

Computational Steering in the CAVE

Jurriaan D. Mulder^{a,1}, Robert van Liere^{a,1},
Jarke J. van Wijk^{b,2}

^a *Center for Mathematics and Computer Science CWI, P.O. Box 94079, 1090 GB
Amsterdam, the Netherlands*

^b *Netherlands Energy Research Foundation ECN, P.O. Box 1, 1755 ZG Petten,
the Netherlands*

Abstract

Scientists can gain much more insight from their simulations if they are enabled to change simulation parameters on the fly while observing the results immediately. A crucial aspect of such *computational steering* is an intuitive user interface. We have developed an environment that enables researchers to construct such interfaces efficiently and effectively for graphical workstations. In this paper we report on our next step towards more intuitive user-interfaces: We have modified our system for use in the CAVE.

The CAVE is a projection-based virtual environment. Virtual environments are designed to provide the effect of immersion in an interactive three-dimensional computer-generated environment. We show that the use of virtual environments for computational steering interfaces can improve interaction with the simulation and immersion in the computational process. We present our system, the methods we have developed for improved 3D interaction, and describe three applications.

Keywords: Computational steering, virtual environments, 3D graphics and interaction.

1 Introduction

Scientific visualization has traditionally been a post-processing process. Upon completion of a simulation the gathered data is processed and visualized. In this case, the user interaction with the simulation is limited. Computational steering enables the user to study a simulation's behavior interactively: results

¹ {mullie|robertl}@cwi.nl

² vanwijk@ecn.nl

are visualized while the simulation is in progress and, simultaneously, the user can manipulate the simulation’s input parameters. In this way, the user can interactively explore parameter spaces and experiment with different settings to obtain additional insight in the underlying computational model.

Computational steering is a highly interactive process. Therefore, an intuitive user interface is one of the key requirements for a steering application to be effective. It is important that the user interface provides intuitive methods for interaction, both for the visualization of the simulation results and for the manipulation of the simulation input parameters. These aspects become particularly critical for three-dimensional simulations.

In recent years, the use of virtual environments in scientific visualization has been explored. Virtual environments are designed to provide the effect of immersion in an interactive three-dimensional computer-generated environment [3]. For scientific visualization, the aspect of immersion can make a valuable contribution to the perception of a visualization compared to 3D graphics displayed on desktop graphics workstations. For computational steering, both the aspects of immersion and interaction can contribute to the effectiveness of the interface.

The main differences between interactive 3D desktop graphics and virtual environments are listed in table 1 (as formulated by Cruz-Neira [5]). Each of the aspects listed in the table concern user interaction, both viewing and manipulation. The table indicates that virtual environments in potential can provide more intuitive means for viewing and manipulation. Therefore, it can be expected that virtual environments will lift user participation in 3D computational steering applications to a higher level. However, both viewing and intuitive manipulation in virtual environments are still current research issues.

<i>Interactive 3D Graphics</i>	<i>Virtual Environments</i>
Screen used as canvas	Displays used as 3D space
Tele-manipulation	Direct manipulation
3D to 2D mapping of user’s actions	3D to 3D mapping of user’s actions
No multiple simultaneous interactions	Multiple simultaneous interactions
Users always aware of interaction	Interaction is transparent

Table 1

The main differences in interaction and display between interactive 3D desktop graphics and virtual environments (from Cruz-Neira [5].)

The CAVE is a virtual reality system developed at the University of Chicago [6]. The goals that motivated the design of the CAVE included “. . . to produce a

virtual reality environment that was suitable for scientific research and to provide a user-interface to steer high performance computing applications running on remote super computers.” [17]. Stereoscopic images are rear-projected on three walls and down-projected on the floor. The correct projection is determined via the position and orientation of the user’s head. Users wear shutter glasses for the stereo depth perception. Input is realized through a six-degree-of-freedom wand with three buttons and a joystick.

Many researchers have also used head mounted displays (HMDs) for scientific visualization applications. However, there are a number reasons why using the CAVE may be more adequate than HMDs. First, the CAVE provides a wider field of view than current HMDs. This can be a benefit for the perception of large scenes. Secondly, with HMDs each head movement requires a re-rendering of the scene according to the new line of sight. This can lead to severe latencies between head movement and actual rendering. This can be very disturbing and it is considered one of the major causes for cyber sickness. In the CAVE, simple head movements such as rotations or nodding, require the scene to be re-rendered with only an adjustment to the stereo perspective. The CAVE user can still see the image projections on the walls, with only a slight distortion in the stereo perspective. Thirdly, multiple users can share the same immersive experience in the CAVE, resulting in a natural environment for collaborative work. Note however that, since only one user is tracked, the stereo projections are not fully correct for the other users.

In this paper, we describe how we have modified the Computational Steering Environment (CSE [11]) to take advantage of the opportunities of the CAVE. We found that virtual environments can provide very intuitive interfaces to steering applications. As a result, well designed virtual environments enable users to actively participate with ongoing simulations. In section 2, we discuss related work on computational steering and the use of virtual environments. Then we discuss the CSE in the CAVE. We describe how the PGO editor has been extended to provide enhanced viewing and input capabilities. Finally, we present three applications and discuss our experiences.

2 Related work

Many research groups have studied computational steering. Reported results vary from presentations of single computational steering applications (e.g. the steering of a 3D turbulence model of Lake Erie developed by Marshall et al. [13]) to the development of general environments for computational steering (e.g. the SCIRun environment by Parker et al. [16] and the VASE environment by Jablonowski et al. [10]). Vetter [19] has composed an annotated bibliography that provides a broad overview of computational steering

research. Relatively little research however, has been done on using virtual environments in computational steering applications.

Virtual environments have been used in scientific visualization to exploit the benefits over interactive 3D desktop graphics. Bryson et al. [4] have developed the Virtual Windtunnel, a system designed to visualize precomputed simulated unsteady three-dimensional fluid flows through the use of a Binocular Omni Orientation Monitor (BOOM). The user can interactively position collections of seedpoints for streamlines and streaklines with the use of an instrumented glove, and in addition control the passage of time and the scale of the display and the objects through the use of menus and sliders. The author states that virtual environment interfaces "...are useful (...) for the exploration and hopefully discovery of phenomena ..." [2], primarily because of the more natural and intuitive interface between human and computer for viewing and manipulation.

A more recent example is presented by van Teylingen et al. [18]. They describe the Virtual Data Visualizer (VDV), an interactive immersive environment for visualizing and analyzing data through a head mounted display. In this environment, users can map data variables to glyph elements. The data is organized hierarchically in classes. The user can modify the data organization and the mappings to the glyphs from within the virtual environment, and he has control over display attributes such as animation and zooming. User interaction is primarily focussed at the viewing of the data and the manipulation of the glyph mappings through an interface based on menus and icons.

Roy et al. [17] have presented a computational steering application implemented in the CAVE. The main focus is on the computational requirements of computational steering. They discuss a distributed virtual reality system where the simulation is run on a remote supercomputer and the steering interface in the CAVE. Interaction is limited to starting, stopping, pausing, continuing, and restarting the simulation. Wand-controlled sliders are used to change simulation input parameters, but this does require the simulation to be stopped and restarted.

Besides more natural viewing of 3D scenes, immersive virtual environments offer more intuitive 3D object manipulations through the use of multiple degree-of-freedom input devices. However, such *spatial input* techniques are still in the development phase and it has not been determined yet which input device and manipulation technique are best for a particular manipulation task. Recent research results have been reported on techniques for object rotations [21], two handed input [20,7,9], and remote object transformations [1]. A survey of earlier research on spatial input techniques is given by Hinckley et al. [8].

3 The Computational Steering Environment in the CAVE

The CSE has been developed to allow scientists to easily integrate a simulation and create a custom interactive graphical user interface to steer a simulation. Interfaces can be run both in the CAVE and on conventional desktop graphics workstations. This provides an excellent opportunity to explore the added value of virtual environments over desktop workstations.

The CSE's architecture is based on a central Data Manager surrounded by satellite processes, which communicate with each other via the Data Manager, see figure 1. The modular architecture allows satellites to execute in a distributed environment. This allows simulations to be run on remote super-computers. Careful design of the communication and data transfer protocols ensures that data is only transported between the Data Manager and the satellite processes when needed. Many data manipulation satellites have been developed, such as a logging satellite, a calculator, and a data slicer.

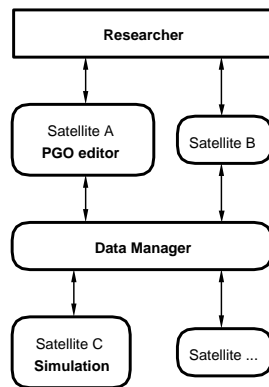


Fig. 1. Architecture of the Computational Steering Environment.

The PGO editor [14] is a graphical editor that enables the user to create custom 2D, 3D desktop, and 3D immersive user interfaces for the visualization and manipulation of the data present in the Data Manager. PGO interfaces are constructed out of *point-based Parametrized Geometric Objects*. The user can construct composite objects out of a set of simple basic objects. The bidirectional binding between objects' geometry and data in the Data Manager can be specified. As a result, objects can be used for the visualization of the simulation output data and the user can manipulate the simulation input parameters by directly manipulating the objects.

User interaction in a virtual environment differs from conventional desktop steering interfaces on two major aspects: viewing and object manipulation. To exploit the benefits virtual environments provide over desktop systems on these aspects, we have developed viewing and manipulation techniques for the PGO to be used in the immersive interfaces.

3.1 Viewing

In an immersive virtual environment, users have a natural view of the 3D scene. Users can physically move about in the CAVE or can navigate through a large 3D scene. In the PGO editor, basic navigation functionality (viewing direction, scene rotation, or scene scaling) is provided through the use of the wand.

However, such basic navigation functionality is often not sufficient. The goal of the PGO editor is to allow the user to actively participate with the ongoing simulation. For user viewpoint movements, this implies that navigation and viewing become active elements in the interface. More specifically, the following navigation and viewing functionality should be available in a computational steering interface:

- Definition of multiple viewing configurations;
- Coupling of the viewpoint to entities in the interface;
- Parametrization of the viewpoint to data;
- Construction of user defined controls over the viewpoint.

To accomplish this in the PGO editor, the parametrizable camera object [15] was developed. The camera objects define viewpoint parameters such as position and orientation, and behave identical to the other basic PGO objects. As a result, the position and/or orientation of camera can be parametrized to data that is controlled by other interface objects in the interface or by the simulation. In addition, a camera can be connected to an object present in the interface. Different viewing configurations can be defined by placing multiple camera objects. By selecting a different camera, the user can easily switch between viewing configurations.

The cameras are fully integrated in the PGO editor, and provide an easy and intuitive way to define different viewing configurations valuable to computational steering applications. Combined with the aspect of immersion provided in virtual environments, a higher level of user integration in the interface is achieved. In stead of displaying the view of the camera in a separate window, the user himself is moved to the selected camera.

3.2 Object manipulation

To provide intuitive object manipulations, seven different techniques have been developed and implemented. The CAVE's primary input device is a six degrees-of-freedom wand with three buttons and a joystick. Each of the techniques derive object translations from the movements of the wand. The tech-

niques can be classified according to two important characteristics regarding the mapping of the wand movements to object movements:

- Derived Object Motion: The performed object movements are position based, velocity based, or a combination of these two. With position based methods, the position of the object is directly coupled to the wand. Velocity based methods apply a velocity on the object to be translated. The speed and direction of the object motion is derived from the wand movements.
- Involved Wand Parameters: The second characteristic determines whether the object translations are derived from the (changes in the) wand position, orientation, or both.

Table 2 lists the developed methods. The first three of the methods use position based motion control, out of which one relates to the wand position only, while the other two relate to the wand position and orientation. In the Slave method, the object to be translated follows the exact position movements of the wand. In the Stick method, the wand is used as a pointing stick to position the object. The Crosshair method is derived from the method used for the desktop PGO editor. Here, the 3D object translations are obtained by mapping the wand movements to a 3D crosshair cursor.

The next two methods use velocity motion control, one based on the wand position and one on the wand orientation. In the Flying method, the wand is used to steer the object as it is flying through the 3D space. The throttle method uses a motorcycle throttle grip metaphor to obtain the velocity of the object as it moves in a direction indicated by the ray emitted by the wand.

The final two methods use a combination of position and velocity control, where one method is solely based on the wand position, and one on the wand position and orientation. The first method is a combination of the Slave and Flying methods, the second method is a combination of the Stick and Throttle methods.

	<i>Name</i>	<i>Motion Control</i>	<i>Wand Parameters</i>
1.	Slave	position	position
2.	Stick	position	position and orientation
3.	Crosshair	position	position and orientation
4.	Flying	velocity	position
5.	Throttle	velocity	orientation
6.	Slave & Flying	position and velocity	position and orientation
7.	Stick & Throttle	position and velocity	position and orientation

Table 2

Seven manipulation methods developed for the CAVE.

These manipulation schemes are currently being evaluated on criteria such as speed, accuracy, and user friendliness through usability tests.

4 Applications

With the use of the CSE and the PGO editor, several computational steering applications have been constructed and used in the CAVE. In this section, three applications are described. The accompanying figures are snapshots of the applications running in the CAVE simulator. The distributed architecture of the CSE allows the simulations to be run on remote (super-) computers. Furthermore, the PGO editor can run asynchronously of the simulation. These aspects allow the interfaces to be rendered at interactive frame rates.

4.1 *Bouncing Balls*

Figure 2 shows an interface that was constructed to steer a simulation of bouncing balls. The balls are visualized by spheres with an additional arrow to depict their velocity vector. Numerous steerable parameters are present in the simulation. In this particular interface, the user can grab a ball and move it to a different location, or use the sliders to manipulate the dampening property of the medium in which the balls move, adjust the number of balls, and adjust the radius of the balls. The user can also steer a field force by manipulating the arrow in the small box. The direction of the arrow depicts the direction of the force, while its length represents the strength of the force. When confronted with this interface, the user immediately feels the urge to participate in the simulation and to start manipulating the interface objects.

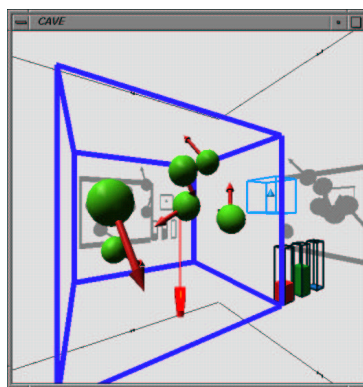


Fig. 2. Snapshot of a steering interface to a simulation of bouncing balls.

4.2 Smog Prediction

Figure 3 shows an interface to a high performance simulation for smog prediction over Europe. Previously, we have used a desk top visualization tool especially designed and implemented for this application [12]. However, the PGO editor has recently been extended with visualization functionalities (i.e. mesh visualizations with parametrizable transparency) that allows us to use it for this type of applications as well.

The simulation calculates the smog dispersion over four layers according to numerous input parameters concerning emission fields and meteorological and geographical parameters. The governing equations of the model are a set of partial differential equations that determine the advection, diffusion, emission, wet and dry deposition, fumigation, and chemical reactions. In this interface, the user can examine the different atmospheric layers and steer the simulation by manipulating pollution emission sources.

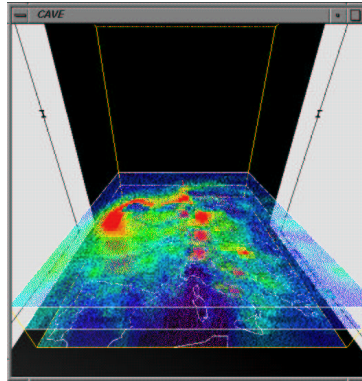


Fig. 3. Snapshot of a steering interface to a simulation for smog prediction.

4.3 East-Indiaman

Figure 4 shows an interface to a simulation of the sailing characteristics of an East-Indiaman. At each time step, the simulation calculates the new state of the ship. This new state is derived from the prior state and the wind conditions (direction and force), the wave spectrum of the sea, the position, orientation, and surface area of the sails, the orientation of the rudder, etc.. In this interface, the user can literally steer the ship: sails can be set, the sail area can be enlarged or reduced to adjust to different wind forces, the sails can be aimed to different angles to adjust to different wind directions and ship courses, and the user can use the rudder and the sails to change the course of the ship. In addition, the wind force and direction can be changed. Cameras can be

positioned aboard the ship or in the masts to obtain an ‘on-board’ sailing experience.

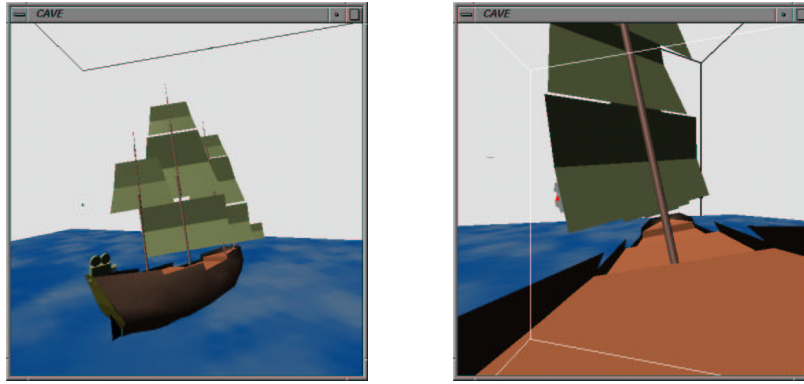


Fig. 4. Snapshot of a steering interface to a simulation of a sailing East-Indiaman. Left an overview, right an on-board view from the camera mounted on the rear deck.

5 Discussion

The major differences between virtual environments and conventional interactive 3D graphics are the aspects of immersion and interaction. We expected that these aspects could lift the user participation in a computational steering application to a higher level. Although we have not yet conducted any formal experiments we can make some remarks about our experiences.

Hinckley et al. stated that “... people do not innately understand three dimensional reality, but rather they experience it.” [8]. From our experience, we found that immersion can contribute significantly to experiencing the 3D scene compared to the desktop 3D graphics approach, and therefore lift user participation to a higher level. For example, in the bouncing balls application it is easier to perceive the 3D positions and velocities of the balls, and relate the individual ball movements to the state of the simulation. Another illustrative example is found in the East-Indiaman application. By placing the user’s viewpoint aboard the ship (through the use of the camera object), an on board sailing experience was obtained that could not be accomplished on a conventional desktop graphics workstation.

For object manipulation in the CAVE, seven different techniques are currently being evaluated. More effective object manipulation contributes to a higher degree of user participation. For comparison, the same usability tests are also conducted with the manipulation techniques used on the desktop workstations. Our experiences obtained during the development and testing of the methods

indicate that object manipulations in the CAVE can be performed faster and more accurate than on the desktop workstations.

These observations indicate that the use of virtual environments allow more intuitive and effective interfaces. However, much research still is needed. For instance, it is not yet known under which exact conditions steering will benefit from virtual environments. Some applications may still be more effective when used on desktop systems.

6 Conclusion

Computational steering allows a user to get better insight in and understanding of the behavior and results of the simulation in less time. To achieve this, the user interface plays a crucial role. The user should be able to view and interact with the simulation in an easy and intuitive manner to actively participate with the ongoing simulation. Virtual reality systems, and the CAVE in particular, can provide such an interface. Combined with the CSE and the PGO editor, the CAVE offers a general purpose computational steering environment that allows a user to create and use immersive 3D interfaces for computational steering.

Future work we intend to address concerns the development of automated navigation strategies to use in conjunction with the camera objects and further development and evaluation of 3D object manipulation techniques. In general, we believe that more research is needed in formal comparisons between the use of virtual environments and interactive 3D graphics on desktop workstations.

References

- [1] D.A. Bowman and L.F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In S.N. Spencer, editor, *1997 Symposium on Interactive 3D Graphics*, pages 35–38, 1997.
- [2] S. Bryson. Virtual reality in scientific visualization. *Computers & Graphics*, 17(6):679–685, 1993.
- [3] S. Bryson. Approaches to the successful design and implementation of VR applications. In R.A. Earnshaw, J.A. Vince, and H. Jones, editors, *Virtual Reality Applications*, pages 3–15. Academic Press, 1995.
- [4] S. Bryson and C. Levit. The virtual windtunnel: An environment for the exploration of three-dimensional unsteady fluid flows. In *Visualization '91*

- (*Proceedings of the 1991 Visualization Conference*), pages 17–24, 1991.
- [5] C. Cruz-Neira. Making VR a useful technology: From immersive displays to applications. Invited talk at the 1997 EURO-VR Mini Conference, Amsterdam, the Netherlands, November 10–11 1997.
 - [6] C. Cruz-Neira, D.J. Sandin, and T.A. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the CAVE. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 135–142, 1993.
 - [7] L.D. Cutler, B. Fröhlich, and P. Hanrahan. Two-handed direct manipulation on the responsive workbench. In S.N. Spencer, editor, *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, pages 107–114, 1997.
 - [8] K. Hinckley, R. Pausch, J.C. Goble, and N.F. Kassell. A survey of design issues in spatial input. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '94)*, pages 213–222, 1994.
 - [9] K. Hinckley, R. Pausch, and D. Proffitt. Attention and visual feedback: The bimanual frame of reference. In S.N. Spencer, editor, *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, pages 121–126, 1997.
 - [10] D.J. Jablonowski, J.D. Bruner, B. Bliss, and R.B. Haber. VASE: The visualization and application steering environment. In *Proceedings of Supercomputing '93*, pages 560–569, 1993.
 - [11] R. van Liere, J.D. Mulder, and J.J. van Wijk. Computational steering. *Future Generation Computer Systems*, 12(5):441–450, April 1997.
 - [12] R. van Liere and J.J. van Wijk. Steering smog prediction. In *Proceedings of the 1997 Conference on High-Performance Computing and Networking*, pages 241–252. Springer Verlag, April 1997.
 - [13] R. Marshall, J. Kempf, S. Dyer, and C.-C. Yen. Visualization methods and simulation steering for a 3D turbulence model of Lake Erie. *Computer Graphics*, 24(2):89–97, 1990.
 - [14] J.D. Mulder and J.J. van Wijk. 3D computational steering with parametrized geometric objects. In G.M. Nielson and D. Silver, editors, *Visualization '95 (Proceedings of the 1995 Visualization Conference)*, pages 304–311, 1995.
 - [15] J.D. Mulder and J.J. van Wijk. Parametrizable cameras for 3D computational steering. In W. Lefer and M. Grave, editors, *Visualization in Scientific Computing '97, Proceedings of the eighth Eurographics Workshop*, pages 165–175. Springer Verlag, 1997.
 - [16] S.G. Parker and C.R. Johnson. SCIRun: a scientific programming environment for computational steering. In *Proceedings of Supercomputing '95*, 1995.
 - [17] T.M. Roy, C. Cruz-Neira, and T.A. DeFanti. Cosmic Worm in the CAVE: Steering a high performance computing application from a virtual environment. *Presence: Teleoperators and Virtual Environments*, 4(2):121–129, 1995.

- [18] R. van Teylingen, W. Ribarsky, and C. van der Mast. Virtual Data Visualizer. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):65–74, January-March 1997.
- [19] J. Vetter. Computational steering annotated bibliography. *SIGPLAN Notices*, 32(6):40–44, June 1997.
- [20] R.C. Zeleznik, A.S. Forsberg, and P.S. Strauss. Two pointer input for 3D interaction. In S.N. Spencer, editor, *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, pages 115–120, 1997.
- [21] S. Zhai, P. Milgram, and A. Rastogi. Anisotropic human performance in six degree-of-freedom tracking: An evaluation of three-dimensional display and control devices. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 27(4):518–528, 1997.