AN AUGMENTED HAPTIC INTERFACE

AS APPLIED TO

FLOW VISUALIZATION

BY

STEPHEN SCHAEFFER

BFA, University of Illinois, 1990

THESIS

Submitted to the University of New Hampshire

in Partial Fulfillment of

the Requirements for the Degree of

Master of Science

In

Computer Science

December, 2007

This thesis has been examined and approved.

_____

Thesis Director
Colin Ware
Professor of Computer Science

_____

R. Daniel Bergeron
Professor of Computer Science

_____

James Weiner
Associate Professor of Computer Science

_____

Date

DEDICATION

This document is dedicated to the people who have made my venture in graduate school possible: my wife Astrida, my daughter Helena, my mother Liz, and my in-laws Yvonne and Witek, who have all been amazingly supportive, tolerant, and patient while I did this. I can only apologize for dragging you all through the obsessive self-indulgence that is a graduate degree. Finally, Dad, I wish you had been around... I could have used some help on my homework from time to time.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

AUGMENTED HAPTIC INTERFACE

FOR FLOW VISUALIZATION

by

Stephen Schaeffer

University of New Hampshire, December, 2007

A novel 3D computer interface is proposed in which a physical handle containing force sensors and capable of simulating virtual touch through force feedback is coupled to a variety of virtual tools in a 3D virtual environment. The visual appearance of each tool reflects its capabilities. At one moment a user might feel they are holding a virtual grabber, activated by squeezing, and at another moment they are holding a virtual turntable activated by physical motion of a virtual wheel. In this way it is intended that form and function can be combined so that users rapidly learn the functional capabilities of the tools and retain this learning. It is also intended that the tools be easy to use because of intuitive mappings of forces to actions. A virtual environment is constructed to test this concept, and an evaluation of the interface conducted.

CHAPTER I

INTRODUCTION

The problem with many computer interfaces is that their form does not follow their function. A mouse-based interface can be in many states and, depending on the state, a mouse click can result in selecting an object, deleting an object, or running a program to name just a few results. In none of these examples does the mouse cursor provide clues as to the actions available, and users often make mistakes because they are not aware of the state of the interface.

On the other hand, there is no uncertainty about what physical tools like a pair of pliers, a hammer, or a knife are capable of. Their forms tell us about their functions. Although Virtual Reality systems are intended to simulate "reality" they fail to provide interfaces in which the form follows function in this way. Even when force feedback devices such as the PHANToM are used, the input device usually appears as a stylus that can either sculpt or move an object with no visible indication of the underlying system state, leading to confusion on the user's part.

This thesis is an exploration of a novel kind of interface to 3D virtual environments, and has as its core idea the design and implementation of a set of tools in which the visible form does follow the function. An example of the new tool is a virtual ink dropper used to place dye. This tool looks like an eyedropper and responds to grip pressure in a way analogous to a real one. Another tool serves as a gripper to move objects around in virtual

space, and appears as a pair of pliers. Moreover these tools can be activated by literally squeezing an instrumented handle, naturally mapping the physical action into the virtual world. The application domain chosen to explore this concept is an interactive visualization system to explore an ocean current model.

Prior Research

Virtual reality interfaces are those computer interfaces which attempt to immerse the user in some kind of simulated world. Many computer games, such as driving or flight simulators, create virtual worlds through which users navigate using controls like steering wheels, joysticks, or simple mouse and keyboard commands. Adding alternate

Figure 1: An image showing the state of the art in virtual reality, from 1989. Note the lack of physical feedback for the user (Carl06). Originally presented in (Fish89).

input and output tools, such as head-mounted displays and body sensing input devices, may enhance the user interface but only if properly designed. Figure 1 shows a system that was state-of-the-art in 1989, including position sensing gloves, head tracked stereoscopic display, and sound input and output. Conspicuous by its absence is any means of generating a virtual sense of touch. Partly because of this lack of controllable tactile feed back to match the visual imagery, early experiments in virtual reality were lacking, and the technology was not adopted.

More recently, force-feedback devices have appeared that can both generate force as a form of computer output and accept user generated forces as a form of input, thereby allowing the computer to simulate a virtual sense of touch. These are called *haptic* devices.

Haptic devices come in many forms, but can be divided between active haptic devices, which are capable of generating specific force feedback, and passive haptic devices, which rely only on their inherent physical characteristics to provide feedback and can not change the force they produce. Of necessity, all active haptic devices have some sort of mechanical component to generate force as well as capture position information. Such devices can be further classified according to their frame of reference. Some devices are anchored to a workspace which provides a fixed reference frame, and are able to generate forces independent of the user's body. Other devices are anchored to

Figure 2: The Immersion CyberGrasp glove. This is an example of a haptic device which uses the body as its anchor (Glas07).

the body and produce forces on the the user's fingers, wrist, or arm. Figure 2 shows a glove anchored to the user's body for its frame of reference. In contrast, Figure 3 depicts a device with a fixed reference frame. See Burdea (Burd00) for a more complete summary of the various kinds of haptic devices and their histories.

What makes haptic devices so powerful is that they can impose constraints on motion in virtual spaces. When putting an object down onto a table in the real world, a person feels the object making contact with the surface of the table. Once on the table, the surface constrains movement to a horizontal plane. Because of this, it is far easier to arrange

objects on a physical plane than on a virtual plane with no such constraint, although the latter is required in some VR systems. Such feedback applies naturally to use in navigating a virtual world, because walls and floors feel solid and impenetrable, constraining the user to moving between rather than through the walls.

Mechanical tools have haptic constrains that are critical to the way we use them, for example the user can feel when a screwdriver is firmly seated in the head of a screw. Many tools have rotating parts, and users can feel the rotation about a fixed axis through proprioceptive feedback in their own joints. An example if this is the rotating platform used by many sculptors to position a piece for the best working angle. A particularly useful kind of mechanical feedback is a kind of breakable position constraint known as a *detent*. This kind of feedback occurs with some cupboard doors that click open and shut, the balance knob on a radio often has a detent for its center position, and some fan control knobs have multiple speed settings defined by detents
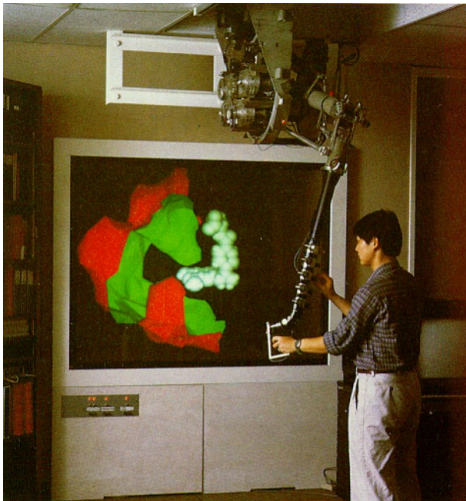
Figure 3: The GROPE-III haptic system. Project GROPE began in 1967, and has gone through many stages of development. This image is from 1990 or slightly earlier (Broo90).

In Figure 3 a user is shown trying to fit a chemical molecule into a receptor. By adding haptic feedback to this process, the user can feel the simulated forces from other molecules. This can guide and constrain their actions in a way analogous to feeling a key going into a lock, a process which is familiar enough to many people to be done in the dark.

Another way of representing objects haptically is through the use of props, as proposed by Hinckley et al. (Hinc97). In this school of

thought, virtual objects are represented by actual physical objects similar in form to the virtual ones. In Figure 4, a doll head and a sheet of Plexiglas, with appropriate tracking mechanisms, are used to interface with MRI brain scans for planning surgeries. The doll head is a proxy for the scanned head, and the sheet of Plexiglas acts as a cutting plane. This interface
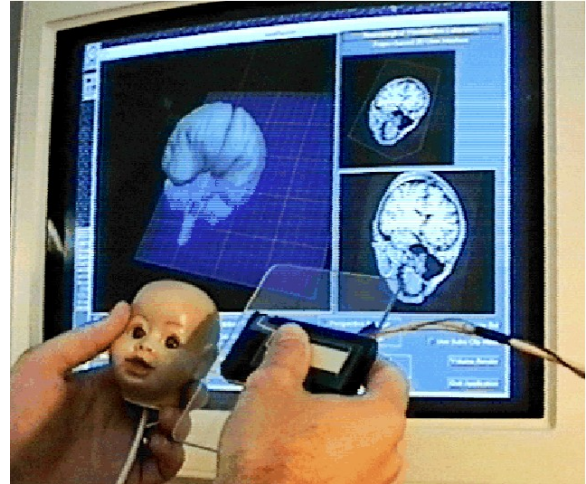
Figure 4: Props-based interface. MRI data is controlled by the doll head, and the cutting plane by the Plexiglas sheet (Hinc97).

allows doctors to freely manipulate both head data and cutting plane in a very natural and intuitive manner, although the force feedback is entirely passive.
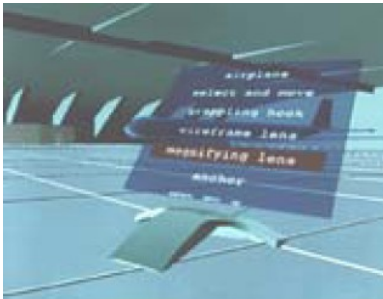
Figure 5: The Virtual Tricorder. The top portion shows it in the virtual realm with context dependent information displayed. The lower portion shows the physical object and its virtual counterpart. Image from www.cs.ucsb.edu/ ~holl/CS290I/handouts/ slides2-metaphors.pdf.

An extension of the props idea is the Virtual Tricorder proposed by Wloka and Greenfield (Wlok95). In this design, there is a single physical prop which can be transformed into a variety of virtual tools depending on the state of the system. The visual appearance of the tool takes on different forms representing the virtual tool selected (Figure 5). The Tricorder, at least in concept, then becomes a universal tool capable of accessing and manipulating many kinds of data in many different ways. They did not, however, attempt to change its appearance in any interesting ways to make its form follow its function, leaving the user likely to make state errors, and the haptic feedback is entirely passive.

Another example of an interface capable of both blending the real and the artificial, and of transforming itself, is the Chameleon tool described by Fitzmaurice and Buxton (Fitz94) and Tsang, et al. (Tsan02). An LCD screen is used in the real world to display a window onto the virtual world (Figure 6). As the display is

Figure 6: The Boom Chameleon interface. As the user moves the monitor around, the image on the display updates as if the user were moving a picture frame around a physical object (Tsan02).

moved through space, the scene changes to reflect the user's viewpoint, as if the user were looking through a picture frame, which they can move, while walking around a real object. The important aspect of the Chameleon, however, is it ability to to transform into a variety of different tools, each of which uses the small touch screen interface in different ways. The original chameleon (Fitz94) could be a context sensitive interface to a weather display panel, could transform to a 3D viewer, or could be a visual interface giving detail about a rack of electronics.

Haptic Flow Visualization

Because flow visualization is our chosen application domain for testing the interface ideas proposed here, we now review prior work in the are that has used haptics. Most of this focuses on using the flow data to generate forces either pushing the stylus through the workspace or generating resistance as the user moves the stylus.

Aviles and Ranata's voxelNotepad (Avil99) was designed for exploring geophysical data. It allows the user to choose some scalar variable in the data for use in generating virtual viscosity. The user can then drag the haptic stylus through the virtual world feeling

greater or lesser drag while moving through virtual oil deposits.

Iwata and Noma (Iwat93) used scalar, vector, or tensor fields to generate forces or torques to apply to their haptic platform. The user's hand was then either pulled to areas of lower potential or rotated
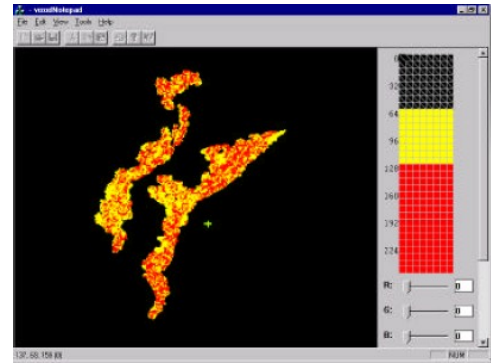
Figure 7: The voxelNotepad interface (Avil99).

depending on the data being used. This was extended by Nesbitt et al. (Nesb01), who used the modeled flow of metal in a blast furnace to calculate forces which were then used to push the haptic stylus through the virtual space in the direction of the modeled flow. The work of van Reimersdahl et al. (vRei03) extends these ideas further, creating streamlines through the data which then pull the stylus tip in the computed direction of flow.

Ikits et al. (Ikit03) use flow data to constrain user motion. In their design, the user finds it easier to move with the virtual flow than across it. Figure 8 shows a model of a human heart, with user motion constrained to a single layer of muscle. Yellow sections of the
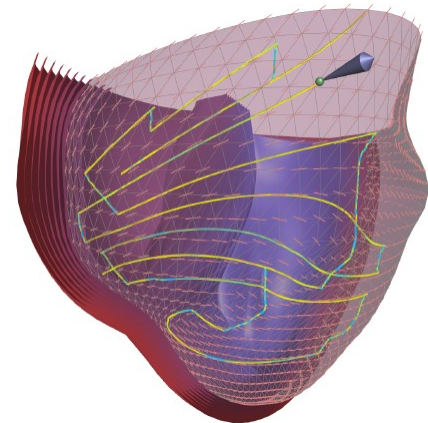
Figure 8: Constraint based haptic volume exploration (Ikit03).

path indicate areas where the user moved the stylus following the path of the muscle fibers, and blue sections show where the user crossed from one fiber to another.

Lawrence et al. (Lawr00), (Lawr04) also use the flow data to generate constraints for motion. Shock waves in the flow data are used to create haptic shells which the stylus can move over freely, but which are more difficult to penetrate. The user can then feel intersections between primary and secondary shock waves, as well as explore the space

7

between the leading and trailing edges of a shock wave. They also describe a case where rather than creating haptic surfaces, they generate torques on the stylus so that the user feels the stylus being pulled to point "upstream" into the flow rather than pointing across it.

Baxter and Lin (Baxt04) propose haptically interacting with virtual fluids, but in their case the fluids in question are simulations of paint which are then virtually brushed on by the user, so the user is causing the fluid interaction rather than trying to explore it.

Figure 9: A sequence of images from (Baxt04) showing interactive fluid rendering with haptic interaction. Green arrows show instantaneous force, while magenta shows velocity vectors.

We believe that using forces or resistances to "visualize" flow should be used with caution. In the everyday world humans use the perception of forces to get the general range of a value rather than a specific numerical one. For example, on a windy day there is an immediate sensation that wind is blowing outside and not indoors. As humans, we do not perceive such forces with numerical precision, but rather would feel subjectively that the wind is blowing "gently", "moderately", or "like mad". Humans can sense the position of their fingertips to within 2.5 mm, but according to Srinivasan and Basdogan (Srin97), "The resolution for velocity and acceleration of the fingertip, measured in terms of the Just Noticeable Difference (JNDs), are about 11% and 17% of reference values, respectively." On the other hand, with vision we can capture one million points of information per eye and process them to

find patterns at a rate of approximately four per second (Ware04). Thus vision will always be better than haptics for perceiving patterns, and therefore haptics should not try to compete against graphics but should seek to complement it.

A more useful design approach may be to use haptics to support *interaction* with data objects positioned in the data. In this way, we rely on the strength of the human proprioceptive-haptic system for manipulating tools and positioning objects in space, and on the eye's ability to interpret numerical information through visualization.

Research Strategy

A central problem in VR research is that computer technologies can simulate visual information with photo-realistic quality, but cannot simulate haptic information nearly as well. There is a fundamental difficulty in creating arbitrary virtual shapes that can be touched. This means that we can create any kind of tool we like with respect to appearance, but only a very limited set of mechanical properties.

This presents a challenge in following the design precept that form should follow function. In meeting this challenge researchers have developed the props concept, enabling a real object with haptic input to be combined with various visual appearances. However, physical props have a fixed set of haptic affordances. For example, the doll head prop affords intuitive rotating or translating of a virtual brain, but is not useful to represent a cutting plane. For this another prop must be introduced.

On the other hand, devices such as the PHANToM are programmable but can only produce a single point force vector. This means that the device is more versatile with

regard to the affordances it can be used to represent, but more limited with regard to the richness of haptic feedback it can provide.

In the case of the current project, we take an existing force-feedback tool, the SensAble PHANToM, as the basic platform and add force sensors and a mouse wheel/button assembly. By augmenting the interface input mechanisms, we aim to capture the best of all of these schools of thought, constructing a tool which can provide a physical entity similar to Hinckley's props-based interface, but with capabilities similar to the chameleon tool of Fitzmaurice and Tsang, and with haptic constraints to aid manipulation. Our goal is to create a novel blend of the real and the virtual where users can physically touch virtual objects, combined with the ability to change virtual tools on the fly – a kind of Virtual Tricorder with active haptic support.

In this work the problem of making form follow function is addressed through the following set of design principles:

1) The visual appearance of a tool should indicate its functionality.

2) The passive haptics of a tool will be supported through the use of an instrumented handle. This allows for forces on the handle to be transmitted to the computer.

3) The active haptics of a tool, achieved through the use of the PHANToM, will be used to reinforce interaction constraints.

4) There should be a natural mapping between forces on the handle and visual appearance of the tools. This is intended to make the available actions of the tools self-evident.

As mentioned, the chosen application domain is flow visualization. However, the approach taken here is unlike previous work using haptics to explore flow data, in that the tools described are not intended to allow users to feel the flow. Instead force feedback of both the passive and the active kind is used to provide constraints and is intended to make a set of tools that are easy to manipulate. By so doing, the visualization remains predominantly in the realm of the visual, and the manipulation remains in the realm of the haptic.

The remainder of this thesis represents an exploration of the design possibilities arising from this set of principles. In general the idea is that the visual form and the haptic affordances should act in concert and therefore a reasonably diverse set of easy to use tools can be created. The research is carried out as a design exercise to produce a proof-of-concept prototype environment, using ocean flow data because it provides time-varying 3-dimensional flow.

CHAPTER II

THE DESIGN OF THE PROTOTYPE
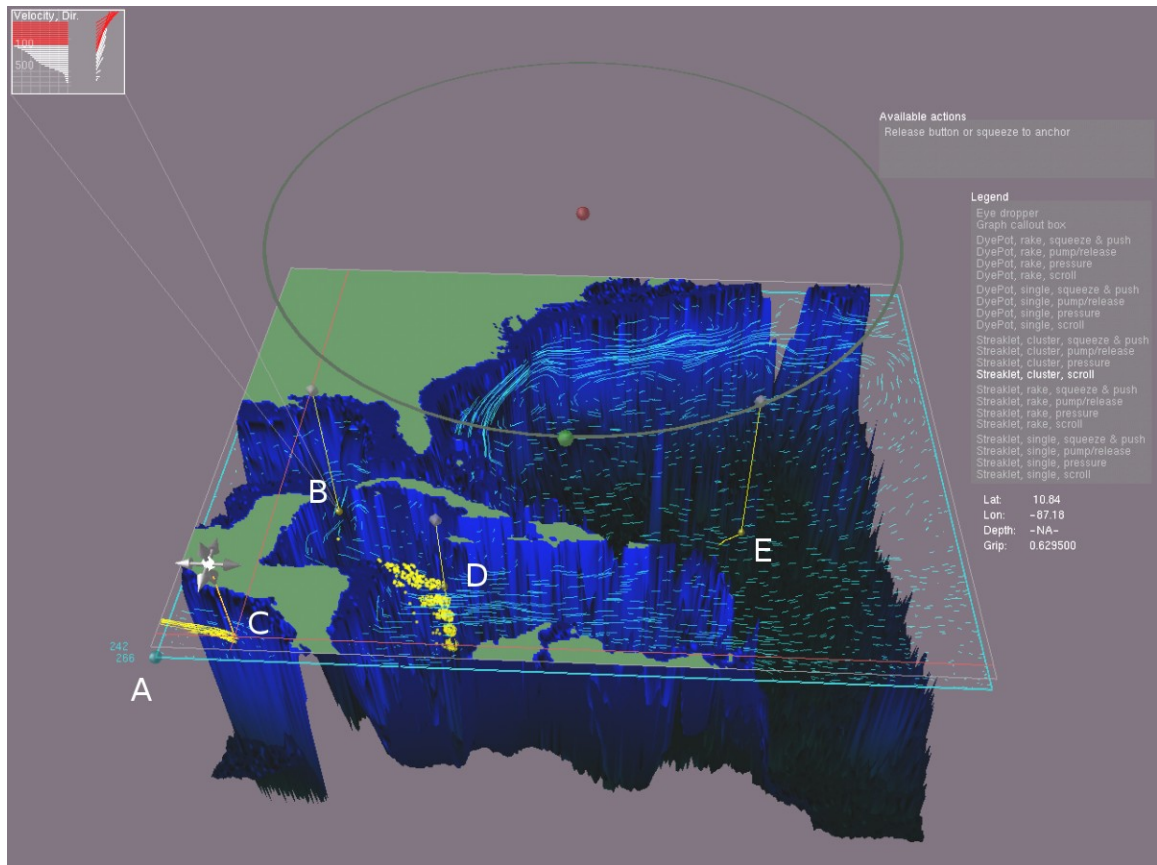
Figure 10: The prototype haptic interface visualizing ocean currents in use, showing the Caribbean Sea and part of the Atlantic Ocean. Five different types of visualization tool are visible, as are the view control tools:
(A) Shows a blue rectangle containing streaklets depicting the overall current flow.
(B) Shows flow velocity and direction through the water column in a separate window.
(C) A streaklet emitter with many trails emerging from a single detent. It is shown being moved.
(D) A dyepot rake tool, with five emitting nodes but a single handle.
(E) A single trail streaklet emitter.

A research prototype was developed to evaluate methods for combining haptic and visual feedbacks. As with all user interface research, the goal is to investigate ways of making computer interaction more effective and easier to learn. In the case of this prototype one

of the key design principles is to make state changes in the system apparent using both visual and haptic feedback. The prototype design was iteratively refined as challenges or opportunities became apparent. Figure 10 shows an image of the prototype haptic ocean exploration software.
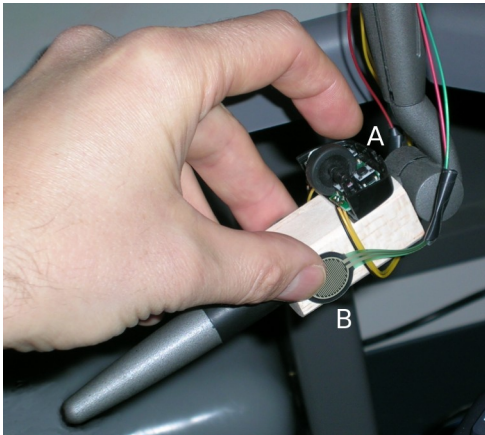
Figure 11: The chameleon handle. On top is a modified mouse wheel assembly (A). On the sides are pressure sensors (B).

All interaction in the system is done through the *chameleon handle* (Figure 11). This is like the Virtual Tricorder in being a physical handle with a virtual manifestation in the 3D graphic environment. The goal is to have a single interface device in the real world which can mutate, on demand, into a variety of tools in the virtual world. This physical handle has the capacity for active force feedback, so it can produce forces on the tool tip appropriate to the tool being instantiated, and has other inputs such as pressure sensors, buttons and thumb wheels that can be mapped to different features of the virtual tools. Two pressure sensors are placed on either side of the handle affording a pinch grip, which allows for isometric force sensing. The user receives proprioceptive fingertip feedback relating to how hard they are gripping the sensors, and the computer receives a corresponding pressure value. A mouse wheel assembly is located on top of the handle, which incorporates a series of detents felt as clicks when the user scrolls. This allows for scrolling through choices, and the mechanism used for cycling the chameleon handle through its various tool transformations.

13

The Chameleon Tools

The visual image of the handle is chameleon-like in that it changes appearance to portray whatever virtual tool is appropriate at the time. Some of the tools are a pincer head, a translation widget, an
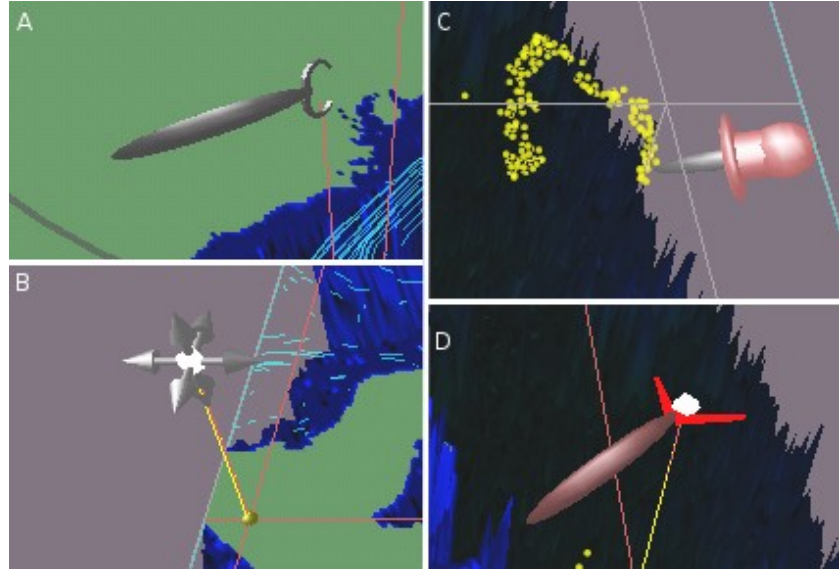
Figure 12: Some of the possible handle manifestations. (A) Pincer head. (B) Moving an element. (C) Eyedropper tool. (D) Deleting an element.

eye dropper for placing virtual dye, and a cutter for deleting elements, as shown in Figure 12.

Detents

An essential part of most of the tools is a haptic *detent*, which we describe here before continuing to develop the chameleon tools concept. A detent is a point in space that pulls in the phantom stylus tip and holds it there. Selecting points in 3-dimensions without haptic support is known to be difficult, and adding detents makes the selection task much easier (Hern94). Detents in the prototype are more than simply selectable points in space, and are used to guide the user when manipulating tools in a manner similar to the haptic fixtures described by Payandeh and Stanisic (Paya02).

Detents are the main link between the chameleon handle and view controls or previously instantiated visualization tools. Each already instantiated tool has a detent associated with

it, shown as a sphere. Touching this detent causes the chameleon handle to transform into that particular tool. If the stylus tip is near a detent, it will snap to the location of the detent. Given moderate force, the stylus tip snaps free from the detent. Attaching to detents in this way the stylus helps the user attach to view controls or visualization tools, which can then be moved or otherwise interacted with as the basis for most of the other haptic interactions.

When attached to a detent, a *three finger grip*, where the user simultaneously squeezes the grip sensors with the middle finger and thumb and clicks the mouse wheel button with the index finger, can be used to move visualization tools or view controls.

Grabber Tool

When the chameleon handle is not attached to any detent, it takes the appearance of a pair of pincers which parallels the motion of the stylus through the virtual world. From this state, the user has the potential to attach to either control detents or visualization tool detents. Figure 13 shows the handle in this state.

Figure 13: The handle not attached to a detent.

Eyedropper Tool

The eyedropper tool deposits a cloud of dye particles into the flow which are then carried by the flow until they either leave the display area or expire. Only grip pressure (not the mouse wheel button) is needed to place dye particles and, once placed, the user cannot modify them in any way.

If the chameleon is in this state, gripping the pressure sensors places dye particles into the flow with the volume of dye corresponding to the pressure on the sensors. Visually, squeezing causes the bulb of the eyedropper to constrict in the middle, as if being squeezed by a virtual hand. It also causes the color to redden, with the
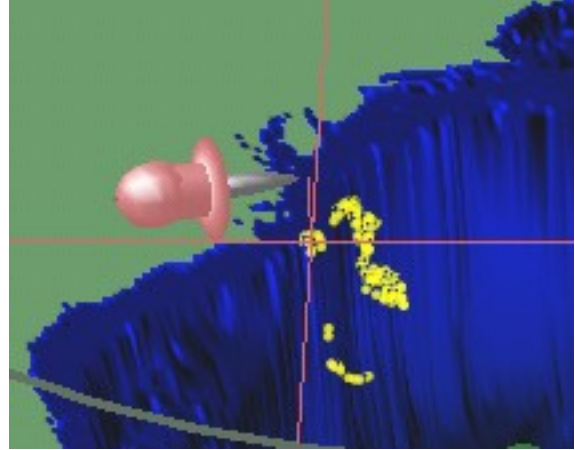
Figure 14: Eyedropper tool.

amount of constriction and the change in color corresponding to the strength of the grip. Once the pressure sensors are released, emission stops.

Anchored Flow Visualization Tools

Unlike the eyedropper, several of the flow visualization tool have the ability to be detached from the handle and continue to emit dye in various forms. These are called *anchored tools*.

There are three basic anchored tools that use either streaklet tracing or particle advection for visualization: s*treaklet emitters, virtual dyepots,* and the *streaklet field*. A fourth tool, the *callout window tool*, provides a more schematic view onto the flow data. All these tools are interactively placed in the flow field, anchored, and manipulated using the chameleon handle. When the handle is attached to a tool detent, it becomes visually transformed to that tool. Tool manipulation techniques differ depending on tool type.

Alternate Mechanisms

A major design challenge relating to the chameleon concept is the method by which dye emission rate is adjusted for anchored tools. Four different control mechanisms were

designed and implemented in order to evaluate different haptic/visual mappings for this basic task, with the intention that an actual production system would only make use of the best of these. Tool head designs for each adjustment mechanism are shown in Figure 15.

The four mechanisms are as follows.

1) Scroll: The tool tip transforms into a representation of a scroll wheel as show in Figure 15A. The mouse wheel directly controls dye volume: scrolling forward increases volume, backward decreases it. The emission rate is visually represented by the color of the tool, with gray indicating low emission and red indicating high flow. The tool includes a gray wire-frame box to indicate the orientation of the handle. Arrows, indicating the adjustment actions, rotate on the same axis as the wheel as the user adjusts the tool. When emission is high, the arrows are rotated to the front of the tool, and the wheel is bright red. When emission is low, the arrows are rotated to the back, and the wheel is gray.

2) Squeeze: The tool tip transforms into a representation of a box with two pads on the sides, as shown in Figure 15B. These are intended to correspond to the pressure
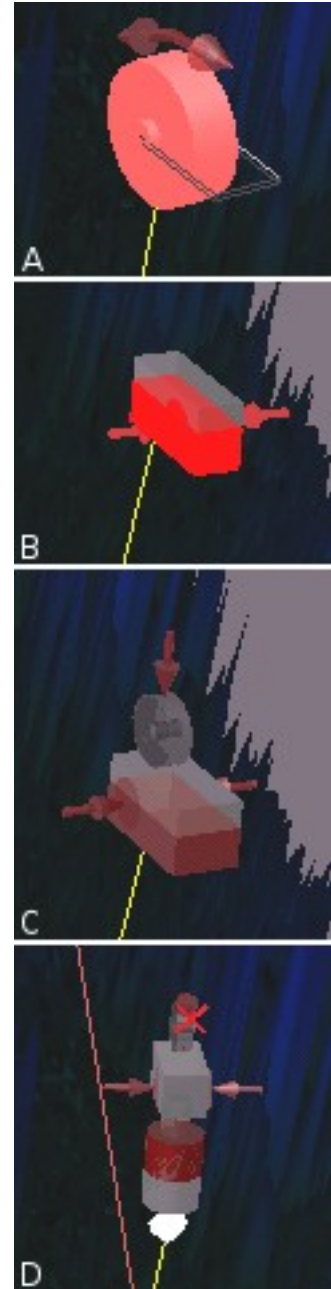
Figure 15: The adjustment mechanisms.
(A) Icon for tools using scroll to set volume or frequency.
(B) Icon for tools using grip pressure directly to set frequency or volume.
(C) Icon for tools using pressure to increase and wheel button to decrease.
(D) Icon for tools using squeeze and lift to increase flow and squeeze and lower to decrease it.

17

sensors between the middle finger and thumb. Pinch force controls the volume of dye being emitted, with more pressure increasing dye volume and vice versa. The tool squishes as more pressure is exerted, the "reservoir" of dye contained within the box raises and lowers with increase or decrease of pressure, and the color of the level indicator in the reservoir changes from dark red to a bright red when being adjusted. As an additional cue to action, arrows are shown pointing at the pressure sensing pads.

3) Pump/release: The tool tip transforms into a box with two pads on the sides and a representation of the scroll wheel on the top, as shown in Figure 15C. Squeezing the tool increments the dye emission by one step, with a corresponding raise in the level of the dye reservoir in the box. The color also changes from dull red to bright red. Holding the wheel button down reduces the emission rate as long as the button is held, causing the reservoir to slowly drain.

4) Raise/lower: The tool tip transforms into a reservoir cylinder with a box mounted above with pressure pads on the sides. Above this is a representation of the scroll wheel. To increase flow, the user grasps the handle and presses down. To decrease flow, the user pulls up. The level in the cylinder moves up and down accordingly. The scroll wheel acts as a lock, with scrolling forward enabling adjustment, and scrolling back locking the flow volume at the current rate. The scroll wheel representation includes an X which highlights when adjustment is disabled, and the pressure pads indicate grip pressure by changing color.

Tool Flow Visualization Techniques

The actual flow visualization techniques built into the tools are of two types. One type uses the continuous emission of *streaklets*. An individual streaklet is based on a *pathline*, which is the trajectory of a particle as it moves forward in space and time. A streaklet shows a short section of the computed pathline that is animated along the pathline as time progresses. This type of visualization tool is called a *streaklet emitter*. The other type of tool uses continuous emission of dye particles which are then carried by the flow model as it progresses through time. This is called a *virtual dyepot*. Refer to Appendix C, Flow Visualization Techniques for more detail.

Streaklet Emitters

Streaklet emitter tools consist of a detent and one or more emitting node that periodically emits a streaklet which flows downstream from the origin point, eventually fading out. Emission rate and detent location can be adjusted using the chameleon handle.

Figure 16: Single streaklet emitter tool. Since the detent is above water, emission takes place at the surface of the water, with a reference mark on the water. Because of the level of the red reservoir and the narrowness of the icon, this indicates that slightly over half of the possible pressure is being applied.

The detent is positioned where the stylus tip was when the element is released. If the detent is above water, streaklets are emitted at the surface of the water. On the other hand, if the detent is in the water streaklets are emitted at the location of the detent. In either case, a reference sphere is drawn at the surface of the water, with a line connecting to the detent (Figure 16).

Streaklets are drawn in yellow, but the actual color depends both on the depth of the streaklet and by the upwelling or downwelling of the specific section of streaklet. Streaklets deeper in the water column are shown in darker shades (Figure 17, A, B). Areas where water moves either up or down are rare in the model, but are indicated by coloring the streaklet red where flow moves upward and green where it move downward (Figure 17, C, D).

One variation of this tool, called a *rake*, creates a string of emitters aligned vertically, having a single detent. In this variation, if the handle is above water the depth of the the deepest emitter matches the height of the detent above water. The remaining emitters are spaced evenly from the surface of the water to the deepest emitter. The top half of Figure 17 shows a rake streaklet emitter with the detent above the surface. The top-most emitter is at the surface,

Figure 17: Streaklet behavior. (A) Shows an emitter at the surface of the water. (B) Shows an emitter at the same latitude and longitude near the bottom of the ocean. (C) Shows an emitter in an area with upwelling and downwelling currents. (D) Shows a detail of (C) enlarged 2.5x.

the bottom-most emitter and the haptic handle are

equidistant from the surface, with the remaining nodes equally spaced between top and

bottom emitters. If the handle position is below water, the bottom emitter is at the

location of the detent.

A second variation of this tool type, called a *cluster*, allows the user to place a cluster of

streaklets scattered randomly in 3 dimensions around the location of the detent (Figure
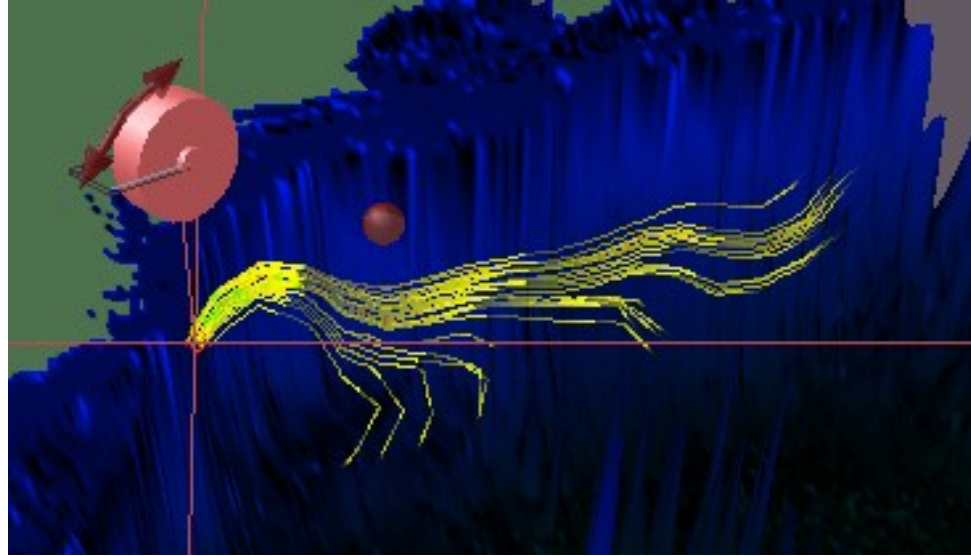
Figure 18: Cluster streaklet emitter tool.

18). If the detent is above water, the source is taken to be the water surface and streaklets originating above water are reflected below the water surface. For this tool, the volume control mechanism affects the total number of streaklets being displayed while the emission frequency remains constant.

Virtual Dyepots

A virtual dyepot continuously produces a stream of particles which act as dye injected into the flow model. Emission rate and origin location can be adjusted using the chameleon handle. This tool has the same four volume control mechanisms as were described with the streaklet emitter.

As with the streaklet emitter tool, the rake dyepot variation of this tool creates a string of dyepots aligned vertically, with the top-most dyepot at the surface of the water, and the bottom-most dyepot being either at the detent or at a depth equal to the height of the detent. Other emitters are arranged linearly between the top-most and bottom-most elements.

Figure 19: Rake dyepot tool.

Non-Chameleon Tool Types

Two types of tools breaking with the chameleon principles were also implemented. They followed the same rules for creating, moving, and deleting, but were not adjustable via the chameleon handle. These tools were intended to provide context, if needed, for the other tools by giving the user a different means of visualizing the flow data.

Streaklet Field Tool

The streaklet field tool is used to show the overall pattern of flow, displaying a horizontal range of streaklets distributed above and below the depth of the detent (Figure 20). The tool can be positioned at any depth underwater to chose a horizontal slice of flow for display, and

Figure 20: Streaklet field tool being adjusted.

the F1 key scans through high density display, low density display, and no visible display.

The user can neither create nor delete this tool, but can position the detent with the constraint that motion falls along a line aligned with the right front corner of the haptic workspace, with stops at the surface of the water and at the deepest point in the flow data. In case there is land in that corner, a depression is made in the haptic "floor" to allow the stylus all the way down to the bottom. Numerical depths in meters are shown next to the detent for the top and bottom depths of the highlighted slice. As the depth increases, the size of the range displayed increases as well. More on this in Chapter III, under Depth Mapping.

Callout Window Tool

The callout window tool displays flow velocities and directions throughout the water column for a given latitude and longitude (Figure 21).

Figure 21: Callout window tool in use.

This information is shown in a sub-window containing two graphs: the total speed of the flow at each available depth is on the left of the window, and an indicator for direction at the corresponding depth on the right. If the speed or directional lines cross the border of the sub-window or if the directional lines cross over onto the velocity graph, they are colored red as an indication that clipping is occurring and the actual value is greater than can be shown (Figure 22).

23

The velocity portion of the sub-window has horizontal depth indicator lines drawn at 100, 500, 1000, 2000, and 3000 meters. The density of these reference lines increases deeper in the data, reflecting the non-uniform sampling in the data[1]. There are also vertical lines indicating velocities of 0, 100,
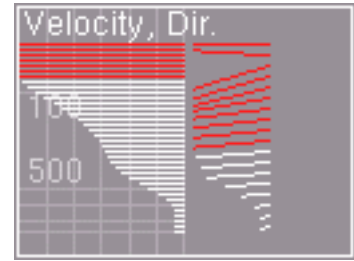
Figure 22: Detail of callout window data.

200, 300, 400, and 500 centimeters per second. The edge of the frame, and clipping, falls at 600 cm/s.

The direction portion of the sub-window shows the direction of the flow for every other data cell as if directly overhead looking down at the ocean. Clipping on this window happens at the edge of the allocated space rather than at a fixed speed.

Creating, Moving, and Deleting Tools

To instantiate a flow visualization tool the user clicks the mouse wheel button. Scrolling the mouse wheel at this point transforms the head of the chameleon tool into different tools, such as a dye pot emitter, or streaklet emitter. A three finger grip instantiates the selected tool type, while clicking the mouse wheel button again ends the tool instantiation session.
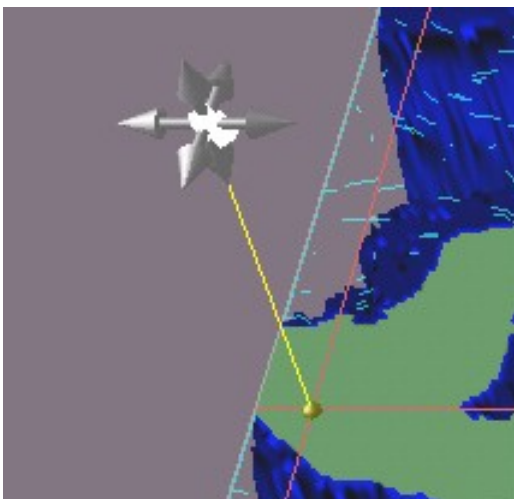
Figure 23: Moving an element.

Once a flow visualization tool is created, the three finger grip puts it into positioning mode, and a set of translation arrows are added. The stylus transforms into an icon of arrows aligned with the 3D coordinate axes (Figure 23).

---

1   This is covered in more detail in Chapter III, Depth Mapping.

Because controlling motion in free space is inherently difficult, motion in the virtual world is constrained by the sides of the virtual box containing the workspace and the haptic "floor." In this way, the chameleon handle tool is forced to remain on the screen and prevented from being in the regions "below ground".

Moving a tool which has already been instantiated requires attaching to the tool's detent, and applying the three finger grip. The tool then transforms to the 3D arrow icon shown in Figure 23. Releasing the three finger grip anchors the tool, and restores its appearance to that of the initial visualization tool.

Deleting a tool is done by transforming the chameleon handle into a clipper-like object. This is done by snapping to the detent of a tool, holding the shift key, and applying the three finger grip (Figure 24). The tool is removed, and the stylus transitions back to a pincer.
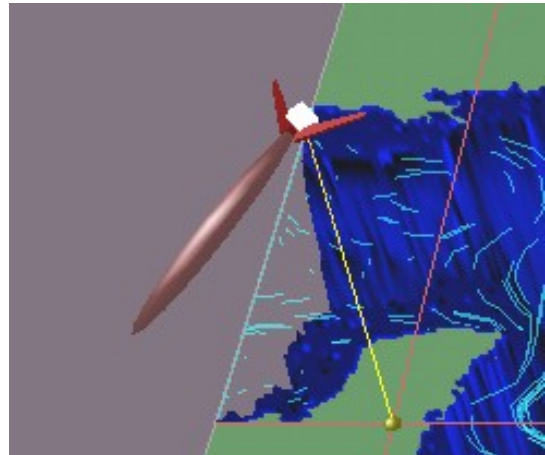
Figure 24: Deleting an element.

Haptic View Controls

A set of fixed controls, permanently instantiated in the scene, are used to control the view angles. Figure 25 gives a view of the overall design, highlighting the view control tools. Pitch is controlled by the haptic detent shown as a red sphere at A in Figure 25. If the user grasps the control detent with the gripping tool, the chameleon tip is transformed into a pair of 3-D arrows showing the constraints on movement. A three finger grip
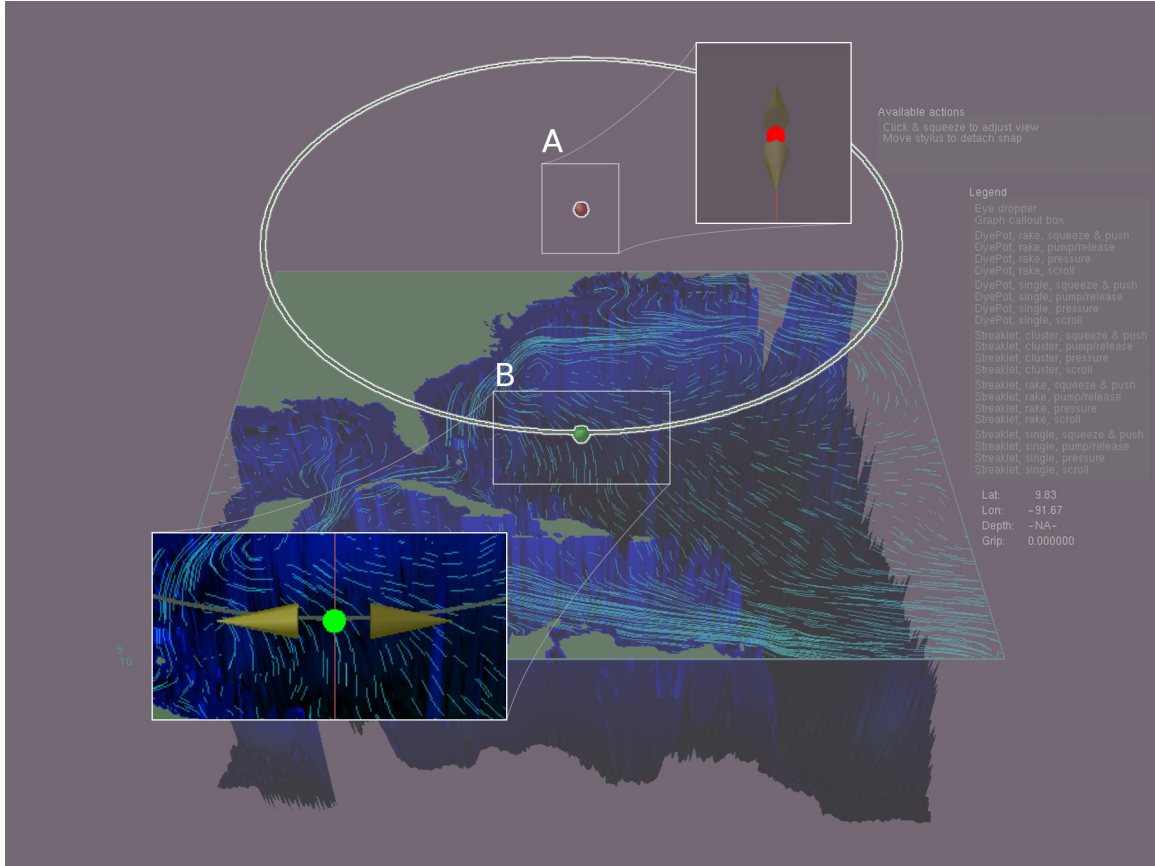
Figure 25: Interface highlighting the haptic view control elements. Controls include pitch (A), and yaw (B). The appearance of the tools in their active state is shown in highlighted windows. Note that the controls are shown highlighted for the sake of visibility.

allows the user to rotate the scene with stylus motion constrained to follow the arc. Pushing rotates the world so the user is looking more edge-on, and pulling rotates so that the user is looking more perpendicular to the plane. When the user releases the three finger grip, the control operation is finalized and the detent is again stationary. At this point, the user can detach from the detent, and move on to other tasks.

Yaw is controlled in a similar way. The user grabs the detent on the edge of the ring, which changes the handle into a horizontal pair of arrows. Motion is constrained to fall on the ring. The haptic and visual feedback combine to give the user the sense that they are directly manipulating a turntable containing the virtual world.

26

Interface State Transitions

The various transitions between the chameleon handle states are illustrated in Figure 26. In this diagram, each of the nine possible states is shown as a node. Edges represent potential user actions, with corresponding changes to both the visual appearance and functional behavior of the chameleon handle. Images next to the node give examples of the appearance of the chameleon handle in different states. For the sake of clarity, only the single streaklet emitter with the scrolling adjustment is shown associated with SELECT_TOOL_KIND, CREATE_TOOL, and ATTACHED_NOT_GRIPPED.
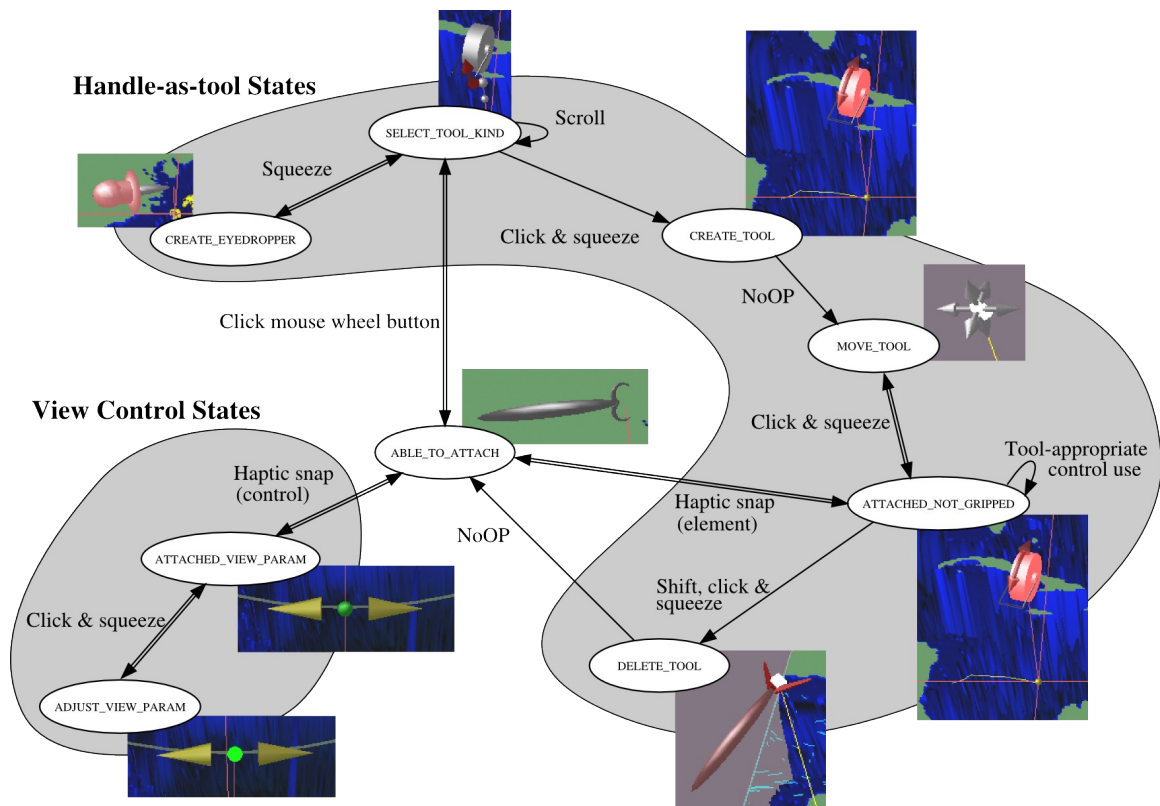
Figure 26: System states and transitions, together with the visual appearance that accompanies each state. The single streaklet emitter with the scrolling mechanism is shown, but the other tools have similar behavior.

The default state is ABLE_TO_ATTACH, which is the state when the tool is a grabber, and not attached to anything. From this state the user can move through the workspace, which remains in the default state, or can change state by attaching to the detent of a control tool, attaching to the detent of a visualization tool, or by clicking the mouse wheel button to begin the process of instantiating a visualization tool of some kind.

If the user attaches to the detent of one of the haptic view control tools, the system transitions to ATTACHED_VIEW_PARAM which activates the detent, generating moderate forces to prevent the stylus from being moved. The tool tip takes on the appearance of arrows indicating the appropriate motions. In order to change the view, the user performs a three finger grip, causing the system to transition to ADJUST_VIEW_PARAM which allows the user to move the detent, and which "lights up" the detent like an LED to indicate that it is active. The motion of the detent is constrained to move along a circle while in this state, and motion along the circle causes a corresponding rotation of the workspace. Releasing the three finger grip locks the view position and transitions back to the ATTACHED_VIEW_PARAM state, from which the user can return to ABLE_TO_ATTACH by pulling the stylus away from the detent.

On the other hand, if the user attaches to the detent of an existing visualization tool the system moves to ATTACHED_NOT_GRIPPED, binding the stylus to the detent with moderate force. At this point the tool can be adjusted using the appropriate control inputs, as described in Chapter III, under Adjustable Emission Tools. Alternately, the user can apply the three finger grip to change the system state to MOVE_TOOL, which allows the user to move the detent anywhere in the workspace that the stylus is permitted to go. Releasing the grip anchors the visualization tool in a new location and transitions back to

ATTACHED_NOT_GRIPPED. If the user holds the shift key and performs the three finger grip, the system transitions momentarily to DELETE_TOOL, which destroys the visualization tool and transitions automatically to ABLE_TO_ATTACH.

From ABLE_TO_ATTACH the user can also click the mouse wheel button, which transitions to SELECT_TOOL_KIND. In this state scrolling the mouse wheel causes the chameleon handle to cycle through the available tool types.

If the state is SELECT_TOOL_KIND and the currently selected tool is the eyedropper, simply gripping the force sensors moves to the CREATE_EYEDROPPER state, causing particles to be emitted into the flow model. Releasing the force sensors returns the system to SELECT_TOOL_KIND. If the currently selected tool is anything other than the eyedropper, and the three finger grip is applied, the system moves to CREATE_TOOL, causing the currently selected tool type to be instantiated, then immediately transitions to MOVE_TOOL, allowing the user to position the tool. As described above once the three finger grip relaxes, the system transitions into ATTACHED_NOT_GRIPPED.

Conclusion

The most "pure" instances of the haptic chameleon concept outlined in the introduction are the grabber tool, the eyedropper, and the various flow control mechanisms. Experience with the system suggests that detents are extremely useful in supporting transformation to the various tools and connection to the view control handles. Squeezing as a method of changing to a moving state seems natural, although a two finger grip

would probably be preferred to a three finger one. Squeezing is also a natural way of increasing the rate of flow.

One of the design goals was that users would not become confused about the state of the interfaces because of the clarity of the form-follows-function design. As shown in Figure 26, the system has nine states, yet informal observations with a number of users suggests that state confusion is relatively rare, and what there is diminishes rapidly with experience. When the user is controlling the view pitch control widget, for example, they are never in doubt of the effect of a hand motion. When they are holding the eyedropper tool, there is no doubt about the consequences of squeezing.

Some limitations to the chameleon handle concept became apparent through the iterative design process. Ideally the chameleon handle should be the only tool needed to use the system, but in practice it is extremely difficult to design tools which need nothing else. For example, when removing a visualization tool, the user currently presses the shift key, which breaks with the pure haptic concept. This could be replaced by having a trash can, in which users can place unwanted tools for deleting.

The four different flow control mechanisms were implemented in order to compare design alternatives, and a final design would only use the most successful of them. Chapter IV, User Testing, describes a user study in which these alternatives are evaluated.

CHAPTER III

HARDWARE AND SOFTWARE ARCHITECTURE

This section details the physical hardware platform and the major software components of the research prototype. While Chapter II, The Design of the Prototype, described the user experience of the operation of the system, this chapter describes the underlying system used to produce those experiences.

Hardware Platform

The research prototype uses a desktop model SensAble PHANToM (Sens06), a 1280 by 960 pixel stereo display, and our enhancements to the haptic input (Figure 27).

The PHANToM is a commercially available haptic input device that collects data with six degrees of freedom and
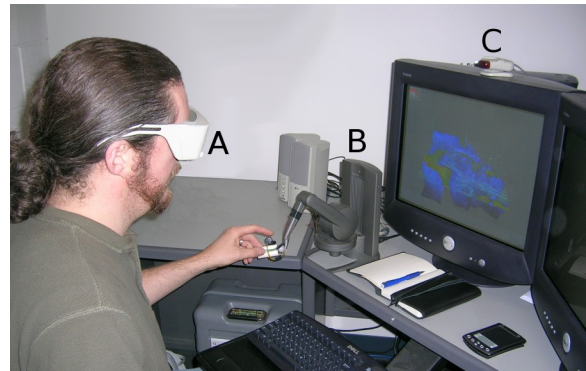
Figure 27: The system in use. The subject wears NuVision shutter glasses (A), and holds the modified PHANToM stylus (B). Resting on the monitor is the device for synchronizing monitor refresh cycles with the glasses (C). Behind the monitor is the CPU and other infrastructure.

generates forces on three. Three of the six input degrees of freedom capture the location of the stylus tip in space and the remaining three represent the stylus orientation. The

three degrees of freedom for output control forces on the stylus tip. While stylus orientation is measured, no output torques are generated.

To provide the extra degrees of freedom needed for this research, a mouse wheel from a miniature USB laptop mouse and two pressure sensors have been attached to the stylus (Figure 29).

Figure 28: The SensAble Desktop PHANToM, a commercially available haptic interface.

The mouse was constructed on two separate circuit boards: one for the buttons and scroll wheel, the other for optical tracking and logic. Our modifications consisted of cutting the outer shell to separate the wheel from the logic board, and extending the wires attaching the two boards. At present the prototype uses the 1-dimensional scrolling action of the mouse wheel and the button under the wheel.
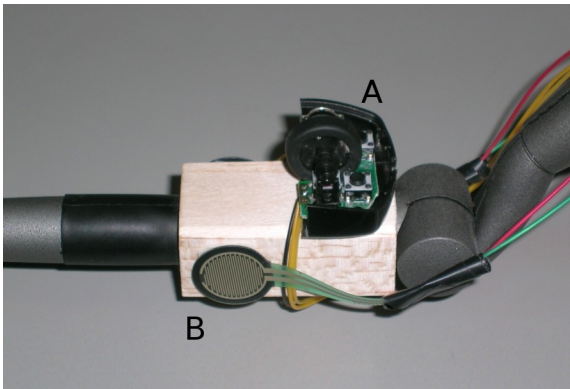
Figure 29: Modifications to the PHANToM stylus. On top is a small mouse wheel (A), on the sides are force sensing resistors (B). For reference, the force sensing resistors have a 0.75" (~19 mm) diameter.

Grip pressure is sensed using the Phidgets rapid prototyping system (Phid06), which is a modular system consisting of a controller board and a number of input and output ports. We replaced the original pressure sensors with low profile flexible force sensing resistors obtained through the US Phidgets distributor.

32

All components are controlled by a 3.2 GHz Pentium 4 computer with 1GB of memory running Windows XP. Graphics are driven by an Nvidia QuadroFX 3000 card (Nvid06), and displayed on a monitor capable of 1800 by 1440 pixels at 32 bit color and 75 Hz. For the sake of stereo rendering, resolution is reduced to 1280 by 960 pixels to achieve 120 Hz refresh rates with NuVision 60GX shutter glasses (NuVi06). Frame rates vary depending on the amount of data being displayed and the number of particles in place, varying from over 60 frames to about 5 frames per second.

Software Platform

This project was written in C++ using Microsoft Visual Studio 6.0, and built using OpenGL 2.1 (OpGL06), the SensAble GHOST 4.0 API (Ghos06), Phidgets 2.0 C++ API (PAPI06), and netCDF C++ 3.6.1 (NCDF06).

The prototype is broken into five major modules (Figure 30), one being a relatively simple control component to initialize and coordinate the other modules (Figure
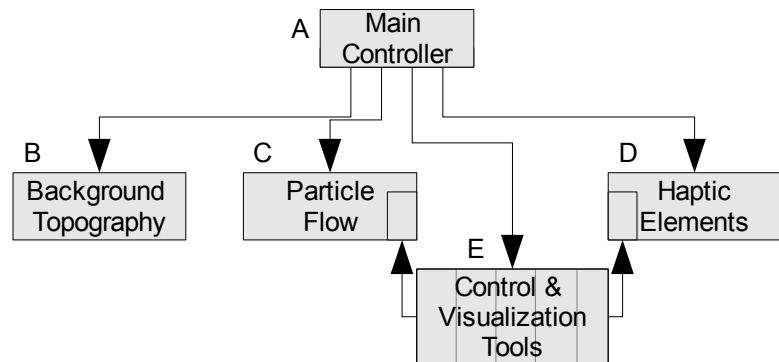
Figure 30: System architecture. Arrows indicate dependencies.

30:A), three which provide general infrastructure (Figure 30:B, C, D), and the finally the *Control and Visualization Tools* module (Figure 30:E), which is itself a collection of classes, each encapsulating the behavior of a single type of tool. These modules are described in detail in the remainder of this chapter.

Main Controller Module

The *Main Controller Module* sets up the viewing frustum, parses mouse and keyboard inputs, creates, initializes and synchronizes the other modules, and causes the display to refresh. The viewing frustum is based in part on system parameters such as the number of pixels displayed, and stereoscopic and monoscopic modes.

Background Topography Module

The *Background Topography* module handles loading and display of a given set of terrain topography data to serve as a backdrop giving geographic context to the rest of the system. This data is US Geological Survey GTOPO5, (USGS06), which is a simple height map consisting of an array of 2160 cells East to West by 870 cells North to South, each cell containing a height or depth in meters. This corresponds to six cells per degree along both North-South and East-West axes, covering the complete circumference of the Earth up to 72.5º North or South.

Background generation begins with reading an XML configuration file containing geographic bounds specified in degrees of latitude and longitude, which is used for defining the boundaries of the virtual world. Based on these bounds, the relevant subset of GTOPO5 data is loaded and used both for visual display and for constructing a haptic "bottom" to the display.

A representative section is shown in Figure 31. This section was used in the user testing. It shows the Caribbean Sea, from 45º N to 10º N, and centered on 70º W. This area was chosen because it contains many interesting features in the flow and is at least somewhat familiar to the test subjects.
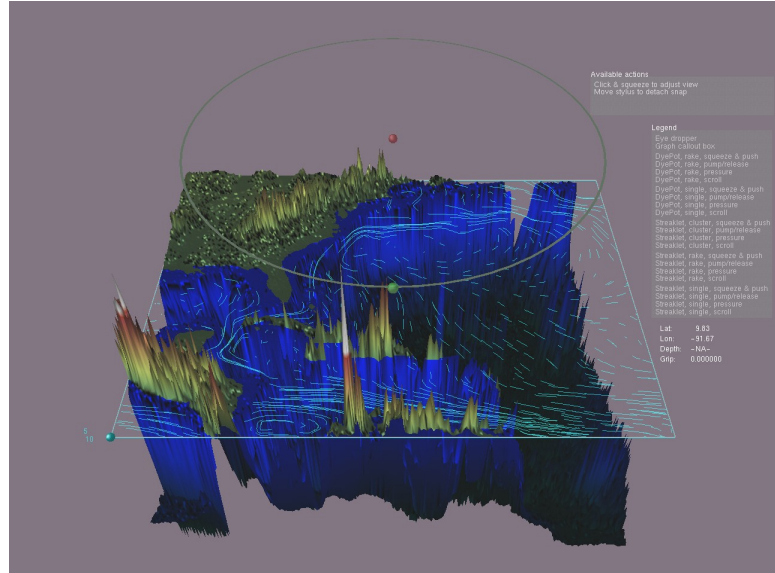
Figure 31: Interface overview. This image covers nearly 4000 Km North to South, while the deepest point in this area is around 5 Km, which means that the vertical scale of the terrain is grossly exaggerated. This image shows topography above the water because that data is loaded by the Background Topography Module. By default, features above water are not displayed.

Vertical scale is exaggerated by 220 times normal scale for the sake of visibility. This is the largest value which would be guaranteed to fit all the flow data on the screen. Given that the primary focus of this prototype is ocean flow and that displaying mountains on the landmasses often obscures flow features, values above 0 meters are by default drawn as having no topography. The F2 key toggles this to display heights at the same scale as the bathymetry.

Particle Flow Module

The *Particle Flow* module handles the creation and initialization of the underlying structures and the display of particles used for visualization. The data is stored as a 4

dimensional grid of flow vectors. This prototype relies on particles moving through the flow model over time to show the characteristics of the underlying flow. The source of the flow data was the NOAA GODAS ocean flow model used in five day weather and ocean forecasting (NOAA07).

A series of GODAS data files are read, each corresponding to a separate time. These files are in netCDF format (NCDF06) and contain a 3D array consisting of 360 cells East to West, 200 cells North to South, and 40 cells top to bottom. This covers a geographic range from 0º to 360º East to West, from 64.5º North to 74.5º South, and down to a depth of 4478 meters. Each cell in turn contains vectors for speed of flow along X, Y, and Z axes, which are taken as Δx, Δy, and Δz for the cell.

The latitude and longitude used for defining geographic bounds of the background topography, described above, are also used to define the bounds of the GODAS flow data. Because the GODAS data is an array with locations specified by index numbers while the boundaries are specified by latitudes and longitudes, tables must be constructed to convert from degrees to cell indices. The relevant subset of flow data is then loaded based on the geographic bounds. A buffer of two cells is included on the North, South, East, and West edges so flow can pass in and out of the world without generating bizarre artifacts at the edges.

Latitude Mapping

The GODAS data grid is non-linear with respect to latitude, being more closely spaced near the equator than at higher latitudes (Figure 32). Between 10º North or South is a zone of uniform high density data, with three samples per degree of latitude. Above 30º North or South are zones of uniform low density, with only one sample per degree. Latitudes from 10º to 30º are transitional regions, where data sampling changes smoothly from three samples per degree to one per degree.

Particles are traced in the grid corresponding to the GODAS data, and must be transformed into Lat/Lon coordinates for display. This is done by means of a high-resolution lookup table that converts GODAS grid coordinates to Latitude.
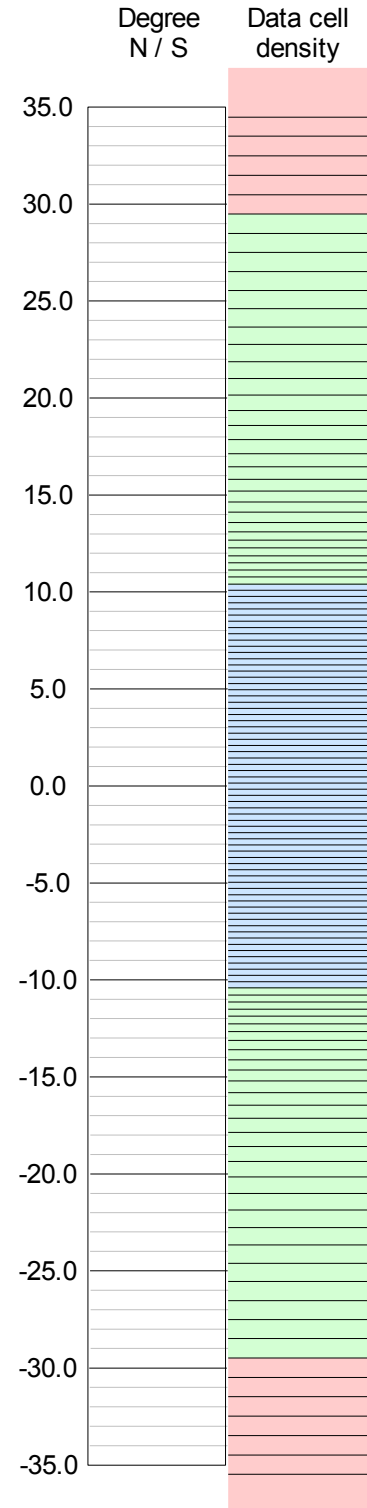
Figure 32: Latitude mapping. Latitudes are on the left. The right shows bounds of individual data cells. The high density region is blue, the low density regions red, and the transitional regions green.
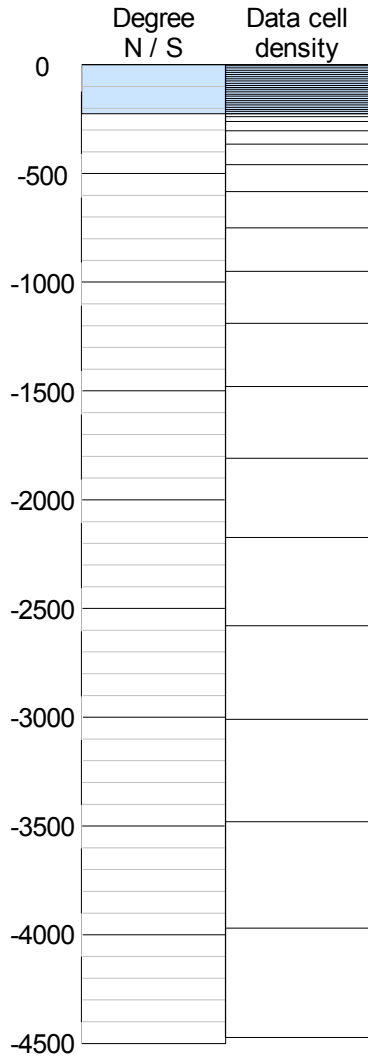
Figure 33: Depth mapping. On the left are depths in meters. On the right are the bounds of individual depth cells. The blue region at the top represents cells with a uniform spacing of 10 meters. This includes 23 of the 40 depth cells and covers depths from 5 meters to 225 meters.

Depth Mapping

Depth is also non-linear. Figure 33 compares actual depths in meters on the left with grid cell boundaries on the right. Mappings are constructed to translate from data cell index value to depth in meters and vice versa.

Resolution of Flow and Topography

The resolutions of the topography and the flow data are significantly different and this can result in inconsistencies in the representation (Figure 34).
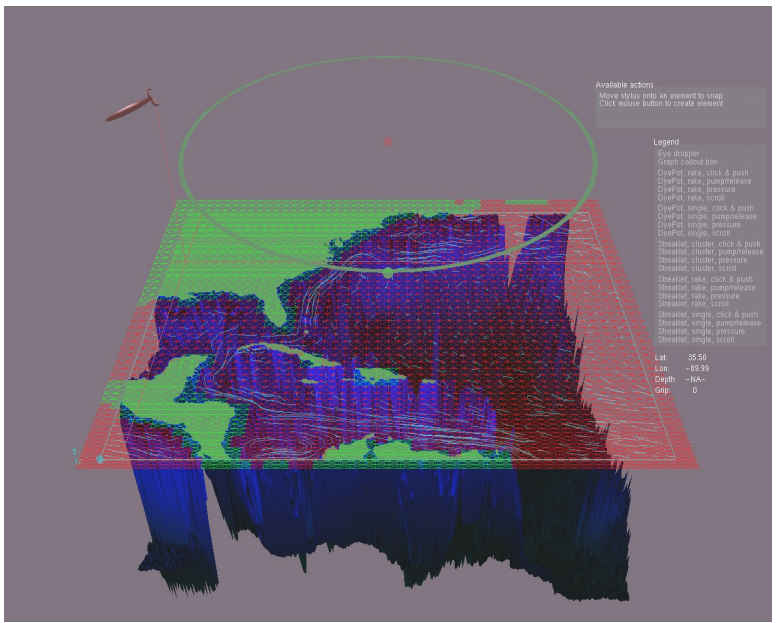
Figure 34: Comparing topographical data and flow data resolutions. Green boxes indicate cells considered to be land by the flow data, red boxes indicate water. Note that boxes become more tightly packed near the equator.

Figure 35: Topology data resolution compared to flow data resolution for Cuba, Jamaica and the tip of Florida.

Figure 35 highlights this phenomenon: the flow representation of Cuba is depicted by the four green boxes to the right of center, each of which represents one voxel of flow data at the surface. The red boxes are all considered to be water, and so ocean currents can move through what appears to be solid land. Jamaica, shown to the right of center near the bottom of the figure, has no depiction at all at the resolution of the flow data. This effect is unavoidable without more closely matched data resolutions. The resolution of the topographic data could have been down-sampled to match that of the flow, but the resulting image would hardly have been recognizable

Flow Tracing in Control and Visualization Tools

In this prototype, streaklets are used for representing flow. These are an animated representation based on pathlines. For a discussion of streaklets, pathlines, and other methods for visualizing flow see Appendix C, Flow Visualization Techniques.

There are three related but distinct types of flow tracing used in the system. One is used for the Streaklet Field Tool, one for tracing user placed tools that display streaklets, and one for user placed tools which display particles that follow in the current. While these are similar, they have important differences.

39

Particle Flow and the Streaklet Field Tool

The Streaklet Field Tool is a collection of streaklets intended to give the user an overview of flow around a given depth in the water. These streaklets are created by generating seed particles in the flow model, then projecting the motion of these particles forward and backward from their origins.

The Particle Flow Module handles creation and drawing of the streaklets used in the Streaklet Field Tool, while selecting the subset to animate is handled by the Control and Visualization Tools module, described below in the Haptic Elements Module section.

Seed Point Generation for the Streaklet Field

Streaklet generation begins with a single seed particle placed in each 3D cell representing water. These are jittered randomly in three dimensions to reduce any artifacts arising from the underlying grid structure being present in the final display. Cells on land have no flow, so particles there will never move and are ignored. If the motion for a cell causes a particle to move out of water cells onto a land cell, the particle will remain at that location for the remainder of its lifespan, effectively running aground.

Tracing for the Streaklet Field

Streaklets are curved line segments animated along pre-computed pathlines. Pathlines are computed as 3D trajectories in the flow data, starting from a particular point in space and time. At each subsequent time, a new location is generated and used as the origin for the next iteration. More detail on this process can be found in Appendix C, Flow Visualization Techniques.

The path is traced over time both "downstream" and "upstream" from the original location, then streaklets are animated along the path starting from the upstream position.

When the data structures are constructed and populated, each cell in the array is given a value taken from the GODAS data for the motion vector contained in each cell, Δx, Δy, and Δz. These are used to compute the location of the next point along the pathline. This puts the particle at a new location in space, and this process is repeated using the updated location and time until the end of the life of the particle. Once the "downstream" half of the particle is traced, the "upstream" half is done by starting again at the initial seed location, with the signs of Δx, Δy, and Δz, being reversed, working backward in time.

Note that tracing for the Streaklet Field only uses the first time slice of data, and is therefore not time varying. This is due to memory restrictions imposed by an unfortunate data structure design.

Drawing the Streaklet Field

For the sake of performance streaklets are drawn in a connect-the-dots manner, as shown in Figure 36. Drawing the streaklet begins at the first point in the lifespan of the particle and skips some interval, X, before drawing a line connecting the first point
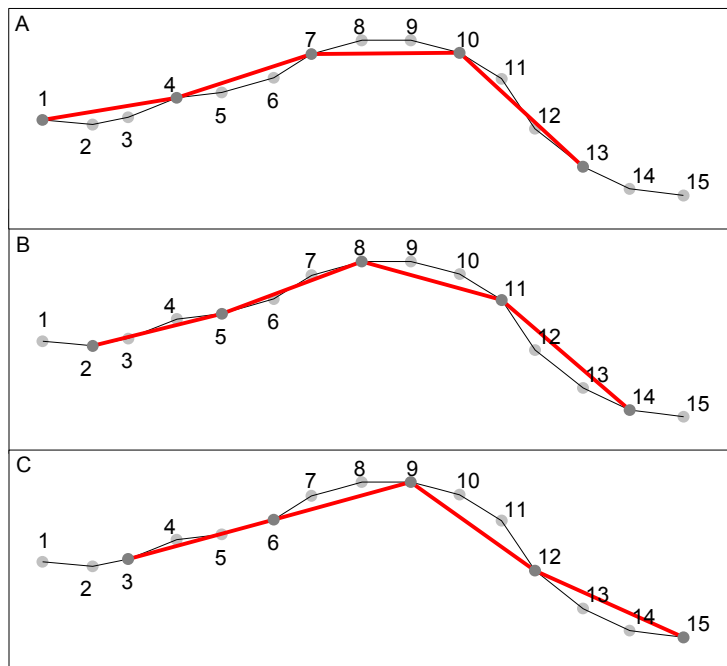
Figure 36: Drawing the flow. A) shows the initial refresh, with a skip spacing of 3. Numbers indicate the index value for a given spatial location. The red line shows the line as drawn. B) shows the second refresh, and C) the third. Drawing then begins again with A.

and the Xth. point. From this point, the same interval is skipped before another line is drawn, going from the Xth. point to the (2*X)th. point, to the (3*X)th. point, and so on for the entire length of the streaklet. Once a streaklet reaches the end of its path, a new path is computed and animation begins over.

Particle Flow and the User-Placed Tools

As mentioned above, there are two mechanisms for handling flow visualization for user placed tools: streaklet-based, and particle-based. The streaklet-based mechanism is very similar to that described above for the Streaklet Field tool, except the motion of streaklets is only computed in the "downstream" direction. These elements also take advantage of the time-varying nature of the data.

The particle-based mechanism for User-Placed tools does not use any of the streaklet tracing functions provided by the Particle Flow Module, relying instead on the motion vectors stored when the module reads the GODAS data. Each particle is simply moved by the Δx, Δy, and Δz values for the cell it finds itself in.

Haptic Elements Module

The chameleon handle concept depends on virtual objects having a physical presence, and the *Haptic Elements* module provides the pieces to use in constructing such a presence. This module provides a means to construct, manipulate, and destroy detents, a means to construct and maintain the haptic bounding box, and the means to manage the heterogeneous linked list on which all such tools are stored.

These haptic features are constructed using the GHOST API (Ghos06) to interface with the PHANToM hardware, and the haptic rendering loop is handled by that API independently of the OpenGL graphics rendering loop, with a refresh rate around 1000 cycles per second.

This module also manages the state of the chameleon stylus in its interaction with Control and Visualization tools, and updates the world in response to the user manipulating these tools.

On each screen refresh, the Main Controller module instructs the Haptic Elements module to draw itself. The Haptic Elements module in turn instructs all objects on the linked list to draw themselves. Each tool on the list handles its own visual display and, if needed, haptic rendering. Haptic rendering of a tool is only done if the stylus is attached to the tool's detent, but visual display is done as long as the tool is on the linked list.

Haptic Detents

A detent is a point in 3D space which serves as an origin around which forces are generated. When the tip of the stylus is within a certain radius from this point, forces are generated to move the tip onto the origin point. These forces are proportional to the distance from the origin, so at the edge of the sphere the pull towards the origin is the strongest, and in the center the pull is the weakest. At most these forces are still relatively weak, so that with mild exertion the stylus tip can be detached from the detent and return to moving through free space. When this occurs, the detent is removed from the system and the resources released. Should the stylus tip move back within range of some point with a detent enabled, the detent is recreated based on the parameters for the new detent.

43

Detents can also be modified in the following ways:

A normal detent has a threshold force value and the stylus snaps free if the force is above that threshold. One of the variations on detents sets this threshold force value to infinity, which prevents the user from detaching from the detent. The reason for this variation will be discussed below in the Control and Visualization Tools section, under the Raise/Lower Interface Mechanism tool.

All detents are static points in space, and once the stylus haptically attaches to the detent its origin remains stationary. Controls with normal detents can be manipulated at this point by disabling the detent, which allows the origin point to be freely moved through space to a new location within the haptic workspace. Once the relocation is finished, the detent becomes active in the new location, again constraining the stylus tip.

In addition to this generic behavior there are also two kinds of detents with constrained motions: One for following the circumference of a circle and one for following a line. In both these cases, forces are generated by the system to resist motion of the stylus tip away from the constraining shape, while motion along the shape is not resisted. As with detents in general, these constraints are created only when needed and destroyed when unneeded.

In the case of the constraint circle, a plane on which the circle falls is defined as is the origin and radius of the circle. Motion is frictionless along the circle, and is limited to a range between two stops, if specified. Attempts to move the stylus off the circle result in spring forces being generated to return the stylus to it.

In the case of the constraint line, a point through which the line passes and a vector are defined. Motion is frictionless along the line, and the stylus is constrained as above. End stops can be added if needed.

Haptic Bounding Box

This module creates and maintains a haptic workspace which prevents the stylus tip from moving outside the virtual world. The top, front, back, left, and right sides of this are simple planes. The stylus is free to move within this space, but forces are generated to prohibit motion outside this box.

The bottom of this box is constructed based on the bottom bathymetry, but at a lower resolution than the visual image. This was found to be necessary because of the exaggerated height scale of the virtual world. If the haptic resolution was the same as the visual one, the user frequently got trapped by peaks and valleys in the terrain which became quite frustrating when navigating near the bottom of the virtual world. With lower resolution on the haptic representation the user can navigate with much less difficulty.

Control and Visualization Tools

The *Control and Visualization Tools* bring together attributes of the other modules in unique ways so that each tool provides a different set of abilities. As mentioned above, these tools are stored on the linked list maintained by the Haptic Elements module, which handles adding and removing tools. The linked list also passes to the tools redraw instructions originating with the Main Controller module. Depending on the type of tool

being considered, different functions provided by the Particle Flow module are used for computing and displaying of streaklets or particles.

Of these tools, only the Haptic View Control and the Streaklet Field Tool can not be removed by the user, although the Streaklet Field Tool can be hidden.

Haptic View Control

The haptic view control tools are for controlling pitch and yaw, and are illustrated in Figure 25. These tools only take advantage of haptic detents and their ability to be moved which are provided by the Haptic Element module, and do not use any functions provided by the Particle Flow module. The tools provide handles to control the rotation of the haptic and visual displays of the virtual world in space. Both these tools take advantage the Haptic Elements module ability to constrain motion to a ring, which comes into play when the user is actively changing the view parameters, forcing the stylus to follow a circular motion while adjusting the scene rotation. Adjusting the pitch affects the axis of rotation of the yaw tool, but the yaw tool does not alter the rotation of the pitch tool.

Streaklet Field Tool

The streaklet field tool uses streaklet field data generated by the Particle Flow module. This tool displays a subset of the streaklets described in Flow Tracing in Control and Visualization Tools above. It displays one vertical grid cell worth of data centered on the current position of the control, and extending horizontally over the entire data set. The control for this tool is represented by a haptic detent in the front left corner of the virtual world, and is constrained to move along a vertical line. A depression is made in the haptic bottom to allow the control full range of motion.
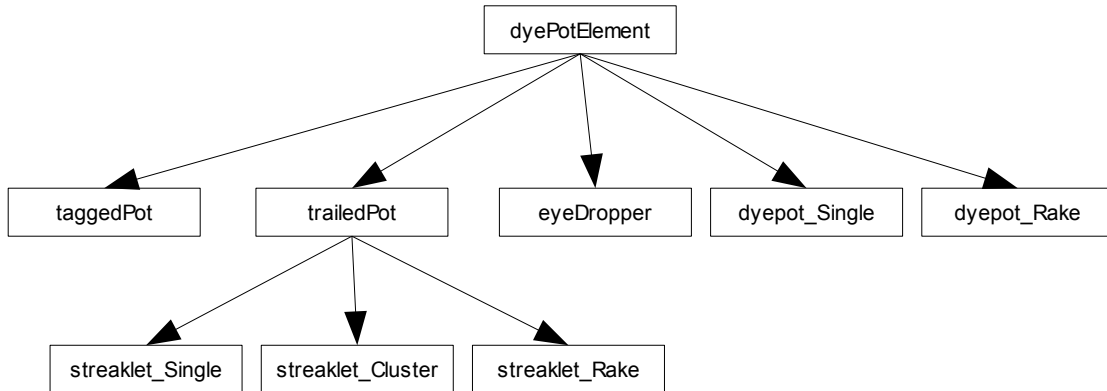
Figure 37: Inheritance structure for user placed elements.

The Adjustable Emission Tools can be created, manipulated, and deleted by the end user. These tools have the inheritance hierarchy as shown in Figure 37.

The dyePotElement class is a generic object, providing functions for creating, positioning and deleting elements, and to enable or disable haptics for the element. This is the parent class for everything that can be included on the linked list. The taggedPot class adds to the parent the ability to display a callout window, as is described below under Callout Window Tool. The trailedPot class adds capabilities to trace and draw streaklets, and is itself a parent of the three streaklet classes. The eyeDropper class is used for a single particle emitter which has no haptic detent associated with it. The remaining classes, dyepot_Single and dyepot_Rake, are used for placing objects which periodically emit particles and which have haptic detents and can be manipulated both in terms of position in the world and of the rate of particle emission.

The streaklet emitter type, dyepot emitter type, and eyedropper type are very different from the user's perspective, but are very similar in terms of the underlying software. They all make use of the streaklet or particle tracing and display functions provided by the

47

Particle Flow module. Single emitter types have one emitting location created, rake emitter types have a constant number of emitting locations, and the cluster emitter type has a variable number of emitters, up to some maximum set at initialization. The eyedropper type can be though of as a single emitter dyepot type without a haptic detent. With the exception of the eyedropper type, these types of tools make use of the detent creation, deletion and positioning functions provided by the Haptic Elements module.

Dye Emission Control Methods

The significant differences between these tools are the adjustment mechanisms used to control the amount of dye emission. Each mechanism behaves differently, but all rely on an dye control parameter (DCP) which is normalized to fall in the range of 0.0 at the minimum emission and 1.0 at the maximum.

Streaklet tools express their DCP by the number of streaklets visible over the entire lifespan of the streaklet. The emission is given by:

$$Streaklet\ count = 1 + (DCP \times 10)$$

This results in between one and eleven streaklets being visible at any time.

Dyepot tools express their DCP by releasing particles into the stream. The rate of particle emission for dyepots is controlled by changing the time interval between the emission of successive particles, with the delay between particles set by the inverse of the DCP:

$$Delay = \frac{1}{DCP}$$

Giving a rate of particle emission that is linearly proportional to the DCP. In the case that the DCP is less than 0.01, the delay is explicitly set to 101. The delay is given in units

48

defined by the graphical refresh rate, which is typically roughly 30Hz., with the result that the particle emission rate varies from 30 particles per second at the highest down to one particle every 3.3 seconds.

We will now discuss the user interface mechanisms for adjusting the DCP.

Scroll Interface Mechanism

The scroll mechanism increases the amount of emission when the mouse wheel is scrolled forward and decreases it when scrolled backward. For streaklet tools, the DCP varies from 0.1 to 1.0 in steps of 0.1 with each click of the mouse wheel. This results in between one and ten streaklets visible at any one time.

For dyepot tool types a function was chosen to gave finer control at low dye emission rates than at high rates. On each scroll wheel click, the DCP was changed as follows:

$$DCP_{new} = (\sqrt{DCP_{old}} \pm 0.0625)^2$$

This results in 16 steps from minimum to maximum over a quadratic curve:
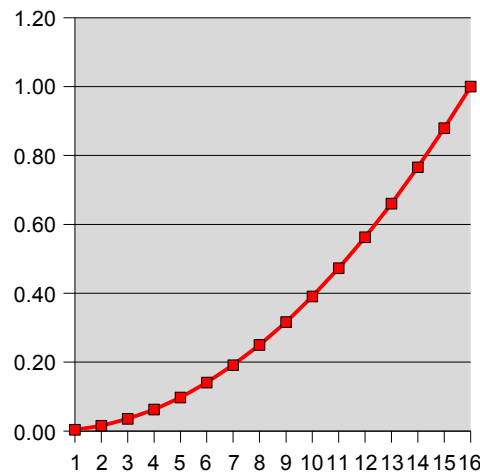
Figure 38: Scrolling mechanism dye control parameter curve.

Squeeze Interface Mechanism

The squeeze mechanism sets the emission value by simply normalizing the pressure readings from the left and right pressure sensors into the range 0.0 to 1.0:

$$DCP_{new} = \frac{sensor_L + sensor_R}{2 * sensor_{max}}$$

Pump/Release Interface Mechanism

The pump/release mechanism increments the DCP by 0.0625 per squeeze event, for a total of 16 squeeze events over the complete range. Squeeze force on the sensors has to drop back to zero before another squeeze event can be registered. Holding the mouse wheel button reduces the flow DCP by 0.005 per screen refresh, taking 200 refreshes from 1.0 to 0.0. The actual time this takes varies depending on how much is being drawn at that time, but is generally on the order of two seconds.

Raise/Lower Interface Mechanism

This adjustment mechanism uses the forces being applied to the stylus to set the emission rate. The grip force sensors act as a switch, with a normalized force of 0.5 or above activating the adjustment mechanism. The scroll wheel acts as a lock, preventing adjustment if the last scroll action was downward. If the pressure sensors are 0.5 or above and the last scroll action was not downward, the stylus is prevented from snapping off of the haptic detent and emission adjustment takes place. The force on the stylus along the Z axis is used to set the emission rate, with force upward decreasing the flow and force downward increasing it. Note that the pitch angle controls the direction of the Z axis, so "upward" or "downward" are relative to the ground plane of the virtual world.

Callout Window Tool

This tool makes use of the haptic detent manipulation provided by the Haptic Elements module, and uses the Particle Flow module to retrieve flow information for a column at a single latitude and longitude, rather than to construct particles or streaklets. This column of flow data is then presented as a graph in a callout window at the top of the display. Moving the tool to a new latitude and longitude causes the tool to refresh the column of data displayed.

CHAPTER IV

USER TESTING

The central idea in this project is that by combining force feedback with careful design and visual feedback, tools for flow visualization can be made that are easy to learn and use. This evaluation focuses mainly on the specific issue of controlling the *rate of emission* of virtual dye using haptics and visual feedback. This aspect was chosen for detailed evaluation because it was a critical aspect of the design interface and a number of design alternatives seemed plausible. As previously described, four different flow control mechanisms have been implemented and it was important to choose between them. Part of the study was also designed to test the overall ease of use of the system.

Method

The evaluation was carried out in three phases. The first of which was intended to introduce subjects to the basic operation of the interface. The second phase was a detailed comparison of four different dye emission mechanisms, and the third phase was a semi-structured interview designed to gain overall impressions regarding the ease of use of the system.

Subjects

Twelve subjects were recruited, seven of whom were researchers, predominantly graduate students, in a variety of earth-science related fields. They were familiar with 3D numerical data, although not with an interface similar to the one presented here. The remaining five test subjects were non-scientific students or university staff members.

Phase 1: Introduction to the environment

Subjects were shown the basic operation of the system, including manipulation of view controls, creation, placement, and deletion of tools, the toggling of stereo, topography, etc. They were then asked to perform specific small tasks:

- Adjusting the pitch and yaw of the view.

- Adjusting the streaklet field tool.

- Creating, moving, and deleting tools. One tool from each of the available control mechanisms was randomly chosen, placed, its operation explained in detail, and then deleted.

If the subjects had any questions about the interface or the operation of a specific tool, an explanation was given.

Phase 2: Evaluation of flow regulation methods

Of the five tool types with adjustable flow rate, three were chosen as for evaluation: the single dyepot, the rake streaklet emitter, and the cluster streaklet emitter types. The other types were commented upon by users, but only these three were used in the experiments.

Four different mechanisms for controlling the rate of emission were tested. These have been described in detail in Chapter III, Adjustable Emission Tools.

1) Scroll: The mouse wheel directly controls dye volume: scrolling forward increases volume and backward decreases it.

2) Squeeze: User squeeze pressure controls the volume of dye being emitted: more pressure increases dye volume and vice versa.

3) Pump/release: Each squeeze event increments the dye emission by one step, while holding the wheel button down reduces emission as long as the button is held.

4) Raise/lower: Squeezing the pressure sensors allows the user to raise or lower the PHANToM stylus to set dye volume, with emission proportional to the force of raising or lowering. The scroll wheel is used as a lock to enable and disable the adjustment mechanism.

Procedure

Subjects were given four instances of a single type of tool (single dyepot, streaklet rake, or streaklet cluster) each instance having a different emission control mechanism. These tools were randomly placed in four locations known to have either high flow or low flow, and the tools were initialized to a random emission level within the valid range.

Figure 39: Test matrix of tool head types and their control mechanisms.

Subjects were asked to adjust each tool in turn to its maximum emission rate, its minimum emission rate, and an approximate midpoint. After all four mechanisms had been explored, subjects were asked to rank them from easiest to use to hardest to use.

54

This same procedure was repeated for each of the tool types and current flow conditions, following a different random order for each subject.

Next a set of specific questions about the visual appearance and function of the four emission adjustment mechanisms were asked while subjects had the opportunity to interact with the tools. These questions were:

1. Do the icons help you remember how the specific tool works?
2. Do the icons show you the actions to adjust the tool?
3. Do you have good feedback as to the emission rate of the tool?

Other questions were also directly related to stylus and icon behavior:

1. Do you find in general that the icons help or interfere with your use of the system?
2. Can you tell from the display of a tool that you are adjusting it?

Finally, subjects were asked to comment freely on aspects of the flow control mechanisms, asked for suggestions to improve the design, and asked to comment on anything they found particularly difficult.

Phase 3: User comments & survey

This section probed the success of the interface as a whole in ease of use, in representing the data in an understandable way, and looked for input into successes, failures, and directions for future work.

In this phase of the experiment subjects were asked a series of questions while interacting with the system. These questions covered the interface in general, the stylus in particular, tool manipulation, the utility of the visualization tools, and prior experience with virtual

reality and topographical maps. These questions are presented in Appendix B, Survey Questionnaire.

Results

Emission control mechanisms

Because the results were similar regardless of tool type and whether the tools were in a high or a low flow region, results were combined. These results are illustrated in Figure 40. An ANOVA was run to compare the four flow control mechanisms, and the difference was

Figure 40: Mean ranking and standard deviation of all current flows and tool types. Lower is better.

highly significant, with [$F_{(3, 240)} = 4.323$, $p = 0.011$]. A Tukey Honestly Significant Difference (HSD) test showed three groups. The Scroll interface mechanism was most preferred, followed by the Squeeze mechanism and the Raise/Lower mechanism, which were not significantly different from one another. Last came the Pump/Release mechanism.

Flow adjustment icons

As part of the semi-structured interview process subjects were asked to rate the quality of the icons representing each of the

Figure 41: Number of times each mechanism fell into each rating category.

56

adjustment mechanisms from best to worst. Figure 41 shows the preference rankings as to how well a given icon serves as a reminder of the action required to adjust a tool.

Users were also asked to rate from best to worst how effectively the icons represented actions available to the user when adjusting a mechanism (Figure 44). This question differs from the previous one in that rather than addressing an icon as a memory aid, it addresses an icon as instruction on the operation of a tool.

Finally, users were asked to rate from best to worst how effective an icon was at giving the user a sense of the current emission level of a tool (Figure 45).

Stylus behavior

As part of the semi-structured interview subjects were asked to rate the hardware and user interface components of the system design. Subjects generally rated the location of the scroll wheel and the force sensing pads quite favorably (Figure 42, Figure 43). However, many subjects commented on how fatiguing or ergonomically unfriendly the handle was. One subject had tendonitis in the relevant elbow, and commented that using the system was uncomfortable. Another subject noted that people with arthritis would not be able to use the handle as designed. Several thought that gripping the stylus like a pen might be a more comfortable grip, but that only by using the interface with a pen configuration would they know for sure.
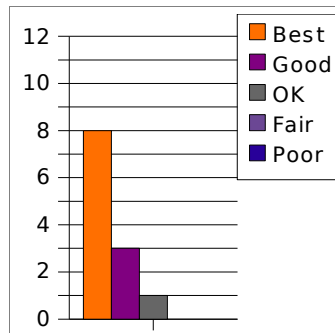
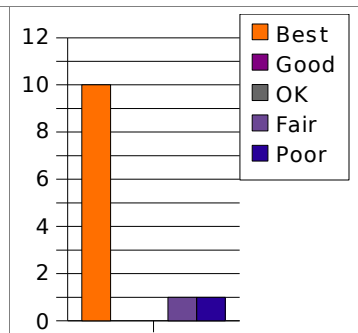*Figure 42: User rating of location of mouse scroll wheel.*

Figure 43: User rating of location of force sensors.
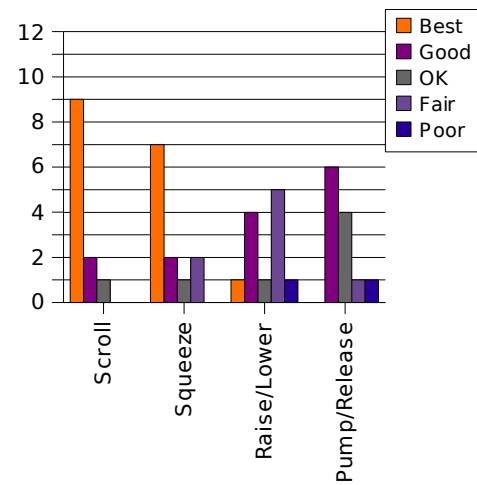
Figure 44: User rating of how well an icon depicted the actions available when adjusting a tool.
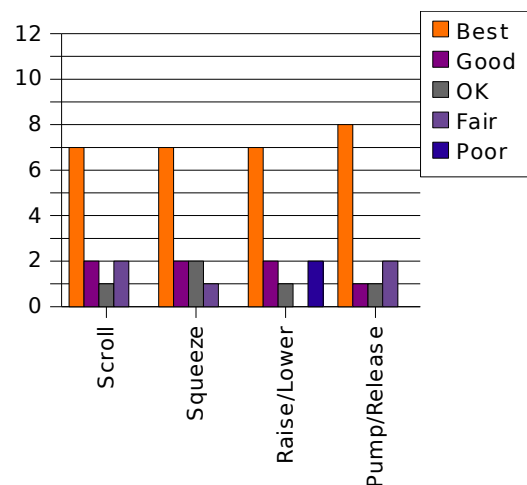
Figure 45: User rating of how well an icon depicted the emission level for a mechanism.

58

Overall, subjects were pleased with the operation of the haptic view controls and with the streaklet field control. The streaklet field itself was considered to be quite useful, especially because it could be turned on for getting a sense of the overall flow and turned off when focusing on tools within the space.

The fact that flow field and background image were vertically exaggerated was considered to be useful for understanding the flow. Two subjects requested numerical display of the exaggeration, and one of them also requested control over the amount of the exaggeration.

Subjects considered the look of the interface to be relatively clean and uncluttered, although two subjects wanted more information to be displayed, depending on the nature of the research being done.

The textual window of actions available to the user was largely ignored, and it was often the case that subjects were shown it during the initial introduction to the system, but reacting as if seeing it for the first time when asked about it. Two users commented that it would be more useful if it were larger and more graphic, akin to the weapon selection information found in many first-person shooting type games. Two users wanted the textual information to be in pop-up windows, and thereby be both movable and resizable.

All subjects were asked for suggestions on new tools or features to add to the system. These suggestions include being able to zoom in on a specific area of the flow, the ability to place tools with fully specified latitude, longitude, and depths, (more than one depth in

the case of the rake type tools) and a tool for inserting notes into the virtual world. One user proposed a window to give hints and tips about features in the system.

Tool manipulation

All subjects experienced some degree of trouble with button clicks interfering with scrolling when manipulating controls, stumbling over the fact that it was relatively easy to generate clicks when trying to scroll and vice versa.

There was consensus among the users that the interface is attractive and pleasant to use. However, the icon used when deleting a visualization tool was considered ineffective. Users definitely liked the haptic nature of detents on the tools, with one subject declaring "That's the best part." The strength and size of the detents was considered to be appropriate, although three users wanted the sizes to be either under user control or set based on the proximity of other detents.

Tool comparisons

Both streaklets and dyepots were rated as being effective at providing different views onto the data. Streaklets were considered to show more history and speed of a particle, while the dyepots were felt to show particle motions and amounts. All but two users recognized that streaklet type tools indicated upwelling or downwelling with color coding, but most did not remember which color signified which motion.

Individual differences

Early on an interesting split seemed to appear in the opinions of the test subjects. Those with scientific backgrounds were observed to prefer tools with the capacity for numerically precise adjustment, and disliked the tools which were not capable of this. On

the other hand, the non-scientific group favored the tactile and kinesthetic nature of the more haptically focused tools, preferring the feel of the tools more than the precision of them. One subject reported favoring the raise/lower mechanism because it felt analogous to a sculpting tool.
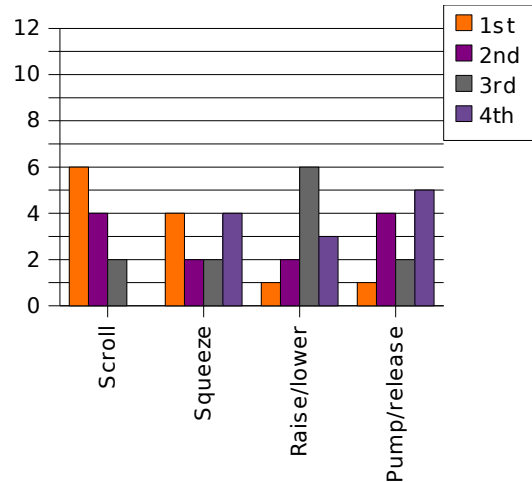
Figure 46: User preferences for the streaklet rake in the high current flow condition.

This dichotomy is best shown in Figure 46, under the Squeeze tool section. Opinions are strongly divided on this mechanism – four subjects thought it was the best available, four thought it the worst, and four were relatively ambivalent about the tool.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

At the beginning of this project, we set out to design and construct an interface onto a virtual world which took the best aspects of prior work in the field, synthesized it and extended it. We sought to incorporate Hinckley's passive haptic ideas where tools in the virtual world are represented by a similar physical proxy in the real world, and interacting with the virtual world is facilitated by interacting with the physical artifact. Added to this base is a layer inspired by both the Virtual Tricorder and the Chameleon interfaces, in which tools change their virtual behaviors depending on the needs of the moment. In this way a single physical tool can transform to become many tools in the virtual world.

These design ideas are somewhat at odds with each other. For pure passive haptics, a tool is represented by a single physical entity. This matches our intuition from the real world, in which a pencil is represented by a pencil, a screwdriver by a screwdriver, and so forth. On the other hand, a chameleon tool implies that all possible virtual tools should be represented by a single physical entity. One universal tool then can become a pencil or a screwdriver, etc. This removes a lot of clutter from the toolbox, but requires a certain suspension of disbelief when changing tools in the virtual world.

The solution we explored involved giving the chameleon tool a variety of different visual appearances corresponding to each tool's function. This was done through the following design principles: that the visual appearance of a tool should indicate its functionality,

that the passive haptics of a tool should be supported by designing the virtual tools to complement the stylus metaphor of the PHANToM, that the active haptics of a tool should be used to reinforce interaction constraints, and that there should be a natural mapping between forces on the handle and visual appearance of the tools.

Design principles

We tried to design the tools in this virtual environment to be visually simple, conceptually easy to understand, and self-explanatory in their operation. Further, we tried to separate the interaction of the thumb wheel from the interaction of the pressure sensors, and both of these from the operation of the haptic stylus. In some cases this design appears to have been quite successful, and in other cases less so.

The interaction methods designed for basic control operations of the virtual environment were relatively successful. Control of the viewing angle onto the virtual world was simple and straightforward. The users generally were able to successfully perform the tasks of creating, moving, and deleting tools in the virtual environment. The exception to this was that of changing tool heads to or from the eyedropper tool. This task changed the grip operation from a simple squeeze to a three-fingered grip and changed the use of the thumb wheel from a scrolling tool to a clicking button, which often resulted in the actual state of the system not matching the user's anticipated state.

The task chosen for detailed study and evaluation was a comparison of different virtual dye emission control mechanisms. Scrolling worked the best, perhaps because it was the most familiar action to users with scrolling mice. The direct squeeze adjustment was simple and straightforward, and while not the preferred mechanism was at least readily understood by the users. The remaining two mechanisms, pump/release and the haptic

lever mechanism, were the least successful in conveying their actions through their appearance, with corresponding user confusion. In part these designs failed because the interactions being represented were fairly far removed from everyday activities, and in part because the tools had too many states within themselves which were poorly communicated to the users. In short, the designs were unable to successfully convey status information to the user.

"Magic"

The strength of the chameleon handle approach is that any number of operations can be hidden behind a single tool handle, because the tool can become a pencil or a screwdriver as needed. The weakness of it is that this type of interaction by definition does not exactly match a user's intuition developed by years of interacting in the physical world. Any such overloading of a tool's abilities needs to be designed and implemented in a way that makes as much intuitive sense to the user as more mundane interactions. In the case of tools with real-world predecessors, this is simply a question of matching the virtual behavior to the anticipated physical behavior based on the appearance of the tool and the user's expectations. For tools with multiple abilities, however, this can be a difficult design problem.

Part of the challenge of designing virtual tools is that tools with no real-world analogue can be created. These can be very powerful in that they can be designed to perform "magical" functions, like creating dye-emitting elements from thin air. They can be difficult to use because they lack grounding in a real-world tool, and users may have no intuitive feel for their operation. These magical interactions are a key advantage of working in the virtual world, so designing their actions will be a critical part of future

work if the technology is to make it out of a research environment. The challenge for the designer is to find or construct extensions of real-world actions which make logical sense to the users and so make these magical interactions feel commonplace.

As more research is done in this field, rules of thumb will emerge as they did for the 2D computer interface world. For example, in a 2D interface when a user clicks on the scrollbar slider by the side of a text window and moves it up or down the accompanying text scrolls accordingly. In the 3D haptic world, there is not yet such a collection of practices.

Some of the techniques designed and implemented in this research may be good candidates for standardized haptic interfaces of the future. Such actions are relatively intuitive and can make the jump from the real world to the virtual one easily. These include the controls used to change the angles of the world in this prototype – the user simply grabs the control and pushes or pulls. Other controls, such as volume of dye emission, could follow a number of possible metaphors and therefore require more thought to find the "right" solution.

Successes and failures

Users of the system were able to come to it with no prior experience with virtual reality of any kind, and within a few minutes of trial and error exploration of the ocean flow model could gain understanding of the currents and bathymetry of the region displayed. With this in mind, the system as a whole can be seen as a success. Within the prototype system, however, there were four variations on controlling the emission of the tools in the virtual space, and these had differing degrees of success. In a production system, there would most likely be a single mechanism based on the most successful of those tried in the prototype. This was the scroll wheel.

The three fingered grip was the most problematic part of the system. It was intended to be a virtual equivalent to the real world's pinch grip, which is an extremely common means of interacting with physical artifacts. Anytime someone picks up a fork or a pencil they most likely use a two fingered pinch grip. The third finger on the scroll wheel was the cause of the difficulty. As implemented in this system it was in a relatively awkward physical position on the stylus handle. This would be easy enough to fix by moving the sensor pads to a better location, but a solution based on a two fingered grip would be preferable.

The three fingered grip was initially introduced as a means of allowing the user to push or pull virtual handles to change the flow rates, without changing the position of the tool being adjusted. Because the scroll wheel was seen to be the most favored emission control mechanism over all of the push-pull mechanisms, the three finger grip is not actually needed.

Future work

We conclude with some suggestions regarding possible research should the development of this system continue. First the proposed system without a three finger grip should be implemented and evaluated.

One of the major limitations of the existing prototype is that it is limited in how users can interact with the environment. The viewpoint control is relatively primitive, allowing the user only to pitch the world forward and backward and to rotate the world around an axis perpendicular to that world plane. The center of the view is always the center of the world, and there are no controls to allow the user to move that center of view. Extending the system to include tools capable of panning the view left and right or up and down, or capable of zooming in, for more detail, and out, for a more global perspective, would be useful. One possible future path would be enhancing the augmented stylus to include navigational tools capable of such maneuvers. Such experimentation should make use of data sets with higher resolutions however, because the current ones do not have the underlying data to support getting much closer.

Other future work might include exploring the use of the haptic chameleon interface in other application areas. These could include nearly any field where an existing physical structure is being explored in virtual space. An example might be a 3D virtual sculpting tool, where shape was specified using the haptic stylus. Virtual material could be removed with cutting tool heads that could be changed by scrolling the thumb wheel. Cutting speed might be controlled via the pressure sensors. Or virtual material could be added with a toothpaste-tube-like emitter, or pushed around with pressure-sensitive virtual trowels.

67

Another possible application could be keyframe animation, where the stylus is used to grab a joint on a virtual armature and pull it into position for the next frame. In this case, grip pressure could map to a physical characteristic of the character being animated, like body tension. The harder the user grips the pressure sensors, the more effort the animated character is putting behind the motion being animated.

Doctors performing remote surgical procedures would likewise benefit from such interfaces. A single haptic tool could become a scalpel, a suction tool, endoscope, or any other tool needed for a given surgery.

When all is said and done, the world of haptic interfaces is a very young and emerging one. While this project can not possibly have provided all solutions in a field that is hampered by the technical limitations of current haptic devices, we believe that important questions have been posed, and while perhaps not answered conclusively, at least answers were hinted at which future research will either confirm or disprove. In passing, other questions have been raised which will hopefully provide grist for future students of the art.

BIBLIOGRAPHY

Avil99: Aviles, Walter A and Ranata, John F, Haptic Interaction with Geoscientific Data, 1999

Baxt04: Baxter, William and Lin, Ming C., Haptic Interaction with Fluid Media, 2004

Broo90: Brooks, Frederick; Ouh-Young, Ming; Batter, James; and Kilpatrick, Jerome, Project GROPE: Haptic displays for scientific visualization, 1990

Burd00: Burdea, Grigore C., Haptic Feedback for Virtual Reality, 2000

Carl06: Carlson, Wayne, A Critical History of Computer Graphics and Animation, 2007, http://accad.osu.edu/~waynec/history/lesson17.html

Fish89: Conn, Coco; Lanier, Jaron; Minsky, Margaret; Fisher, Scott; Druin, Allison, Virtual Environments and Interactivity: Windows to the Future, 1998

Fitz94: Fitzmaurice, George and Buxton, William, The Chameleon: Spatially Aware Palmtop Computers, 1994

Ghos06: SensAble Technologies, Inc., PHANTOM and GHOST: GHOST Overview, 2006

Glas07: Glass, J., The Haptic Community Web Site, 2007

Hern94: Herndon, Kenneth; van Dam, Andries; and Gleicher, Michael, The Challenges of 3D Interaction: A CHI '94 Workshop, 1994

Hinc97: Hinckley, Ken; Pausch, Randy; Downs, Hunter; Proffitt, Dennis; and Kassell, Neal, The Props-Based Interface For Neurosurgical Visualization, 1997

Ikit03: Ikits, Milan; Brederson, Dean; Hansen, Charles; and Johnson, Christopher, A Constraint-Based Technique for Haptic Volume Exploration, 2003

Iwat93: Iwata, Hiroo and Noma, Haruo, Volume Haptization, 1993

Lawr00: Lawrence, Dale; Lee, Christopher; Pao, Lucy; and Novoselov, Roman, Shock and Vortex Visualization Using a Combined Visual/Haptic Interface, 2000

Lawr04: Lawrence, Dale; Pao, Lucy; Lee, Christopher; and Novoselov, Roman, Synergistic Visual/Haptic Rendering Modes for Scientific Visualization, 2004

NCDF06: Unidata, NetCDF (network Common Data Form), 2006, http://www.unidata.ucar.edu/software/netcdf/

Nesb01: Nesbitt, Keith; Gallimore, Randall; and Orenstein, Bernard, Using Force Feedback for Multi-Sensory Display, 2001

NOAA07: Climate Prediction Center Internet Team, NCEP Global Ocean Data Assimilation System (GODAS), 2007

NuVi06: MacNaughton, Inc., Stereoscopic Viewing Solutions: 60GX Wireless Stereoscopic Glasses, 2006, http://www.nuvision3d.com/the60gx.html

Nvid06: NVIDIA Corporation, High-End Products, 2006, http://www.nvidia.com/page/quadrofx.html

OpGL06: OpenGL.org, What's New in the Latest Version of OpenGL, 2006, http://www.opengl.org/documentation/current_version/

PAPI06: Phidgets Inc., Unique and Easy to Use USB Interfaces, 2006, http://www.phidgets.com/modules.php?op=modload&name=Sections&file=index&req=viewarticle&artid=76&page=1

Paya02: Payandeh, Shahram and Stanisic Zoran, On Application of Virtual Fixtures as an Aid for Telemanipulation and Training, 2002

Phid06: Phidgets USA, USB sensors, USB robot sensors, robot kits, hobby robots, robot parts, 2006, http://www.phidgetsusa.com/

Sens06: SensAble Technologies, SensAble, 2006, http://www.sensable.com/

Srin97: Srinivasan, Mandayam A. and Basdogan, Cagatay, Haptics in Virtual Environments: Taxonomy, Research Status, and Challenges, 1997

Tsan02: Tsang, Michael; Fitzmaurice, George W.; Kurtenbach, Gordon; Khan, Azam; and Buxton, Bill, Boom Chameleon: Simultaneous Capture of 3D Viewpoint, Voice and Gesture Annotations on a Spatially-Aware Display, 2002

USGS06: U.S. Geological Survey, Earth Resources Observation and Science, 2006, http://edc.usgs.gov/products/elevation/gtopo30/gtopo30.html

vRei03: van Reimersdahl, Thomas; Bley, Florian; Kuhlen, Torsten; and Bischof, Christian H., Haptic Rendering Techniques for the Interactive Exploration of CFD Datasets in Virtual Environments, 2003

Ware04: Ware, Colin, Information Visualization: Perception for Design, 2004

Wiki06: Wikipedia contributors, Streamlines, Streaklines and Pathlines, 2007, http://en.wikipedia.org/w/index.php?title=Streamlines_streaklines_and_pathlines&oldid=97215956

Wlok95: Wloka, Mathias and Greenfield, Eliot, The Virtual Tricorder, 1995

APPENDICES

APPENDIX A

REVIEW BOARD APPROVAL

IRB Approval

APPENDIX B

SURVEY QUESTIONNAIRE

Stylus:

1. Do you like holding the stylus from the top? Would you prefer a grip like a pen? Is there some other grip you would prefer?

2. Is the location of the scroll wheel comfortable and easy to use?

3. Are the locations of the squeeze pads comfortable and easy to use?

4. Does the handle feel bulky or awkward?

5. Do you feel that you are in direct control of the tools in virtual space, or is the computer putting a layer of some kind between you and the tool?

6. Are any actions in the system surprising or confusing?

General interface:

1. Do you find the interface easier to use when stereo is enabled? (Stereo is disabled at this point, and the subject asked to create a callout tool at a random location, and detach from it. The subject was asked to reattach to the tool and position it at a randomly selected latitude, longitude, and depth, then detach from it. Finally the subject is asked to reattach to the tool and delete it. This process is repeated with stereo enabled.)

2. Do you think the latitude, longitude, and depth values on the right side are useful?

3. Do you like the way the red ball (pitch) control looks and operates?

4. Do you like the way the green ball on the ring (yaw) control looks and operates?

5. Do you like how the blue control for ocean current depth looks and operates?

6. Do you like how the blue ocean current itself looks?

7. Which level of ocean current display is the most useful?

8. Does the vertical scale exaggeration interfere with your understanding? Does it help? (The reasoning behind the vertical scale exaggeration will have been explained during the introduction.)

9. Do you find the textual tool tips in the upper right corner useful? Or do the icons display that information graphically?

10. What tools, if anything, would you like to see added to the interface?

11. Does the display feel cluttered? Would you like to see other information displayed as well?

Tool manipulation:

1. How could the icon used for moving a tool be improved?

2. How could the icon used for deleting a tool be improved?

3. Can you tell when the button is clicked?

4. Can you tell when the pressure sensors are gripped?

5. Can you tell when you are attached to a tool?

6. Do you find selecting a tool type confusing?

7. Are the attachment points too easy or to hard to snap off of?

8. Should the radius of the "grab" from the attachment points be larger or smaller?

Tool comparisons:

1. Do you find the streaklet type tools to be useful?

2. Do you find the dyepot type tools to be useful?

3. Do you find the streaklet tools and the dyepot tools convey the same information? If not, how do they differ?

4. Can you find areas of upwelling or downwelling?

5. Do you feel you have a better understanding of ocean flow in the area shown?

User experience:

1. How often do you play video games?

2. Have you used virtual reality systems in the past?

3. Do you consider yourself to be tech-savvy?

4. Have you used a haptic feedback system in the past?

5. Have you ever used shutter glasses before? Used any kind of 3D viewing setup?

6. Have you ever used a topographical map before?

APPENDIX C

FLOW VISUALIZATION TECHNIQUES

Because the application domain is ocean flow, a brief description of some techniques used to reveal motion in flow is in order here. While the focus of this project is on user interaction with haptic interfaces, a basic understanding of flow visualization is needed before proceeding. There are a number of different ways of visualizing motion through a flow, and each has benefits and drawbacks. (Ware04), (Wiki06). The various techniques are shown graphically in Figure c47.

1) A *free particle* is a point in the flow advected from one moment to the next, for example, a piece of driftwood carried on the ocean. No trail is generated.

2) A *pathline* is the path traced by a particle as it moves through space in a time-varying flow field, such as the path traced by a piece of driftwood carried on the ocean.

3) A *streaklet* is a small section of a pathline animated along the pathline as time progresses. It is a free particle that shows the local pathline at a given instant.

4) A *streakline* is the line defined by connecting a series of free particles that have been emitted from a fixed point over time into a flow field, for example, the trail of smoke from a cigarette.

5) A *streamline* is a curve tangent to the velocity vector of the flow at a given point in time. In other words, it is a line starting at some source point and running through the flow, but only at a given instant.

In steady-state flow, which does not vary with time, pathlines, streaklines, and streamlines are equivalent.
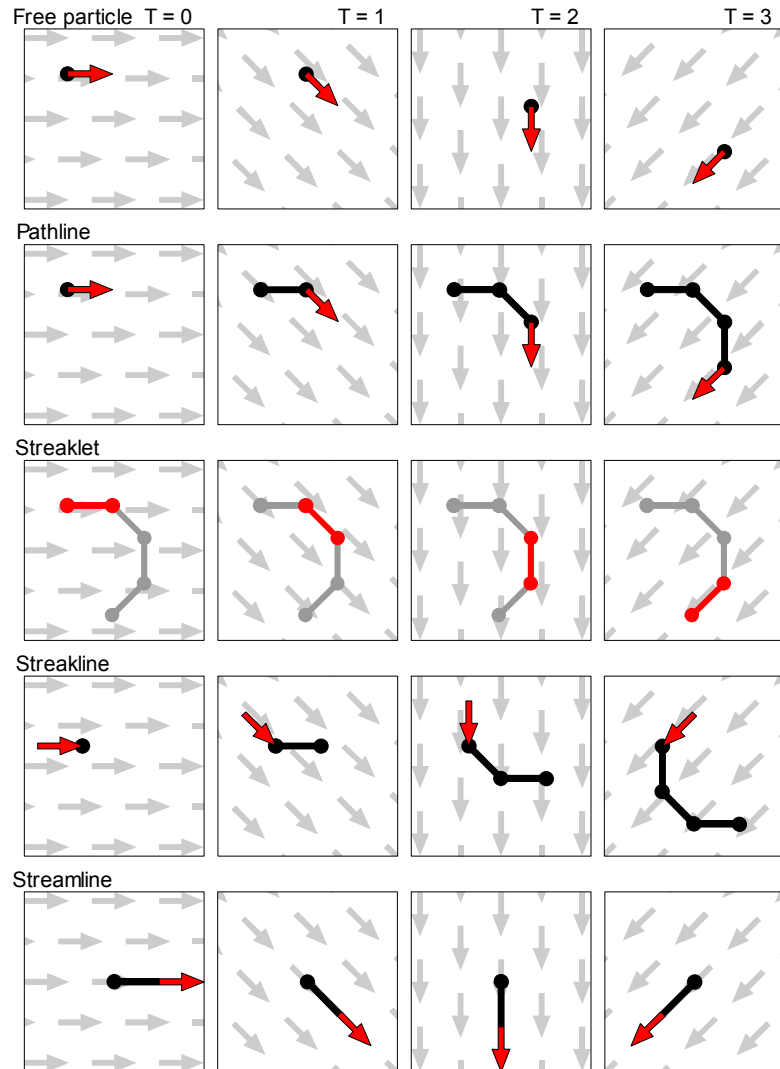

Figure c47: Differences in showing flows. Free particles simply move with the flow. Pathlines trace a particle from an initial location, leaving a fixed trail behind it. Streaklets animate a section of a pathline. Streaklines move the trail, leaving the particle fixed in space. Streamlines indicate the current direction of flow, while the origin remains fixed in space.

This interface design uses either free particles and streaklets for all the visualization needs, with the other types described here for context. Free particles were chosen as the closest analogue to "particles" of dye, as the particles would flow and diffuse in the current. Similarly, streaklets were chosen as the most reasonable way of showing motion while retaining a visual history of the path of the particle.

77