

HAPTIC-GEOZUI3D: EXPLORING THE USE OF HAPTICS IN AUV PATH PLANNING

Rick Komerska and Colin Ware
Data Visualization Research Lab
Center for Coastal and Ocean Mapping
University of New Hampshire
Durham, New Hampshire
komerska@ccom.unh.edu
colinw@cisunix.unh.edu

ABSTRACT

We have developed a desktop virtual reality system that we call *Haptic-GeoZui3D*, which brings together 3D user interaction and visualization to provide a compelling environment for AUV path planning. A key component in our system is the PHANTOM haptic device (SensAble Technologies, Inc.), which affords a sense of touch and force feedback – *haptics* – to provide cues and constraints to guide the user’s interaction. This paper describes our system, and how we use haptics to significantly augment our ability to lay out a vehicle path. We show how our system works well for quickly defining simple waypoint-to-waypoint (e.g. transit) path segments, and illustrate how it could be used in specifying more complex, highly segmented (e.g. lawnmower survey) paths.

INTRODUCTION

AUV path planning is typically done today using either commercially available 2D visualization packages designed for surface ship hydrographic survey planning, or with custom in-house applications having limited interaction capability. Although many aspects of current AUV mission design are relatively limited in nature and capable of being handled with 2D planning tools, the requirement for specifying more complex AUV missions requiring 3D interaction is growing. For example, mapping a large-scale plume event will likely require the planning of complimentary routes for multiple AUVs working together within the 3D water column.

Given this need, it would seem that working in a 3D virtual reality (VR) environment would provide a more intuitive and natural setting for the AUV mission planner. Experience has shown though that interacting in 3D VR environments is difficult. One problem is that many 3D environments, notably the large CAVE type immersive environments, lack the high-resolution stereo imagery that enables good depth perception. Another major problem is that many 3D environments do not provide haptic feedback to the user. As human beings though we rely heavily upon various force cues and constraints imposed by our real world environment to support our interactions. For example, we constantly employ 2D surfaces such as floors and countertops to help us to position items around us.

Haptic devices, which allow fine force constraints to be applied in VR environments, now commercially exist. The question then arises, how should force be used to support user interaction in a VR environment? The answer depends upon the haptic input device and its role in the application. In designing a medical simulator that uses a pen-based device such as the PHANTOM to model a virtual scalpel, it is appropriate to use force feedback to mimic those *physically based* forces created by the contact of scalpel against human tissue. For applications in which the interaction is not so obvious, such as AUV path planning, the idea of haptically modeling *task constraints* offers a solution.

It has long been recognized that in many user interface problems, adding task-related constraints can improve a user interface. Computer-aided design programs employ concepts such as snap-dragging, for example, to force objects to visually line up or rotate about certain fixed axes (Bier 1990). Adding force feedback enables users to feel these constraints embodied in a virtual element. Thus, for example, if a particular widget should only be allowed to rotate about a certain axis, then that constraint can be physically imposed to restrict the range of motion of the input device. In the field of teleoperations, the notion of task constraints has lead to the idea of using force feedback embodied in “virtual fixtures” to constrain a user’s motion when carrying out manual and supervisory control tasks (Sayers and Paul 1994; Stanisic, Jackson et al. 1996). There are of course many constraints inherent in real world interaction; e.g. physical objects do not in general interpenetrate each other when they come into contact.

Haptic VR systems have demonstrated their capability to provide a compelling interaction environment while enhancing user productivity across several application areas, including petroleum exploration, medical training and industrial design. In the petroleum exploration industry, for example, VR and haptics have been shown to improve the speed and accuracy of seismic data analysis (McLaughlin and Orenstein 1997). Commercial firms now market procedural simulators for endoscopy and laparoscopy for surgical training, where the sense of touch plays a critical role (Tendick, Downes et al. 2000). In the

field of product design and development, haptic VR systems are being used to provide a more natural and intuitive way of defining concepts in a completely digital environment (Grahl 2003).

This paper describes a VR system our lab has built called *Haptic Geographic Zoomable User Interface 3D*, or Haptic-GeoZui3D (Komerska and Ware 2003), which is based upon this idea of using task constraints to support user interaction. We have chosen to demonstrate these ideas in an AUV path planning application because we believe that 3D haptic interaction and visualization technologies, when appropriately applied, can greatly assist and enhance a task such as this.

SYSTEM DESCRIPTION

Our Haptic-GeoZui3D application is built upon a visualization system and a haptics interface device, which are integrated together in a physical workspace.

The visualization component we use is a modified version of our lab's *Geographic Zoomable User Interface 3D*, or GeoZui3D, application (Ware, Plumlee et al. 2001). It is used within our lab as a platform for exploring basic research questions in 3D interaction and as a practical tool for analyzing bathymetric data. GeoZui3D uses OpenGL for graphics rendering, and can display stereo imagery when paired with appropriate hardware. GeoZui3D runs under Windows, Irix and Linux operating systems.

We use a SensAble Technologies PHANTOM 1.0 haptic input device in our workspace. The PHANTOM was chosen because its pen interface provides a simple and intuitive pointing device that is similar in function to a mouse in a 2D environment yet provides for 3D selection and application of fine force constraints. It allows for 3 degree-of-freedom (dof) position and 3-dof orientation tracking of the pen, while providing the capability for application of 3-dof point force output. The PHANTOM 1.0 provides a haptic workspace comprising a rectangular volume 17 cm (width) by 14.5 cm (height) by 8 cm (depth).

In Haptic-GeoZui3D, the visualization and haptic components are unified in a Haptic Fish Tank VR arrangement, as illustrated in Figure 1. Fish Tank VR refers to the creation of a small but high quality virtual reality that combines a number of technologies, such as head-tracking and stereo glasses, to their mutual advantage (Ware, Arthur et al. 1993). A horizontal mirror is used to superimpose virtual computer graphics imagery onto the PHANTOM workspace. The placement of the mirror also means that the PHANTOM and the user's hand are hidden from view. However, a proxy for the pen that the user holds is shown and, because the user's actual eye position is used to compute the computer graphics imagery, visual and haptic imagery are co-registered at all times. To accomplish this, we use a 17-inch monitor set at a 45° angle above the mirror. Stereoscopic display is provided using NuVision Technologies stereo glasses with a monitor refresh rate of 100 Hz. We also provide head-tracking capability, through

the use of a Polhemus FASTRAK system with a sensor mounted to the stereo glasses.

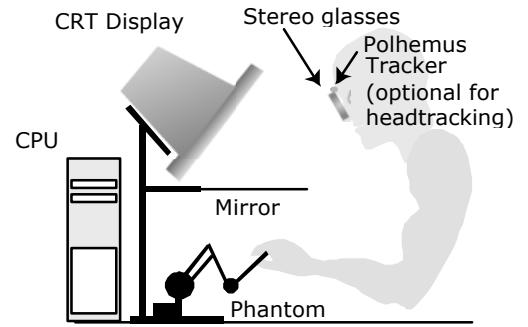


Figure 1. Haptic Fish Tank setup.

There is a high degree of synergy between the elements that comprise our Haptic Fish Tank setup. The workspace volume is similar in size to the localized workspace that we interact with in our everyday experience. GeoZui3D works particularly well in our Haptic Fish Tank because it uses what is known as *center-of-workspace* interaction (Ware, Plumlee et al. 2001). In this interaction style, objects are brought to and operated upon at a fixed point located conceptually at arm's length from the user. We align this point with the center of the physical PHANTOM workspace, also at arm's length, such that interaction in the virtual environment matches with what our body (proprioceptive) sensors tell us. Figure 2 shows the actual system in use.



Figure 2. Working in the Haptic Fish Tank.

DESIGN PHILOSOPHY

In general, research suggests (Flanagan and Lederman 2001) that touching objects, especially with a single point of contact, provides little useful information about object shape. That task is best left to visual display. However, a number of studies have shown that considerable benefit can be gained by feeling constraints that are relevant to task performance (Unger, Nicolaidis et al. 2002). For example, placing a peg in a hole is done faster if the force constraints are provided. Accordingly, our research work has focused on finding ways of adding haptic constraints in such a way that they improve task performance.

To this end, we have evolved the following set of design principles to guide us in development of haptic 3D interaction elements:

- Haptically represent constraints rather than objects
- Display constraints both visually and haptically (constraints are possibilities for movement, limits on motion)
- Visually emphasize potential for interaction (manipulation hot spots)
- On contact, visually reveal additional constraints
- Make state information both haptically and visually accessible

An interesting way of combining constraints with a direct manipulation interface is to create haptic widgets (Miller and Zeleznik 1999). The idea of a widget is to encapsulate both behavior and affordances in a single object. Thus if an object looks like a handle, and behaves like a handle when clicked on with a mouse, learning time will be minimized. We have extended this notion to our set of domain-specific haptic elements as well.

In our AUV path planning application, we have implemented a number of haptically enhanced data objects and interaction widgets. Widgets include elements designed to support object layout and scene navigation, as well as haptically enhanced in-situ menus and slider controls for mode selection and parameter adjustment, respectively. Objects include representations of the AUV, as well as for transit and survey behaviors.

We also differentiate between the notion of “passive” and “active” constraints. Passive constraints are force fields surrounding static elements. Active constraint forces guide a user, typically along a specific trajectory, in repositioning elements that have been selected.

Haptic-GeoZui3D leverages the center-of-workspace metaphor to let the user directly manipulate virtual objects and widgets in the environment. Many of the elements that can be manipulated in our application possess a visual and haptic hotspot by which the user can interact through the PHANTOM proxy (visually modeled as a pen). To select such an element, for example, the user moves the PHANTOM pen tip until its virtual proxy falls within a 3 mm capture radius of the hotspot. At this point, the pen tip

is subjected to a spring force that snaps the tip to the widget center. This is analogous to a 3D detent. This spring force constitutes a “passive” constraint and signals to the user that they can now manipulate the element, if they wish to. Additional visual cues, such as an element state numerical values and/or permissible manipulation track, are also displayed. To manipulate the element, the user presses the PHANTOM stylus switch while moving the pen. Appropriate haptic “active” constraints are imposed which properly guide the user during this interaction. The nature of these active constraints is based on the type of element being manipulated and its context with respect to other elements in the environment. Visually, the element changes to a standard color (we use green). Releasing the switch locks the haptic (and visual) position of the element and re-enables the passive constraints. The object or widget element changes back to its default coloring. To detach, the user then simply pulls the pen away from the attached element, to beyond the 3 mm radius, where the attractive force drops to zero.

While an element is being manipulated, the passive capture forces of certain other elements must be temporarily deactivated, while others must be left active. For example, while manipulating a transit waypoint object, the user may want to use the grid widget to help place the object, but will not want to be captured by any of the scene navigation widgets.

HAPTIC ELEMENTS

As previously mentioned, we have developed various interaction elements that rely heavily upon haptic constraints to guide the user’s actions in carrying out specific tasks. We can divide these elements into four major categories: scene navigation widgets, object placement widgets, mode and parameter selection widgets, and objects representing the AUV and the behaviors it supports. It is these behaviors that define the planned path for the vehicle.

SCENE NAVIGATION WIDGETS

Scene navigation widgets are used in Haptic-GeoZui3D to allow the user to quickly and intuitively translate, rotate and scale the virtual environment about the center of the workspace. Figure 3 shows the rotation and scale widgets, which are located near the top of the vertical axis at the center of the workspace.

The yaw widget is visually modeled as a tab handle anchored on a compass arrow passing through and perpendicular to the vertical scale axis. Its role is to allow rotation of the world about the vertical axis. The widget hotspot is the tab center. Once attached to the handle, a circular band appears as shown in Figure 4, providing a visual cue as to how the handle will be actively constrained. When the user presses the pen switch, the active constraint forces restrict the PHANTOM tip movement along a 12 mm radius ring co-registered with the surface of the band.

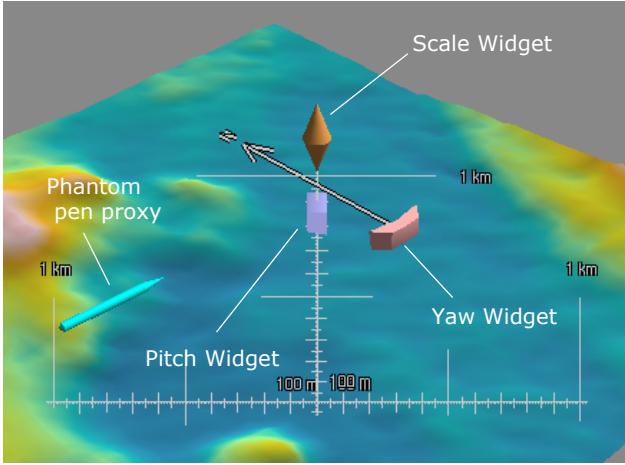


Figure 3. Scene navigation widgets.

Haptic detents are established at 1° increments to provide additional position cues.

The pitch widget appears similar to the yaw widget and is located near the top of the vertical axis, underneath the scale widget. The purpose of this widget is to allow the user to rotate the world about the horizontal axis. When the user attaches to the handle, a circular band appears as shown in Figure 4. This band has its center at the crosshairs and lies in the plane parallel to the vertical axis and perpendicular to the horizontal axis. In a similar fashion as with the yaw widget, the active constraint forces restrict the PHANTOM tip movement along a 27 mm radius ring co-registered with the surface of this band. Haptic stops are imposed at +90° (plan view) and -40° to help prevent the environment surface from hiding the widgets. Haptic detents are established at 1° increments. As the pitch changes, the orientation of the vertical axis, along with the location and orientation of the yaw, pitch and scale widgets, also changes.

The scale widget is used to allow uniform scaling of the environment about the center of the workspace. It is visually modeled as two opposing cones atop the vertical axis. Once attached to the widget center, the user presses the pen switch and pulls up or pushes down *along* the axis direction to zoom in or out, respectively. Visually, the cones alter shape to indicate the direction of scaling, as shown in Figure 4. Haptically, the hotspot remains fixed in space; in this case the dot product of the spring restoring force vector with the vertical axis controls the magnification and minification rate of the widget. Note that scaling the environment does not alter the scene navigation widgets. They always maintain their fixed location and size at the center of the workspace.

Translating the environment within the workspace is handled in one of two ways. In the first method, the user, while not attached to any of the other widgets, simply presses the pen switch and directly drags the scene. Visually, the PHANTOM pen proxy changes color to

green, while maintaining a fixed position with respect to the dragged world. A small amount of inertia is imposed while dragging to give the world a sense of “weight”. The second method is needed because our visual workspace size is a viewing frustum that is larger, most notably in depth, than the haptic space. This can lead to the case where elements are visible but beyond the touchable space. When the user wishes to select an element that lies outside of the haptic wall boundary, he or she simply reaches for the object until the PHANTOM tip encounters the boundaries of the haptic workspace. Upon contact, the scene moves along a vector formed by the boundary contact point and center of workspace; the effect is to “push” the workspace boundary in the direction of the object (although in fact the virtual environment translates in the opposite direction). The speed of translation is proportional to the wall reaction force. If the location of interest lies far outside the reachable haptic workspace, the user will typically employ the scaling widget to zoom out such that the location is reachable, then drag the location of interest to the workspace center. Scaling in on this new center permits more detailed study and manipulation.

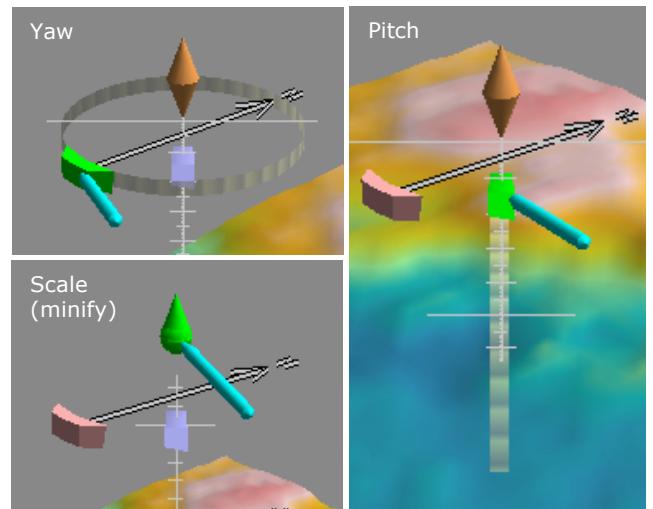


Figure 4. Scene navigation widgets activated.

OBJECT PLACEMENT WIDGETS

To support the user in placing objects in the virtual environment, we have developed a constant depth grid and a constant altitude grid. These grids are particularly suited for placing transit waypoints, allowing for the easy creation of vehicle paths with a constant depth or height above the sea bottom. The grid extents match the bathymetric surface extents and are oriented along the principal axes of the surface. Figure 5 depicts the grids and associated bathymetry. The depth grid is visually represented as a flat transparent surface overlaid with square gridlines. The altitude grid is similar to the depth grid except that the surface contour is identical to the bathymetric surface contour. The grid gap spacing is adjustable by the user by means of an appropriate slider control widget. Touching a

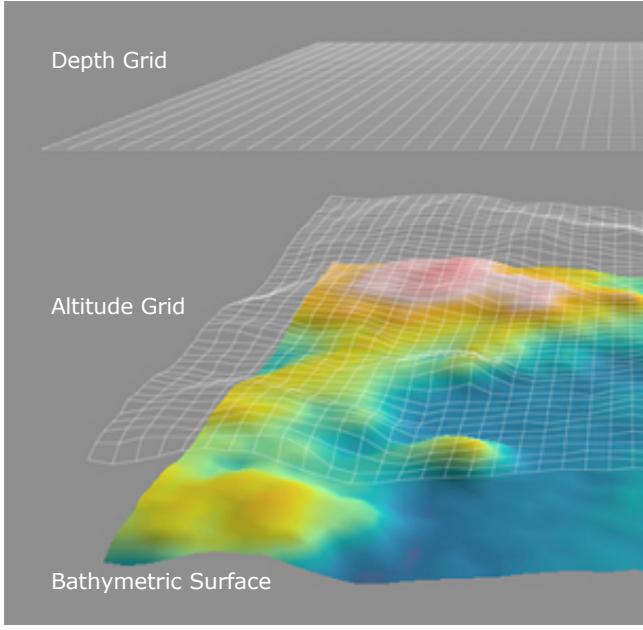


Figure 5. Object placement widgets.

grid and pressing the pen switch allows the grid to be moved in the vertical (depth) direction; haptic constraints restrict the user motion to this axis. The grids are created and removed through a menu selection.

The grid widgets implement force functions, co-registered with the visual grid, to provide both a surface force as well as a snap-to effect when the pen tip is near a gridline. The haptic effect is similar to sliding a pencil tip over a glass surface, except near a gridline, where a crisp detent force snaps the pen tip into a simulated groove aligned with the grid.

If bathymetry is available, the user can load this into Haptic-GeoZui3D. This provides visual context for path planning and can be haptically rendered as a unidirectional constraint surface, which blocks the user from moving the pen tip from the topside down. This provides a useful task constraint of restricting the user from performing the undesirable action of placing transit waypoints or vehicles below the ocean bottom surface. User motion is not constrained when moving from the underside of the surface upwards; this helps prevent the pen from becoming trapped under the surface. The bathymetric surface transforms appropriately as the scene is scaled, translated and reoriented. It cannot be manipulated.

MODE AND PARAMETER SELECTION WIDGETS

We have developed a system that employs haptic pie menus to allow the user to perform mode selection using the pen interface. Menus are context sensitive and generated in-situ by means of a second pen switch, located in-line behind the primary switch. This behavior is analogous to how the right-hand button on a mouse pops up a menu in a Windows environment. When the menu switch

is pressed, a pie menu is created centered in 3D space about the pen tip and positioned at right angles to the users view direction, to account for the Fish Tank VR perspective. Forces are imposed to guide the user in making a selection. Figure 6a shows an example of a menu for editing a depth grid widget. Our use of pie menus rather than conventional linear menus was inspired by research showing that users find the gesture-based option selection inherent in pie menus to be quicker and more intuitive than in the linear menu style (Komarska and Ware (unpublished); Kurtenbach and Buxton 1994).

Our pie menu layout can display from 1 to 8 options. We utilize a wedge size that subtends a 45° angle, with wedges aligned along the 8 ordinal compass points. This, combined with our use of semi-transparent wedges, helps to reduce their tendency to visually obstruct the view of the environment. The menu layout has inner and outer radii of 5 mm and 16 mm, respectively.

When a menu is activated, we first disable all other environment forces. We then superimpose three assistive force components: (1) a 2D planar constraint, (2) edge boundary constraints, and (3) a wedge selection force.

The haptic plane constraint is aligned in the same plane as the visual representation and acts to constrain the user to this 2D plane while making a selection. The edge boundary prevents the user from moving the PHANTOM tip outside of the valid menu widget region. This region includes the octagonal “home” space and visible wedge option spaces. Finally, the wedge selection force is activated when the user moves the pen tip more than 3 mm from the pie center. A spring force is created between the current tip location and a point centered within the closest wedge. This force acts to pull the PHANTOM pen tip toward this hotspot, at which time the wedge changes color from translucent white

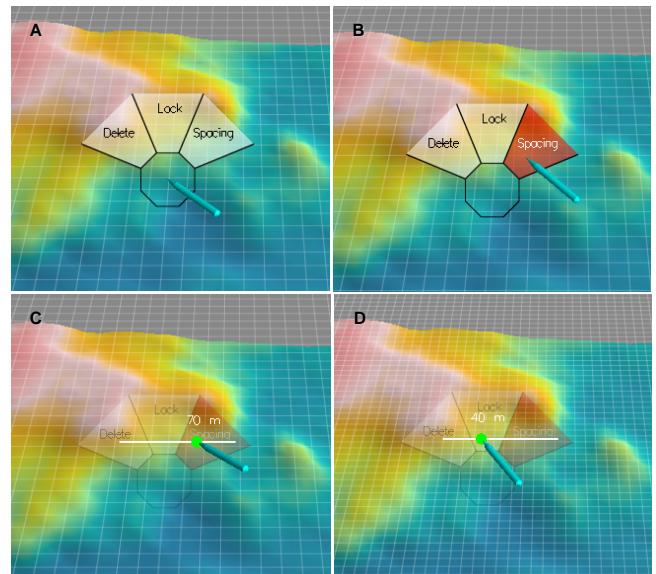


Figure 6. In-situ menu and slider control for adjusting grid spacing.

to red, indicating the option is ready to be selected (see Figure 6b). If the user decides to choose another option, the user moves the pen tip until it crosses the wedge option boundary, at which point the selection force of the new wedge becomes active and pulls the pen tip to its center.

Force is also used to indicate menu option selection as well as disengagement without selection. When the user is on an option wedge, they have the choice to push into the menu plane to make the selection. Pushing into the option causes the pen tip to “pop-through” the menu plane and cause the option to be selected. The user can also choose another option, as described above, or exit the menu completely by pulling away (toward their head location). In either case, the menu will disappear, the normal environment object and widget forces are reactivated and the appropriate selection logic is executed.

In addition to haptic pie menus, we have developed a haptically enabled slider control widget for adjusting parameter values. The slider control represents a type of very simple general purpose control widget, of which other examples include dials and buttons.

Our slider control is modeled visually as a thin rod with a sphere-shaped handle attached to the rod, located in 3D space. The rod endpoints represent the available parameter range while the handle position on the rod represents the current parameter value, in a similar fashion to the traditional 2D slider control. The exact parameter value is also displayed next to the handle.

Haptically, our slider control draws upon the same passive/active force paradigm that we utilize with our other haptically enabled objects and widgets. The sphere handle exerts a spring force to capture the PHANTOM pen tip. Once captured, the handle color changes to green and the user then has the option of adjusting the parameter value. The user does this by pressing the PHANTOM switch, which triggers active force constraints to confine the pen tip to a co-registered haptic line. This also activates appropriately spaced detent forces along the line length. Releasing the switch locks the position of the haptic sphere handle and parameter value, allowing the user to pull away and detach from the slider control.

We have closely integrated the slider with our haptic pie menus. This is illustrated in Figure 6(c,d), where we show a slider control for adjusting the gap spacing of a depth grid widget. While attached to the grid widget, the user activates a pie menu as described in the previous section. The user moves the pen tip to the appropriate menu pie wedge and pushes through the wedge to make the selection. At this point, the slider control is created with the PHANTOM tip initially attached to the control handle, slightly behind the menu. The menu forces are deactivated while the menu transparency is increased significantly. This provides a sense of context while not obscuring the slider control. The user manipulates the slider handle as described above, while direct feedback is provided through haptic detent forces, changing parameter value label, and immediate

visual resizing of the depth grid widget gap spacing. After completing the parameter adjustment, the user detaches from the control, which removes both the pie menu and slider control and places the pen proxy and user interaction back into the scene space.

AUVS & BEHAVIOR OBJECTS

We have created several objects for use in our path planning application, which have been visually and haptically rendered. These include a vehicle representation and a vehicle transit behavior object. Figure 7 shows an example of each of these elements.

The vehicle object is designed to represent the start location of the AUV path. Visually, the vehicle is shown as a simplified 3D model. It is capable of being manipulated, and is haptically represented for selection as a 3D detent as described previously, with its hotspot being the center of the visual icon. As the environment is scaled out (minified), the AUV object maintains its original visual and haptic size. This visual/haptic sizing model works when the interaction volume is much larger than the size of the object, allowing us to essentially treat the object as a point. Since the primary interaction task on the AUV is selection and relocation, the interaction typically is done at a minified scale and the model works well. When selected, state information including the AUV name and georeferenced position is shown next to the vehicle icon.

In addition to providing a handle for selection, the AUV object provides a contextual anchor for menu options representing vehicle-specific behaviors available to the planner when building a mission profile. The idea here is that vehicle capabilities would be defined a-priori (e.g. within a configuration file), and would be used to set limits on interaction. Examples include only displaying in menu options those behaviors supported by the selected vehicle, and using the vehicle maximum depth limit to define a haptic boundary that would constrain the user from creating a path below this limit.

We take the approach in our program that a vehicle path is composed of a string of path sections, where each section represents a specific behavior. The most basic behavior in our system is the *transit* behavior. The transit behavior encapsulates a single path segment having a starting and ending waypoint location, and a speed parameter. To initiate a transit behavior, the user pops up a menu while attached to the AUV object, and selects the “Transit” option. A line segment will appear, with the starting waypoint anchored at the AUV and the other waypoint rubberbanded to the pen proxy tip. State information similar to that shown for the AUV appears alongside the moving waypoint. The user moves the pen (and waypoint) to the desired location, using the object placement widgets if desired, and anchors the point in space by depressing the pen switch. This end waypoint can now be selected and manipulated in a similar fashion as the AUV, and represents the handle by which the user interacts with the

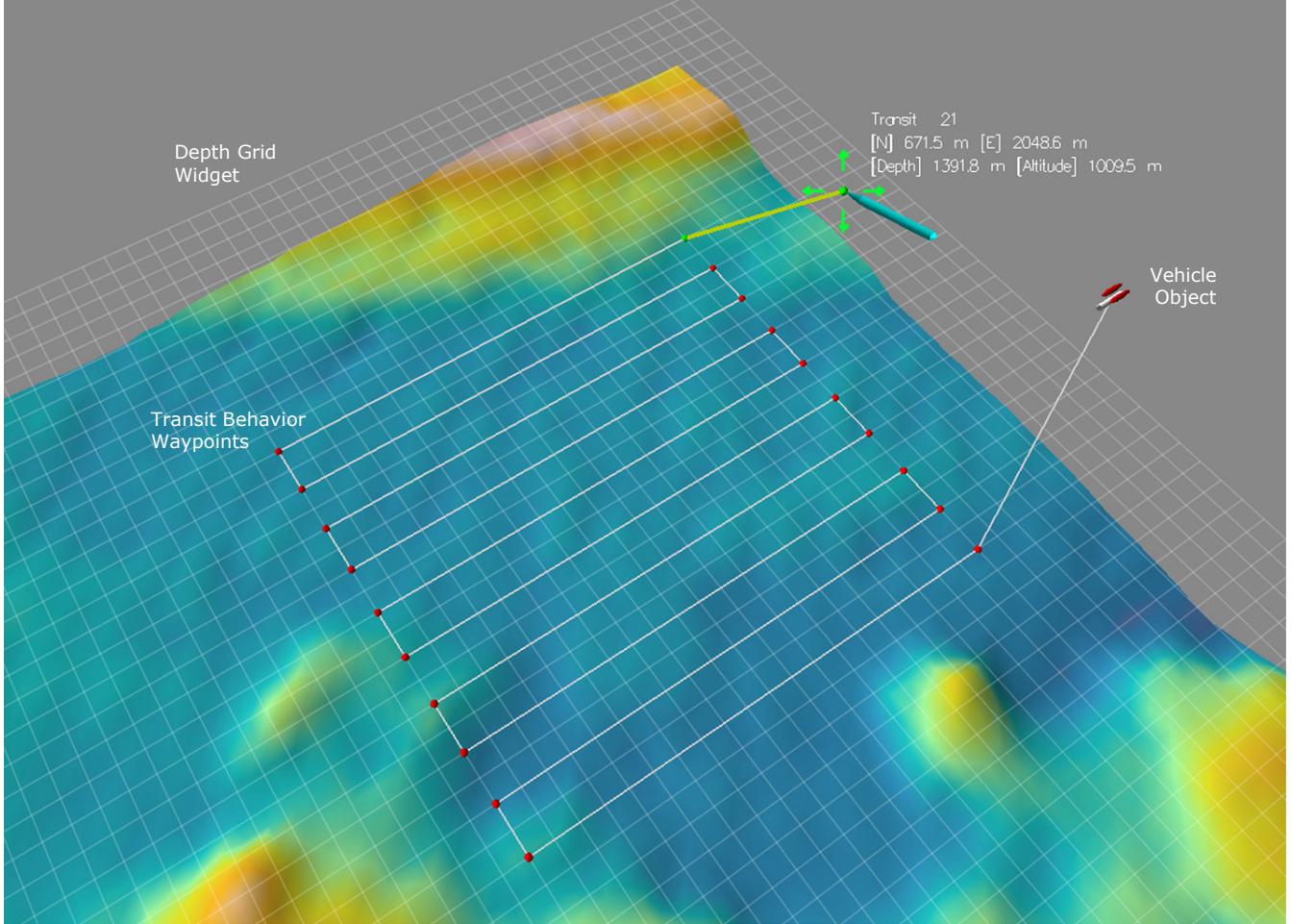


Figure 7. Creating a vehicle path.

transit behavior (and path segment). Figure 7 shows a transit behavior selected within a path. To add another transit behavior, or to delete a transit behavior, the user selects a transit behavior and pops up a menu; the options will be “Speed”, “Transit” and “Delete”. Selecting “Transit” inserts a new transit behavior after the selected one, whereas selecting “Delete” removes this behavior and stitches together the behaviors behind and ahead of the deleted behavior. Selecting “Speed” creates a slider control that allows the user to set the desired transit speed for that path segment.

The transit behavior works fine for specifying simple path segments, but is too slow and unwieldy for defining larger scale behaviors of interest to AUV path planners. A good example is a lawnmower type survey, which could include more than 100 waypoints. For our application to be of real value, it must be able to support construction of these higher-level behaviors with the same ease and direct manipulation style that we employ in our other haptic elements. To this end, we are currently developing the *survey* behavior, which will provide the capability for the user to quickly lay out and reconfigure a generic fixed depth lawnmower style survey. In addition, we also wanted

to allow for customization of the survey waypoints once we had defined its basic parameters. To support these requirements, our design supports two interaction modes: (1) survey object editing, and (2) survey waypoint editing.

To create a survey, the user would select the “Survey” menu option from a selected AUV or transit behavior object. The initial survey layout is very similar to the transit behavior layout. The start point is anchored at the location where the behavior was created. The end point follows the stylus tip in the x,y plane for although the user can move the pen in 3D, we constrain the end point in the z (depth) plane; a vertical line connecting the pen tip to the end point shows the correspondence between them. Visually, the survey appears as a transparent rectangular surface overlaid with survey lines. Lane spacing and orientation are initially set using default values. Figure 8a illustrates this initial creation phase. As with the transit behavior, depressing the pen button anchors the end point, allowing the user to leave or continue editing the behavior.

Having laid out the survey, the user will likely want to edit various parameters for it. The entire survey can be repositioned by touching the surface of the survey, pressing the pen button, and directly dragging the object to a new

location. Manipulating the handle that appears at the end point allows the user to rotate the survey about the start point z-axis to change the orientation of the survey. Haptic forces constrain the user's interaction to an arc that lies in the depth plane, whose origin is the start point and whose radius is the straight-line distance between the end and start points. Two additional handles, located at the survey edges, are available for the user to stretch the survey region. Again, haptic constraints are used to confine the interaction to a line that lies in the depth plane and perpendicular to the edge being stretched. Figure 8b illustrates these interaction hotspots and task constraints. Additional parameters including lane spacing width, lane orientation with respect to survey boundary frame, and survey speed can be set using survey behavior menu options.

Using the interaction methods described above, the user can rapidly create and edit a simple lawnmower style survey. In some cases though, the user may wish to refine a particular portion of the survey. To accommodate this, the user can select a menu option called "Edit Waypoints"; this effectively allows the survey object to be treated as a sequential set of transit behaviors. The user is free to add and delete transit segments, as well as adjust individual

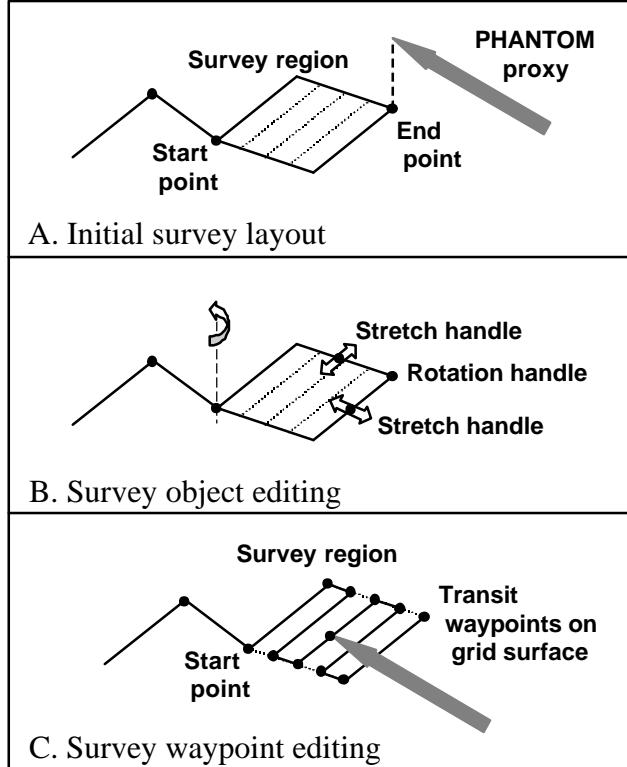


Figure 8. Storyboard concepts for survey behavior interaction.

transit behavior parameters. Grids aligned along the principal axes of the behavior region can be used to help position new transit segments. Figure 8c illustrates this

interaction mode. The user can revert back to the survey object editing mode (using a menu selection) but any modifications made while in the waypoint editing mode will be lost.

PATH PLANNER OUTPUT

Once the user has defined the mission, they can output the mission to a file. We currently save the data in XML format, suitable for translation to an AUV or later editing in Haptic-GeoZui3D.

FUTURE WORK

We are presently exploring new concepts for haptically enabled elements to extend the capability of Haptic-GeoZui3D. One such element would allow us to easily create simple geometric force fields to keep the PHANTOM outside or inside a specified region. This would provide the ability for one user to demarcate a region, such as a minefield, so as to prevent another user tasked with planning a vehicle path from inadvertently plotting a course through that (unsafe) region.

Another area of interest is to expand the types of supported behaviors. This could include useful low-level behaviors, such as "maintain position", as well as higher-level behaviors similar to the survey behavior described earlier. The requirements for future behavior support will of course depend upon the AUV platform developers, but may also take into account work being done to standardize the command language for AUVs.

CONCLUSION

This paper describes a virtual reality system our lab has built called Haptic-GeoZui3D to investigate 3D haptic interaction in the domain of AUV path planning. In our system, we have leveraged the concepts of haptic task constraints and center-of-workspace interaction in developing the user interface elements, and we demonstrate how these elements support our ability to rapidly lay out transit path segments in 3D. We discuss a technique for quickly defining a lawnmower type survey path, which could be manipulated directly or decomposed into individual transit segments for custom tailoring. The output of Haptic-GeoZui3D is a waypoints file that comprises the entire defined vehicle path; this could then be appropriately formatted and downloaded to an AUV for execution.

As AUVs become more prevalent and more capable, the need to better interact with them is going to increase. Mission planners will likely be tasked to plan more sophisticated routes, using multiple vehicles, within more challenging environments. We believe the 3D interaction techniques we have developed constitute a powerful new way of interacting with these underwater assets, particularly in the planning stages. We believe we have demonstrated how haptic constraints not only make path planning in 3D easier and more intuitive, but also provide direct feedback in limiting a user's ability to perform unsafe or undesirable

operations, such as placing a waypoint below a maximum depth threshold or under the sea bottom. These force and display techniques have the potential to minimize user learning time and interaction errors, and to reduce the time to carry out many AUV planning tasks.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of NSF Grant 0081292, NOAA, and Matthew Plumlee and Roland Arsenault of the Data Visualization Lab. We would also like to thank Rick Brennan and Brian Calder of CCOM for their valuable feedback.

REFERENCES

- Bier, E. A. (1990). Snap-Dragging in Three Dimensions. Symposium on Interactive 3D Graphics, Snowbird, UT.
- Flanagan, J. R. and S. J. Lederman (2001). "Neurobiology: Feeling bumps and holes." Nature **412**: 389-391.
- Grahl, C. L. (2003). ONLINE EXCLUSIVE: Design On Demand. Ceramic Industry. April 2003.
- Komerska, R. and C. Ware "A Study of Haptic Linear and Pie Menus in a 3D Fish Tank VR Environment." submitted for publication.
- Komerska, R. and C. Ware (2003). Haptic Task Constraints for 3D Interaction. 2003 IEEE Virtual Reality - Haptics Symposium, Los Angeles, CA.
- Kurtenbach, G. and W. Buxton (1994). User Learning and Performance with Marking Menus. ACM CHI'94 Conference on Human Factors in Computing Systems, Boston, MA, ACM Press.
- McLaughlin, J. P. and B. J. Orenstein (1997). Haptic Rendering of 3D Seismic Data. Second PHANToM Users Group Workshop, Cambridge, MA.
- Miller, T. and R. Zeleznik (1999). The Design of 3D Haptic Widgets. Symposium on Interactive 3D Graphics, Atlanta, GA.
- Sayers, C. P. and R. P. Paul (1994). "An Operator Interface for Teleprogramming Employing Synthetic Fixtures." Presence **3**(4): 309-320.
- Stanisic, Z., E. Jackson, et al. (1996). Virtual Fixtures as an Aid for Teleoperation. 9th Canadian Aeronautic and Space Institute Conference.
- Tendick, F., M. Downes, et al. (2000). "A Virtual Environment Testbed for Training Laparoscopic Surgical Skills." Presence **9**(3): 236-255.
- Unger, B. J., A. Nicolaidis, et al. (2002). Virtual Peg-in-Hole Performance Using a 6-DOF Magnetic Levitation Haptic Device: Comparison with Real Forces and with Visual Guidance Alone. 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, Orlando, FL.
- Ware, C., K. Arthur, et al. (1993). Fish Tank Virtual Reality. INTERCHI '93 Conference on Human Factors in Computing Systems, Amsterdam, The Netherlands.
- Ware, C., M. Plumlee, et al. (2001). GeoZui3D: Data Fusion for Interpreting Oceanographic Data. Oceans 2001.