

University of New Hampshire

University of New Hampshire Scholars' Repository

Center for Coastal and Ocean Mapping

Center for Coastal and Ocean Mapping

7-2003

Integrating Multiple 3D Views through Frame-of-reference Interaction

Matthew D. Plumlee

University of New Hampshire, Durham, Matthew.Plumlee@unh.edu

Colin Ware

University of New Hampshire, Durham, colin.ware@unh.edu

Follow this and additional works at: <https://scholars.unh.edu/ccom>



Part of the [Computer Sciences Commons](#), and the [Oceanography and Atmospheric Sciences and Meteorology Commons](#)

Recommended Citation

Plumlee, Matthew D. and Ware, Colin, "Integrating Multiple 3D Views through Frame-of-reference Interaction" (2003). *Coordinated and Multiple Views in Exploratory Visualization (CMV)*. 289.
<https://scholars.unh.edu/ccom/289>

This Conference Proceeding is brought to you for free and open access by the Center for Coastal and Ocean Mapping at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Center for Coastal and Ocean Mapping by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact Scholarly.Communication@unh.edu.

Integrating Multiple 3D Views through Frame-of-Reference Interaction

Matthew Plumlee, Colin Ware
Center for Coastal and Ocean Mapping
University of New Hampshire
{mdp@ccom.unh.edu, Colin.Ware@ccom.unh.edu }

Abstract

Frame-of-reference interaction consists of a unified set of 3D interaction techniques for exploratory navigation of large virtual spaces in non-immersive environments. It is based on a conceptual framework that considers navigation from a cognitive perspective—as a way of facilitating changes in user attention from one reference frame to another—rather than from the mechanical perspective of moving a camera between different points of interest. All of our techniques link multiple frames of reference in some meaningful way. Some techniques link multiple windows within a zooming environment while others allow seamless changes of user focus between static objects, moving objects, and groups of moving objects. We present our techniques as they are implemented in GeoZui3D, a geographic visualization system for ocean data.

Keywords—interaction techniques, multiscale, ZUI, multiple 3D windows.

1. Introduction

Many scientific visualization problems require fluid interaction with 3D data spaces that enable the user to examine data at various levels of magnification. Examples of such problem domains include geology [23], molecular modeling [16], and oceanography [21]. For instance, in the domain of oceanography, it is sometimes necessary to understand how biological organisms function in the context of the topography of the ocean floor, ocean currents, physical oceanographic parameters such as temperature and salinity, and the presence of other organisms. The scientist must be able to rapidly and effortlessly change the viewpoint, rotate the scene about a particular point and move from place to place, sometimes operating over orders of magnitude in scale. For example, one might wish to follow the path of a migrating whale in the context of a much larger landscape. Alternatively, one may wish to replay the voyage of an autonomous undersea vehicle that had

explored submarine seamounts to understand what was seen in the context of data gathered in previous voyages. Generally the challenge is to understand local information (such as that available regarding the whale or collected by the vehicle) in the larger context of information from various sources. We call the general problem the focus-in-context problem [14].

We are engaged in building interactive 3D stereoscopic visualization environments for dealing with exactly these kinds of visualization problems. We have worked closely with oceanographers and ocean engineers to support their visualization needs, and our main challenge has been the construction of an interactive environment that meets the following set of requirements:

1. Ability to rapidly change scale by many orders of magnitude, in order to examine data objects of different sizes.
2. Ability to rapidly select a point of interest and rotate the view around that point.
3. Ability to have multiple simultaneous views of a 3D data space, in order to integrate information from views at multiple scales or locations. For example, one should be able to see a submarine vehicle path in the larger environment while a local view of the vehicle in its immediate context simultaneously shows local data. There are a number of more specific requirements associated with this.
 - a) The spatial relationship between these views should be clear (including relative position, scale, and view direction).
 - b) It should be possible to have a viewpoint tethered to a moving object.
 - c) It should be possible to rapidly “hop-on” and off a moving object or change view from one object to another.
 - d) It should be possible to link views in various ways, such as coupling their view directions, in order to make it easier to cognitively integrate spatial information.
4. Ability to support stereoscopic viewing for all views.

There have been a number of previous approaches to the focus-in-context problem that have informed our work. One way of showing detail in context has been to use fisheye views [5, 7, 15]. In these, an area of interest is magnified relative to the surround to show detail in that region. One drawback to this approach is distortion of the data space, which can cause confusion when trying to understand spatial relationships. A more serious problem is that the detail area can only be magnified by a relatively small amount (a factor of five or so) before the distortion becomes unacceptable. Also the focus area cannot be rotated with respect to the surrounding area. Generally, the fisheye approach seems more suitable to 2D scenes than 3D scenes, although Carpendale has applied it to 3D models [3].

Another way of providing detail in context is to facilitate rapid scaling or *zooming* of the information space [2]. Zooming affords detail by zooming in and context by zooming out, but the user must hold relevant information in visual working memory to make a perceptual synthesis. We have shown that if more than a small amount of visual information must be transferred cognitively from one view to another, then multiple simultaneous views are necessary for efficient task completion [14].

A number of experimental systems have explored the possibilities for multiple 3D views of the same data space. The Worlds in miniature (WIM) system is a fully immersive environment that contains a small model of the virtual world within it [17]. Navigation in WIM can be accomplished by moving a viewpoint proxy within the model world. In a similar vein, Holm et al. [6] used a "Greek god metaphor". A god user could move an actor from one part of the scene to another, and thereby change the viewpoint upon switching views to that of the actor. While these previous studies were sources of inspiration, we are interested in an easy-to-use desktop environment that has a more unified method for viewpoint change.

Another source of inspiration is work on tethered and dynamically tethered views [22, 18]. A tethered view is a view attached to a moving object for the purposes of following the object and controlling the view's path. It has been found that for some vehicle control applications, it is best not to adopt a viewpoint within the vehicle but to have a viewpoint somewhat above and behind it, tethered to its motion. One of our contributions in the present paper is to provide a general framework that encompasses tethering together with a number of different methods for linking viewpoints to each other and to moving objects.

Finally, we have been influenced by image plane interaction techniques [10], which suggest that even though the environment is 3D, it can be simple and effective to use 2D interaction methods. We use the image plane not only for selection, but also for 3-dimensional placement of items and integration of information from multiple views.

We have previously given partial reports of some of the concepts presented here. In the UUST workshop, we presented an overview of system capabilities and how they could be applied to autonomous undersea systems [11]. At UIST '02, we gave a demonstration of the linked view techniques but did not provide details or the conceptual framework [13]. This paper represents the first full treatment of frame-of-reference interaction.

GeoZui3D is the platform on which we have developed frame-of-reference interaction. GeoZui3D stands for **G**eographical **Z**ooming **U**ser **I**nterface. GeoZui3D has two primary purposes. First, it is a geographic visualization system, especially designed for interaction with three-dimensional ocean and ocean-floor data. Second, it serves as a proof-of-concept system for exploring and developing frame-of-reference interaction. GeoZui3D runs under Windows and Linux operating systems, and supports the simultaneous rendering of geo-referenced objects from landmasses, to tracks of underwater vehicles, to temperature information in the water volume.

2. Frames of Reference (FoR)

There are two principles we have used to guide and unify the set of techniques we have developed.

1. *Frames of Reference: 3D views should be characterized by a frame of reference at a designated point of attention, rather than at the viewpoint.* This principle allows both views and objects of attention share the same valid operations, and opens up a whole new way of thinking about linking views to the target information and to each other.
2. *The Image Plane: The 2D image plane should be used to reduce the complexity of 3D views whenever possible.* Many things can be done well in the 2D image plane, including selection, 3D positioning, window management, and linkage of 3D views.

This section describes several techniques based on the first principle, but makes reference to some application of the second principle. The section that follows addresses the second principle in fuller detail.

A reference frame is characterized by three composite quantities, namely position (x , y , and z), orientation (heading, pitch, and roll, or quaternions), and scale. In FoR interaction, the origin of the reference frame takes on a particular significance as a device for mediating 3D interactions with objects and scenes, as well as linking different views of the same scene.

2.1. Window FoR: Center of Workspace

In many 3D interfaces there are two kinds of reference frames that concern interaction. One kind is for viewpoint control. A common technique is to fly a virtual camera around the scene, which uses a viewpoint-

centered frame of reference. The second kind is for object manipulation. When objects are to be moved within the scene, an object-centered frame of reference is adopted that specifies how the object rotates and scales. These two kinds of reference frames are not immediately compatible, and can make it difficult to properly align views with objects of interest.

Our method for viewpoint control is based on the center of the workspace as a unifying construct. Rather than putting the origin of a reference frame at the virtual camera for the window, the origin is placed at a “look-at” point at a fixed location in front of the camera. More specifically, the center of workspace is established at the center of the screen, conceptually at a depth corresponding to arm’s length from the user, as illustrated in Figure 1. To move a different point in the scene to the FoR origin (or equivalently, navigate the view to a different point), the user simply clicks on that point with the middle mouse button. This causes a smoothly animated translation of the selected point to the center. Once there, the user can rotate and scale the view using a set of 3D widgets, as shown in Figure 2. These widgets allow for direct manipulation of the FoR’s orientation and scale by grabbing various handles. As we describe later, such interaction allows for scene navigation and object-centered views to have the same interface.

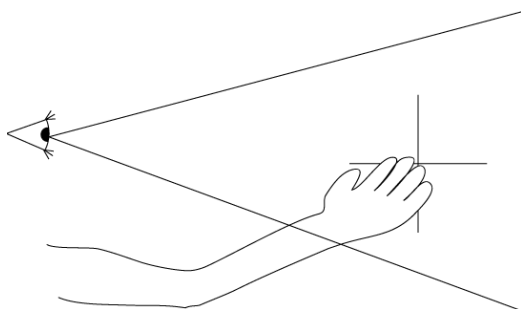


Figure 1: Illustration of a workspace, relative to an observer. The cross is at the “center” of the workspace and defines the frame of reference for interaction.

This kind of center of workspace interaction is partly based on ideas presented in the NV3D system of Parker, Franck, and Ware [9]. In that system the interaction was based on object centers, while in GeoZui3D it is based on surface points. The use of the center of workspace was not extended to multiple windows in NV3D.

The center of workspace acts as a convention for focusing user attention. The user brings objects to the center of workspace because that is where the system provides mechanisms for further inspection. Meanwhile, the system can assume a user’s attention is focused at the workspace center because that is often where the system’s behavior causes the user to focus. The center of workspace then becomes a natural location to place

additional tools that the user may wish to use with the focus item.

We have taken advantage of this in GeoZui3D. For instance, GeoZui3D can optionally place scales at the center of workspace, as illustrated in Figure 2. Normally, a scale in a 3D perspective view is difficult to place, because distances vary with depth. Scales at the center of workspace provide an excellent way to both convey a sense of scale, and measure distances to objects near the focus. Some have raised objections to cluttering the display at the focal point of attention, because it occludes part of what the user may be looking at. In practice, users have not complained about the tools at the center of the workspace. Furthermore, the tools can be quickly hidden and redisplayed using menu items or a keyboard key. Users have no problem hiding the widgets in the rare occasions that they do get in the way, and then reinstating them when they need to use them.

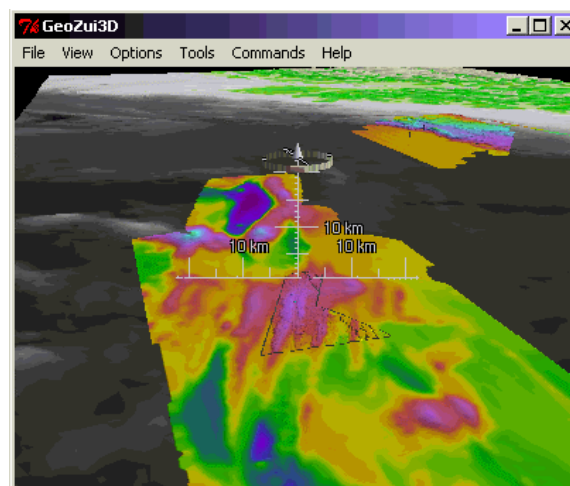


Figure 2: GeoZui3D bases tools at the center of each workspace, such as the horizontal and vertical scales. The ring at the top of the scales facilitates rotation in heading and pitch.

2.2. Application to Multiple 3D Windows

The center-of-workspace concept extends to multiple windows by endowing each 3D window with its own attentional focal point and its own set of tools at the workspace center. We call each of these 3D windows a *zoomport*. Once there is more than one zoomport, it becomes necessary to enable users to clearly see the spatial relationships between them. It also becomes possible to enforce geometric links between different FoR’s depending on task requirements.

We have developed a number of techniques that use FoR’s to link multiple views. Before describing these techniques, consider two illustrative applications for such linkages. The first application is scientific exploration. Linked views allow scientists and engineers to explore complex data spaces at multiple levels of detail, making

it possible to see some magnified area within a larger context. This can also be valuable when the scientist wishes to present results to others. It is this application that motivated the development of zoomport proxies. The second application is monitoring one or more vehicles, such as ships or ROVs (Remotely Operated Vehicles). In this case a scientist may wish to have a close-up view of a vehicle and the data it is actively collecting, and at the same time have an overview of the surrounding environment. Alternatively, a harbor-master may be required to monitor the progress of ships as they come in and out of port, and be ready to take corrective action when ships get too close to each other or start wandering into restricted areas. It is such an application that motivated reference-frame couplings and frame-of-reference operations.

2.2.1. Zoomport Proxies. We have developed *zoomport proxies* as graphical devices for showing the spatial relationships between views. A zoomport proxy is a representation of one zoomport displayed as part of the scene within another zoomport, as illustrated in Figure 3. The child zoomport can be translated by dragging its proxy in the parent window. This is an example of image plane interaction. During dragging, the cursor hot-point is projected back to the nearest surface in the parent zoomport, and the intersection becomes the new origin for the child zoomport's FoR.

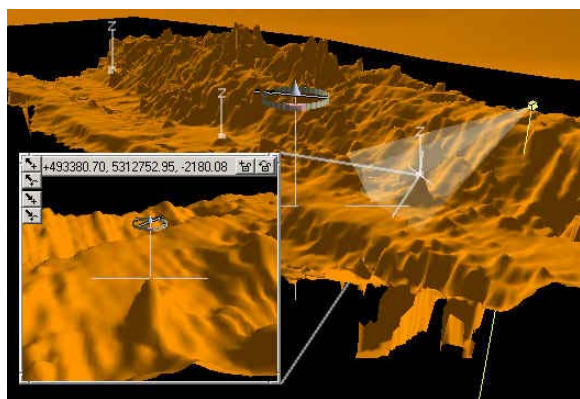


Figure 3: A child zoomport, with proxy and tethers linking its center of workspace to a parent zoomport in GeoZui3D.

The most important part of a zoomport proxy is the proxy center: a marking that represents the child's center of workspace in the parent window. It gives an immediate indication to the user as to where the focal area of the child is with respect to the area of interest in the parent. It also provides an unobtrusive way of marking what the object of attention is in zoomports that are minimized or are for some other reason off-screen. Furthermore, it provides an affordance for dragging the focus of attention for the child zoomport: the user can

grab the proxy center and move it to a desired location in the context of the parent zoomport.

Other parts of a zoomport proxy include the directional indicator and the tethers. The directional indicator provides an indication of which way the child zoomport is oriented, and how wide the field of view is for that zoomport. In GeoZui3D, this is done by drawing a representation of the camera for the child zoomport, with a "headlight" that suggests the shape of the viewing frustum for the child zoomport (see Figure 3). A line is dropped from the camera through the surface to provide enough depth cues so that the user can distinguish 3D orientation. The tethers are simple lines drawn from the proxy center to the near corners of the corresponding child zoomport window. The tethers provide a way of quickly determining which zoomport belongs with which tether when there is any doubt.

Our use of proxies with tethers is inspired both from artist renditions in magazines such as National Geographic, and from the use of similar devices in the DragMag system of Ware and Lewis [20]. One significant innovation over these precursors is the identification of the point of interest in the proxy itself, and then using this point to tie tethers to the corresponding view. In two-dimensional settings, the convention has been to draw a box representing the extent of the child 2D view, and connect the box to the child view at the corners.

2.2.2. Reference-Frame Couplings. Reference-frame couplings enforce relationships between zoomports for creating effects such as tethering a zoomport to a moving object, or coupling two zoomports to create a forward-up overview. For example, suppose that we wish to see a detailed view of a remotely operated vehicle in the context of an overview. A "wingman" view or an "over-the-shoulder" view can be accomplished easily by attaching the FoR of a zoomport to the FoR of the moving object. As implemented in GeoZui3D, FoR couplings make it possible for the user to attach a view to a moving object simply by clicking on that object with the middle mouse button. This causes the zoomport FoR to become linked to the moving vehicle in such a way that further view manipulations occur in the reference frame of the moving object. Without interaction, the zoomport rotates as the object rotates and its center stays on the vehicle. Interaction with zoomport widgets can rotate the view around the moving vehicle as if the vehicle were static, and selection of other points on the object with the middle mouse button provide a sense of navigating with respect to the moving object. To return to a static view or transfer to another moving vehicle, it is only necessary for the user to click on the appropriate object with the middle mouse button.

A reference-frame coupling is a mathematical constraint on a component, such as position or heading, such that a change in one frame of reference induces a

change in the other. For instance, if two zoomports were to be coupled in position, any motion in one would be matched by the same motion in the other. To support vehicle-linked views and other interactions, we have designed three basic types of coupling: absolute, relative, and localized.

Absolute coupling is the simplest. If an absolute coupling exists between reference frames P and Q on an attribute a , then whenever $P.a$ changes value, $Q.a$ is updated to have the same value. For instance, for a coupling on heading between two zoomports, if the heading of a zoomport characterized by P changes to 45° , the heading of the zoomport associated with Q is also set to 45° . Figure 4a shows a zoomport (Q) and object (P) at the initialization of a coupling, while Figure 4b shows what would happen if no coupling were enabled and the object moved. The effect of an absolute position coupling between the zoomport and object is illustrated in Figure 4c.

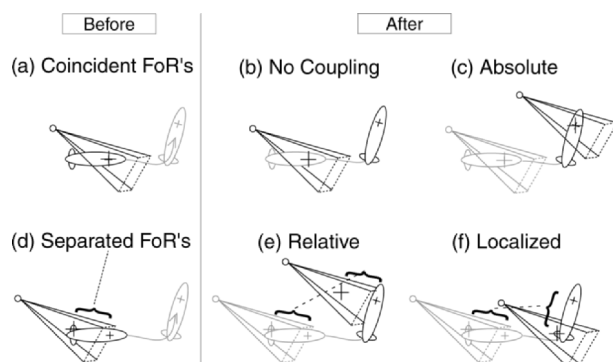


Figure 4: Illustration of various couplings of position between zoomport and object in the position attribute. In the first row, the zoomport and object share the same position to begin with (a). After the object moves, the effects of no coupling (b) and an absolute coupling (c) are shown. In the second row, the zoomport is positioned behind and slightly above the object initially (d). After the object moves, the effects of relative coupling (e) and localized coupling (f) are shown.

Relative coupling is more general than absolute coupling. If a relative coupling exists between reference frames P and Q on an attribute a , then whenever $P.a$ changes by some amount, $Q.a$ changes by as well. For instance, consider what happens if the heading of the zoomport characterized by P starts at 45° and the zoomport associated with Q starts at 130° . If the heading of P changes to 25° ($= -20^\circ$), then the heading of Q changes to 110° . Figure 4e shows the effect of a relative position coupling between a zoomport (Q) and object (P), after movement of the object from its initial position (Figure 4d).

Whereas absolute and relative couplings operate over a single attribute and are valid for any attribute of a

reference frame (in position, orientation, or scale), a localized coupling operates over all attributes. The purpose of a localized coupling is to “fix” one reference frame with respect to another, just as would occur if one were to rigidly attach a camera to a moving object. Localized coupling works as follows. If a localized coupling exists between reference frames P and Q , then whenever P changes in position, orientation, or scale, Q is updated so that its position remains unchanged with respect to P . The effect of a localized coupling is to “fix” the position of Q into P , as if Q were rigidly attached to the origin of P . Figure 4f shows the effect of a localized coupling between a zoomport (Q) and object (P), after movement of the object from its initial position (Figure 4d). The position of the zoomport is always in the same place on the tail of the object, regardless of how the object moves (or how it is scaled).

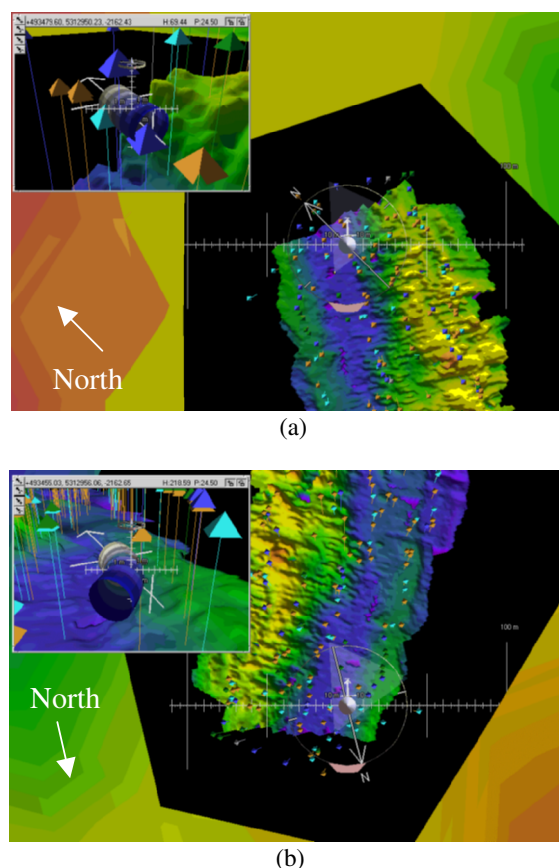
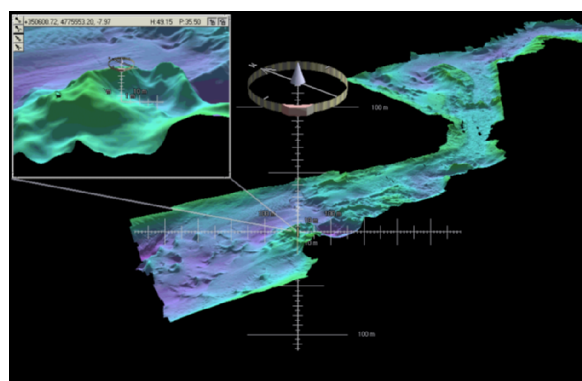


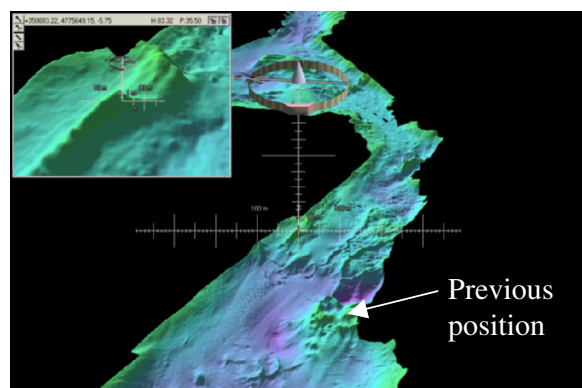
Figure 5: The main zoomport is coupled with the inset zoomport to yield a forward-up view, while the inset zoomport is coupled to a moving vehicle using a localized coupling and a relative coupling in heading to implement a tethered view. (a) and (b) show how both windows translate and rotate as the vehicle moves and turns.

Couplings are generally most useful in combinations. For instance, a forward-up overview map

can be achieved by absolute-coupling two zoomports in position and heading attributes. Figure 5 shows a similar coupling instantiated between the inset zoomport and its parent in GeoZui3D. As another example, a magnified view can be created by absolute-coupling two zoomports in position and orientation, and relative-coupling them in scale. Such a coupling arrangement would provide a way of rapidly operating at two disparate scales (see figure 6). As a third example, a tethered view can be created by establishing a localized coupling between a zoomport and an object, combined with a relative coupling in orientation. Such a coupling is shown in Figure 5 between the inset zoomport and the moving vehicle (arrow with tube). In GeoZui3D, couplings such as these are temporarily suspended during interaction, and are reinstated after interaction, essentially establishing a tethered view at a different location. This is done in order to give the user the feeling that they can navigate on the moving object just as they would navigate through the static virtual world.



(a)



(b)

Figure 6: Zoomports coupled in a magnified view arrangement. Any movement in one is matched in the other, as in the translation, rotation, and scaling from (a) to (b), allowing the inset zoomport to act as a magnifying glass for the center of workspace in the main zoomport.

Behaviors similar to such combinations of couplings have certainly appeared before. Forward-up views have been around for some time and have been studied as to their benefits for particular tasks [8, 4, 12]. Similarly, “tethered views” or “dynamic tethers” have been proposed and implemented by others [18, 22]. While the behavior of dynamic tethers is not directly representable in terms of couplings, couplings do provide a way to unite various coordinated views in a single framework. Our introduction of frame-of-reference operations later in this paper makes it possible to address dynamic tethering in the frame-of-reference paradigm.

We have experimented with a number of different subsets of possible couplings between a zoomport and its parent, and have provided a set of buttons on the zoomport borders for this purpose. One of our informal results is that we find azimuth coupling to be more useful than elevation coupling. Other results indicate that both relative and absolute couplings are useful in heading, while absolute coupling in position is more useful than relative position coupling. Our selection of buttons reflects these results in our choice of a button for relative-coupling in heading and one for absolute-coupling in position (upper right corner of zoomports in Figure 3), supplemented by position-setting and heading-setting buttons (upper left corner). In general we feel that coupling requirements are quite task specific and are likely to be most useful under specialized applications, such as steering a vehicle.

2.3. Frame-of-Reference Operations

There are many instances in which it is not so important what individual objects are doing as what their emergent behavior is. For instance, the user may want to monitor the extent of a fleet of vessels for their progress in a survey mission or a search-and-rescue mission. Alternatively, the user may wish to be alerted to potential collision conditions in the management of a busy port, and have views update dynamically to monitor close calls that may require intervention. It is for these sorts of situations that we have developed frame-of-reference operations.

A *frame of reference operation* (FoR-op) is a generalization of a reference-frame coupling. A FoR-op is itself a reference frame that in some way aggregates position, orientation, and/or scale information from one or more other frames of reference. We refer to the reference frame of the FoR-op itself as the *resultant* reference frame, and the frames of reference that it aggregates as *operand* reference frames. We now describe two examples of FoR-ops from GeoZui3D to help illustrate the concept. The first is the *overview* FoR-op, and the second is the *closest-proximity* FoR-op.

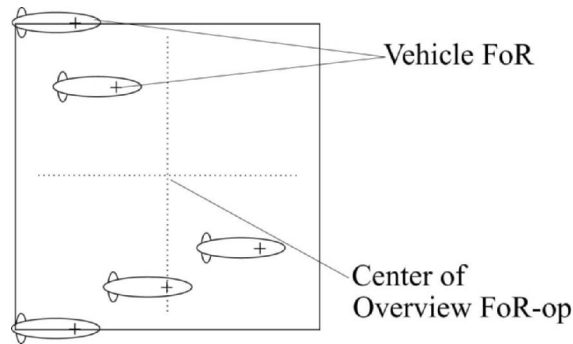


Figure 7: 2-dimensional representation of the overview FoR-op, applied to five vehicles.

Figure 7 shows a 2-dimensional representation of the idea behind the overview FoR-op as implemented in GeoZui3D. A cube is created that encompasses the operand reference frames, generated from the minimum and maximum x , y , and z coordinates among the operands. The coordinates of the center of this cube become the origin for the resultant of the overview FoR-op, and the size of the cube is stored as the scale of the resultant. The average orientation of the operands is stored as the orientation of the resultant, as well. The resultant of the overview FoR-op characterizes all the information needed to monitor the operands as a collection.

The reference frame of the overview resultant is designed to be the target of a coupling with a zoomport. It makes it possible, for example, to couple a zoomport to a group of vehicles and follow them no matter where they go, or how far apart they stray from each other (see Figure 9 for an illustration). With a coupling in position and scale, the zoomport widens its view as the vehicles spread apart, since the cube containing the vehicles' reference frames gets larger. When the vehicles move closer together, the cube contracts, as does the view. If a zoomport coupling on orientation is enabled, the view points in the emergent direction in which the vehicles are heading.

Figure 8 depicts a 2-dimensional representation of the idea behind the closest-proximity FoR-op as implemented in GeoZui3D. A cube is created that encompasses the FoR's of the two closest operands at a given time, with position and scale treated in a fashion similar to the overview FoR. The closest-proximity FoR-op is parameterized by danger thresholds. If the distance between any two operands crosses a danger threshold, the FoR-op generates a system event. The resultant of the closest-proximity FoR-op characterizes all the information needed to monitor the two closest operand reference frames at a given time and alert the user when user attention is required.

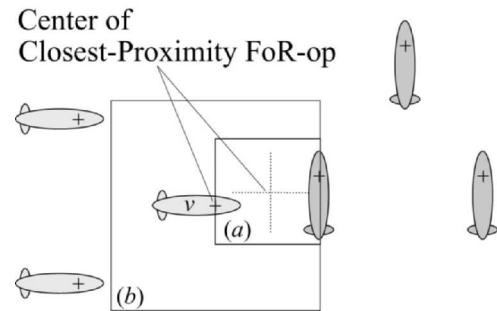


Figure 8: 2-dimensional representation of the closest-proximity FoR-op, applied to six vehicles. (a) Extent of zoomport coupled directly to the resultant (absolute coupling). (b) Extent of zoomport position-coupled to vehicle v and scale-coupled (relative coupling) to the resultant.

The reference frame of the closest-proximity resultant can be used in a couple of ways for coupling with a zoomport. A zoomport could be coupled to the resultant with absolute couplings in both position and scale, with an effect similar to an overview on the two closest operands at given time (Figure 8a). However, when there is an emergency situation, the user will often want to take quick action, and the interface should facilitate such action. This is where the event triggers come into play. For instance, consider a situation in which two fleets are passing near each other and two of the member vehicles are coming dangerously close, as shown in Figure 8b. The event generated by the closest-proximity FoR-op could trigger a warning message and create a zoomport focused on the event. This zoomport could be position-coupled to one of the vehicles (vehicle v in the figure) and scale-coupled to the resultant of the FoR-op, giving the user the opportunity to steer vehicle v to safety. Alternatively, if vehicle v were in a docking situation, as illustrated in Figure 9, the scale coupling could be especially useful in making certain that optimal alignment is maintained.

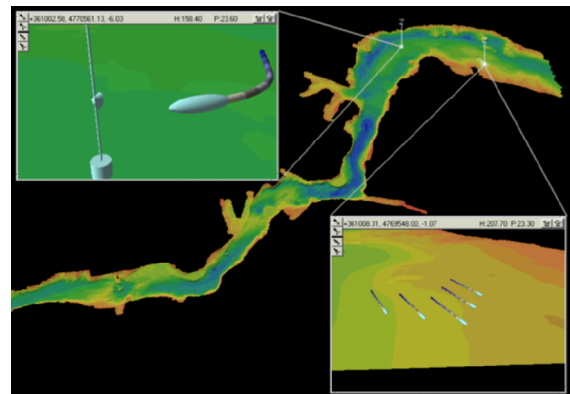


Figure 9: Zoomport coupled to a closest-proximity FoR-op (top) and another coupled to an averaging-overview FoR-op (bottom).

FoR-ops can be further extended to account for additional information associated with operands, such as velocity, fuel remaining, and repair status. The guiding principle in the use of a FoR-op is that it should directly map to a higher-level relationship between objects that the user is likely to have interest in. When used in this way, the FoR-op resultant acts as a sort of “chunk”, aggregating the information and relationships of the operands into a single focus of interest. Because the FoR-op resultant is itself a reference frame, it can also be the target of even higher-level relationships represented by FoR-ops. Ideally, each level of FoR-op should reduce cognitive load on the user by automatically monitoring important relationships and alerting the user only when specific conditions requiring user attention are met.

2.4. Additional FoR Benefits

The characterization of a view by a frame of reference about a designated center of workspace has several advantages. We have already seen how well user attention harmonizes with this characterization, and how easily multiple such windows can be integrated with each other. Other benefits include exceptional suitability to stereo viewing and a natural region in which special rendering can be done.

Humans are used to investigating objects directly in front of them, within arms’ reach—approximately where the center of workspace is located. This location maps especially well to stereo display environments because it is also exactly where stereoscopic depth perception works best. This is in contrast with flying interfaces where the viewpoint is usually a long way from objects in the scene (at least without artificial manipulations such as the cyclopean scale discussed in Ware et al. [19]).

Rendering of 3D data often runs into various problems like occlusion and high computation demands. Having a designated point of attention in a view makes it easier to decide how rendering might best be done. For instance, in the field of oceanography, there may be important information both on the surface of the sea bottom, and in the structure of the sediment layers beneath that surface. In order to focus on information beneath the surface, it is necessary to remove occluding information. The center of workspace provides a natural place to designate a high degree of interest, and remove anything that would occlude that information [5, 3]. Likewise, using the workspace center to determine degree of interest can inform rendering as to what is most important to render in high resolution when resolution is at a premium.

3. The Image Plane

The image plane is the term given to describe the perceived 2D layout of 3D information seen by a human observer. It is an implicit frame of reference that is

present at all times. When using a conventional monitor for display, this image plane maps directly to the screen. Frame-of-reference interaction makes use of the image plane for simplifying interaction in 3D space in a variety of ways. We have already seen how the image plane can be used for selection, both in the selection of a new location to be brought to the center of workspace, and in the dragging of zoomport proxies. We now present two other applications of the image plane design philosophy to multi-view environments.

3.1. Zoomports vs. Worlds in Miniature

In our initial design discussions for zoomports, we considered using something like the Worlds in Miniature (WIM) concept of Stokley et al. [17]. In their system, a complete miniature version of the 3D environment was implemented within an immersion VR system. However, our requirements are for a desktop VR system, and we decided on a more conventional extension of windows for a number of reasons.

If WIMs were to be embedded in the scene, they would become rotated and scaled during navigation, making it hard to keep them visible. If they were to be placed in the scene space, but kept in-place with respect to the user viewing frustum, they could be more easily managed, but they could also be punctured or occluded by the scene itself. By placing zoomports in the image plane, all of these problems are avoided, maintenance and implementation are simplified, and user experience from conventional 2D windowing systems can transfer to a 3D environment.

3.2. Integration of Zoomport Views

Often when working with multiple views, it is desirable to identify an item or location seen in one view, within the context of another view. For objects that the computer can highlight, highlighting the desired objects can often be sufficient, as is done with the technique of *brushing* [1]. However, there are times when the user wishes to pick out a portion of an object, or even a location on a surface, which the computer cannot possibly know is the current focus of user attention. This is where a multi-view 3D cursor finds its niche.

The multi-view 3D cursor marks in every zoomport what the mouse cursor is pointing to in the zoomport the mouse cursor is currently in. The first component of the cursor is a line drawn from the world-coordinate position of the cursor (in the screen plane of the current zoomport) in the direction of the ray from the eyepoint through the cursor. Figure 10 illustrates this. Since this line indicates the space that can potentially be rendered in the pixel under the cursor, we call this line the *pixel ray* for the 3D cursor. The remaining components, also shown in Figure 10, appear only in the following case. If the pixel ray intersects a surface, an indicator point is drawn at the intersection, and vertical and horizontal

lines are added through this point. The horizontal line is parallel to the image plane of the zoomport containing the mouse cursor and indicates the horizontal extent of the zoomport at that depth, while the vertical line serves as an indicator of the intersection location even in zoomports of much larger scale (where the other components of the 3D cursor may be hard to see).

A multi-view 3D cursor constructed in this way is good for more than integrating selected locations across 3D views. The pixel ray provides much more specific directional information than a zoomport proxy does, and can therefore help the user make decisions that might otherwise be more difficult. For instance, the user may have the cursor over a 2D video feed coming in from a camera whose position and orientation are tracked by the computer. The pixel ray can point into the virtual world to indicate what is most likely to be under the cursor in the video feed.

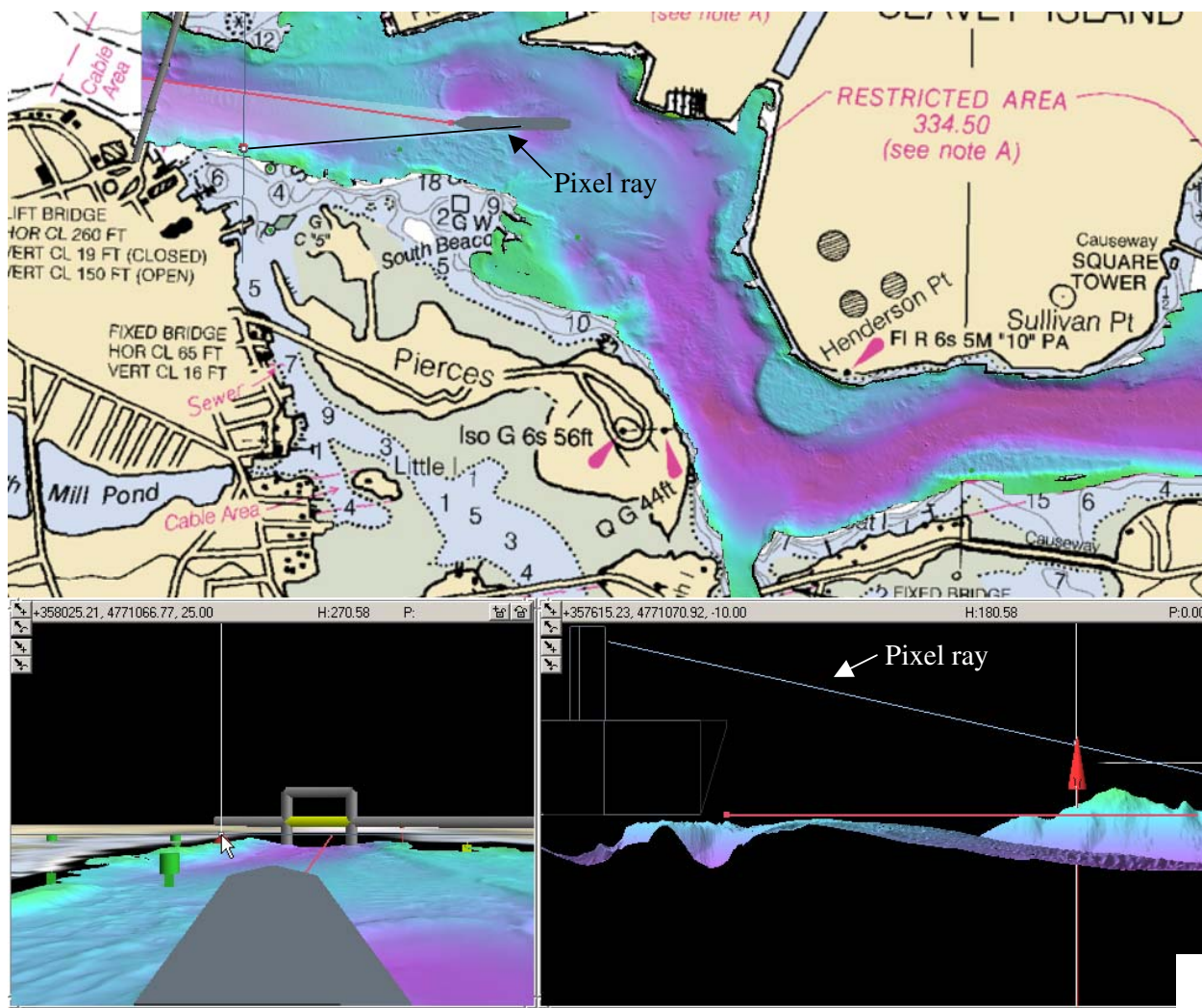
Another advantage of this multi-view 3D cursor is that it updates continuously, and can be used to help in comprehending direction of motion in a rotated view. For instance, it may be clear in one view what path a vehicle is likely to take in following a surface, but unclear in another view that is rotated significantly with respect to the first. By tracing out the motion with the

3D cursor in the first window, the motion can be readily seen in the context of the second.

One potential disadvantage of such a cursor is that the user may want to temporarily mark a location, and then perform an operation on that location in the context of another view. This disadvantage is obviated by allowing in GeoZui3D by allowing the user to “anchor” the cursor by clicking with the left mouse button. This causes the components of the cursor to remain in place until the user activates the 3D cursor again. When the 3D cursor is anchored in this manner, it is anchored with respect to the reference frame of the first object to intersect the pixel ray (if one exists). This assumes that the user is interested in the surface or object, rather than the position at which that surface or object happened to be in at the time of anchoring. While not always the case, this is generally a safe assumption to make.

4. Conclusions

We believe that frame-of-reference interaction provides a useful unifying concept for designing multiple 3D windowing environments. The use of a FoR as a center of workspace provides a meaningful location for dragging the proxy of a zoomport in the context of its



parent. The capability of coupling FoR's together provides ways to "hop-aboard" moving objects, making it possible to navigate seamlessly in dynamic environments. Couplings also provide ways to quickly construct linked views such as forward-up overviews, and can be combined with FoR-ops to construct views of aggregate groups of objects. Finally, the reference frame of the screen is a powerful tool in making it possible to use a 2D input device such as the mouse to interact with multiple 3D views.

In an effort to determine the effectiveness of FoR linkage techniques in particular task domains, we are undertaking a series of experiments, the first of which have been recently published [12]. Our first experiments investigated the effectiveness of three linkage mechanisms in supporting user decision-making in identifying a target across two views. The linkage mechanisms explored were zoomport proxies, tethers, and absolute-couplings in position and orientation. The results indicated that proxies were extremely valuable in reducing user errors, and that zoomport coupling was helpful as well. We expect similar future experiments will help quantify the effectiveness of the multi-view 3D cursor, and other experiments will help determine what situations benefit most from the ability to "hop-aboard" moving objects or establish couplings between zoomports.

5. Acknowledgements

This research was funded by NSF Grant 0081292 and NOAA Grant NA17OG2285.

6. References

- [1]. Becker, R. A. and Cleveland, W. S. 1987. Brushing Scatterplots. *Technometrics*, 29(2), 127-142.
- [2]. Bederson, B. B. and Hollan, J. D. 1994. Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. *Proceedings of UIST '94*, ACM Press, 17-26.
- [3]. Carpendale, M. S. T., Cowperthwaite, D. J., and Fracchia, F. D. 1997. Extending Distortion Viewing from 2D to 3D. *IEEE Computer Graphics and Applications*, July/Aug, 42-51.
- [4]. Darken, R. P. and Cevik, H. 1999. Map Usage in Virtual Environments: Orientation Issues. *Proceedings of IEEE Virtual Reality '99*, 133-140.
- [5]. Furnas, G. W. 1986. Generalized Fisheye Views. *Proceedings of CHI '86*, ACM Press, 16-23.
- [6]. Holm, R., Stauder, E., Wagner, R., Priglinger, M., and Volkert, J. 2002. A combined immersive and desktop authoring tool for virtual environments. *Proceedings of IEEE Virtual Reality 2002*, 93-100.
- [7]. Lamping, J., Rao, R., and Pirolli, P. 1995. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. *Proceedings of CHI '95*, ACM Press, 401-408.
- [8]. Levine, M., Marchon, I., and Hanley, G. 1984. The Placement and Misplacement of You-Are-Here Maps. *Environment and Behavior*, 16, 2, 139-157.
- [9]. Parker, G., Franck, G., and Ware, C. 1998. Visualization of Large Nested Graphs in 3D: Navigation and Interaction. *Journal of Visual Languages and Computing*, Vol 9, Academic Press, 299-317.
- [10]. Pierce, J. S., Forsberg, S., Conway, M. J., Hong, S., Zeleznik, R., and Mine, M. R. 1997. Image Plane Interaction Techniques in 3D Immersive Environments. *1997 Symposium on Interactive 3D Graphics*, ACM Press, New York, 39-43.
- [11]. Plumlee, M., Komerska, R., Arsenault, R., Ware, C., and Mayer, L. 2001. Visualization Techniques To Support Monitoring And Control Of Autonomous Platforms. *12th International Symposium on Unmanned Untethered Submersible Technology UUST '01*, CD Proceedings.
- [12]. Plumlee, M. and Ware, C. 2003. An Evaluation of Methods for Linking 3D Views. *Proceedings of ACM SIGGRAPH 2003 Symposium on Interactive 3D Graphics*, Monterey, California, ACM Press, New York, in press.
- [13]. Plumlee, M. and Ware, C. 2002. Frame of Reference Interaction. *UIST 2002 Companion*, Paris, 41-42.
- [14]. Plumlee, M. and Ware, C. 2002. Zooming, Multiple Windows, and Visual Working Memory. *Proceedings of AVI 2002*, Trento, Italy, ACM Press, New York, 59-68.
- [15]. Sarkar, M. and Brown, M. H. 1994. Graphical Fisheye Views. *Communications of the ACM*, 47, 12 (December) ACM Press, 73-84.
- [16]. Sayle, R.A. and Milner-White, E.J. 1995. RasMol: Biomolecular graphics for all, 20, 374-376.
- [17]. Stoakley, R., Conway, M. J., and Pausch, R. 1995. Virtual Reality on a WIM: Interactive Worlds in Miniature. *Human Factors in Computing Systems CHI '95 Proceedings*, Denver, ACM Press/Addison Wesley, New York, 265-272.
- [18]. Wang, W. and Milgram, P. 2002. Viewpoint Optimisation for Navigation Using Dynamic Tether. *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting*, 2164-2168.
- [19]. Ware, C., Gobrecht, C., and Paton, M. 1998. Dynamic Adjustment of Stereo Display Parameters. *IEEE Transactions on Systems, Man and Cybernetics*, 28(1), 56-65.
- [20]. Ware, C. and Lewis, M. 1995. The DragMag Image Magnifier. *CHI '95 Companion*, Denver, ACM Press, New York, 407-408.
- [21]. Ware, C., Plumlee, M., Arsenault, R., Mayer, L.A., Smith, S., and House, D. 2001. GeoZui3D: Data Fusion for Interpreting Oceanographic Data. *Oceans 2001*, Hawaii, Proceedings, 1960-1964.
- [22]. Wickens, C. D., and Prevett, T. 1995. Exploring the Dimensions of Egocentricity in Aircraft Navigation Displays. *Journal of Experimental Psychology: Applied*, 1(2), 110-135.
- [23]. Wolff, R.S. and Yaeger, L.. (1994) Visualization of Natural Phenomena, TELOS, The Electronic Library of Science, Springer-Verlag, New York.