

# Simulation Exploration through Immersive Parallel Planes

Nicholas Brunhart-Lupo\*

National Renewable Energy Laboratory

Brian W. Bush†

National Renewable Energy Laboratory

Kenny Gruchalla‡

National Renewable Energy Laboratory

Steve Smith§

Los Alamos Visualization Associates

## ABSTRACT

We present a visualization-driven simulation system that tightly couples systems dynamics simulations with an immersive virtual environment to allow analysts to rapidly develop and test hypotheses in a high-dimensional parameter space. To accomplish this, we generalize the two-dimensional parallel-coordinates statistical graphic as an immersive “parallel-planes” visualization for multivariate time series emitted by simulations running in parallel with the visualization. In contrast to traditional parallel coordinate’s mapping the multivariate dimensions onto coordinate axes represented by a series of parallel lines, we map pairs of the multivariate dimensions onto a series of parallel rectangles. As in the case of parallel coordinates, each individual observation in the dataset is mapped to a polyline whose vertices coincide with its coordinate values. Regions of the rectangles can be “brushed” to highlight and select observations of interest: a “slider” control allows the user to filter the observations by their time coordinate. In an immersive virtual environment, users interact with the parallel planes using a joystick that can select regions on the planes, manipulate selection, and filter time. The brushing and selection actions are used to both explore existing data as well as to launch additional simulations corresponding to the visually selected portions of the input parameter space. As soon as the new simulations complete, their resulting observations are displayed in the virtual environment. This tight feedback loop between simulation and immersive analytics accelerates users’ realization of insights about the simulation and its output.

**Index Terms:** I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality

## 1 INTRODUCTION

Parallel-coordinates statistical graphics [12] have proven value in the interactive exploration of moderately large multivariate datasets [13]. However, for sufficiently large datasets they can suffer from overplotting of the polylines representing individual observations, which obscures the underlying correlations in the observations and introduces spurious visual correlations between the observations. Various solutions have been proposed to the over-plotting problem, including the clustering and sub-setting of the observations [5, 14] or replacing the polylines with density representations [16, 19, 9]. Embedding such graphics in three dimensions provides an extra degree of freedom to separate observations, offering an opportunity to reveal a better view of correlations by visually “untangling” the overlap present in two dimensions. Several approaches to utilizing this extra dimension have been implemented in the past by other researchers: Wegenkittl et al. [25] extended parallel coordinates for dynamical

systems trajectories by extruding the polylines into the third dimension as a function of time. Johansson et al. [15] map vertical axes to the edge of cylinders each connecting to a single “focus” axes in the center. Streit et al. [23] extrude parallel coordinate line density into the third dimension. Multiple researchers [4, 22] have implemented versions of “parallel planes” where the single line-segment axes of parallel coordinates are replaced by pairs of axes arranged as a series of rectangles or planes.

In this paper we adopt the notion of parallel planes into a fully interactive immersive virtual environment, and extend the notion of the interface to not only explore existing data, but to explore the system parameter spaces by interactively spawning parallel simulations directly from the immersive environment. The extension of the parallel coordinates from a surface into a volume alleviates the over-plotting problem and enables users to better spot correlations between observations; however, that feature separation can easily be obfuscated when projecting the three-dimensional construction back onto a two-dimensional display. Likewise, interacting with a three-dimensional parallel coordinate construction in a two-dimensional projection becomes cumbersome at best. It is reasonable to suspect that an immersive environment would improve these visualizations. The three-dimensional stereo and motion parallaxes naturally increase the visibility of the separated lines, and users can directly interact with the visualization by brushing the coordinate planes with a virtual “brush” or “flashlight”, easily highlighting the observations temporarily of interest. Additionally, it has been hypothesized [8] that the egocentric perspective may actually improve spatial reasoning through embodied cognition (i.e., the natural body movement and body-relative spatial judgments provided by an immersive environment engages well-practiced automatic brain function acquired by what we have encountered and manipulated with our bodies as we move within and examine the world.) Specific to the visualization of statistical graphics, there have been multiple controlled user studies [2, 21, 10] that have shown benefits to visualizing statistical graphics in immersive environments.

To support the exploration of complex high-dimensional parameter spaces, our immersive implementation of parallel planes is tied to a cluster of on-demand simulations that emit the multivariate time series. In the context of such exploration of the output from a simulation, an impediment to exploring correlations in its inputs and outputs through the parallel-coordinates technique occurs when the brushing of multiple axes for simulation-input dimensions results in filtering the dataset until no more simulated observations remain. This leaves the user without information regarding what output would correspond to the filter on input. When this occurs in practice, the next step would be to run additional simulations for the inputs domain of interest and then to restart the parallel-coordinates visualization with the expanded datasets. In our simulation-linked implementation, however, the brushing and selection actions on the simulation’s input variables trigger the launching of additional simulations corresponding to the visually selected portions of the input parameter space. As soon as the new simulations complete, their resulting observations are displayed in the virtual environment. This tight feedback loop between simulation and immersive analytics accelerates users’ realization of insights about the simulation and its

\*e-mail: nicholas.brunhart-lupo@nrel.gov

†e-mail:brian.bush@nrel.gov

‡e-mail:kenny.gruchalla@nrel.gov

§e-mail:sas@lava3d.com

output.

## 2 APPROACH

### 2.1 Parallel-Planes in an Immersive Environment

We map multivariate observations  $(x_1, x_2, \dots, x_n)$  into three-dimensional space by treating each observation as a line that intersects the rectangles representing coordinate axes at the observation's coordinate values on those axes (see Figure 1). The intersections of the observations with a plane serves as a two dimensional scatter plot. Consequentially, the overall visualization of parallel planes can be thought of as a series of scatter plots where the same observations on each plot are joined by a polyline. The planes are embedded in the immersive environment somewhat like books on a bookshelf, with one of their axes pointing upward and the other axis pointing depth-wards. The names of the variables appear next to the corresponding axes of each plane.

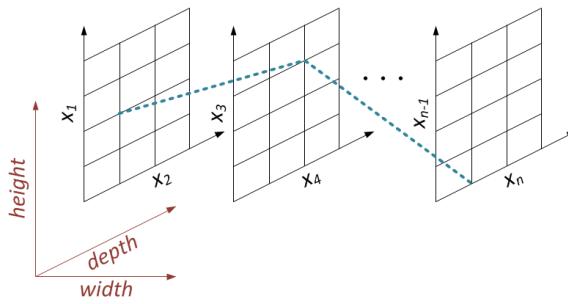


Figure 1: Representation of the observation  $(x_1, x_2, \dots, x_n)$  using parallel planes in an immersive environment: the dashed polyline intersects each plane at the corresponding coordinate values.

Brushing and selection of the observations are accomplished through the use of a virtual tool in the shape of a truncated cone, as shown in Figure 2. In the immersive environment, the base of the cone appears to be connected to the joystick controller (Figure 3) and the cone's tip points in the direction that the controller points. In order to prevent the cone from obscuring the view of the observations, the cone has an opacity gradient from the tip, which is completely opaque, to the base, which is mostly transparent. The intersection of the cone with the parallel planes defines the selection and brushing region. Users can select small regions by pushing just the tip of the cone into a coordinate plane, and they can select larger regions by pushing the cone more deeply into the plane. The opacity gradient is tuned to ensure that the user can see the observations while they are using the tool to select them. Buttons on the control allow the user to select the intersection region, deselect it, or clear all of the selections in the environment. The observations passing through all of the selected regions of the planes have their polylines displayed as thin, opaque white tubes, whereas the observations that do not pass through all of the selected regions appear in transparent gray (see Figure 4). (The planes with no selected regions do not filter or exclude observations.)

Additional buttons on the controller allow the user to scale, translate, and rotate the display. When displaying time series data, a time axis is displayed below and forward of the parallel planes. Buttons on the controller move the time step forward or backwards.

The parallel planes visualization has many of the same strengths and drawbacks as traditional parallel-coordinates plots [13, 17]. In particular, the visualizations are more effective if the pairs of axes are chosen and the planes are ordered in an efficient manner [1].

In addition to simply brushing or highlighting the selected observations, when coupled to a simulation (instead of to a precomputed dataset), this parallel-planes visualization system can request additional simulations to be run for the selected regions of the input

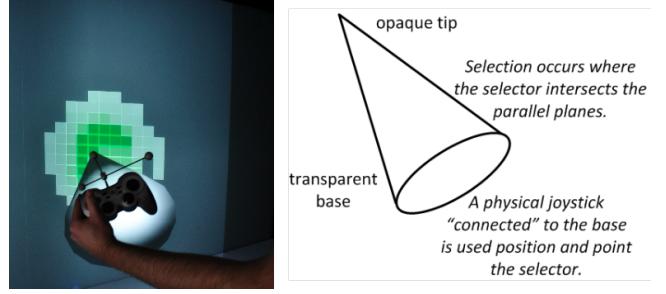


Figure 2: The selection virtual tool, which is anchored in front of a physical joystick. The user can a select region of interest by intersecting the cone with a display plane. The simulation probability density of the selection can be manipulated with the joystick buttons.

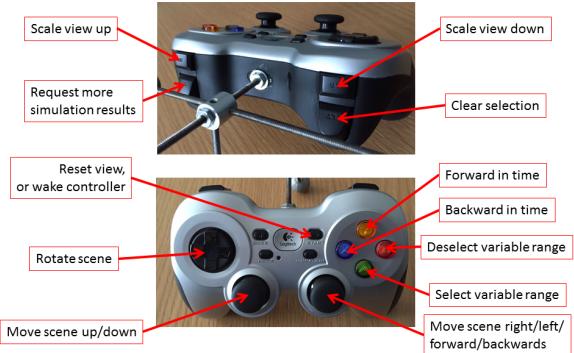


Figure 3: Joystick controller for the selector and other functions.

variables. (Input and output variables are distinguished on the display by the color of their names.) If the user keeps the controller's selection button depressed, then a higher "density" of selection occurs: this higher density of selection appears as the thickness of the selection on the plane. When the button to request more simulations is pressed, the multivariate inputs to the simulation are chosen according to a probability distribution that is in proportion to the visual density of selection. This density-dependent response to simulation requests enables nuanced exploration of the simulation space. The visualizer typically creates a dozen simulation requests each time the simulation-request button is pressed. Pressing the button multiple times, or keeping it depressed, creates correspondingly more simulation requests. As soon as each new simulation completes, the visualizer displays the results of the simulations in the immersive environment.

We have deployed this parallel-planes simulation-visualization system for the immersive environment in the Energy Systems Integration Facility at the National Renewable Energy Laboratory (NREL). Figure 5 shows its appearance in that environment. This is a two-surface environment with four blended stereoscopic HD projectors rear-projecting a  $5m \times 2.5m$  wall (a blended resolution of  $3456 \times 1728$ ) and two stereoscopic HD projectors front-projecting a  $5m \times 1.6m$  floor (a blended resolution of  $3456 \times 1080$ ). The projected space is used in conjunction with an optical motion tracking system that tracks the position and orientation of the user, allowing the visualizations to respond in relation to the movement of the user.

The visualizer was constructed using an in-house immersive graphics engine. This engine was written in C++ using the Qt framework, OpenGL, VRPN [24], and OpenMPI [6] on Linux. Crucially, the engine provides a distributed scene-graph and database integration.

In addition to the aforementioned high performance visualizer for

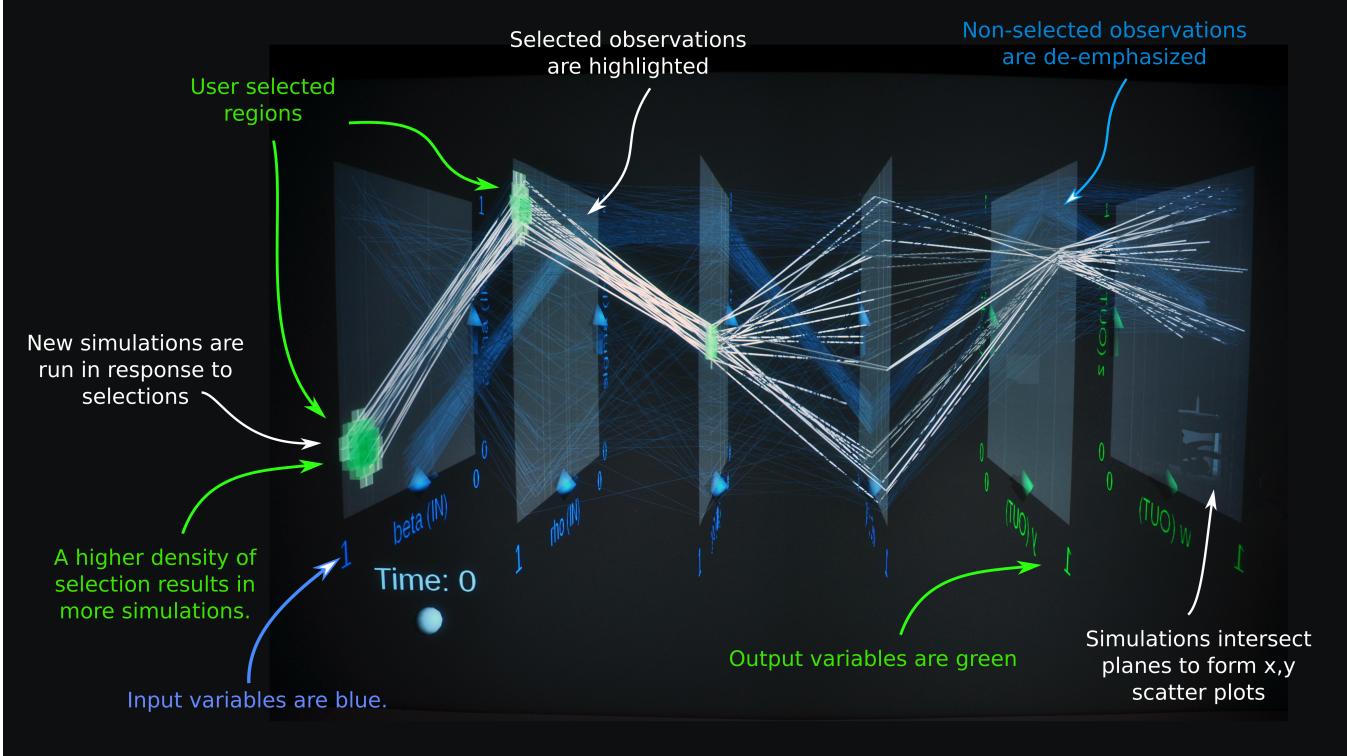


Figure 4: Photograph of the parallel planes in an immersive virtual environment with annotations describing the visualization and user interface.

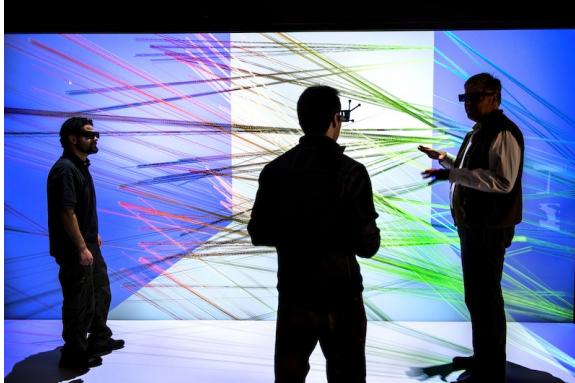


Figure 5: Appearance of parallel planes in an immersive virtual environment. (Photo by Dennis Schroeder / NREL)

immersive virtual environments, we have also implemented a lighter weight parallel planes visualizer that deploys on laptop computers, DLP projectors, Google Cardboard, and the like. Even without the degree of immersiveness provided by a human-scale environment, the parallel planes technique provides significant advantages in decluttering parallel-coordinates plots on these smaller displays.

## 2.2 Visualization-Driven Simulation

Our primary motivation for linking simulation and immersive visualization is to speed the analytic process of developing insights from simulations. By tightly coupling the scenario-definition, simulation, and exploratory-analysis processes (Figure 6), analysts can rapidly test and develop hypotheses regarding the relationships between simulation inputs and outputs and can design scenarios (sets of sim-

ulation inputs) illustrating those hypotheses and insights. Immersive analytics adds a powerful “tangible” dimension to this hypothesis testing and scenario design, and opens possibilities for collaborative exploration and design of simulations. The rapid, concrete feedback between scenario design decisions and seeing their implications, combined with the shared visual record of previous design decisions and their implications, enables developing nuanced intuitions regarding relationships between the simulation input space and the simulation output space.

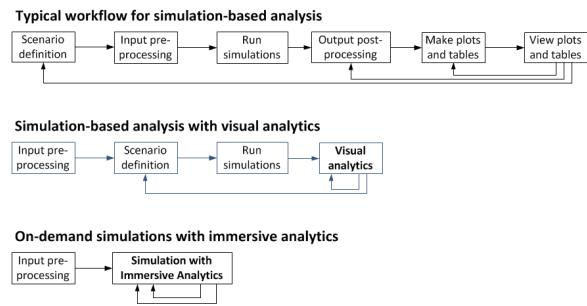


Figure 6: Driving simulation studies from immersive visualization streamlines the simulation-analysis workflow.

Within our parallel-planes visualization framework, three data spaces mediate the exchange of information between the immersive visualization and the simulation cluster (see Figure 7). The *model metadata* space provides information uniquely identifying the model to be simulated, the input variables and their ranges, and the output variables. The parallel-planes visualizer retrieves this information as it initializes its rendering. The *simulation requests* space stores requests from the visualizer for simulations to be run and it records the

completion status of those simulations. The *simulation results* space simply records the multivariate time series output by the simulations. In practice, the parallel-planes simulation framework implements these spaces as three MongoDB collections of BSON documents. This database-mediated communication and the employment of the tuplespace paradigm [7] for distributed, shared memory supports fully asynchronous exchange of information between the visualizer and the simulation cluster and leaves the two only very loosely coupled. Consequentially, this database is the only architectural point in common between the simulation and visualizer. In cases where a precomputed dataset or measured data is only available instead of a live simulation cluster, the visualizer can render those “simulation results” as if there were a live simulation backend. (Because of the static nature of the dataset and the lack of a live simulator, the requests from the visualizer would never be served by the simulator.) In cases of massive precomputed or measured datasets, the “simulation” activity can be replaced with statistical sampling of that primary dataset to create a subset requested for visualization.

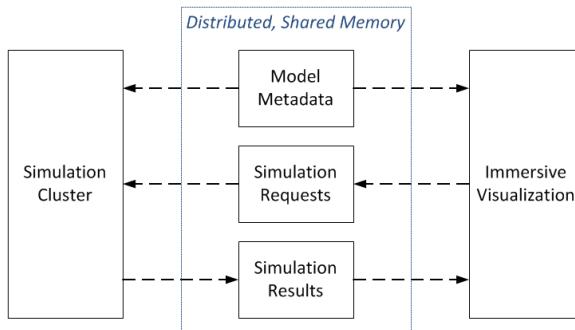


Figure 7: A simple database schema mediates asynchronous exchange of information between the visualizer and the simulators.

As a proof-of-principle for this simulation-visualization system, we have implemented a general purpose simulation for system dynamics (SD) models serialized in the standard XMILE format [3]. The system dynamics (SD) simulation methodology, which emphasizes the representation of time-varying phenomena that involve internal feedbacks and responses to external influences, is well suited for interactive visual exploration. The mathematical and computational straightforwardness of SD, which relies on the solution of systems of coupled ordinary differential equations (ODEs) with Neumann (second type) boundary conditions, also lends itself to rapid computation of the outputs for simulation requests from visualization systems. The simulation has been deployed on NREL’s Peregrine supercomputer [18], running in conjunction with the ESIF immersive environment also at NREL.

The simulation framework, whose components are outlined in Figure 8, consists of eight Haskell packages comprising about eight thousand lines of code that are deployable in heterogeneous computing environments of Linux, Windows, and Mac OS X machines or clusters where simulation load balancing relies on the aforementioned tuplespace mechanism. Communication between the computational units occurs using Erlang-style message passing. We used real-life SD models of varied complexity (dozens to thousands of equations) for verification and validation of the simulator’s correctness and accuracy. The resulting system scales such that hundreds of simulations can be simultaneously executed in response to users’ cues in the immersive visualization. In addition to being displayed in the visualization environments, the simulation results can be exported to statistical analysis packages or data files for further analysis.

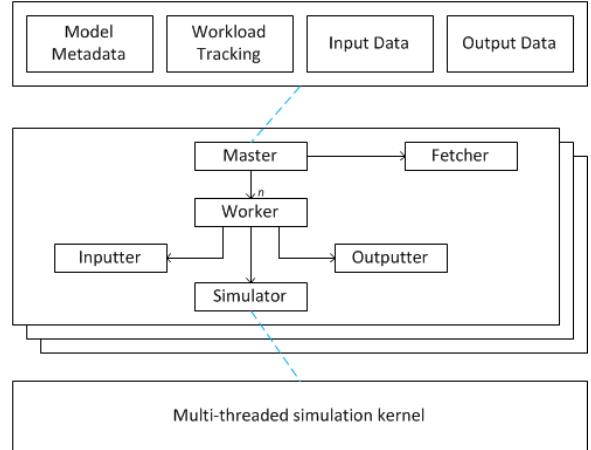


Figure 8: Sketch of major components in the architecture of the simulation framework.

### 3 EXAMPLE WORKFLOW FOR SIMULATION BY PARALLEL-PLANES VISUALIZATION

We now briefly demonstrate—to the extent possible using screenshots in print media—an example workflow for a parallel-planes visualization driving a simulation in response to a users’ brushing of the simulation input space. Figures 9 through 12 illustrate an example exploration for a very simple simulation with eight input variables (shown in the four leftmost planes) and two output variables (shown in the rightmost plane). Initially, the user selects a single region of the leftmost plane, which corresponds to a small rectangular range of the first two input variables. This triggers simulations to be run using that range for the first two input variables and where the other six input variables chosen uniformly at random: Figure 9 shows the visualization once those simulations have completed. At this point, a user would probably want to explore a broader region of the input space. They might brush, as shown in Figure 10, a large region of the second plane from the right, triggering simulations for that range of the third and fourth input variables. In order to explore the rich set of correlations in the simulation output, the user might brush a portion of the rightmost output plane, highlighting the observations that lie both in the previously selected inputs and the particular output regions that has been selected: Figure 11 shows those simulations highlighted and the other simulations displayed faintly. In typical workflows, a user would move between brushing input variables to generate more simulation output (Figure 12) and brushing output variables to identify patterns in the data. This can be thought of as an iterative process of generating and testing hypotheses about the correlation of the outputs among themselves and between the inputs and outputs. Once solid hypotheses have been generated, a user might discard the simulation output and run a rigorously designed statistical experiment and then explore that using the parallel planes visualization.

To date, in our trials of the simulation-linked parallel-planes technique, we have found the user experience to be best when simulation response times are less than thirty seconds, which is the approximate time that a user might easily remember that a particular new set of simulations were requested. Slower responses can be made acceptable to users, however, if visual cues such as markers registering the presence of simulations-in-progress annotate the input planes: essentially, these markers are promises to the user that their requests have been acknowledged and are being simulated. Depending on the immersive environment, tens of thousands of observations can be displayed effectively without causing severe over-plotting. “Zooming in” on fine structure in the results is particularly effective in

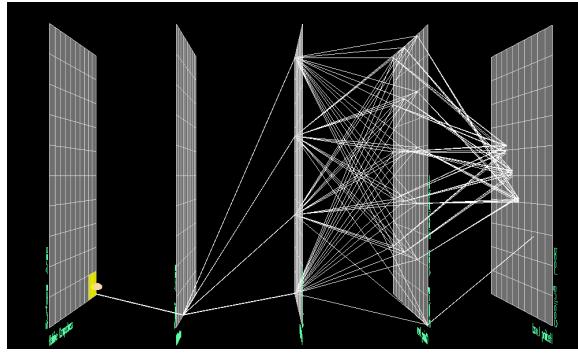


Figure 9: Initial exploration of a example simulation. Simulations have only been run for a small portion of the input space, shown by the selected rectangle in the leftmost plane. In this example, the four leftmost planes show input variables and the rightmost plane shows output variables.

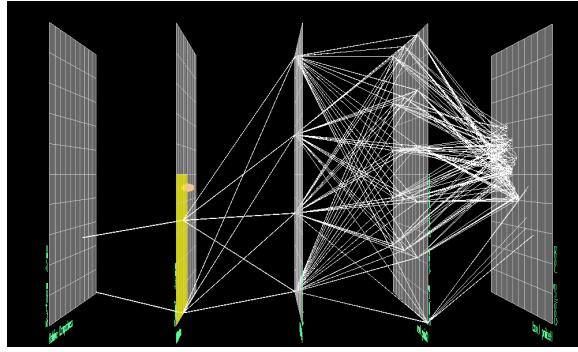


Figure 10: A continuation of the exploration in Figure 9, where additional simulations have been triggered by selecting a large region in the second-from-left plane.

large immersive environments because the user can walk right up to the data in question and brush it to make the observation visible throughout the display.

#### 4 SUMMARY AND PROSPECTS

This software presents a user-provided multivariate time-series dataset or simulation as an interactive three dimensional visualization analogous to the two-dimensional statistical graphic commonly called a “parallel coordinates” plot. Each variable of the input dataset is represented by the horizontal or vertical axis of a plane floating in virtual three dimensional space. Furthermore, each observation of the input dataset is represented by a polyline connecting the observation’s coordinate values on the planes. Users can select, or “brush”, regions of the planes and see highlighted the observations that intersect the brushed regions. This interactive visualization allows users to explore the correlation between variables in the observations and the distribution of observations among the variables. When used in conjunction with a simulation cluster, this software can provide a form of “simulation space exploration” similar “computational steering”, but in the context of ensembles of simulations rather than within a single simulation.

The technology described in this paper is under active use for simulation analysis and also under active development aimed to make it more effective. In particular, the use of a large, immersive space does not completely alleviate the problem of overlapping observations, it simply lets larger datasets be used before it becomes a problem. One approach is to allow the user to thin out the data in the display,

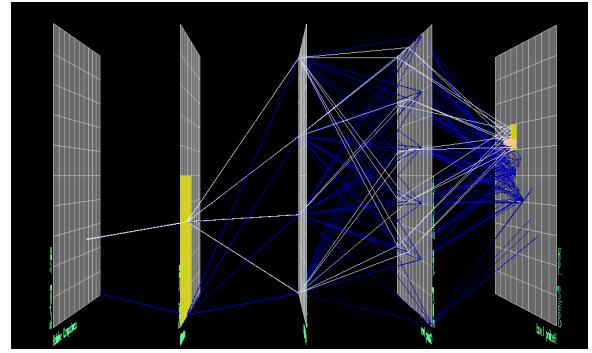


Figure 11: A continuation of the exploration in Figure 10, where the previously run simulations are filtered by a small range of output parameters brushed as the selected region in the rightmost plane.

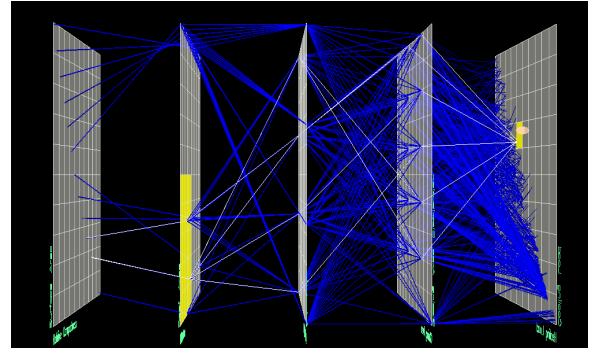


Figure 12: A continuation of the exploration in Figure 11, where many more simulations have been run, but this larger dataset is brushed as it was in Figure 11.

particularly where it is highly redundant. To further relieve the problem of overplotting or occlusion between the lines connecting points on the parallel planes, we propose to explore an adaptation of the edge bundling techniques introduced by Holten and Van Wijk [11] and applied to traditional parallel coordinates by Rübel et al. [22] and Palmas et al. [20]. We also have experimented with alternatives representations to parallel planes, one promising one among which is a “parallel forest” layout. The “trees” in the parallel forest are simply the individual parallel-coordinate axes; these are arranged in the immersive space like trees in a forest, but the locations are carefully chosen to conform to a two dimensional graph layout based on the statistical correlation between variables. Although the parallel forests have somewhat lower information density than the parallel planes, they may provide a more intuitive and inviting, but less abstract, view for users who are not already familiar with parallel coordinates. The axes in parallel forests can also be annotated with information such as histograms of marginal distributions.

Our ongoing work also links this simulation-visualization system with ancillary visualizations and controllers. The loose coupling of the visualization and the simulations allows for the inclusion of other visualization clients and other controllers. For instance, we have successfully linked web-browser-based two-dimensional parallel-coordinate controllers/visualizers to the same simulation clusters while the immersive visualization is active. This allows collaborations where some users are locally brushing and exploring the simulation space, but where the output from the simulations that their exploration launches are available to all other users, whether they are using the immersive parallel-planes, laptop-based visualizations, or other client software. In fact, tools for the design of

experiments, such as R, can also be linked into the collaborative space, so that structured batches of simulations can be requested according to statistically sound principles and then shared among the visualization clients. To further enhance group collaboration in a shared space such as immersive environments, we propose to investigate the use of computer-vision distinguishable physical objects (fiducials) to represent coordinate axes and brush regions on the axes to allow multiple analysts to work with the same visualization at the same time in a shared space. A final amenity for collaborative use of this approach is to allow users to “bookmark” and share the brushing configuration and/or “snapshot” and share the simulations created.

## ACKNOWLEDGEMENTS

This work was supported by the U.S. Department of Energy. The research was performed using computational resources sponsored by the Department of Energy’s Office of Energy Efficiency and Renewable Energy and located at the National Renewable Energy Laboratory. This work was initially supported by the Laboratory Directed Research and Development (LDRD) program at the National Renewable Energy Laboratory.

## REFERENCES

- [1] M. Ankerst, S. Berchtold, and D. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Information Visualization, 1998. Proceedings. IEEE Symposium on*, pages 52–60, 153, Oct 1998.
- [2] L. Arns, D. Cook, and C. Cruz-Neira. The benefits of statistical visualization in an immersive environment. In *Virtual Reality, 1999. Proceedings.*, IEEE, pages 88–95, Mar 1999.
- [3] R. L. Eberlein and K. J. Chichakly. Xmile: a new standard for system dynamics. *System Dynamics Review*, 29(3):188–195, 2013.
- [4] G. Falkman. Information visualisation in clinical odontology: Multidimensional analysis and interactive data exploration. *Artif. Intell. Med.*, 22(2):133–158, May 2001.
- [5] Y.-H. Fua, M. Ward, and E. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Visualization '99. Proceedings*, pages 43–508, Oct 1999.
- [6] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall. Open MPI: Goals, concept, and design of a next generation MPI implementation. In *Proceedings, 11th European PVM/MPI Users’ Group Meeting*, pages 97–104, Budapest, Hungary, September 2004.
- [7] D. Gelernter. Generative communication in linda. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 7(1):80–112, 1985.
- [8] K. Gruchalla, M. Dubin, J. Marbach, and E. Bradley. Immersive Examination of the Qualitative Structure of Biomolecules. In *International Workshop on Qualitative Reasoning about Physical Systems*, pages 36–41, 2008.
- [9] J. Heinrich and D. Weiskopf. Continuous parallel coordinates. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1531–1538, Nov 2009.
- [10] J. A. Henry and N. F. Polys. The effects of immersion and navigation on the acquisition of spatial knowledge of abstract data networks. *Procedia Computer Science*, 1(1):1737 – 1746, 2010. {ICCS} 2010.
- [11] D. Holten and J. J. Van Wijk. Force-directed edge bundling for graph visualization. In *Computer Graphics Forum*, volume 28, pages 983–990. Wiley Online Library, 2009.
- [12] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91.
- [13] A. Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2009.
- [14] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure within clustered parallel coordinates displays. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 125–132, Oct 2005.
- [15] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure in visualizations of dense 2d and 3d parallel coordinates. *Information Visualization*, 5(2):125–136, 2006.
- [16] J. J. Miller and E. J. Wegman. Computing and graphics in statistics. chapter Construction of Line Densities for Parallel Coordinate Plots, pages 107–123. Springer-Verlag New York, Inc., New York, NY, USA, 1991.
- [17] R. Moustafa and E. Wegman. Multivariate continuous data-parallel coordinates. In *Graphics of large datasets*, pages 143–155. Springer, 2006.
- [18] National Renewable Energy Laboratory. NREL High Performance Computing. <http://hpc.nrel.gov/users/systems/peregrine>, 2016. Accessed: 2016-02-03.
- [19] M. Novotny and H. Hauser. Outlier-preserving focus+context visualization in parallel coordinates. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):893–900, Sept 2006.
- [20] G. Palmas, M. Bachynskyi, A. Oulasvirta, H. P. Seidel, and T. Weinkauf. An edge-bundling layout for interactive parallel coordinates. In *Visualization Symposium (PacificVis), 2014 IEEE Pacific*, pages 57–64. IEEE, 2014.
- [21] D. Raja, D. A. Bowman, J. Lucas, and C. North. Exploring the benefits of immersion in abstract information visualization. In *In proceedings of Immersive Projection Technology Workshop*, 2004.
- [22] O. Rübel, G. H. Weber, S. V. Keränen, C. C. Fowlkes, C. L. L. Hendriks, L. Simirenko, N. Shah, M. B. Eisen, M. D. Biggin, H. Hagen, et al. Pointcloudxplore: Visual analysis of 3d gene expression data using physical views and parallel coordinates. In *EuroVis*, pages 203–210, 2006.
- [23] M. Streit, R. C. Ecker, K. Österreicher, G. E. Steiner, H. Bischof, C. Bangert, T. Kopp, and R. Rogojanu. 3d parallel coordinate systems a new data visualization method in the context of microscopy-based multicolor tissue cytometry. *Cytometry Part A*, 69(7):601–611, 2006.
- [24] R. M. Taylor, II, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser. Vrpn: A device-independent, network-transparent vr peripheral system. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST '01*, pages 55–61, New York, NY, USA, 2001. ACM.
- [25] R. Wegenkittl, H. Löffelmann, and E. Groller. Visualizing the behaviour of higher dimensional dynamical systems. In *Visualization '97., Proceedings*, pages 119–125, Oct 1997.