D also offers an open source C++ API, which can be used to extend the official H3D scene graph nodes via sub-classing. Coding in C++ also provides access to lower level APIs such as OpenGL and lower-level haptics APIs, and can be used to compile a custom H3D viewer with additional functionality. For M4-Geo, we created such a custom executable by extending several classes of H3D nodes (such as the indexed triangle set class) to support additional types of manipulation. This includes the addition of a topology database to localize real-time alterations in the mesh

connections and the inclusion of an octree spatial partition to improve performance when locating elements within the mesh.

We make extensive use of H3D's ability to incorporate customized 3D graphics routines, called shaders. Shader programs execute directly on the 3D graphics hardware (the GPU, Graphics Processing Unit) and provide substantial increases in flexibility and performance, especially on modern graphics cards. Our shaders are written in the GLSL high level shading language (based on OpenGL) and are integrated into the H3D scenegraph via shader nodes and routes. This makes it possible for the shader's effects to be directly tied to the Phantom and thus to the user interaction with the data. For example, the stylus orientation in 3D determines the direction of a virtual sun that lights the terrain; this is continuously transmitted to the shader running on the graphics card and changes the 3D illumination of the terrain, similar to hillshading. Slowly reorienting the stylus can bring out minute details of the terrain's morphology that are tied to a prevailing direction. Similarly, a "flashlight" that is connected to the stylus may show certain aspects of the data within its cone, similar to a magic lens. M4-Geo uses several other shader programs to efficiently visualize changes to the terrain as a result of user interaction, for example the visualization of the surface's gradients during its deformation or the application of color directly into the terrain's texture map during the interactive painting process. We use the Phantom's stylus for all interaction tasks, including interactions with the graphical user interface (GUI). The GUI type interactions, like changing the currently active virtual tool, changing the visualization parameters, such as the currently displayed raster or fine tuning the way a tool operates (e.g., changing the size or color of the virtual brush) are performed via a touch sensitive menu panel (Figure 15). When activated, it will appear in the 3-D environment close to the position

of the stylus and can be hidden again once the user has changed the desired setting, such as selecting a different type of virtual tool. Three- dimensional models are used to symbolize the type of the currently active tool, e.g. a brush symbolizes the paint tool, the "claw" shown in Figure 15 indicates that the user will grasp the model to move and rotate it.



**Figure 15. M4-Geo's touchable User Interface panel**

## 2.2. GIS Data preparation

A digital terrain model (DTM, also called digital elevation model or DEM) refers to a digital representation of a part of the Earth's surface derived from various survey techniques. Its representation in 3-D space is based on a set of 3-D points, either with x/y/z-positions (in a Cartesian system) or latitude/longitude/height (in a spherical system). These points are connected into a continuous surface and stored either as the centers of a regular grid or as the nodes of a triangle mesh (or TIN). Although using points with positions in 3-D space, a DTM typically does not contain any points with different z-coordinate (height) values but identical x/y-coordinate, which could represent an "overhang." The typical DTM surfaces are therefore

more precisely called 2.5-D surfaces. The process of creating a 3-D object from values contained in a 2-D height field (or height map) by offsetting them up or down in the elevation (z) direction, is sometimes called "rubber sheeting."

Geospatial raster and vector data that is to be used within the M4-Geo system needs to be converted into a X3D file format. We typically start with a raster DTM in ESRI's grid format; values of the raster's cells are used to create the additional "half- dimension." High resolution data for elevation and many other attributes, are now widely available at different raster resolutions, for example from the USGS Seamless Web site (USGS). Non-elevation raster data, such as USGS topographic maps, satellite imagery or land-use rasters, can be combined with topographic information by draping them over the DTM's geometry (shape). Vector GIS data, such as city outlines, roads, rivers, elevation contours, sample points, etc. may be added to provide visual guidance. To cope with possible differences of projections, extent, and resolutions we first load all the desired raster data as layers into ESRI's ArcGIS and make all the raster data fall into a common area. Additional raster layers may be the result of GIS spatial analysis of the DTM (e.g., slope, azimuth or curvature) or be the result of spatial modeling involving other GIS data (raster or vector data) within the area, for example a view shed raster for several observation points.

Once all the relevant GIS data has been assembled, the DTM is exported from ArcGIS to an ASCII file and then converted into a continuous triangles mesh (a X3D indexed triangle mesh) by a process involving custom written scripts. To prevent geometric distortion, the coordinates of the mesh's vertices are stored in geospatial coordinates (typically UTM). If the terrain data is not stored as a regular grid but rather as a set of irregularly spaced points, as is the case with some forms of LIDAR data, it is first

triangulated into a TIN and then converted into a X3D indexed triangle mesh. To make it possible to drape the additional GIS rasters on to the triangle mesh, each of the mesh's vertices is connected to a 2-D location within these images (texture coordinates). We convert these rasters into high resolution images (textures), with sizes typically ranging from 2048 x 2048 pixels to 4096 x 4096 pixels via ArcMap's "export map" function. Besides creating visual texture maps from the rasters, we can also create grayscale images that may be used as haptic texture maps. These haptic textures also draped over the terrain mesh but add certain haptic information at each of the raster's cell locations; for example, a low cost value may translate into low friction, a high cost value into high friction at a spatial location.

Alternatively, a small amount of additional force may be used to create the feeling of small depressions (low value) or bumps (high value) on top mesh's geometry, this is called a haptic displacement map. As the user moves the stylus over the terrain, such haptic textures may either augment already displayed (visual) information, or convey additional, non-visual information to the user.

It is possible to include GIS vector data in M4-Geo by either converting it into a raster and then into the texture maps or to create 3-D objects by placing the vector data on top of the terrain. While the raster conversion allows the use of vector data as visual landmarks (e.g., major roads or points representing key cities), the conversion into 3-D permits the use of gravitational haptic force effects. For examples, roads that have been converted in a set of 3-D polylines can be configured to attract the tip with a small force when brought within a certain distance, which can be useful in cases where snapping or computer guidance is desired. The final result of this data preparation procedure is the creation of a single "master" data file that contains the desired terrain mesh as indexed

triangle set and any other 3-D geometry within a common 3-D coordinate area. When the M4-Geo system is executed it loads this data file and sets up a series of scaling, rotation and translation operations that fit the data into a roughly 40 cm cube that is centered around the resting point of the Phantom's tip and display accordingly. If a stereo, co-located setup (Figure 13) is used, the cube and it's content will appear to float in front of the user.

### **3. Examples of 3-D interactions with geospatial data**

There has been much work during the last decade in the GIS community to move methods for viewing geospatial data from 2D, the electronic equivalent to paper maps, to 3D. Some of these 3D viewers can also be used with a true parallax based stereo system (i.e., a system that provide slightly different image for each eye), which some consider a hallmark of a virtual environment. For example ESRI's ArcScene/ArcGlobe and Google Earth can be seen as types of 3D visualization applications for GIS data and ArcScene can display this data in different forms of stereo.

While there has been some development to support the 3-D nature of these applications by offering specialized hardware for 3-D navigation, such as the 3DConnection's SpaceNavigator and SpaceExplorer, mouse and keyboard remain the most common devices for 3-D navigation and for interacting with the data as it is displayed in 3-D. This creates a fundamental user interface problem for interactions with 3-D data, including several typical GIS operations. We propose that, even with its technical limitations, point-haptics force feedback devices, like the Phantom, may help solve these problems by providing direct 3-D interaction with geospatial data. Their ability to "display" force may provide substantially more intuitive tools for certain 3-D tasks and permit the users to also use force input to

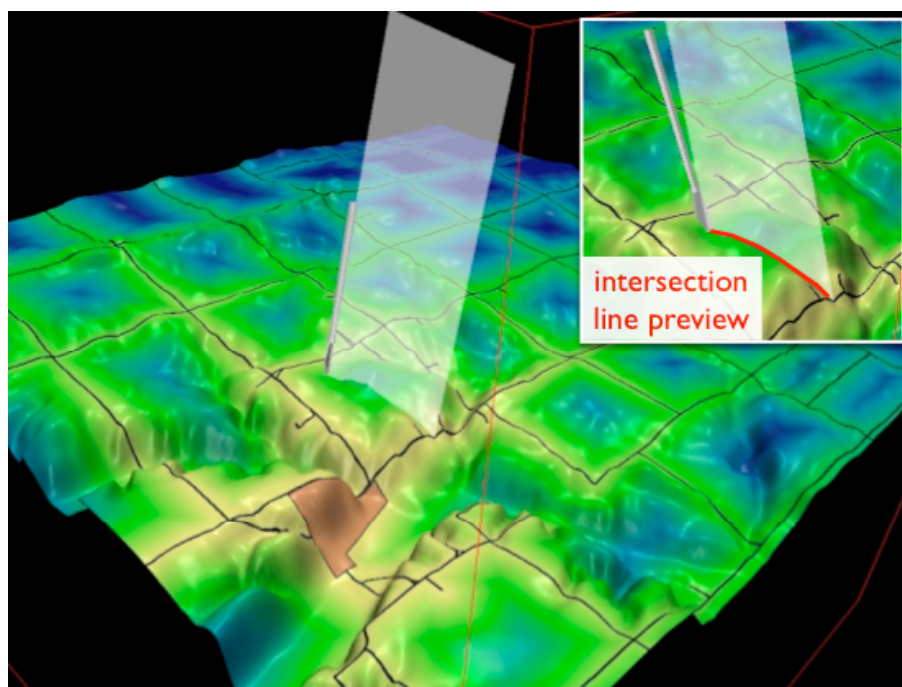


express their intent more efficiently. For example, using the Phantom permits us to create a virtual brush tool in which the amount of paint is tied to the amount of force used during painting.

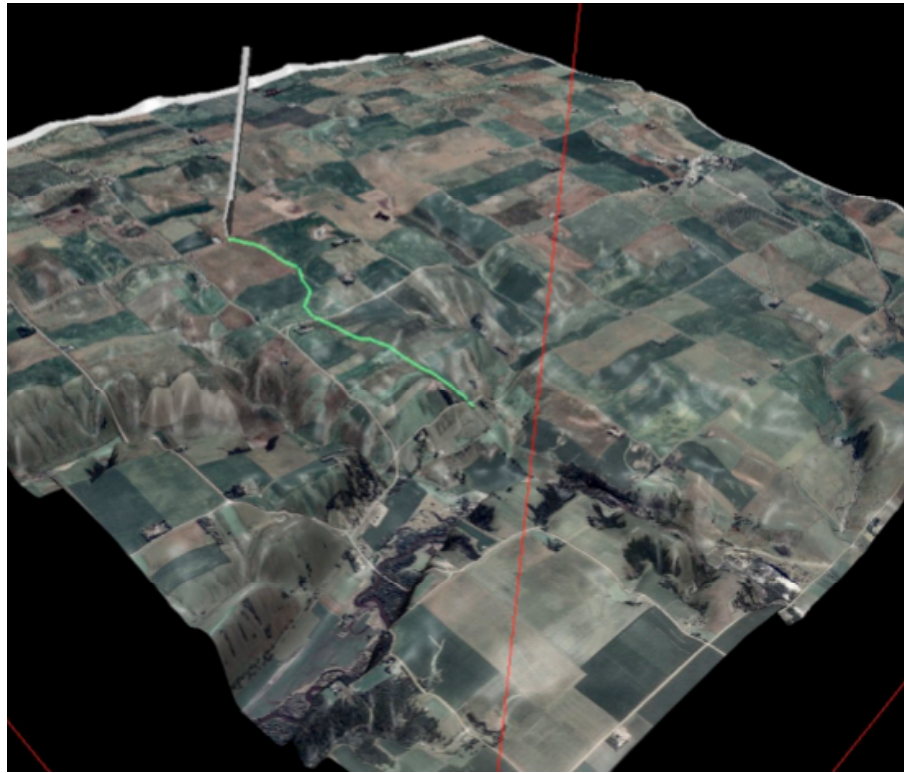
### **3.1. Digitizing line features on a 3-D surface**

Digitizing a polyline or a polygon on a terrain map are common GIS tasks; for example, digitizing a line may be used to create spatial information about the course of a newly planned road or about a geologic fault. The M4-Geo system offers a virtual pen tool for digitizing lines directly onto a 3-D model thus seamlessly combining visual and haptic feedback. The tool's interaction is modeled on the act of drawing on a physical 3-D shape with a pencil. Moving the virtual pencil to a desired location and then clicking the stylus button results in a succession of line segments, similar to digitizing on a 2-D map with a mouse or a digitizing tablet. However, as the 3-D surface may rise and fall between the two points; M4 uses a plane (rectangle) that is oriented in 3-D by the stylus to visualize the rise and fall of the terrain between the two points. In addition to conveying the position of the pencil tool's tip on the terrain surface and with the context of geospatial data draped on the surface, the force feedback of the system informs the user when the tip is in contact with the terrain. As the force feedback keeps the tip precisely on the surface mesh, rubbing the tool over the surface conveys its morphology. Feeling these possibly minute bumps and grooves augments the user's visual perception and helps to find the desired location from which a new polyline segment should be added. While the user moves the stylus around on the surface, the predicted intersection of the line with the mesh is constantly shown as a graphical preview.

Figure 16 shows the intersection of this line segment with the terrain via a white rectangle, which is anchored at the last point and is slaved to the tip as it moves over the terrain. As the stylus provides a directional vector, the user can change this predicted line in subtle ways by reorienting the stylus around the tip and thus changing the intersection angle of the white rectangle with the terrain (inset of Figure 16). Once the user is satisfied with this graphical preview, the new line segment is added to the mesh (Figure 17). In some cases it could be desirable to also convey more information about the touched location during the digitizing process. For example, the user may want to augment the aforementioned haptic feedback about the terrain's shape by also conveying the cost raster, shown draped on the DTM in Figure 16 via friction. This provides a secondary stream of strictly local data that may be useful during certain planning tasks, such as digitizing a road while viewing the air photo shown in Figure 17.



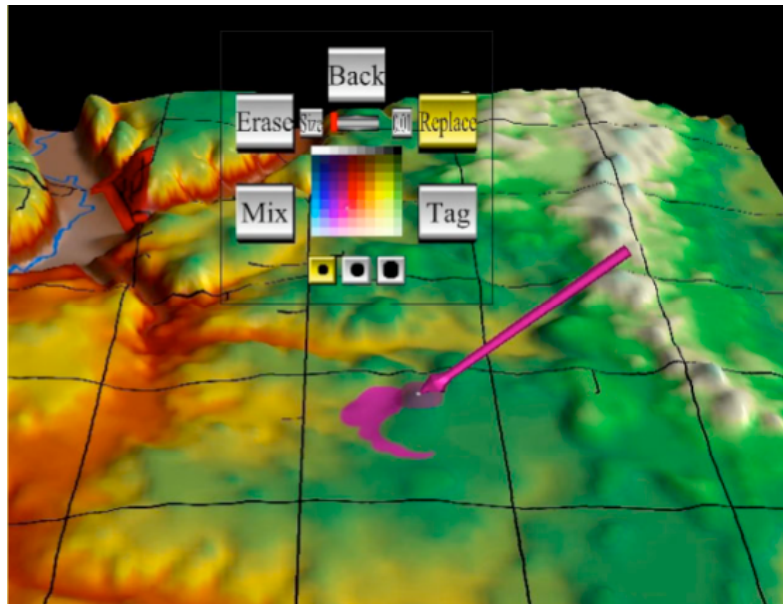
**Figure 16. Using the stylus to orient a rectangle that shows the predicted intersection of a new segment with the terrain**



**Figure 17. The result of digitizing (draping) a polyline on the mesh**

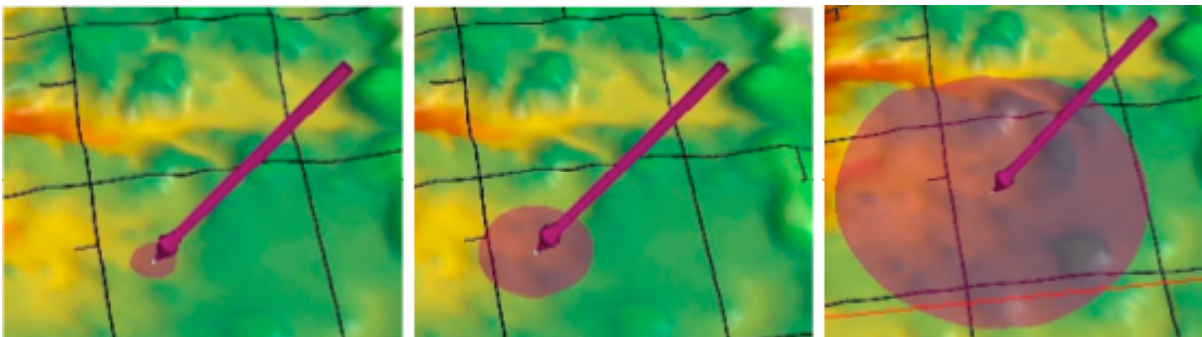
### **3.2. Touch pressure based spatial selection**

Besides digitizing polylines, which uses a drawing/pencil metaphor, M4-Geo implements the spatial selection of raster data via a painting/brush metaphor. The virtual paint tool can be used to select parts of the terrain by applying a color to a separate selection layer that overlays the other raster data layers (textures), as is shown in Figure 18. This selection can later be used to perform a subsequent GIS operation only within its color, similar to a zone in ArcGIS.



**Figure 18. Manual spatial selection of parts of the terrain surface using the interactive paint tool**

To give the user more flexibility, the input force is translated into a brush size, i.e., pressing harder against the terrain surface increases the radius of the paint. A preview mode shows this predicted "footprint" in a semi-transparent color, the images in Figure 19 (left to right) show the effect of increasing the force applied to the object. When the user is satisfied with the preview, pressing the stylus button begins applying color; this produces a smooth paint trail as the tip is moved over the terrain. In addition to painting, the tool can be switched to an erase mode to remove previously applied paint.

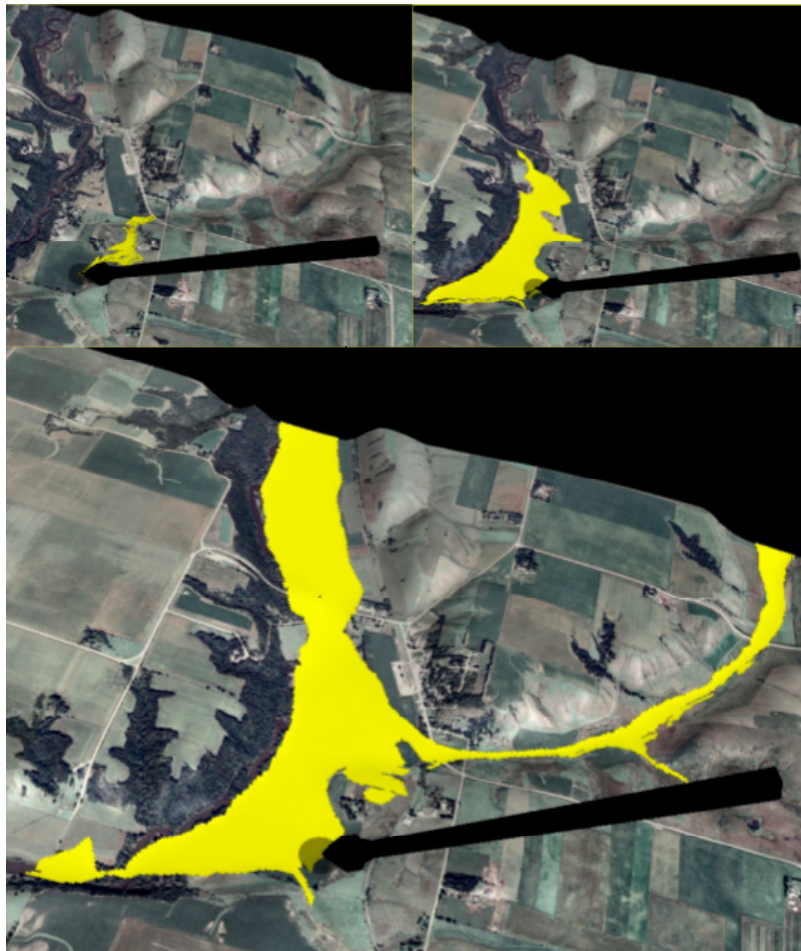


**Figure 19. Adjusting the size of the paint tool's brush by varying the force applied to a location on the terrain mesh**

### 3.3. Spatial selection via slope based flooding

This variation of the paint tool selects a continuous region around the tip (yellow pixels in Figure 20), depending on the position of the tip and the force applied by the user. Beginning at the tip, a flooding algorithm expands this set of pixels outwards, depending on the terrain's slope beneath each new pixel. The flooding terminates where the slope reaches a certain threshold value. This threshold value is again dependent on the force the user exerts on the terrain, i.e., the harder the tip is pressed onto the terrain, the larger the selected region tends to become.

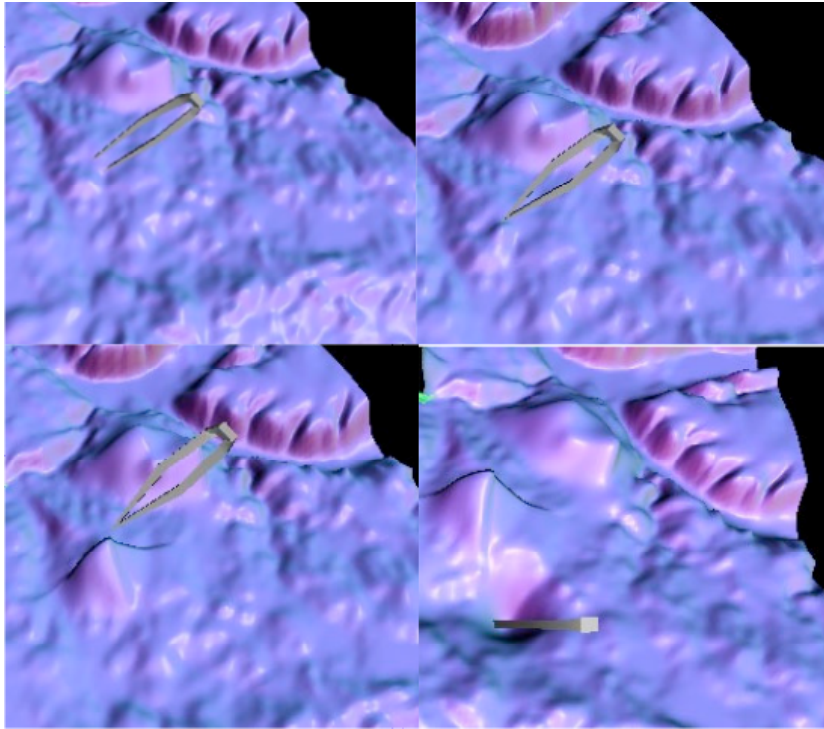
The effect of increasing the force can be seen in Figure 20; the tip's location within a river valley remains constant and the forces increases steadily from the upper left image to the lower image. While the current flooding algorithm is very simplistic it demonstrates how real-time spatial computations can be integrated into touch-based interactions. The current algorithm can be used to investigate the flooding potential for locations that might be part of a river's floodplain.



**Figure 20. Effects of using the flooding selection tool with increasing amounts of force**

### **3.4. Interactive deforming of the terrain surface**

Most of the typical user interactions in 3-D visualization systems relate to changing aspects of the appearance of the data. However, the deformation tool permits the user to affect the terrain's geometry on the vertex level. Figure 21 shows the use of the deformation tool, the user touches the surface at a location (upper-left image), holds down the buttons and moves the tip in any direction.



**Figure 21. Real time terrain deformation**

In the upper-right image, the tip was moved slightly upwards. M4 Geo's implementation of the Generalized ChainMail deformation algorithm (Gibson, 1997; Li & Brodlie, 2003) calculates how far certain vertices of the underlying triangle mesh have to be moved in order to create the geometry of a new hill (lower-left image) or a depression (lower-right image) within the terrain mesh. As the stylus tip continues to move farther away from the initial location, this real-time algorithm will move an increasing number of vertices around in a way that simulates the deformation of a cloth-like material and that distributes the vertices evenly. During the deformation, the terrain's visual appearance is adjusted and the force the user perceives is calculated to indicate the degree of deformation; the farther the tip moves from the origin of the deformation, the larger the deformation becomes, and the larger the counter force that the user feels. The images shown in Figure 21 also demonstrate the effect of a special coloring mode, which temporarily colors the terrain according to its

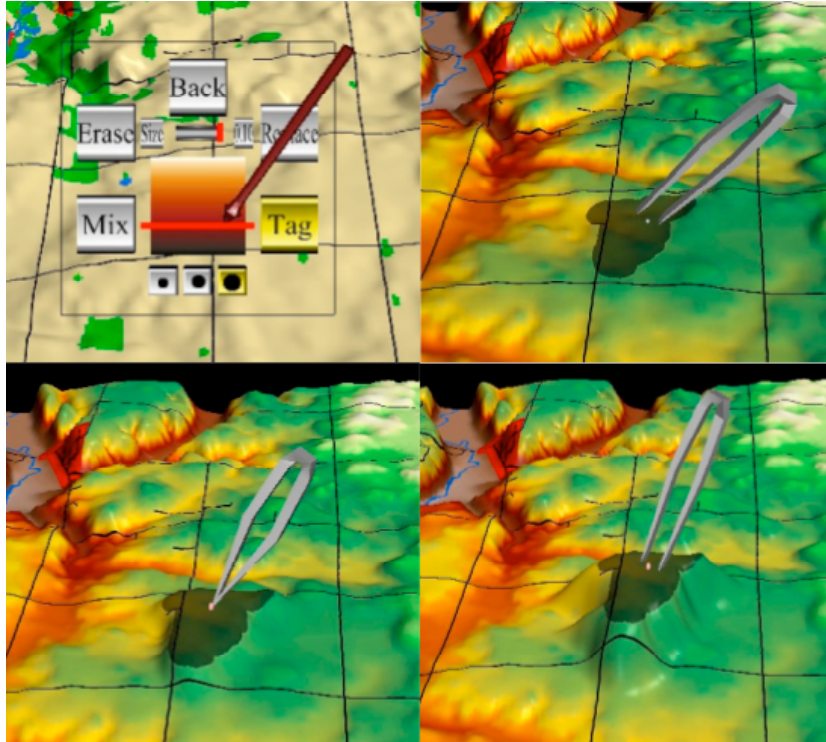
direction (normal). This accentuates the terrain's gradients and enhances the user's perception of the details of the terrain's morphology. As it is implemented through a per-pixel shader, it shows the changes to the morphology created by the use of the deformation tool immediately and helps the user adjust the deformation.

The nature of the deformation is defined by several parameters that the ChainMail algorithm uses to decide which vertices to move and by how much. These parameters can be used to tweak the deformation behavior, i.e. the type of shape the terrain takes in response to moving the stylus tip around. In M4-Geo, these parameters are applied to the vertices of the mesh and thus can be thought of as defining the material properties of the mesh with regard to deformation. For example, moving the tip when deforming a conceptually "soft" terrain results in a steep, "pointy" hill, whereas the identical tip movement for deforming a conceptually "hard" terrain would result in rounder hill. While this degree of softness is initially set globally for all mesh vertices to a medium-soft material, the user can re-define the mesh's type of material (very hard to very soft) locally by painting a special material color onto the terrain.

The upper left image in Figure 22 shows the selection of this material color from a black-red-white color ramp, which was chosen as it mimics the color of iron during heating; black suggest cold and therefore impossible to deform, red and finally white suggest higher temperatures and hence material that is increasingly soft and malleable. The black material color can thus be used to "freeze" parts of the terrain and to prevent any deformation. The images shown in Figure 22 (upper right, lower left, lower right) demonstrate the effect of painting parts of the terrain with this black material color and then moving a point inside this black area upwards to create a hill. Unlike the deformations shown in Figure 21, the



morphology of the terrain painted in black is preserved. Because the digitizing described earlier integrates the topology of its lines into the mesh, they can be used as break lines that limit the spatial extent of the deformation process.



**Figure 22. Constrained terrain deformation by first painting parts of the terrains to become harder (black)**

The combination of assigning specific deformation property to certain parts of the terrain (via manually painting or via the flooding tool) and then guiding the deformation of these parts interactively with the support of force feedback provides a radically different way of modifying terrain in the context of digital landscape design and geotechnical planning.

#### **4. Conclusions and future work**

We have presented M4-Geo, a prototype software system that implements several examples of touch-enabled 3-D interactions with geospatial data. The system is written with

the open source H3D API and uses a point haptics force feedback device as a virtual fingertip for the multi-sensory exploration of terrain data and raster data draped onto its geometry. The system's interactions focus is initially on direct GIS interaction tasks, where the use of a 3-D force feedback device promises significant advantages over the use of mouse and keyboard. These tasks include digitizing of line data, spatial selection via painting and flooding, and real-time deformation of surface meshes. We have conducted an informal evaluation of these tasks with local domain experts, they feedback has been very positive and indicates that a larger multi-sensory GIS system can provide improvements over current methods in the areas of landscape architecture and geotechnical engineering. For example, a landscape architect could create a 3D digital model for a new golf course; or a geotechnical engineer could explore terrain modification for the prevention of local flooding. Currently, the process of assembling and transforming the different types of geospatial data still requires a chain of import/export procedures and handwritten scripts. This process needs to be simplified and should be as automated as possible to widen the number of potential users. Ideally, it should be possible to export GIS layers from GIS systems (such as ArcGIS or GRASS) into X3D/H3D scene, similar to the VRML export offered in ArcScene.

The implementation of a dual-handed system, in which two Phantoms are used, is in the final stages of testing. Preliminary testing shows that, at the very least, the second hand is useful to seamlessly grasp, reposition and rotate the terrain while it is manipulated by the virtual tool; however, there are also several intriguing potential uses of two tools together; for example, one hand could deform the terrain while the other hand changes the deformation properties with a "blow torch" tool, that can soften (or harden) parts of the area under deformation. The use of two Phantoms may also provide interesting possibilities for multi-

user interaction and for the collaborative modification of terrain. A more formal evaluation of certain usability aspects of the system using students is currently underway.

We have also explored the addition of sound as an additional visualization modality. Similar to the communication of data values via friction, a series of sounds can be used to convey data about the currently touched location, for example its elevation. Preliminary results suggest that the changes of pitch in such an elevation melody created by moving the tip over the terrain can help the detection of certain terrain features.

While our research currently focuses on the areas of data exploration, there are several other areas of application into which the system could extend to. One such area may be the multimodal presentation of geospatial information to visually impaired students (Golledge, Rice, & Jacobson, 2006). The use as an educational tool in courses teaching geomorphology has also been suggested.

A variation of the M4 system may become useful for exploring high-dimensional (multi attribute) data and non-spatial data in 3D information visualization context (Ward & Yang, 2004), where it could improve the selection of subsets of data via coloring and modifying the way the data is visualized (attribute-based distortion). Force effects could be added to augment the user's knowledge about the data, e.g., magnetic force effects could be used to convey data relationships (connections) between data elements and motion (vibration) could be used to convey uncertainty.

## 5. Acknowledgements

We acknowledge the helpful feedback and suggestions given by the anonymous reviewers. Part of the work was funded by NSF grant IIS- 534681.

## **CHAPTER 4. SUPPORTING INTERACTIVE HAPTIC SHAPING OF 3D GEOLOGIC SURFACES WITH DEFORMATION PROPERTY PAINTING**

Modified from a paper accepted for publication in  
Proceedings of Eurographics '09, March 2009, Munich, Germany.

Adam Faeth and Chris Harding

### **Abstract**

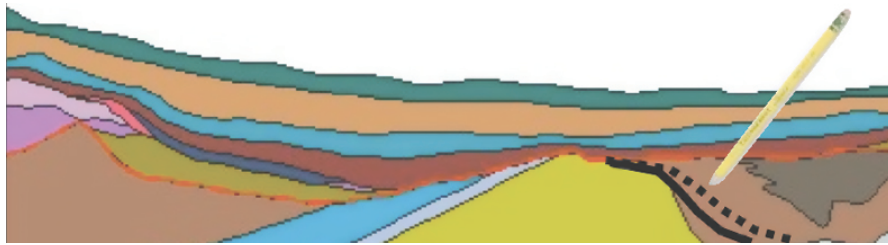
Meshes made from 3D points are used to represent many important geoscience concepts such as the surface of the earth (topography), rock strata (horizons) and faults. When creating a complex computer model, a geoscientist may need to directly manipulate the shape of such a surface to reflect conflicting information from additional data sources. We present a method that allows a geoscientist to precisely interact with these surfaces by painting the surface with colors that represent its local “malleability” (deformation property values) and to interactively deform this surface into the desired shape. Since the deformation property values create an inhomogeneous mesh, we explore several adaptations of the Generalized ChainMail algorithm to allow it to support inhomogeneous mesh deformation at interactive rates. By extending the Generalized ChainMail algorithm to use a FIFO-preserving priority queue to store the candidates awaiting deformation, we can perform inhomogeneous mesh deformations at interactive rates. We present a comparison of this FIFO-preserving priority queue with a simple priority queue and with a simple FIFO queue. Both painting and deforming are part of a larger system for haptic-visual mesh manipulation,

in which we explore combinations of bimanual, touch-enhanced virtual tools for interactions with 3D geoscience data.

## 1. Introduction

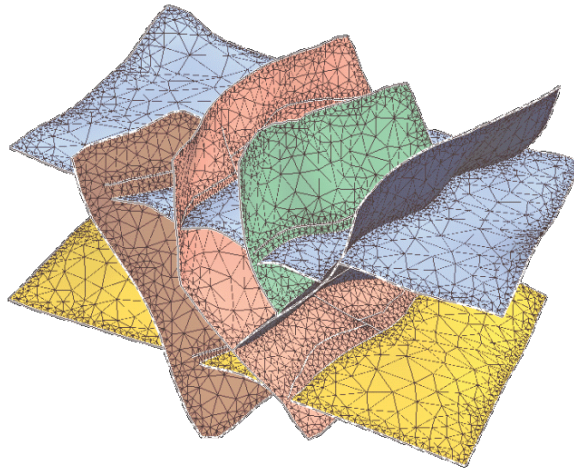
Much of a geoscientist's work focuses around creating 3D models of the Earth's structure from incomplete and imperfect sources of 3D data. For example, it is important to outline the geometry of different rock units within the 3D subsurface by defining horizontal and vertical boundaries, as such structural geologic models form the basis for many academic and commercial projects involving resource extraction, groundwater flow modeling, geotechnical engineering and environmental remediation.

While computational methods for 3D interpolation and 3D scientific visualization play an important role in the creation of 3D geologic models, the expertise and intuition of the project's geologist are equally vital in developing complex models of the subsurface. The nature of geologic data often makes it necessary for the geologist of to "fill in the blanks" in the data by postulating a possible 3D solution from their knowledge of geologic processes (e.g., sedimentation, tectonics). 3D computer systems for geological modeling (such as GoCad or Petrel) are typically focused on offering increasingly powerful 3D graphics but are still based on using 2D mouse interaction with 3D data. Many geologist resort to a comfortable paper and pencil interface for 2D sketches of horizontal or vertical cuts (profiles) to develop mental models of rock structures (Figure 23).



**Figure 23. Vertical cut through a geologic model of rock layers with potential variations of a slope angle**

However, while it is easy to correct the angle of a vertical separation (fault) with this 2D interaction, the 3D equivalent is considerably more difficult, in part due to the lack of an integrated and easy to use 3D user interface. For example, a geologist may want to adjust the green fault shown in Figure 24 by bending its upper part like a sheet of metal but without affecting the part below the blue horizon.



**Figure 24. Example of horizontal rock boundaries (horizons) cut by vertical fault planes**

Our work aims to create such an intuitive “3D canvas” for 3D models that not only overcomes the artificial nature of using 2D mouse interactions to express an intrinsically 3D interaction, but also uses force feedback for freehand cutting, painting and deformations in real time.

In this paper we focus on a part of this larger process that enables a geoscientist to precisely and directly deform these surfaces. The first step uses a paint metaphor to define inhomogeneous deformation property values (“malleability”) on the surface, effectively creating harder, softer and even frozen regions. In particular, we deal with an algorithm for efficiently calculating the shape of a mesh during deformation if the mesh’s deformation property values are not homogeneous, or uniform, throughout the mesh but varies locally as an effect of the user’s painting process. We extend the underlying deformation algorithm to seamlessly incorporate these localized deformation property settings into a cloth-like, interactive deformation process. Note that we do not attempt to accurately simulate specific realistic material properties, such as a specific type of tissue; we employ deformation as a 3D interaction technique to enable the user to more efficiently and naturally translate a mental model of geology into a 3D computer model. Since this paint/deform combination can also be performed simultaneously, i.e., to soften or harden the model while it is being deformed, we also explore the potential for bimanual (two-handed) manipulation of 3D models.

## 2. Related work

Previous research investigated the use of haptic devices for painting and deforming surfaces, bimanual interaction, and interactive deformation. We previously presented higher-level aspects of our system for the manipulation of meshes with 3D graphics, haptics and sound, including the painting and interactive cutting of meshes and an earlier version of the deformation algorithm (Faeth, Oren, & Harding, 2008; Faeth, Oren, Sheller, Godinez, & Harding, 2008). The work presented here focuses on the integration of a paint metaphor into

the deformation algorithm and provides significantly more detail on its implementations and performance.

## 2.1. Haptics interaction with 3D models

SensAble's family of Phantom force feedback devices, shown in Figure 25, offers one method of enhancing 3D interactions with a form of touch (Massie & Salisbury, 1994). These grounded haptic devices provide active, kinesthetic feedback at the tip of a stylus. Although the overall haptic experience conveyed by the Phantom reduces the haptic experience to a single point, it is well suited as an augmentation of a 3D graphics environment with touch-based interactions.



**Figure 25. Phantom haptic force feedback device**

Previous research has explored the ability to paint and deform a 3D model with a Phantom haptics device in related contexts. The inTouch system allowed users to paint colors on a 3D model and perform a simple fixed-heuristic deformation (Gregory, et al., 2000). Foskey et al. extended that system into an art tool for painting textures onto the 3D model (2002).



McDonnell et al. explore a volumetric framework using a control lattice; one interaction of that framework is “stiffness painting,” which allows the user to paint a stiffness value onto the nodes of a coarser control lattice with a haptics device (2001). Cani et al. also present several approaches to manipulating virtual clay, and suggest that some approaches would benefit from using more than a single device (2006). These systems either operate on a control structure rather than directly on the data points, or operate on homogenous models.

## **2.2. Bimanual interaction**

Theoretical models for bimanual user interaction make a distinction between symmetric and asymmetric interaction (Guiard, 1987). The bimanual interactions presented in this paper are all asymmetric interactions, where a dominant hand and a nondominant hand perform separate functions. Other authors have explored approaches for bimanual interactions with 3D shapes, both in 2D interactions (Owen, et al., 2005), and in 3D interactions (Grossman, et al., 2001; Hinckley, et al., 1998). However, manipulations using multiple force-feedback devices, and bimanual manipulations for geoscience-specific tasks remain largely unexplored.

## **2.3. ChainMail deformation**

Gibson developed the original ChainMail algorithm to enable fast deformation of 2D and 3D objects containing hundreds of thousands of nodes (1997). The deformation approximates rigid, deformable, elastic, and plastic materials with three deformation parameters: stretch, compression and shear. It provides a fast, real-time deformation method

in which a limited number of points are moved by a certain distance depending on the displacement of an initial vertex.

Since its introduction, follow-up work on ChainMail diverged. In the Enhanced ChainMail algorithm, Schill et al. adapted the algorithm to operate on inhomogeneous meshes, meshes where the deformation property values could be different for each node on the mesh, however, Enhanced ChainMail was developed for quadrilateral meshes (1998). Li and Brodlie later developed the Generalized ChainMail algorithm to operate on arbitrary meshes in 3D (2003). This extension provides a framework for ChainMail deformation of tetrahedral and triangular meshes, however, Generalized ChainMail did not include the ability to deform inhomogeneous meshes.

In this paper, we present a technique that extends the Generalized ChainMail algorithm towards functionality described in the Enhanced ChainMail algorithm. This technique allows the deformation of tetrahedral meshes with inhomogeneous (nonuniform) deformation properties and includes a way to directly change these deformation property settings locally by painting them onto the mesh before or during the deformation. We also present an example of how this type of direct interaction can be used in a geoscience setting.

### **3. Deformation property painting and deforming of 3D meshes**

Our method for precisely manipulating a surface mesh consists of two actions that are typically performed in sequence but may also be performed simultaneously. The geoscientist first assigns local deformation property values to specific parts of the mesh by painting them with the desired malleability values with a virtual paint brush tool. For this step, the stylus of the haptic device turns into a virtual 3D brush, which the user can move along the surface of

the object to feel the geometry of the surface and to steady the hand. This type of painting tool allows the geoscientist to assign inhomogeneous deformation properties values across the mesh. The inhomogeneous properties can set certain parts to deform more or less easily (react as soft or hard) or to protect them from deformation (react frozen).

In the second action, the geoscientist switches from the paint tool to a deformation tool, grasps the mesh at a certain vertex and gradually displaces it by pulling or pushing this grasped vertex in any direction, The vertex's displacement is continuously propagated to nearby vertices, leading to changes in displacement throughout the mesh. Internally, the deformation is defined with three parameters, a stretch value, a compression value and a shear value, which are stored inside each of the mesh's vertices.

During the deformation process, the geoscientist observes the visual effect of the deformation as the deformation propagates from that part to affect the object. They also feel a haptic effect equal to the magnitude of the deformation and can bump into other objects in the scene. These types of visual and haptic feedback during deformation guide the geoscientist towards their goal of deforming a part of a surface to align it with other 3D objects.

From the time the user initiates the deformation until they release it, the system performs a continuous simulation at interactive rates. During each frame, the deformer calculates the result of displacing the point of contact from its initial position to its current position (given by the Phantom's tip in 3D space). This is vital for intuitive modeling because it allows the user to seamlessly revert changes to the shape of the mesh if they deform too far. If the user moves the stylus too far while performing the deformation, they

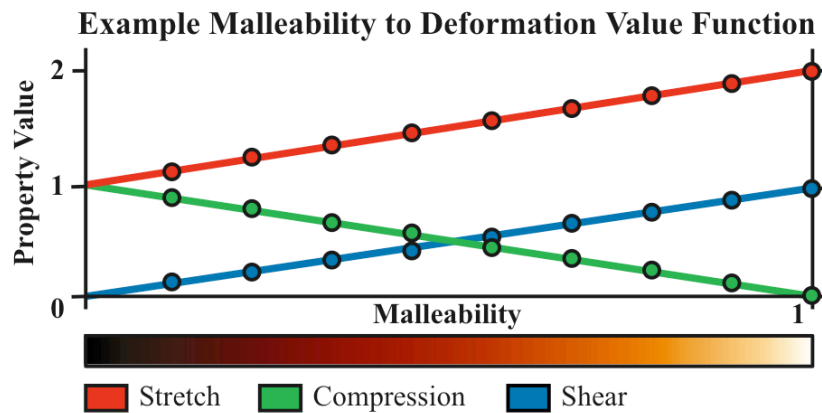
may simply move the point of contact closer to its initial position, and revert to the less severe deformation.

### 3.1. Visualizing the deformation property values

The power of deformation property painting comes from the fact that it allows the geoscientists to plan the deformation in a way that supports the transfer of their geoscience knowledge into the 3D computer model. Painting is a common metaphor for changing the visual appearance of 3D objects. Deformation property painting affects the object's underlying deformation properties instead of color values, and those properties then govern the deformation behavior of specific areas of the mesh. For example, it might be desirable to set parts of a model to deform like a soft, rubber surface, other parts to deform like metal sheets, and other parts to remain rigid. However, the act of material painting requires an immediate visual feedback as well. The geoscientist needs to know ahead of time what the deformation property values are at each point on the object to be able to “design” the distribution of the deformation accordingly.

One obvious solution maps the three deformation parameters, stretch, compression and shear values, to the three RGB primary colors: red, blue and green, to visibly convey information about the mesh to the user. However, the choice of which color to map to which deformation parameter is arbitrary. Even worse, the properties often lie on different numeric ranges. Compression generally lies on the range between [0.0, 1.0], while stretch might be useful to define between [1.0, 2.0] or [1.0, 200.0] depending on the context of the deformation. This makes the use of a RGB color mapping problematic.

An alternative to the color mapping is to use a single value to represent the concept of malleability, ranging from 0.0 to 1.0, and to use a transfer function to convert this single malleability value to the three deformation property values. The advantage of using this grayscale malleability value is that it can be mapped to a color scale that conveys the meaning of deformation more effectively than a translated RGB scheme. We have explored the use of a heat map color scale that mimics the color of iron at increasing temperatures. Our heat map ranges from dark black, corresponding to a rigid cold iron, through warmer reds and oranges, to white-hot iron for the most malleable. This provides a clearer connection between color and the expected deformation behavior at different areas of the object – a white-colored hot area will react in a softer manner than a red-colored moderately-hot area, and a black-colored cold area will preserve its shape.



**Figure 26. Example linear transfer functions from malleability to deformation property values**

The transfer function underlying the heat map allows the user to control how the malleability value range maps onto the deformation property values. Figure 26 shows a simple configuration of transfer functions that map the malleability value to the three deformation property values (the red, green, and blue lines). The example shown is a set of

linear transfer functions, which creates a believable progression from hard to soft malleability. However, the use of nonlinear or spline transfer functions could help to fine tune the mapping to suit the application or to deliberately introduce jumps or steps.

### **3.2. Bimanual interaction**

In addition to sequentially switching between using the paint and using the deform tools, the user may employ a second haptic device to either grasp the object or paint and deform the object simultaneously. Using a tool to grasp the object with the second device, the user may reposition and reorient the model with six degrees of freedom. This asymmetrical bimanual interaction allows the second hand to provide a frame of reference for the task of the primary hand. The user can use their second hand to reposition or reorient the object to make the painting or deformation task easier to perform.

The user may also use the paint and deform tools simultaneously. This allows the user to change the properties of the deformation while performing the deformation and immediately view the effect of the modified deformation property values. For example, if a user notices that the deformation creates an undesirable shape in certain parts of the model, they could apply a different malleability value on this part and the deformation changes accordingly. The user could also use the paint tool to reduce the malleability of a region, or harden it, to preserve its location relative to its neighbors after starting a deformation.

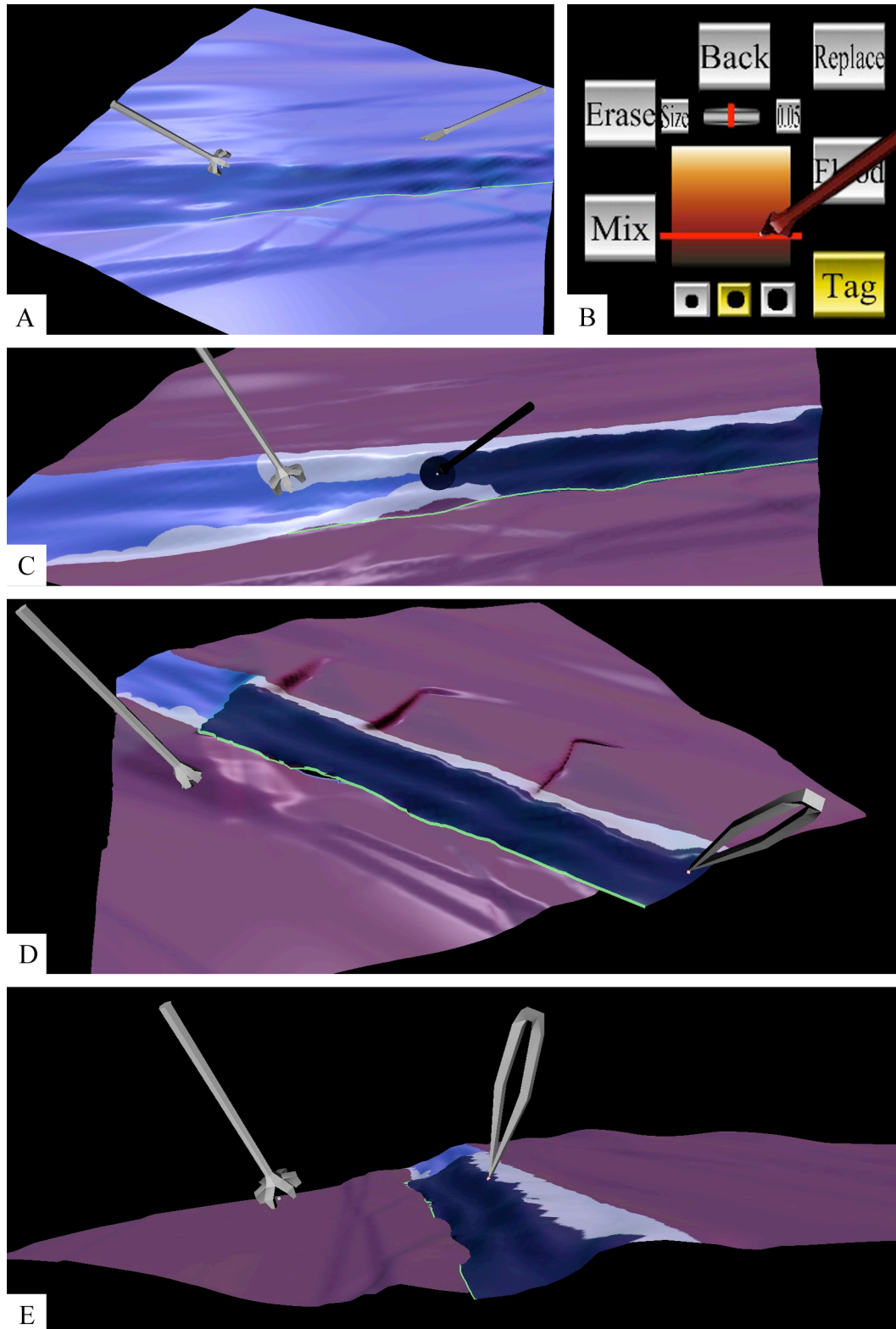
### **3.3. Example of using deformation painting in a geoscience context**

The following will illustrate an example of deformation property painting and subsequent deformation in the context of adjusting a horizontal rock boundary (horizon) to

better model a vertical “step” (fault). To aid clarity, we deliberately use a simplistic 3D geological subsurface model and ignore many of the complicating factors encountered in real-world cases.

We assume that the geologist has generated an approximation of a horizontal boundary between two types of sedimentary rock (horizon) from one data set and, extrapolating from the Earth’s surface downwards, has created a mental model of where the step should be located. Looking at the horizon in Figure 27A, she determines that the position and shape of the step needs to be moved to be correct. Using the mesh as a 3D canvas, she prepares its deformation by painting three different deformation colors on it. She selects a malleability value to paint by bringing up the 3D haptic user interface underneath the stylus, shown in Figure 27B. She first hits a red color on the color selection palette with the virtual brush to permit gradual deformations of the mesh. She then hits the color black on the palette and paints the vertical region (corresponding to the fault plane) black to ensure that its shape remains constant (Figure 27C). She then presses on the color palette again but now selects white. She paints a white border around the vertical region to enable a very abrupt change of shape.

The geologist is now able to shape the vertical part of the 3D model. Using the Phantom’s stylus, she touches the mesh directly with the deform tool (tweezers), holds down the stylus’s button and moves the tip laterally up or down, thus creating an offset from the initial position. The deformation algorithm translates the offset into a mesh deformation that honors the underlying deformation properties indicated by the colors painted on the mesh earlier (Figure 27D). Because this deformation is updated in real time, the geologist can adjust it until it conforms to the desired shape.

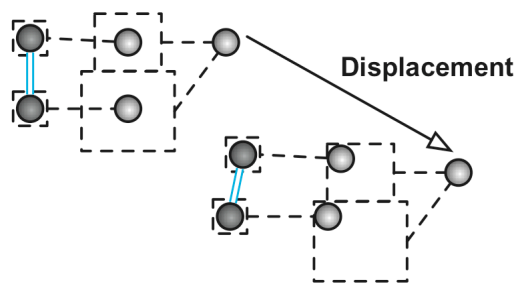


**Figure 27 (A-E). Applying deformation parameters to a subsurface mesh, and interactive deformation**



## 4. ChainMail algorithm for inhomogeneous mesh deformation

The problem with using the Generalized ChainMail algorithm when deforming an object with inhomogeneous deformation properties is that it can allow malleable points to determine the final position of any stiffer points. Instead of preserving a constant relative position to neighboring points in the rigid areas, it causes that relative position to change. This results in ripples through the region that is supposed to remain rigid, as the deformation propagates from malleable points to stiff points. The Generalized ChainMail algorithm does not differentiate between points based on malleability, but instead propagates using a tree traversal or an outward spiral. Before detailing the necessary modifications to the Generalized ChainMail algorithm, it is useful to explain the operation of the algorithm.

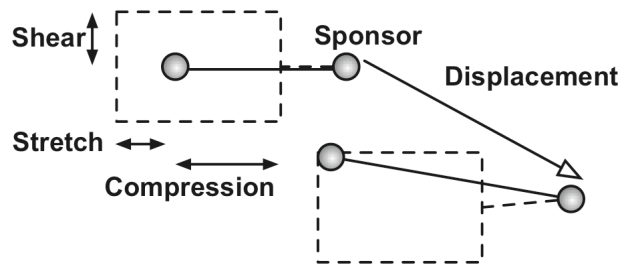


**Figure 28. A ChainMail displacement that propagates from the malleable grey nodes to the rigid black nodes fails to preserve the fixed displacement between the black nodes.**

### 4.1. Details of Generalized ChainMail

The ChainMail algorithm was designed for fast simulation of deformable objects; it focuses on achieving interactive rates to provide the user with immediate feedback about the shape of the deformation. To keep the complexity of the algorithm low, each node is only moved once and not every node needs to be analyzed. Even though a given node often has several neighboring nodes that might affect its movement, only one neighbor sponsors the movement of that node. A node will sponsor the movement of its neighbors only if it is

displaced by the deformation. This means that the algorithm only iterates through the region of nodes affected by the deformation. The deformation spreads outward from the point of contact. Each node displaced by the deformation drags along an imaginary box around each of its connected neighbors, displacing the neighbors as they are caught by sides of the imaginary box. The propagation of the deformation terminates when a given node already lies within the imaginary box, called the valid region.



**Figure 29. Valid region of a point before and after a ChainMail deformation displaces its sponsor**

The valid region is a bounding box around each candidate node for deformation that is influenced by both the sponsoring node and the deformation parameter values. The origin of the bounding box is the original displacement of the candidate, measured from its sponsor. If the sponsor is displaced by the deformation, the origin of the valid region is displaced by the same amount. The dimensions of the bounding box are defined by the deformation parameter values. The stretch parameter elongates the box along each component axis of the displacement. The compression parameter determines how closely the bounding box may approach a sponsoring node. The shear parameter adds a proportion of the perpendicular components of the original displacement to each dimension of the bounding box. Shear allows a node with a displacement with only a z-axis component to move on the x or y axis to absorb the deformation. Note that the origin of the valid region is only at the center of the valid region if the stretch and compression values are the same distance from 1.0.

$$VR_{\text{origin}} = VR_{\text{center}} \text{ iff. stretch} = 2.0 - \text{compression} \quad (1)$$

Since each displaced node may sponsor a number of neighboring nodes, it is necessary to store the candidates awaiting analysis in a data structure. Li and Brodlie suggest the use of a tree or a FIFO queue for Generalized ChainMail (2003). The choice of structure influences the propagation of the deformation, with the tree structure producing vein-like propagations, and the queue producing a smoother spiral propagation.

## 4.2. Malleability gradient propagation

The ChainMail algorithm must preserve the relative displacement between sponsor and neighbor nodes in the rigid portions of the mesh. Enhanced ChainMail satisfies this requirement by calculating the constraint violation of each neighboring node at the time when the algorithm displaces a node (Schill, et al., 1998). The constraint violation is the distance between a neighbor and the closest point in its valid region. The Enhanced ChainMail algorithm then displaces the neighbor with the largest constraint violation and moves on to process its neighbors.

An alternate method to prevent malleable nodes from sponsoring stiffer nodes is to require that a deformation propagate down a malleability gradient, through stiff nodes before spreading to more malleable nodes. This method also preserves the relative displacement between neighboring rigid nodes by ensuring that a node may only sponsor the displacement of an equal or more malleable node.

The exception to this rule is that a malleable node can sponsor a stiffer node for deformation if there is no stiffer candidate awaiting deformation in the queue. This exception prevents a deformation that originates in a softer region from stretching infinitely along the

border of a stiffer region; the first stiff node encountered will propagate the deformation through the rest of the stiff region.

### 4.3. Priority queue for deformation candidates

One straightforward approach to enforcing malleability gradient propagation is to use a priority queue instead of a FIFO queue or a tree for the candidates that have been sponsored for deformation but not yet analyzed. The priority queue would prioritize candidates with lower malleability values over candidates with higher malleability values. This would implicitly prioritize the nodes most affected by a deformation without having to calculate the potential displacement of each node before moving the next as proposed by Schill et al. in 1998, since the valid region around a node decreases as the malleability of a node decreases.

While the priority queue ensures that candidates with lower malleability are displaced first by the deformation, it does not carry any guarantee that it preserves FIFO (first in - first out) ordering among nodes with equal priority. Many priority queue implementations are based on underlying heap operations, such as the C++ Standard Template Library (STL) priority queue implementation (ISO/IEC, 1998). This heap structure can lead to a vein-like propagation through areas of equal malleability, as demonstrated in Figure 30 and the two provided video files. This propagation results in ridge-like shapes through malleable regions and thus in deformations that are inconsistent when the user makes small movements between frames of the interactive deformation.

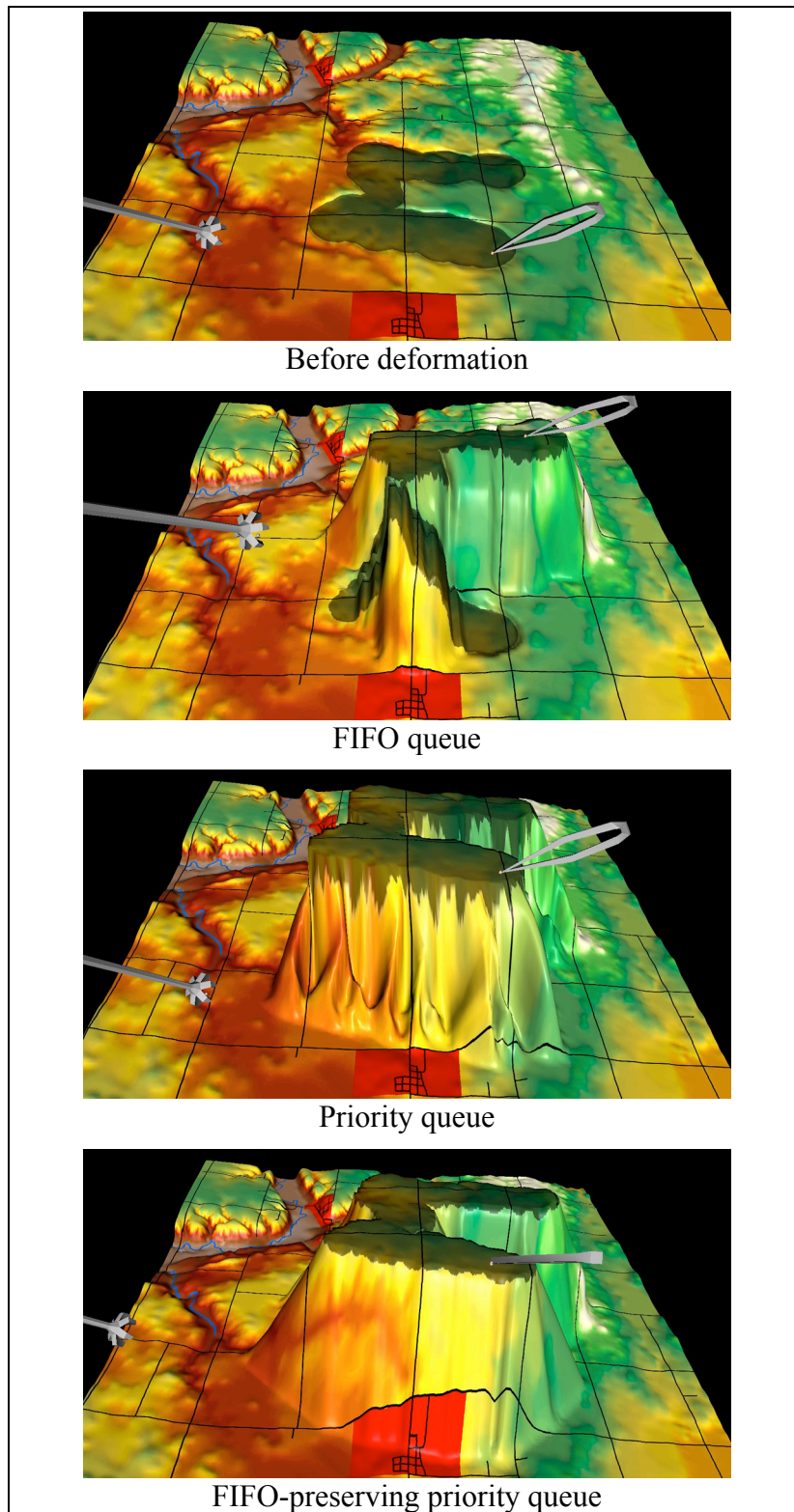


Figure 30. Comparison of candidate data structures for ChainMail deformation

A priority queue that preserves FIFO ordering among candidates with equal malleability values produces a much more consistent deformation, as shown in Figure 30. This type of queue provides a spiral-like propagation through areas of equal malleability, leading to deformation results that are less severe in appearance and slope. They also have the advantage of being much more predictable, which is important for providing precise control to a geoscientist.

The use of a priority queue instead of a simple FIFO queue to store the candidates does impose an increase in complexity. The priority queue we used in our implementation is logarithmic in complexity for insert and removal operations, while a simple queue would be constant in complexity for those operations. However this is proportional to the number of candidates awaiting analysis, and bounded by the size of the deformation, just as the original ChainMail algorithm. The Generalized ChainMail algorithm also suggested the use of a tree structure to store the candidates, which would also carry a similar increase in complexity over the simple queue (Li & Brodlie, 2003). In practice, the deformation remained interactive even with the additional complexity in the queue insert and removal operations, which we will detail further in the next section.

## 5. Results

We implemented this interaction as part of a larger haptic manipulation system using H3DAPI and other open source projects (H3DAPI). H3DAPI provides an efficient and flexible scene graph API (in C++ and Python) to implement sophisticated haptic interactions with 3D objects.

The result of using the FIFO-preserving priority queue on a number of different deformations is presented in Table 2. We compared it to implementations of a normal priority queue and a simple FIFO queue to store the candidates awaiting deformation. We measured the deformation by the number of nodes that it displaced, not the vertex count of the model. The results are reported in millisecond timings collected as the system performed the deformation. During the deformation, there were also two high-priority haptic threads running that calculate the force-feedback for two Phantom haptic devices. We performed these tests on an ordinary graphics PC: dual-core 2.6 GHz AMD 5000+ with 2GB of RAM and a Nvidia 256 MB 8600GT video card.

**Table 2. Comparison of timings in (ms) for deformation candidate structures**

<b>Nodes displaced</b>	<b>9,152</b>	<b>14,088</b>	<b>17,289</b>	<b>18,225</b>	<b>29,241</b>
FIFO queue (ms)	38	76	111	159	195
Priority queue (ms)	46	96	130	164	302
FIFO priority queue (ms)	45	79	125	167	298

The timings suggest that it is possible to improve the quality of mesh deformation with inhomogeneous properties while maintaining interactive rates. Although the FIFO-preserving priority queue has more expensive insert and removal operations, the measured performance hit is not severe. Even for relatively large numbers of vertices (around 29,000 nodes) the increase is only around 50% (from 200 ms to 300 ms).

## 6. Conclusions and future work

We presented a method for giving a geoscientist precise control over the deformation property values of an inhomogeneous mesh. The painting interaction allows precise control

over the malleability values of the mesh to affect locally varying degrees of deformation. We also presented enhancement of the Generalized ChainMail algorithm that allows it to perform a deformation of an inhomogeneous mesh. The implementation allows for interactive exploration of bimanual painting and deformation. A pilot study is underway to determine the effectiveness of this bimanual interaction.

## 7. Acknowledgements

We would like to thank Michael Oren, Jonathan Sheller and Sean Godinez for their contributions to this project. We would also like to thank the anonymous reviewers who provided valuable suggestions for improving this paper. Part of this work was funded by NSF grant IIS-534681.



## CHAPTER 5. CONCLUSIONS AND FUTURE WORK

Computer manipulation of digital shapes provides greater precision, faster replication and wider dissemination compared to traditional manipulation of real world objects. The goal of this research was to increase the speed and expressiveness of three-dimensional shape manipulation by providing direct three-dimensional interactions. This thesis presented mesh cutting, mesh deformation, and mesh painting manipulations, and the specialized combination of mesh painting and mesh deformation to enable deformation property painting.

The manipulations applied Guiard's Kinematic Chain theory for asymmetrical bimanual interaction by providing both macrometric and micrometric roles (1987). One hand assumed the macrometric role to control the position and orientation of the digital shape with a Phantom. The hand with the macrometric role aligned the digital shape in a favorable position for the detailed manipulation. The second hand fulfilled the micrometric role to perform the precise manipulations with a second Phantom. The user chose which hand to assign each role according to their lateral preference.

Mesh cutting is one of the micrometric manipulations available to the user. This thesis presented a mesh-cutting algorithm designed for cut path planning on the surface of a triangle mesh. The algorithm presented defers re-meshing until the user chooses to embed the cut in the mesh. This allows the user to see how the cut path will drape across the mesh and move or delete segments of the planned. To enable large portions of the mesh to be cut at interactive rates, the algorithm also pre-calculates the mesh elements that will need to be re-

meshed. These adaptations created a cutting algorithm that is suited to applications where the user wants to plan a precise cut rather than simulate contact with a cutting tool.

This thesis also presented a unified extension of the Generalized ChainMail algorithm and the Enhanced ChainMail algorithm. The unified algorithm allowed the deformation of both inhomogeneous and triangle meshes. Enhanced ChainMail calculates a constraint violation for each neighbor before selecting the neighbor to displace next (Schill, et al., 1998). The algorithm presented in this thesis employed an alternative method to ensure that the displacement of malleable nodes did not alter the relative displacement of neighboring stiff nodes. The use of a FIFO-preserving priority queue to store candidates awaiting deformation did not adversely affect the execution time of the deformation operation. This gave the system a way to interactively deform inhomogeneous meshes.

This research also presented a method of defining inhomogeneous deformation properties by painting a malleability value directly into a deformation property map, and the use of transfer functions to map the malleability value to three deformation property values. By using a single malleability value, instead of three deformation parameters, the system was able to use a temperature metaphor to visualize the local malleability of the mesh. The color of iron at cold temperatures designates rigid regions of the mesh, while the color of white-hot iron indicates the most malleable regions. The combination of deformation property painting and inhomogeneous mesh deformation created an interaction where the user can precisely move malleable regions of the mesh into place without changing the local shape of any stiff regions.

While these examples of asymmetric bimanual interactions have shown potential in a number of informal demonstrations, these manipulations have not yet been formally

evaluated. Other research demonstrated that asymmetrical bimanual interaction reduced errors and completion time in alignment tasks (Hinckley, et al., 1998; Hinckley, et al., 1997). However, the use of point-based haptic devices like the Phantom introduces a reduction in haptic information compared to the tracked tangible objects used by Hinckley et al. Another variable that remains unexplored is the value of haptic feedback in the asymmetric bimanual interaction for both hands. Existing experiments have only tested the use of haptic feedback devices for one of the two hands in a bimanual system (Fiorentino, et al., 2008). Future studies could determine whether any conditions exist where haptic feedback to the macrometric hand provides a benefit to the user.

Another area that deserves further exploration is the use of the temperature metaphor for malleability visualization. One future user study might explore whether the single malleability value and set of transfer functions to obtain deformation property values is more useful than allowing the user to directly apply stretch, shear, and compression parameters to the mesh. Another related study could evaluate the clarity of the temperature-to-malleability mapping compared to a more direct mapping of the three deformation parameters to the red, green, and blue components of a color for visualization. The careful study of these two design choices would inform future development of similar systems.

As three-dimensional digital shape creation and manipulation continues to be an important component of product design, character animation, and geoscientific manipulation, the need for expressive interactions will likely grow. This research presented three manipulations that support asymmetric bimanual roles for three-dimensional interaction. These manipulations represent a step towards creating more expressive interactions for digital surface manipulation.

## REFERENCES

- Atlan, S., & Garland, M. (2006). Interactive Multiresolution Editing and Display of Large Terrains. *Computer Graphics Forum*, 25(2), 211-223.
- Autodesk. 3D Studio Max Retrieved March 4, 2009, from <http://www.autodesk.com/3dsmax/>
- Autodesk. Maya Retrieved March 4, 2009, from <http://www.autodesk.com/maya/>
- Baxter, B., Scheib, V., Lin, M. C., & Manocha, D. (2001). *DAB: Interactive Haptic Painting with 3D Virtual Brushes*. Paper presented at the ACM SIGGRAPH.
- Bendels, G. H., Kahlesz, F., & Klein, R. (2004). *Towards the next generation of 3D content creation*. Paper presented at the Proceedings of the working conference on Advanced visual interfaces.
- Boeck, J. D., Raymaekers, C., & Coninx, K. (2006). Exploiting proprioception to improve haptic interaction in a virtual environment. *Presence: Teleoper. Virtual Environ.*, 15(6), 627-636.
- Bowman, D. A., Kruijff, E., LaViola, J. J., & Poupyrev, I. (2001). An Introduction to 3-D User Interface Design. *Presence: Teleoperators & Virtual Environments*, 10(1), 96-108.
- Bruyns, C., & Senger, S. (2001). Interactive cutting of 3D surface meshes. *Computers & Graphics*, 25(4), 635-642.
- Bruyns, C., Senger, S., Menon, A., Montgomery, K., Wildermuth, S., & Boyle, R. (2002). A survey of interactive mesh-cutting techniques and a new method for implementing generalized interactive mesh cutting using virtual tools. *Journal of Visualization and Computer Animation*, 13(1), 21-42.
- Cani, M.-P., & Angelidis, A. (2006). *Towards virtual clay*. Paper presented at the ACM SIGGRAPH 2006 Courses.
- Dachille, F. I., Qin, H., & Kaufman, A. (2001). A novel haptics-based interface and sculpting system for physics based geometric design. *Computer-Aided Design*, 33(5), 403-420.
- De Boeck, J., Raymaekers, C., & Coninx, K. (2006). Exploiting Proprioception to Improve Haptic Interaction in a Virtual Environment. *Presence: Teleoperators & Virtual Environments*, 15(6), 627-636.
- Dean, A. (2006, January). SensAble Freeform Modeling *Product Development and Manufacturing Solutions*.

- Dewaele, G., & Cani, M. (2003). Interactive global and local deformations for virtual clay. *Computer Graphics and Applications, 2003. Proceedings. 11th Pacific Conference on*, 131 - 140.
- Döllner, J. (2005). Geovisualization and Real-Time 3D Computer Graphics. In J. Dykes, A. M. MacEachren & M.-J. Kraak (Eds.), *Exploring Geovisualization* (pp. 343-366). Oxford: Elsevier.
- Dräger, C. (2005). *A ChainMail Algorithm for Direct Volume Deformation in Virtual Endoscopy Applications*. Technische Universität Wien, Vienna.
- Ebert, D. S., & Shaw, C. (2001). Minimally immersive flow visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 7(4), 343 - 350.
- Faeth, A., Oren, M., & Harding, C. (2008). *Combining 3-D geovisualization with force feedback driven user interaction*. Paper presented at the ACM Geographic Information Systems.
- Faeth, A., Oren, M., Sheller, J., Godinez, S., & Harding, C. (2008, Feb 29). *Cutting, Deforming and Painting of 3D meshes in a Two Handed Viso-haptic VR System*. Paper presented at the IEEE Virtual Reality.
- Fikkert, W., D'Ambros, M., Bierz, T., & Jankun-Kelly, T. (2007). Interacting with Visualizations *Human-Centered Visualization Environments* (pp. 77-162).
- Fiorentino, M., Uva, A. E., & Dellisanti Fabiano, M. (2008). Improving bi-manual 3D input in CAD modelling by part rotation optimisation. *Computer-Aided Design*.
- Foskey, M., Otaduy, M. A., & Lin, M. C. (2002). *ArtNova: touch-enabled 3D model design*. Paper presented at the IEEE Virtual Reality.
- Fröhlich, B., Barrass, S., Zehner, B., Plate, J., & Göbel, M. (1999). *Exploring geo-scientific data in virtual environments*. Paper presented at the Proceedings of the conference on Visualization '99: celebrating ten years.
- Gibson, S. F. (1997). 3D chainmail: a fast algorithm for deforming volumetric objects. *Proceedings of the 1997 symposium on Interactive 3D graphics*, 149-154.
- Golledge, R., Rice, M., & Jacobson, R. D. (2006). Multimodal Interfaces for Representing and Accessing Geospatial Information *Frontiers of Geographic Information Technology* (pp. 181-208).
- Gregory, A. D., Ehmann, S. A., & Lin, M. C. (2000). *inTouch: Interactive Multiresolution Modeling and 3D Painting with a Haptic Interface*. Paper presented at the IEEE Virtual Reality.

- Grossman, T., Balakrishnan, R., Kurtenbach, G., Fitzmaurice, G., Khan, A., & Buxton, B. (2001). Interaction techniques for 3D modeling on large displays. *Proceedings of the 2001 symposium on Interactive 3D graphics*, 17-23.
- Guiard, Y. (1987). Asymmetric division of labor in human skilled bimanual action: the kinematic chain as a model. *Journal of motor behavior*, 19(4), 486-517.
- H3DAPI. Open Source Haptics Retrieved February 23, 2009, from <http://www.h3dapi.org>
- Haan, G. d., Koutek, M., & Post, F. H. (2002). *Towards intuitive exploration tools for data visualization in VR*. Paper presented at the Proceedings of the ACM symposium on Virtual reality software and technology.
- Harding, C., Kakadiaris, I. A., Casey, J. F., & Loftin, R. B. (2002). A multi-sensory system for the investigation of geoscientific data. *Computers & Graphics*, 26(2), 259-269.
- Hay, R. J. (2003). *Visualisation and Presentation of Three Dimensional Geoscience Information*. Paper presented at the 21st International Cartographic Conference.
- Hinckley, K., Pausch, R., Proffitt, D., & Kassell, N. F. (1998). Two-handed virtual manipulation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 5(3), 260-302.
- Hinckley, K., Pausch, R., Proffitt, D., Patten, J., & Kassell, N. (1997, Jan 1). *Cooperative bimanual action*. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems, Atlanta, Georgia, United States.
- Ikits, M., Brederson, J. D., Hansen, C. D., & Johnson, C. R. (2003). *A Constraint-Based Technique for Haptic Volume Exploration*. Paper presented at the Proceedings of the 14th IEEE Visualization 2003 (VIS'03).
- ISO/IEC (1998). Programming languages: C++ (Vol. 14882, pp. 480-481): American National Standards Institute.
- Johnson, D., II, T. V. T., Kaplan, M., Nelson, D., & Cohen, E. (1999). *Painting Textures with a Haptic Interface*. Paper presented at the IEEE Virtual Reality.
- Keefe, D. F., Zeleznik, R. C., & Laidlaw, D. H. (2007). Drawing on Air: Input Techniques for Controlled 3D Line Illustration. *Visualization and Computer Graphics, IEEE Transactions on*, 13(5), 1067-1081.
- Kim, L., Sukhatme, G. S., & Desbrun, M. (2003). *Haptic Editing of Decoration and Material Properties*. Paper presented at the Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'03).
- Kowalik, W. (2001). *3D Device Interaction with Force Feedback*. Paper presented at the Gocad 3DH.

- Kraak, M.-J., & MacEachren, A. M. (2005). Geovisualization and GIScience. *Cartography and Geographic Information Science*, 32, 67-68.
- Kron, A., Schmidt, G., Petzold, B., Zah, M. I., & Hinterseer, P. (2004). Disposal of explosive ordnances by use of a bimanual haptic telepresence system. *2004 IEEE International Conference on Robotics and ....*
- Krum, D. M., Omoteso, O., Ribarsky, W., Starner, T., & Hodges, L. F. (2002). *Evaluation of a multimodal interface for 3D terrain visualization*. Paper presented at the Visualization '02.
- Lawrence, D. A., Pao, L. Y., Lee, C. D., & Novoselov, R. Y. (2004). Synergistic Visual/Haptic Rendering Modes for Scientific Visualization. *IEEE Comput. Graph. Appl.*, 24(6), 22-30.
- Li, Y., & Brodlie, K. (2003). Soft Object Modelling with Generalised ChainMail-Extending the Boundaries of Web-based Graphics. *Computer Graphics Forum*, 22(4), 717-727.
- Lim, Y.-J., Jin, W., & De, S. (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: Teleoper. Virtual Environ.*, 16(6), 563-583.
- Lundin, K., Cooper, M., Persson, A., Evestedt, D., & Ynnerman, A. (2007). Enabling design and interactive selection of haptic modes. *Virtual Reality*, 11(1), 1-13.
- MacEachren, A. M., & Kraak, M.-J. (2001). Research Challenges in Geovisualization. *Cartography and Geographic Information Science*, 28, 3-12.
- Massie, T. H., & Salisbury, J. K. (1994, November). *The PHANTOM Haptic Interface: A Device for Probing Virtual Objects*. Paper presented at the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems.
- Maxon. Cinema 4D Retrieved March 4, 2009, from [http://www.maxon.net/pages/products/cinema4d/cinema4d\\_e.html](http://www.maxon.net/pages/products/cinema4d/cinema4d_e.html)
- McDonnell, K. T., Qin, H., & Wlodarczyk, R. A. (2001). Virtual clay: a real-time sculpting system with haptic toolkits. *I3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*, 179-190.
- Mitasova, H., Mitas, L., Ratti, C., Ishii, H., Alonso, J., & Harmon, R. S. (2006). Real-Time Landscape Model Interaction Using a Tangible Geospatial Modeling Environment. *IEEE Comput. Graph. Appl.*, 26(4), 55-63.
- Nöllenburg, M. (2007). Geographic Visualization *Human-Centered Visualization Environments* (pp. 257-294).

- Ott, R., De Perrot, V., Thalmann, D., & Vexo, F. (2007). MHaptic : a Haptic Manipulation Library for Generic Virtual Environments. *Cyberworld '07*, 338-345.
- Owen, R., Kurtenbach, G., Fitzmaurice, G., Baudel, T., & Buxton, B. (2005). When it gets more difficult, use both hands: exploring bimanual curve manipulation. *Proceedings of Graphics Interface 2005*, 17-24.
- Pihuit, A., Kry, P., & Cani, M. (2008). Hands on virtual clay. *Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on*, 267 - 268.
- PTC. Pro/Engineer Retrieved March 9, 2009, from <http://www.ptc.com/products/proengineer/>
- Qi, W. (2006). *Geometry based haptic interaction with scientific data*. Paper presented at the Virtual reality continuum and its applications.
- Ratti, C., Wang, Y., Ishii, H., Piper, B., & Frenchman, D. (2004). Tangible User Interfaces (TUIs): A Novel Paradigm for GIS. *Transactions in GIS*, 8(4), 407-421.
- Roberts, J. C., & Franklin, K. (2005). *Haptic Glyphs (Hlyphs) " Structured Haptic Objects for Haptic Visualization*. Paper presented at the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems.
- Roberts, J. C., & Paneels, S. (2007). *Where are we with Haptic Visualization?* Paper presented at the Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems.
- Salisbury, K., Conti, F., & Barbagli, F. (2004). Haptic Rendering: Introductory Concepts. *IEEE Comput. Graph. Appl.*, 24(2), 24-32.
- Schill, M. A., Gibson, S. F. F., Bender, H. J., & Manner, R. (1998). Biomechanical simulation of the vitreous humor in the eye using an Enhanced ChainMail Algorithm, *LNCS* (Vol. 1496, pp. 679-687).
- Sener, B., Pedgley, O. F., Wormald, P. W., & Campbell, I. R. (2003). *Incorporating the Freeform Haptic Modelling System into New Product Development*. Paper presented at the EuroHaptics.
- SensAble. SensAble Technologies Retrieved March 4, 2009, from <http://www.sensable.com/>
- Shaw, C., & Green, M. (1997). THRED: a two-handed design system. *Multimedia Systems*.
- Shneiderman, B. (1983). Direct Manipulation: A Step Beyond Programming Languages. *Computer*, 16(8), 57-69.



- Slocum, T. A., Blok, C., Jiang, B., Koussoulakou, A., Montello, D. R., Fuhrmann, S., et al. (2001). Cognitive and Usability Issues in Geovisualization. *Cartography and Geographic Information Science*, 28, 61-75.
- Teschner, M., Heidelberger, B., Muller, M., & Gross, M. (2004). *A Versatile and Robust Model for Geometrically Complex Deformable Solids*. Paper presented at the Proceedings of the Computer Graphics International.
- The Blender Foundation. Blender Retrieved March 4, 2009, from <http://www.blender.org/>
- USGS. The National Map Seamless Server Retrieved March 5, 2009, from <http://seamless.usgs.gov/>
- van Dam, A., Laidlaw, D. H., & Simpson, R. M. (2002). Experiments in Immersive Virtual Reality for Scientific Visualization. *Computers & Graphics*, 26(4), 535-555.
- Viet, H., Kamada, T., & Tanaka, H. (2006). An Adaptive 3D Surface Mesh Cutting Operation *Articulated Motion and Deformable Objects* (pp. 366-374).
- Ward, M., & Yang, J. (2004, May 2004). *Interaction Spaces in Data and Information Visualization*. Paper presented at the Joint Eurographics -- IEEE TCVG Symposium on Visualization.
- Wood, J., Kirschenbauer, S., Döllner, J., Adriano, L., & Bodum, L. (2005). Using 3D in Visualization. In J. Dykes, A. M. MacEachren & M.-J. Kraak (Eds.), *Exploring Geovisualization, Exploring Geovisualization* (pp. 295-312). Oxford: Elsevier.