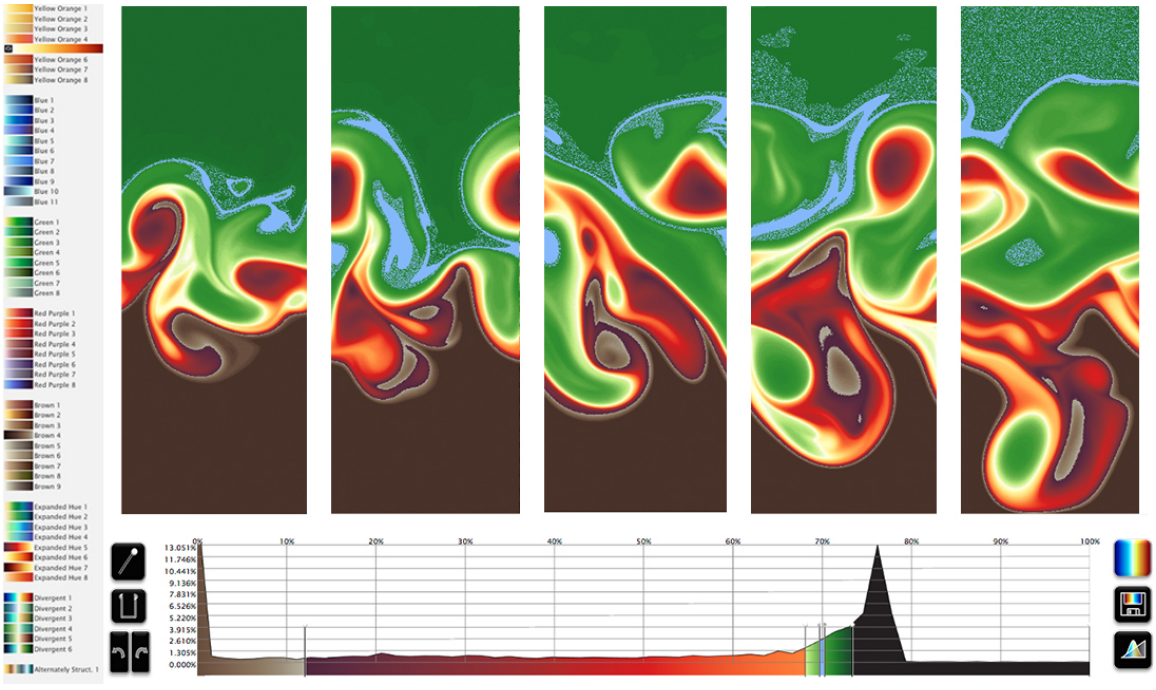
Optimizing the Energy Usage and Cognitive Value of Extreme-Scale Data-Analysis Approaches

Final Report

October 1, 2017

PI	James P. Ahrens	Los Alamos National Laboratory
Co-PI	David H. Rogers	Los Alamos National Laboratory
	Roxana Bujack	Los Alamos National Laboratory
	Anne S. Berres	Los Alamos National Laboratory
PI	Wu-chun Feng	Virginia Tech
	Vignesh Adhinarayanan	Virginia Tech
PI	Colin Ware	University of New Hampshire
PI	Francesca Samsel	University of Texas at Austin
PI	Gregory Abram	University of Texas at Austin
	Terece L. Turton	University of Texas at Austin



Contents

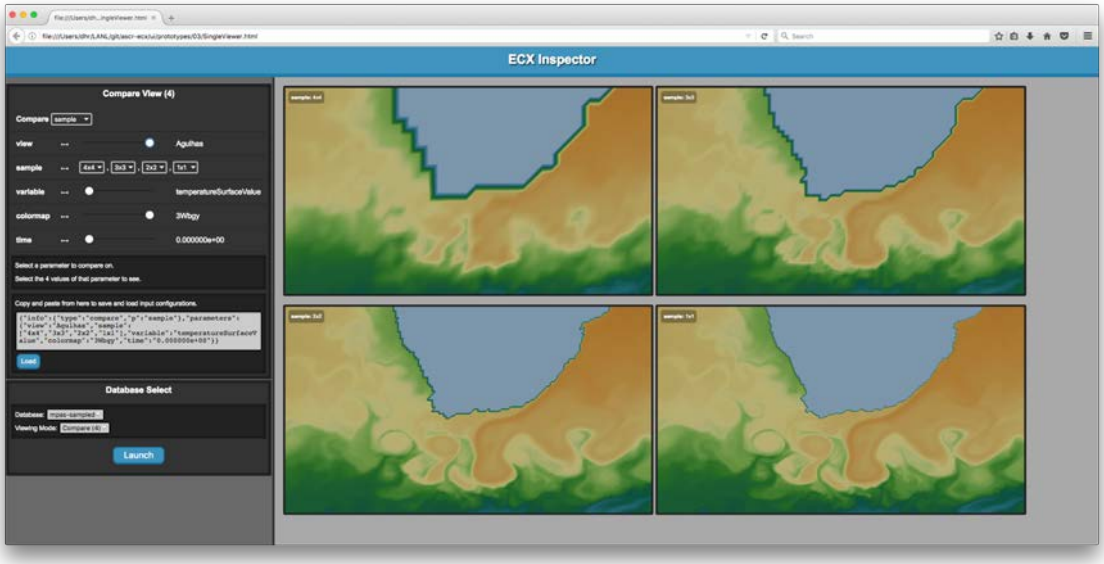


Figure 1: The ECX Inspector, showing a comparison of four different approaches for computing, sampling, and visualizing the results of an MPAS Ocean simulation. This tool embodies the experiments, cognitive and perceptual work, power and energy estimation, and color theory that spans the expertise domains of the project team. This report describes the findings, artifacts, publications, and research that embody an end-to-end approach and toolset for understanding and optimizing the energy usage and cognitive value of extreme-scale data-analysis approaches.

1 Executive Summary

1.1 Overview of Successful Completion

This report serves as a record that we have successfully completed work on the proposal *Optimizing the Energy Usage and Cognitive Value of Extreme Scale Analysis Approaches*, or, more succinctly, *ECX*. This includes successful completion of all three phases of the project, findings as summarized below, strong publications across the fields represented in the proposal, and a set of artifacts (code, websites, and other documentation) available to the public. In total, the results of this project represent significant findings in each domain, and a toolkit that will allow the community to engage with this work, advancing it and contributing to it in the years ahead. They also form the basis for a successfully funded follow-on project.

This executive summary presents findings, tools and publications resulting from the ECX project. The balance of the report then provides more detail on these results.

1.2 Overview of ECX Project Goals

Scientific discovery at the extreme scale is a unique technical challenge, requiring the reduction of massive amounts of data into compact analysis products that capture key scientific insights. This analysis process needs to occur under extreme-scale computational constraints including minimizing 1) data movement, 2) energy usage and 3) storage usage. Put simply, extreme-scale computing platforms are to achieve a three orders-of-magnitude increase in computational performance while consuming only two times the electrical power of current platforms. Data movement costs will dominate energy usage at this scale, so the HPC community expects extreme scale analysis algorithms will be utilized to reduce simulation results in-situ – that is, during the simulation run. This reduction will occur, broadly speaking, via some type of adaptive sampling, such as signal, statistical or feature-based sampling.

For this project, we considered the computational platform and the scientists using the platform as a system to be optimized. Our goal was to maximize scientific insight from sampled results while minimizing power. We proposed to investigate how changes in our sampling algorithms – necessary because of exascale power constraints – impacted the cognitive value of the resulting data. Our goal was to provide tools and methods for scientists and others to explore, understand, and utilize the high dimensional tradeoff space embodied in the inputs or controls to these workflows, e.g. data sampling approach, visualization pipeline, job architecture, and visualization approach. Our work progressed from basic initial experiments through more detailed explorations, and resulted in both research results and a set of tools that can be used to explore these areas further.

A key component of the success of this project was our approach of integrating the expertise of a diverse set of researchers. Experts from computing, perception, cognition, energy and power, and color theory came together to devise and execute experiments and develop tools to address the complex problem of maximizing scientific insight under constraints. The ECX team developed a close, effective working relationship across four disparately-located institutions, developed open source software together, collaborated on papers, and crowdsourced user testing methods. This team and its work is the basis for the successfully funded proposal *A Continuously-Running, Asynchronous, Sampling Engine for the Perceptual and Cognitively-driven Visual Analysis of Massive Scientific Data.* In particular, the ECX toolkit will serve as the initial development base for fundamental portions of that proposal.

1.3 ECX Findings

One question central to ECX is if are there ways of sampling, visualizing and compressing data at extreme scale that do not impact the cognitive value of the resulting artifacts. The short answer is that, yes, there are ways available across science domains that do this. The longer answer is that such benefits are specific to data type, machine architecture, and the question that the scientist is asking of the data. One benefit of this work is general methods for performing *what-if* evaluation of options in what is a very large sample space. Our tools provide methods for exploring this design trade-off space, across many domains.

Over the course of our team's research, several important findings have emerged:

- **In-situ analysis can reduce the cost of discovery**. In an experiment comparing in-situ versus post-processing pipelines, we found that using in-situ pipelines, we can process 100x as many time steps under the same storage budget and 3x as many time steps under the same energy budget compared to a post-processing pipeline. Thus, discovery can be much more cost effective. More detail can be found in Section 2.1.3.
- **We cannot expect in-situ analysis to lower the overall power consumption of an existing data center**. However, there are techniques that can make power-efficient HPC system designs possible by making HPC data centers with smaller storage subsystems possible. More detail can be found in Section 2.1.2.
- **Job layout can have a significant impact on energy and performance.** How the simulation process and visualization task is mapped onto the underlying resources can have a serious impact on application performance and energy consumption. An optimized layout can help obtain multi-fold improvements in performance and energy without affecting the quality of results. Details can be found in Section 2.1.4.
- **Raycasting rendering technique can deliver lower energy for large problem sizes at scale**. Raycasting ran slower than traditional geometry-based techniques for smaller problem sizes. For larger problems at scale it performed significantly better. Therefore we expect the technique to consume lower energy than traditional geometry-based techniques for exascale simulations. More detail can be found in Section 2.1.4.
- **Both 2D and 3D visualization benefit from motion**. Our work on cognition and visualization shows that motion parallax is more effective than stereo views for perception of

3D structures. For 2D vector fields, an animated approach is more effective than static representations. A summary of cognitive studies and publications from ECX can be found in Section 2.2.

- **Colormaps are a very low-cost way to resolve more detail in scientific data**. Color theory has historically been tapped to expand the power of color in visualizations. Our work goes further, engaging directly with design processes and an artist specializing in color relationships. Perceptual science informed the colormap development, validated results, expanding research into feature and pattern detection and optical sampling. Colormaps with greater resolving power were developed together with a flexible tool to develop data-specific colormaps. Colormap improvements can be deployed with no increase in energy or power budget, adding significantly to the perceptual and scientific value of the resulting artifacts. More detail on this work can be found in Sections 2.2, 2.3, and 2.4.
- **Colormap taxonomy can be significantly improved through discrimination of local versus global properties**. The taxonomy in the research field regarding colormapping is not consistent. Some terms, such as uniformity, order, discriminative power, are used to describe different concepts and likewise some concepts are denoted by multiple terms. This makes collaborations across fields very difficult. We found that a distinction between local (neighboring colors in a colormap) and global (colors from anywhere in the colormap) interpretations of the terms significantly improved the unambiguity of inter-disciplinary collaborations. More detail can be found in Section 2.3.7.
- **A new method is available to objectively evaluate colormaps in terms of their ability to resolve features in data.** This methodology provides a resolution function, showing where a color map has good resolving power and where it is poor. It is easy to implement and use and is based on theories of human pattern and color perception. Since the main purpose of colormapping data is the perception of features, this fills a major need. For more detail see Section 2.2.4.
- **There are clear trade-offs to be made between sampling and perceptual value**, and it is possible to save energy (through sampling) without sacrificing the value of the resulting artifacts. Our research shows that when one samples to reduce the size of the data, there are clear early gains in sampling that do not noticeably degrade the resulting artifacts. Thus, it's worth it to spend energy sampling before this threshold. More detail on this work can be found in Section 2.1.6.
- **Very large compressions are possible with no perceptual loss for vector field data.** At least a factor of 20 times compression is possible for the MPAS vector field data with the results being visually indistinguishable from uncompressed data. This means that for visualization purposes the volume of data that must be saved can be considerably reduced. Smaller data volumes can result in large cognitive gains, because of shortened access times and the potential for more interactive exploration. Alternatively, more time steps can be saved making it possible to see phenomena at much greater temporal granularity.
- **We can successfully leverage crowdsourced participant pools for perceptual evaluation in visualization,** including evaluations involving color (Section 2.4.1) and can benefit from an automated approach to experimental implementation. We developed a toolkit (ETK) for running experiments in the cloud, which facilitates these experiments, and significantly reduces the time it takes to run user studies for a class of problems. For more detail see Section 3.1.

1.4 ECX Toolkit

As outlined in the proposal, we have created a set of tools that help with understanding the trade-off space encompassed by the project. We designed these tools to help with *what-if* analysis of questions central to this proposal. This set of tools is available online, and can serve as either tools for exploration, evaluation, or examples for further development. Several of the tools will be used in the

follow-on project. These tools can be used individually, or as a set. The tools are fully described in later sections, so we summarize them here.

- **ECX Explorer** The ECX Explorer is a UI that promotes exploration of the large design tradeoff space described above. Building on the previously developed Cinema specifications and viewers, we have developed a set of tools that a user can use to explore this high dimensional tradeoff space. This tool is a model for general exploration of these tradeoff spaces, and can be used by a scientist to explore tradeoffs before running large scale simulations. The tool provides a method for extracting the values of the inputs, so that they can be used by a job management pipeline. See Section 3.4.
- **Experimental Test Harness (ETH)** The Exploration Test Harness (ETH) is a lightweight, open-source testing harness that promotes exploration of a variety of analysis and visualization pipelines in many different operations, work distributions, and mappings onto hardware. See Section 3.2.
- **ECX Estimator** The ECX estimator is a command-line tool implemented in C++ that allows early-stage exploration of visualization parameters. Using this tool, a user can estimate the power, energy, storage, and performance for different visualization pipelines and different configurations of each pipeline (e.g., sampling rates, rendering techniques, job layouts etc.). Precise evaluations of interesting configurations can then be performed with the other tools such as ETH. See Section 3.2.
- **Evaluation Toolkit (ETK)** The Evaluation Toolkit, (ETK) is a set of JavaScript/HTML/CSS modules that automate the presentation of image-based perceptual experiments. ETK streamlines the implementation of visualization user evaluations, and it can be used on any perceptual question that can be posed as a set of images. This is a very powerful approach to user testing, and has provided a huge gain in the speed and efficiency of our user studies. This tool has allowed us to complete a very large number of user studies, as evidenced by the publications that resulted, and fundamentally changed how we can incorporate user feedback and testing in the design and visualization process. This toolkit is supported on the website `http://www.etklab.org`. See Section 3.1.
- **ColorMoves** ColorMoves is a flexible browser-based tool that promotes an interactive approach to exploring color for specific datasets. It is easy to use, and is targeted at scientists themselves. It encodes the color knowledge of this team in a set of colormaps, and an easy-to-use drag and drop interface that provides a way of mapping color onto specific value ranges in a visualization. This and other tools can be found on `http://www.sciviscolor.org`. See Section 3.5.1.
- **Colormeasures** Colormeasures is an online tool that mathematically evaluates colormap properties in multiple uniform color spaces, allowing a user to select a colormap based on optimal characteristics such as uniformity or discriminative power. This tool is available on `http://www.colormeasures.org`. See Section 3.6.

1.5 Additional ECX Websites

In addition to the tool-specific websites noted above, there are a collection of additional sites that provide portals to other aspects of the ECX project. These are detailed below.

- **`www.ecxproject.org`** This is the main website for the results of this project, and includes information about the team, publications, and links to websites noted below.
- **`www.github.com/ascr-ecx`** This is the main location for code associated with this project, and contains open source releases of ETK, ETH, ECX estimator, and EXC inspector.
- **`www.sciviscolor.org`** This is a website for visualization practitioners, scientists and the public providing resources to maximize color's potential to convey scientific data. We provide extensive sets of color scales and colormaps accompanied by recommended applications - specific colormaps depending on the data and or task. The colormaps are accompanied by color strategies, applied

examples and recommendations for addressing problems common across scientific disciplines. The site includes the ColorMoves tool (see Section 3.5.1), which is a means of interactively constructing custom colormaps that align with the statistical distribution of the data, tasks, domain conventions and goals of the scientist. In total, this site provides scientists with a one-stop-shop to quickly and easily explore and communicate their data via color.

- **`www.discoveryjam.com`** This website is the portal for DiscoveryJam, a new model for creating interdisciplinary teams around science problems and data. We've created the Discovery Jam as an exciting, vibrant, inspirational mechanism for creating interdisciplinary teams to address specific challenge problems. A Discovery Jam demonstrates – through example – a culture of collaboration and a method of training a new cohort of scientists, technologists, and artists to work together toward discovery across disciplines. Jams have been held at VisWeek 2016 and VisWeek 2017, with more planned as we refine the curriculum and model for a Jam.

1.6 ECX Publications

ECX publications span the range of domains represented on the ECX team. The work resulted in twenty-four publications across the domains.

- Abram, G., Adhinarayanan, V., Feng, W.-c., Rogers, D., Ahrens, J., and Wilson, L. ETH: A framework for the design-space exploration of extreme-scale scientific visualizations. Tech. rep., Virginia Tech University and Los Alamos National Laboratory, 2017. LA-UR-17-26715

- Adhinarayanan, V., Feng, W.-c., Woodring, J., Rogers, D., and Ahrens, J. On the greenness of in-situ and post-processing visualization pipelines. In *Proceedings of the 2015 IEEE Parallel and Distributed Processing Symposium Workshop (IPDPSW)* (2015), IEEE, pp. 880–887. LA-UR-15-21414

- Adhinarayanan, V. Performance, power, and energy of in-situ and post-processing visualization. Tech. rep., Los Alamos National Laboratory, 2015. LA-UR-15-27749

- Adhinarayanan, V., Pakin, S., Rogers, D., chun Feng, W., and Ahrens, J. Performance, power, and energy of in-situ and post-processing visualization: A case study in climate simulation. Best Research Poster Finalist

- Adhinarayanan, V., Feng, W.-c., Rogers, D., Ahrens, J., and Pakin, S. Characterizing and modeling power and energy for extreme-scale in-situ visualization. In *Parallel and Distributed Processing Symposium (IPDPS), 2017 IEEE International* (2017), IEEE, pp. 978–987. LA-UR-16-22435

- Ashton, Z. C., Wendelberger, J. R., Ticknor, L. O., Turton, T. L., and Samsel, F. Analyzing task-based user study data to determine colormap efficiency. Tech. rep., Los Alamos National Laboratory, 2015. LA-UR-15-25715

- Berres, A. S., Turton, T. L., Petersen, M., Rogers, D. H., and Ahrens, J. P. Video Compression for Ocean Simulation Image Databases. In *Workshop on Visualisation in Environmental Sciences (EnvirVis)* (2017), K. Rink, A. Middel, D. Zeckzer, and R. Bujack, Eds., The Eurographics Association. LA-UR-17-21590

- Berres, A., Turton, T. L., Rogers, D., and Ahrens, J. VideoDB: Reducing large image databases through video encoding and video compression. Tech. rep., Los Alamos National Laboratory, 2016. LA-UR-16-24358

- Berres, A., Adhinarayanan, V., Turton, T. L., Feng, W., and Rogers, D. H. A pipeline for large data processing using regular sampling for unstructured grids. Tech. rep., Los Alamos National Laboratory, 2016. LA-UR-16-24358

- Bujack, R., Turton, T. L., Samsel, F., Ware, C., Rogers, D. H., and Ahrens, J. The good, the bad, and the ugly: A theoretical framework for the assessment of continuous colormaps. In *To appear at VIS2017, IEEE Conference on Scientific Visualization* (2017), IEEE

- Lange, D., Samsel, F., Karamouzas, I., Guy, S. J., Dockter, R., Kowalewski, T., and Keefe, D. F. Trajectory Mapper: Interactive Widgets and Artist-Designed Encodings for Visualizing Multivariate Trajectory Data. In *EuroVis 2017 - Short Papers* (2017), B. Kozlikova, T. Schreck, and T. Wischgoll, Eds., The Eurographics Association

- Samsel, F., Petersen, M., Abram, G., Turton, T. L., Rogers, D., and Ahrens, J. Visualization of ocean currents and eddies in a high-resolution ocean model. Winner, Best Scientific Visualization & Data Analytics Showcase

- Patchett, J. M., Samsel, F., Tsai, K. C., Gisler, G. R., Rogers, D., Abram, G., and Turton, T. L. Visualization and analysis of threats from asteroid ocean impacts. LA-UR-17-20419

- Samsel, F., Petersen, M., Geld, T., Abram, G., Wendelberger, J., and Ahrens, J. Colormaps that improve perception of high-resolution ocean data. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (2015), CHI EA '15, pp. 703–710. LA-UR-15-20105

- Samsel, F., Turton, T. L., Wolfram, P., and Bujack, R. Intuitive Colormaps for Environmental Visualization. In *Workshop on Visualisation in Environmental Sciences (EnvirVis)* (2017), K. Rink, A. Middel, D. Zeckzer, and R. Bujack, Eds., The Eurographics Association

- Samsel, F., Klaassen, S., Petersen, M., Turton, T. L., Abram, G., Rogers, D. H., and Ahrens, J. Interactive colormapping: Enabling multiple data range and detailed views of ocean salinity. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2016), CHI EA '16, ACM, pp. 700–709

- Samsel, F., Patchett, J. M., Rogers, D. H., and Tsai, K. Employing color theory to visualize volume-rendered multivariate ensembles of asteroid impact simulations. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2017), CHI EA '17, ACM, pp. 1126–1134

- Turton, T. L., Ware, C., Samsel, F., and Rogers, D. H. A crowdsourced approach to colormap assessment. In *EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization (EuroRV3)* (2017), K. Lawonn, N. Smit, and D. Cunningham, Eds., The Eurographics Association

- Turton, T. L., Berres, A. S., Rogers, D. H., and Ahrens, J. ETK: An Evaluation Toolkit for Visualization User Studies. In *EuroVis 2017 - Short Papers* (2017), B. Kozlikova, T. Schreck, and T. Wischgoll, Eds., The Eurographics Association

- Ware, C., Bolan, D., Miller, R., Rogers, D. H., and Ahrens, J. P. Animated versus static views of steady flow patterns. In *Proceedings of the ACM Symposium on Applied Perception* (New York, NY, USA, 2016), SAP '16, ACM, pp. 77–84

- Ware, C., Rogers, D., Petersen, M., Ahrens, J., and Aygar, E. Optimizing for visual cognition in high performance scientific computing. *Electronic Imaging 2016*, 16 (2016), 1–9

- Ware, C., Turton, T. L., Bujack, R., Samsel, F., Shrivastava, P., and Rogers, D. H. Measuring and modeling the feature discrimination threshold functions of colormaps, 2017. In submission to *Transactions on Visualization and Computer Graphics*

- Ware, C., Turton, T. L., Samsel, F., Bujack, R., and Rogers, D. H. Evaluating the perceptual uniformity of color sequences for feature discrimination. In *EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization (EuroRV3)* (2017), K. Lawonn, N. Smit, and D. Cunningham, Eds., The Eurographics Association

- Aygar, E., Ware, C., and Rogers, D. H. The contribution of stereoscopic and motion depth cues to the perception of structures in 3d point clouds, 2017. In press *Transactions on Applied Perception*

1.7 Conclusion

In conclusion, the findings, research, and artifacts shown above demonstrate that we have successfully completed the ECX project. The team appreciates the opportunity to have worked on this research, and anticipates advancing this work in the follow-on proposal, *A Continuously-Running, Asynchronous, Sampling Engine for the Perceptual and Cognitively-driven Visual Analysis of Massive Scientific Data*.

2 ECX Research Details

In this section, we provide detail on the major research completed under ECX. The work is highly interrelated, but for clarity we have divided the details into sections addressing the areas of expertise represented by the ECX team.

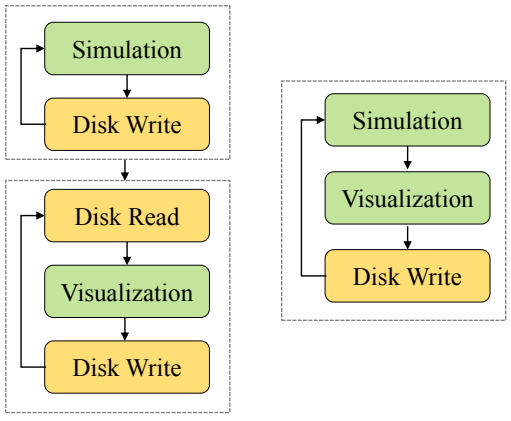
2.1 Power, Energy, and Workflow

Power and energy have emerged as first-order design constraints in high-performance computing systems. An exascale supercomputer built with today's (2016) technology would consume about 350 megawatts (MW) which is sufficient to power 250,000 houses. The energy cost of operating such a machine would far exceed its acquisition cost. For reference, about 25% of the acquisition cost is spent as energy costs in *today's* high-end supercomputers [15]. In this project, we investigate several techniques to reduce power and energy consumption to lower the cost of scientific discovery. We focus on one important class of applications—scientific visualization. Such focused investigation allows us to explore application-specific techniques which can provide significant benefits over generic techniques. On the other hand, the application-specific techniques explored in the project can alter the quality of results which is studied in detail in subsequent sections.

2.1.1 On the Greenness of In-situ and Post-Processing Visualization Pipelines

The following two observations motivated this study:

- The energy cost of off-node data movement is higher than any other operation in a supercomputing system. For instance, fetching data from the disk is 10 times more expensive in terms of energy than performing computation [31].
- Most data movement in scientific visualization can be avoided by running the visualization process alongside the simulation process (i.e., via in-situ visualization) as shown in Fig. 2.



(a) Post-processing **(b)** In-situ

Figure 2: Illustrative example of post-processing (traditional) pipeline and in-situ pipeline. In the post-processing pipeline, raw data is written to the storage subsystem at the end of each iteration of a simulation. After the simulation is complete, the data is read back and rendered. The in-situ pipeline creates an image at the end of each iteration and writes only the image, which is significantly smaller, to the disk.

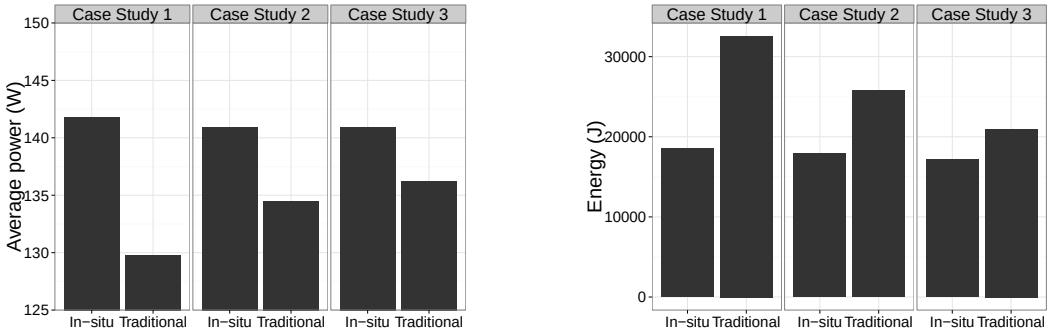
In this publication [4], we study the potential benefits of in-situ visualization by running disk I/O microbenchmarks and a proxy heat-transfer application on an instrumented server machine. Table 1 shows the total power consumed by the server and the dynamic power when running these microbenchmarks. The dynamic power is the portion that can be directly or indirectly attributed to

disk reads and writes. From the table, we can see that less than 10% of the idle system power goes towards data movement to and from the disk, which suggests only a fractional improvement from in-situ visualization.

Metric/Workload	Read μbenchmark	Write μbenchmark
Avg. Power (Total)	115.1	114.8
Avg. Power (Dynamic)	10.3	10.0

Table 1: Power consumed during disk read and write

Next, we used a proxy heat-transfer application to understand the extent of power and energy savings from in-situ techniques. The average power and energy for in-situ and post-processing pipelines for three different problem sizes of this application is shown in Fig. 3. We observe that the maximum energy saved by adopting in-situ visualization is as high as 43% for this application. The corresponding reduction in peak power, however, is less than 1%. This is because most of the energy savings occur because the in-situ version runs significantly faster rather than by reducing the power consumed by the I/O components in the system.



(a) Average power

(b) Energy

Figure 3: Average power and energy consumption of post-processing and in-situ pipelines for three different configurations of the proxy heat-transfer application

2.1.2 Performance, Power, and Energy of In-situ and Post-Processing Visualization: A Case Study in Climate Simulation

This poster and extended abstract [5] builds on the findings from our previous workshop paper [4]. We make four main contributions in this publication: (i) We characterize the power consumed by the disk subsystem in detail using a more sophisticated microbenchmark, namely *fio* (ii) We model the power consumed by the disk sub-system (iii) We confirm our previous findings using a *real-world* application (climate simulation) (iv) We identify indirect ways to save power from in-situ techniques.

Fig. 4 shows the disk power plotted against its bandwidth. We can see that the disk power swings by only 3 W when it is at no load vs. full load. The node power is typically around 300 W when it is at full load. This means, the portion of the system affected by in-situ techniques consume only a small amount of power and therefore, large power savings from such techniques should not be expected. However, we continue to see significant reduction in energy as the in-situ pipeline runs much faster.

This study also points out another source of power savings from in-situ techniques. Our results showed two orders of magnitude less storage required for in-situ visualization. This means fewer storage nodes can be used in the HPC data center and thereby reduce power consumption. Our expectation is that this saved power will be reinvested in compute nodes. Our publication shows that realistically, we can expect up to 10% higher budget for compute nodes, which in turn could

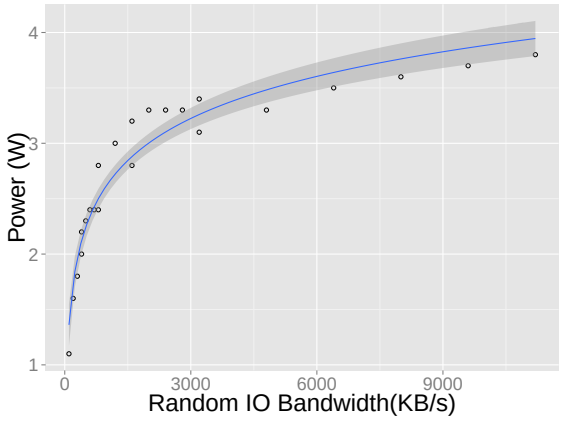


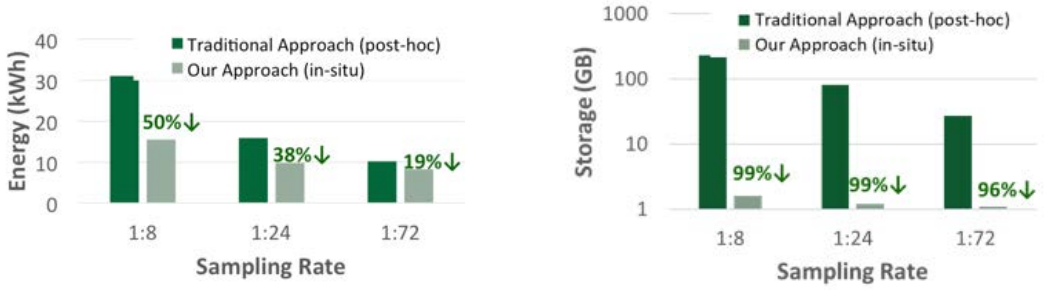
Figure 4: Disk Power increases logarithmically with bandwidth. This means the power consumed by the disk at full load is only slightly higher than the power consumed at no load.

increase the overall performance of our application by 6.3%. Additional details and assumptions are presented in our poster [5].

2.1.3 Characterizing and Modeling Energy for Extreme-Scale In-situ Visualization

In this paper, we expanded our study by performing our experiments at the scale of a supercomputer. In addition, we also studied the impact of temporal sampling on power and energy and constructed some highly accurate energy models which forms the basis of our ECX model.

Fig. 5a shows up to 50% less energy for in-situ pipeline compared to a post-processing pipeline. Using temporal sampling technique, we can save an additional 50% energy by reducing the sampling rate by a factor of nine as shown in the same figure. We can also reduce storage requirements by at least two orders of magnitude as shown in Fig. 5b. These results mean as much as 100x as much time steps can be processed under the same storage budget and 3x as much time steps can be processed under the same energy budget. Thus, scientific discovery can be much more cost-effective using in-situ techniques.



(a) Energy **(b)** Storage

Figure 5: Energy consumed and storage requirements for in-situ and post-hoc visualization at different sampling rates

We also developed an application-aware, architecture-specific model for performance, energy, and storage in this paper. The details of the model are presented in our paper. While the model presented in this paper is application- and architecture-specific, it has been generalized in our ECX estimator tool (Section 3.3) for use in other applications. For more details, see our paper [3] and Section 3.3.

2.1.4 The Experimental Test Harness: A Framework for the Design-Space Exploration of Extreme-Scale Scientific Visualization

This work [1] came out of the realization that performing an end-to-end study (where we integrate the visualization with the application) is time consuming. Considering the range of parameters that we proposed to look at, we developed the ETH framework to enable faster exploration of scientific visualization pipelines, and to provide a tool that the community could use to conduct similar experiments. Using this framework, we studied several other parameters, namely, rendering algorithm, spatial sampling, and workload-architecture mapping techniques. This study uncovered several promising paths to improve the greenness of scientific visualization. The specific contributions of ETH are as follows:

- **A first-of-its-kind toolkit for answering "what if" questions**. This framework will facilitate a deep understanding about the trade-offs among different operations, sampling, visualization pipelines and job layouts.
- **A toolkit that runs on data rather than the coupling to a science code**. While other toolkits can be used to operate directly on the data, the design of ETH ensures that we can answer questions pertaining to domains where we don't have real sim-viz code coupling. This is an important aspect to flexibility, as it means experiments can be run without investment in coupling the in-situ pipeline with a specific code, or compiling the coupled code on new machines. This toolkit adds to the community's ability to run these types of experiments.
- **Built-in exploration of multiple rendering approaches**. In particular, the toolkit includes a raycasting approach that operates on the raw data and a geometry-based approach that performs traditional triangle-based operations. The raycasting approach holds great promise for data at scale, and current work by vendors makes this a viable alternative.
- **Results from experiments on representative data types (grid and points) across different configurations**. In particular, we have conducted experiments on two classes of data — points and grids — that represent the types of data that extreme-scale applications create. Our experiences with both traditional evaluation and ETH showed that design space exploration with ETH was an order of magnitude faster due to the simplified process.

Our findings showed that while the various rendering algorithms consumed similar amounts of power, the raycasting technique showed significantly better energy characteristics for larger problems. Our findings also revealed the importance of job layout as it turned out to be the single biggest factor affecting energy consumption for scientific visualization. Implementation details can be found in Section 3.2 and additional results can be found in our tech report [1].

2.1.5 Power-efficient Scientific Visualization with Integrated GPUs and DVFS

This research goes beyond the work proposed as a part of this project and looks at alternate paths to improve the greenness of scientific visualization. The task of choosing between a CPU and a GPU for running scientific visualization workloads may seem straightforward, as GPUs are specifically designed for graphics processing. However, if cost is an issue, datacenters may restrict (or reduce) the availability of discrete GPUs for application scientists. In the era of post-processing visualization, where the simulation runs on a cluster and the visualization takes place on a different (and separate) node, this may not be an issue. In this case, only the visualization node needed graphics processing capabilities, thus using a more expensive discrete GPU for it is a viable and cost-effective option. For in-situ visualization, the visualization task runs on the same nodes as the simulation. Thus, graphics processing capabilities must be provided on *each* node. In such a situation, the system designer and the users have three options: (i) use CPUs that are optimized for graphics processing for both simulation and visualization, (ii) deploy an expensive discrete GPU into *each* node, or (iii) use smaller integrated GPUs that are soldered onto the motherboard. In this paper, we explore the first and third options.

The motherboard-integrated GPU has a few drawbacks. It only has a few MB of RAM of its own and relies primarily upon system memory instead. Many of these integrated GPUs, including the

Matrox G200 GPU used in our study, do not expose dynamic voltage and frequency scaling (DVFS) settings to the user, and therefore, their operating frequency cannot be tuned to optimize for power and energy. However, they can provide a few benefits. First, they operate at a fraction of the power envelope that other devices do. Second, they are significantly cheaper. While the power savings can be significant for this integrated CPU device, their impact on other greenness metrics such as energy or the energy-delay product (EDP) is not clearly understood. Therefore, we systematically study the *greenness* of the integrated GPU for scientific visualization tasks and compare it against CPUs operating at different frequencies. Ultimately, this work benefits the visualization scientists in making greener and less expensive simulations in the context of high-performance computing (HPC). Our major contributions and findings in this paper can be summarized as follows:

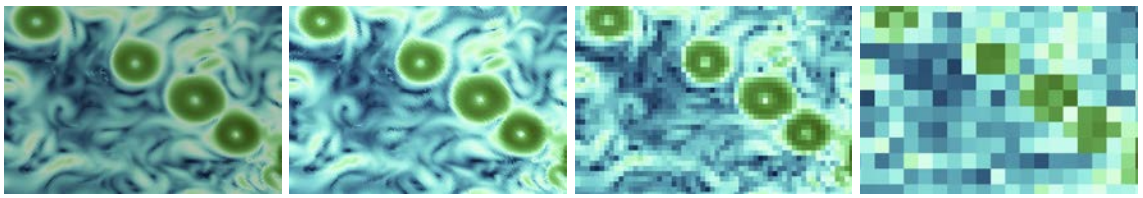
- We performed experiments and provide data to facilitate a direct comparison between the CPU and integrated GPU with respect to performance, power, energy, and EDP for multiple visualization filters.
- Our findings highlighted the scenarios in which the integrated GPU performs better than the CPU and vice versa. For larger data sets and complex visualization operations the integrated GPU delivers significant benefits (up to a 26% better EDP) whereas for smaller data sets and relatively simpler visualization tasks the CPU is a better choice.
- We run the visualization filters on the CPU at different frequencies to help the CPU deliver better characteristics. We identify cases where the DVFS technique helped and where it turned out to be counter-productive.

Additional details about this work can be found in our paper [12].

2.1.6 Sampling, Energy, and Perception

An early pilot study [9] implemented a pipeline to study the trade-offs between data sampling, energy, and perceptual value. The data used was the multi-resolution MPAS-Ocean climate model. The kinetic energy variable allows the scientist to easily see the mesoscale eddies throughout the global oceans.

A regular sampling approach was chosen for simplicity in this initial test of the pipeline process. Since the multi-resolution MPAS approach results in an irregular grid in a latitude/longitude projection, the original simulation was first oversampled to create a regular 15×15 km grid. Subsampling was then accomplished via a standard ParaView filter to increase the grid cell size, lowering the resolution in steps. The original grid resolution corresponds to 1×1 units and the overall resolution can be continuously varied. Sampling was varied from 1×1 to 9×9.



(a) Unstructured grid. **(b)** Supersampled grid. **(c)** Subsampled 3x3 grid. **(d)** Subsampled 9x9 grid.

Figure 6: Comparison of different sampling resolutions, from the original hex-dominant grid, supersampled regular grid, and two different sampling densities.

The post-sampling cognitive impact was assessed via a crowdsourced user evaluation study using a 2-alternative forced choice (2AFC) approach to determine the discrimination threshold. Sampled images from three locations around the globe with varying eddy size were compared to the supersampled regular grid. Participants were asked to choose the image which showed the eddies more clearly. A pilot study with 35 participants recruited from Amazon Mechanical Turk (Mturk) indicated that the 3×3 level of sampling was already beyond the discrimination threshold. Using finer steps from 1×1 to 3.3×3.3 units and 200 participants resulted in the 2AFC ramps of Figure 7. The

perceptual discrimination threshold, defined as the point where 75% of participants can distinguish the subsampled data from the 1×1 baseline, varies as a function of the regions, i.e., as a function of the size of the eddies. A linear fit to the overall average indicates that the overall discrimination threshold is around a sampling level of 2.0×2.0.

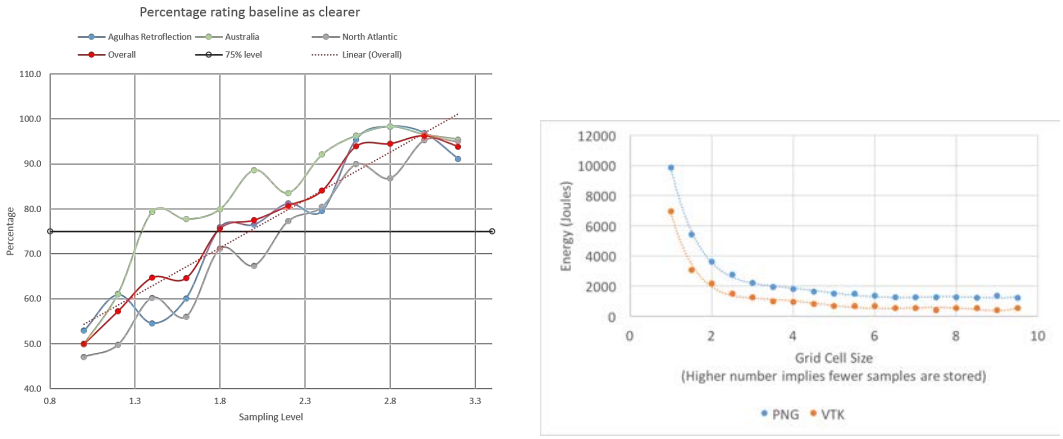


Figure 7: Left: Results of a perceptual evaluation study comparing images with different sampling grid sizes for different ocean regions. In this study, participants were forced to choose which of two images showed the eddies *more clearly*. Right: Comparison between power for PNG image output and power for VTI (slices) output.

We are interested in the energy consumption for the sampling itself in addition to saving as VTK image data (i.e., `.vti` format) or as colormapped images (i.e., `.png` format). Energy measurement used a WattsUp Pro power meter with one-second intervals between measurements. ParaView's `pvpython` library was used to read, sample, and write all data. ParaView's *slice* rendering was chosen as it avoids an unnecessary interpolation inherent in the *surface* rendering that introduces an unintentional smearing of the image. Slice shows the raw data as needed for this study. Hence, we use *slice* rendering for our *png* output. However, writing out raw data as VTK image is much more energy-efficient than rendering a colormapped PNG, as demonstrated in Figure 7.

From these two graphs, we can see that the majority of the energy savings occurs before the perceptual discrimination threshold is reached. Thus, this is the ideal tradeoff point for such a reduction.

2.1.7 Video Compression for Ocean Simulation Image Databases

Increasing supercomputer speeds have led to higher spatial resolution in global climate models. As model resolutions move to higher resolutions, scientists need innovative solutions to minimize storage requirements. Berres and Turton collaborated to study the impact of video compression on Cinema image databases for the MPAS-Ocean simulation. Image databases contain large numbers of visualizations created in-situ. Post-processing, users can interactively explore image databases as if interacting with the data directly. Using image databases, the data volume can be reduced from an order of petabytes to an order of terabytes. Video compression further decreases storage needs through efficient compression of sequences of similar images. The changing features of interest to the climate scientist are the ocean currents and eddies while the land remains a static background in ocean simulation visualizations.

This effort looked at a workflow, visually represented by Figure 8, through which image databases can be compressed by 2-4 orders of magnitude with varying perceptual loss, and which allows for efficient arbitrary access to individual images. We compared databases from all steps of the workflow and across the entire range of encoding and compression parameters. To evaluate the efficacy of our method, we considered compression and access speeds, compressed size, as well as image quality metrics, perceptual evaluation, and expert feedback.

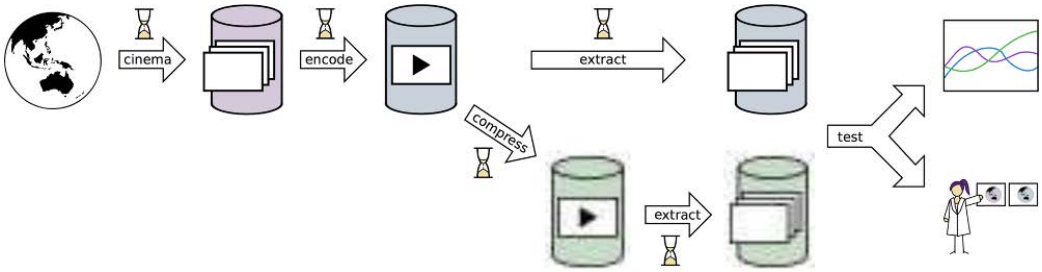


Figure 8: Video compression workflow: From the left, an ocean simulation generates a Cinema database which is then encoded into a video. From there, images can either be directly extracted (top) with no compression or (bottom) compressed and then extracted. The resulting images can be assessed through image quality metrics or through perceptual evaluation.

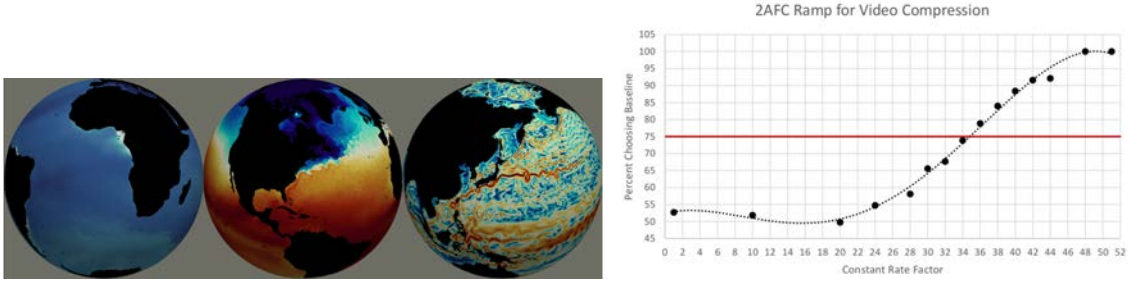


Figure 9: Left: Three of the regions used for analysis, from left to right: Agulhas Retroflection, Gulf Stream, and Kuroshio Current. The variables are (left to right) salinity (SA), temperature (TE), and kinetic energy (KE). Right: The 2AFC ramp for video compression as determined from a crowdsourced participant pool. Note that the perceptual discrimination threshold found from the Mturkers (CRF=34) is in agreement with expert opinion as the point where the images become marginally acceptable.

A pilot simulation run used kinetic energy as the variable of interest over three time steps. This run was used to study optimal traversal order and used in the perceptual evaluations. The full run had 175 time steps and three variables: kinetic energy, temperature and salinity (in different colormaps). Multiple regions around the globe representing a variety of eddy sizes and areas of interest were used. Examples of those regions can be seen in Figure 9. Two databases were generated from the full simulation run, one with high spatial coverage using 1250 Cinema cameras positioned around the globe, the other with only 200 cameras.

The video encoding step is quasi-lossless, providing some compression through similarities between frames. Additional compression can be achieved at the expense of lower image quality through quantization (FFMPEG compression). Quantization is applied through setting a *Constant Rate Factor (CRF)* between 1 (lossless) and 51 (lossiest). After the video encoding, we compressed video databases with the CRF in the range $[1, 51]$. Table 2 gives an overview of size reduction for both image databases across different steps of the workflow.

We assessed computational quality and accuracy based on the color-adjusted version of common image quality metrics: Mean Absolute Error (MAE), Peak Absolute Error (PAE), Root Mean Square Error (RMSE), and Peak Signal-to-Noise Ratio (PSNR). Using the full (175 time step) image database, we compared images extracted from compressed and uncompressed video databases with corresponding ones from the original image database. Image quality metrics are shown in Table 3. As expected, the different errors rise with higher compression, whereas PSNR falls with higher compression. As can be seen, there are only small differences between the purely encoded video and and the videos compressed with CRF=30/34. For CRF=51, there is a significant drop in quality across all metrics.

Perceptually, we considered when a compressed image is no longer useful from the domain expert

Database:	1250 cameras		200 cameras	
	Size	% IDB	Size	% IDB
Image DB (IDB)	359 GB	100 %	20.49 GB	100 %
Video DB (VDB)	6.3 GB	1.6 %	2.0 GB	12.5 %
Video DB, CRF=30	2 GB	0.6 %	684 MB	4.2 %
Video DB, CRF=34	1.1GB	0.3 %	406 MB	2.5 %
Video DB, CRF=51	93.2 MB	0.03 %	50 MB	0.3 %

Table 2: Comparison of compression levels for two databases with different numbers of Cinema cameras positioned around the globe.

CRF	MAE	PAE	RMSE	PSNR
No CRF	0.008 / 0.024	0.79 / 0.67	0.03 / 0.05	80.1 / 78.6
CRF=30	0.009 / 0.025	0.78 / 0.68	0.02 / 0.05	80.2 / 78.3
CRF=34	0.009 / 0.025	0.76 / 0.68	0.03 / 0.05	80.2 / 77.9
CRF=51	0.018 / 0.033	0.84 / 0.83	0.04 / 0.06	77.1 / 74.2

Table 3: Comparison of error metrics by compression rate over all images. All measurements are computed w.r.t. an image from the original database and are given as median / mean.

point of view. Two experts evaluated compressed images from the pilot run. The experts agreed that the images became unusable at CRF=38. They transitioned to "marginally acceptable" between CRF=34 and CRF=36 and were deemed "completely usable" at CRF=30, with some slight variations based on the size and number of visible eddies and the amount of land in the image.

The same images were also used in a crowdsourced psychophysical evaluation using the 2AFC module from the Evaluation Toolkit (Section 3.1). With 282 participants crowdsourced from Mechanical Turk, the perceptual discrimination threshold was determined to be between CRF=34 and CRF=36, Figure 9, in agreement with expert opinion.

Full details of the compression work can be found in the paper presented at the EuroVis Workshop on Visualisation in Environmental Sciences [11] and an associated LANL technical report [10].

A follow-up study is currently exploring the ability to track ocean features using contour and edge detection techniques in similarly compressed images. Based on the work of Banesh et al. [8], the study will explore the relationship between feature detection metrics, image quality metrics, and perceptual evaluation of image quality for compressed images.

2.2 Perceptual and Cognitive Research

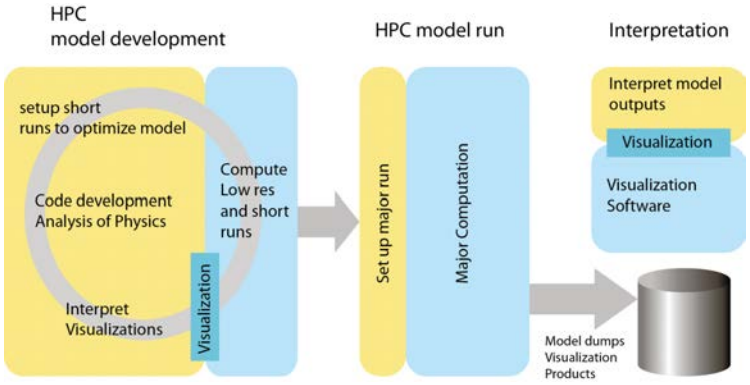


Figure 10: The architecture of the cognitive system that does model optimization for high performance computing. Yellow areas represent human cognition, blue areas represent computation.

To frame our work for this project, it is useful to think of high performance computers and the people who use them as constituting a distributed cognitive system, with part of the work being done in the machine and part in the brains of developer scientists. Figure 10 gives an overview. When scientists and engineers allocate their time to activities necessary for development of computer simulations they must balance the cost of setting up and running informal experiments with the cost of analyzing the results. To better understand how such systems can be optimized we can apply information foraging theory as a framework for predicting how time allocation will be optimized.

To support the study of HPC cognitive systems we borrowed an idea from cognitive economics and created a simplified, easily modeled task, to approximate the cognitive process involved in the optimization of super-computer codes simulating ocean dynamics. The task involved the detection of "anomalous" patterns. In this HPC analytics game, the occurrence of pattern anomalies declines with the number of images viewed to simulate the tendency of developer-scientists to look at the most informative images first. The participants could order a new "run" at any time, although this too had a cost.

Our visual analytics game studies provide a new way of exploring the trade-offs inherent in making cognitive decisions under time constraints. Our results suggest cognitive inefficiencies in the way people perform the task of looking for anomalies in model data [39].

The transition to in-situ generation of visualization products has a major implication for cognitive work flow. There must be a major shift in cognitive work, tasks currently performed after a simulation must now be done prior to the simulation. In-situ products must be chosen ahead of time to provide what are anticipated to be the most useful visualizations. To support this transition software tools will be needed. HPC model users will require tools to show them what in-situ products are available, and to insert code related to the generation of those products into the simulation code.

The following sections detail the work done with this model in mind.

2.2.1 Flow Pattern Identification Study: Animated vs Static Portrayal

The visualization of 2D vector fields has applications including surface ocean currents and surface wind patterns as well as slices through electromagnetic fields. Significant effort has gone into determining the most effective method for statically representing these patterns but the use of animation to show flows and other vector fields has not been previously evaluated despite the fact that this is becoming a common practice. There would seem to be no obvious reason to use animation to represent 2D data based on phenomena that do not change over time; animation, is generally more costly to generate in terms of computation and more difficult to deliver to the viewer. Nevertheless the fact that flow patterns inherently involve movement suggests that animation should be an intuitive model of representation. Also, it is possible that an animated version of a pattern may be easier to perceive than a statically represented version of the same pattern.

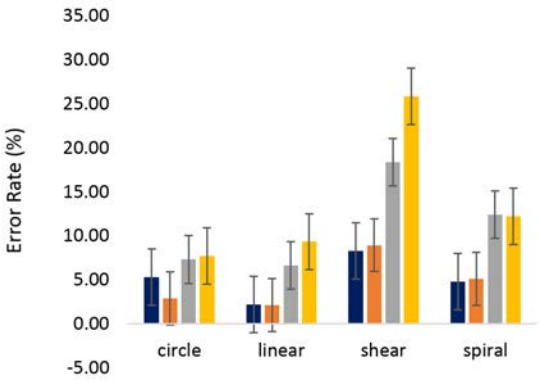


Figure 11: Results from an experiment where the task was searching for patterns in vector fields.

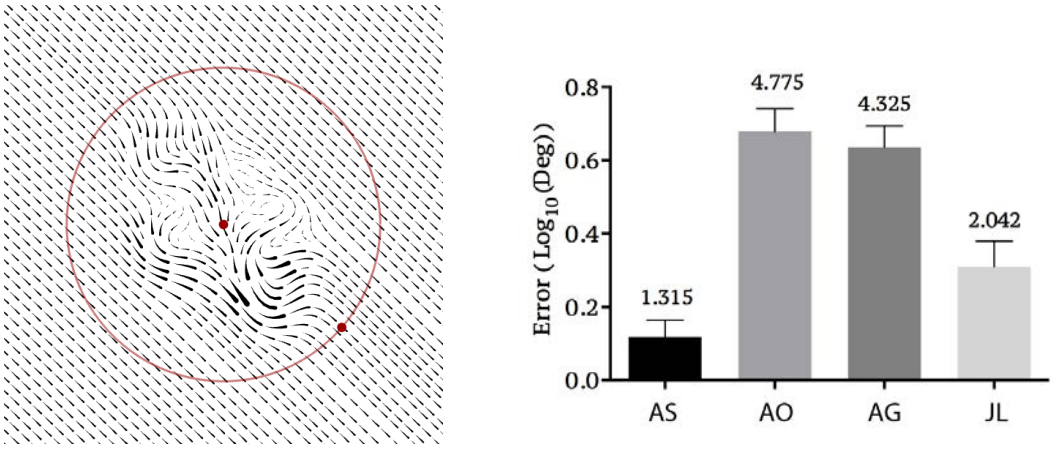


Figure 12: Left: A screen from an advection task. The participant has placed the dot on the circumference of the circle to indicate where a particle dropped at the center would exit the circle. Right: Mean log errors for the advection task experiments. Anti-log values are provided above each bar. AS: Animated streamletes. AO: Animated orthogonal particles. AG: Arrow Grid. JL: Equally spaced streamlines using the Jobard and Lefer algorithm.

We have completed two experiments comparing perception of flow patterns comparing the best static methods with animated streamlets. The results strongly support using animation to show 2D flow patterns. Both of the animated methods cut error rates roughly in half in a pattern search experiment, Figure 11, and one of the animated methods was also the best in the advection path tracing task (Figure 12). The results have been published in [38].

2.2.2 Optimal Viewing for the Perception of Features in Cosmology Simulations

Haloes are theoretical structures used to represent dark matter in the universe and they are modeled as large scale particles systems. Haloes are clusters of particles that have positions, masses, and velocities in three dimensional space.

From a visualization standpoint, haloes can be thought of as three-dimensional point clouds containing various structures of scientific interest including filaments, membranes, and clusters of clusters. The most important depth cues for perceiving structures in three-dimensional point clouds visualization are kinetic depth and stereoscopic depth and this suggests that any visualizations that are generated in situ should support these depth cues.

A straightforward way of providing the motion parallax depth cue for 3D visualization is to compute a set of images from a viewpoint rotating around a structure of interest. But this may result in substantial storage costs and this runs counter to the goal of in situ visualization which is

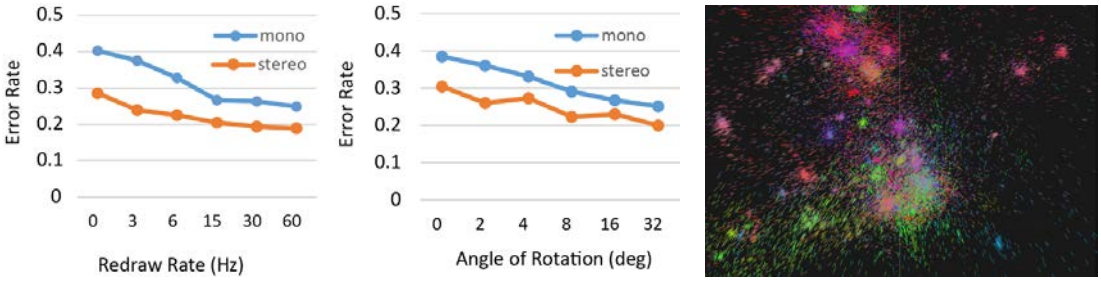


Figure 13: Left: Pattern detection errors with different redraw rates, both with and without stereoscopic viewing. Middle: Pattern detection errors with different rotation amplitudes, both with and without stereoscopic viewing. In both cases it is better to combine motion with stereoscopic viewing. Right: Halo data colored according to the direction of particle movement.

to maximize the potential for insight while minimizing the amount of data that must be saved. We have therefore carried out two experiment to investigate how frame rate and motion angle affects our ability to see 3D structures reliably.

Kinetic Depth Experiment 1: Kinetic depth (frame rate, with and without stereo). The task was to detect patterns in point clouds rotating in an oscillatory fashion about a vertical axis. The frame rate was varied and the display was produced both with and without stereoscopic viewing. Kinetic Depth Experiment 2: Kinetic depth (Rocking amplitude: with and without stereo). The task was the same. In this experiment the amplitude of oscillation was varied. All things being equal, reduced motion is desirable since the viewpoint is preserved. The results of these two experiments are summarized in Figure 13.

The practical implication of these results is that motion parallax is a better cue than stereo for enabling the perception of 3D structures, although ideally both kinds of information should be provided. As a rule of thumb, our results suggest that a redraw rate of 15 Hz and an angle of rotation of at least 8 degrees should be used.

We also experimented with using color to help bring out movement direction for halo particles using color coding. The right-most image in Figure 13 illustrates a frame from a color coded animation of a halo simulation run. A paper describing these results is in press [7].

2.2.3 Perceptually Optimal Compression of Vector Field Data

Large cognitive efficiencies can result from reducing latencies in interactive data exploration. In high performance computing, the key to reducing latencies is reducing the size of the data used in visualization and the key to this is a combination of in-situ sampling and compression.

How do we decide what is an acceptable loss from compression? There are a number of possible criteria based on perception. The most stringent is the just noticeable different (JND). If an image is degraded by less than a JND then it is reasonable to argue that no information is lost. While there has been decades of work on image compression, there has been very little work on the compression of vector fields. In addition, whether compression artifacts can be seen depends on the flow rendering method.

Ware and Turton have developed both a methodology and a series of studies to look at compression issues using both Amazon Mechanical Turk (Mturk) and conventional laboratory experiments. Figure 14 shows a pair of flow patterns based on MPAS data, one set compressed and the other not compressed. The results suggest that it is possible to compress vector field data by a factor of 50 or more without perceptual loss. The consequences of this will be that scientists will be able to review more of the results of a simulation, much more quickly. This work should be taken as a preliminary exploration of the problem. Significant additional work will be required for the result to be applied in operational systems. Further details on the user studies is in Section 2.4.7.

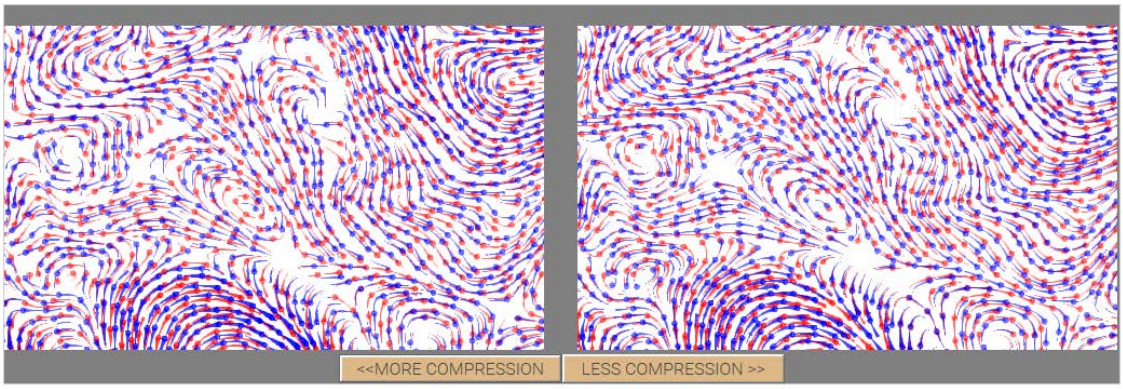


Figure 14: Stimuli images for a vector field compression experimental task. The right hand image is compressed less than the left. The participants task is to determine where there is no perceptible difference between the blue (compressed) and red (uncompressed) patterns.

2.2.4 Evaluating the Perceptual Uniformity of Color Sequences for Feature Discrimination

The design work of Samsel and the mathematical framework of Bujack [14] revealed a large gap in prior work on colormap assessment, namely that there had been little prior research that objectively evaluated colormaps for feature discrimination tasks. The great majority of work on colormap uniformity has relied on uniform color spaces, such as CIElab and CIEDE2000. However, there are good reasons for thinking that these are not suited to the task.

To meet the need for a rigorous and easy to use colormap assessment method, Ware designed the test pattern shownin Figure 15. Sets of these patterns can be used to evaluate the feature discrimination functions of different colormaps for features of particular sizes. Turton and Ware designed and implemented a methodology that used sets of these patterns in a Mechanical Turk study (see Section 2.4.3) using nine base colormaps, Figure 15.

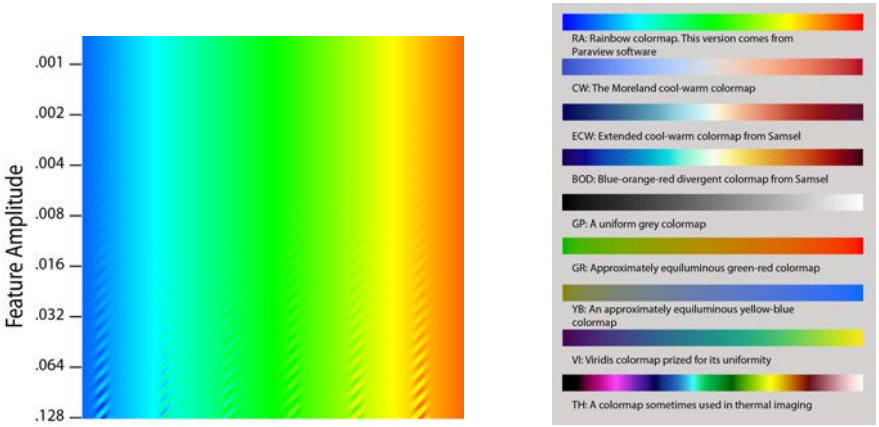


Figure 15: Left: The test pattern we developed is rendered in the rainbow colormap. Six vertical columns of features are shown. In each column the contrast decreases by a factor of two, every 80 pixels. A set of five images per colormap yields 30 sample points across the colormap. Right: The set of nine base colormaps used in the feature discrimination papers. From top to bottom: rainbow, greyscale, green/red, yellow/blue, thermal, cool/warm, Viridis, and two ECX colorscales: extended cool/warm and blue/orange divergent.

The measured feature discrimination functions for the set of nine base colormaps are shown in

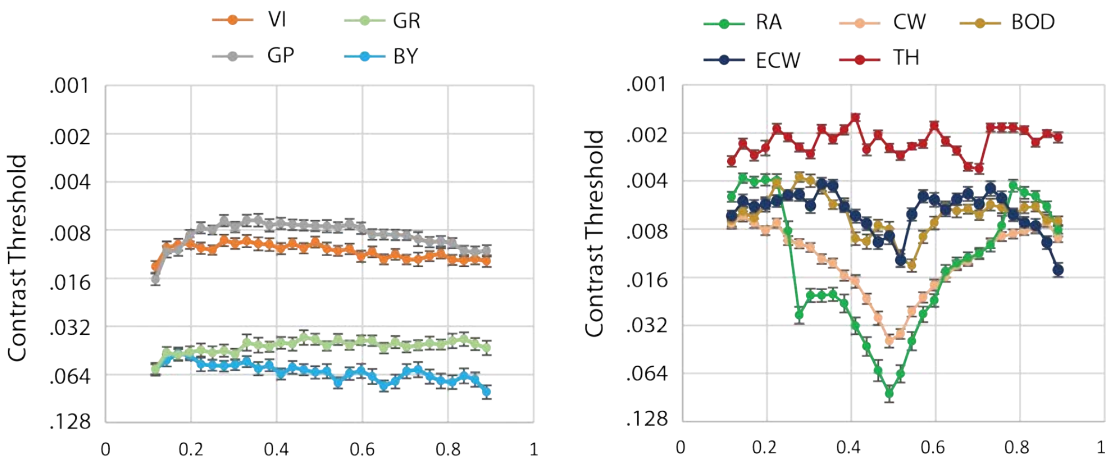


Figure 16: The measured feature discrimination functions for nine colormaps.

Figure 16. Among other things these results show that the rainbow colormap is extremely non-uniform while the Samsel double-ended colormaps have greater feature resolving power than in commonly used dark-to-light colormaps, such as Viridis.

Using the data from these studies, we were able to make a theoretical contribution. Opponent color theory holds that early visual processing transforms the retinal signal into red-green, yellow-blue and dark-light color channels. These channels are known to have very different characteristics in terms of their feature resolving power. We tested the hypothesis that we should be able to model the measured feature resolving power functions by modifying the CIElab uniform color spaces, underweighting the contributions of the green-red and yellow-blue channels. Our results confirmed this; by reducing the color channel weights to 0.15 we were able to generate an excellent fit to the empirical data.

A preliminary account of the experimental work appeared in the 2017 EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization [41]. A longer paper adding features of 10 pixel and 45 pixel size to the original 15 pixel size has been submitted for publication in TVCG [40].

2.3 Perception and Color

Color can convey enormous amounts of information in visualizations of large scientific data, although its potential is seldom fully tapped due to the time-consuming nature of quality colormap construction and the level of expertise required. Within the visualization research community there is a tradition of drawing upon artistic techniques (e.g., color theory, line illustration styles, oil painting layers, or cinematic camera placement) to develop encodings of data. Less common, however, is engaging directly with art/design processes and with artists themselves. The ECX project bridged that gap by bringing in an artist with expertise in color to explore how perception and cognition in scientific visualization could be optimized through improved color approaches.

This process began with collaborative engagement with a wide range of domain scientists. Themes echoed by scientists across domains included the need to see greater perceptual detail in data and the difficulty scientists had in developing data-specific color encodings with current tools. We have made numerous contributions to address these needs: colormaps with greater perceptual depth; linear color scales varying in contrast distribution; structured colormaps; color sets; recommendations for 2D and 3D multivariate data; and the ColorMoves Tool which enables scientists to interactively develop data-driven color encodings. These products, represented in Figure 17, have been brought together in our website, `http://www.sciviscolor.org`.

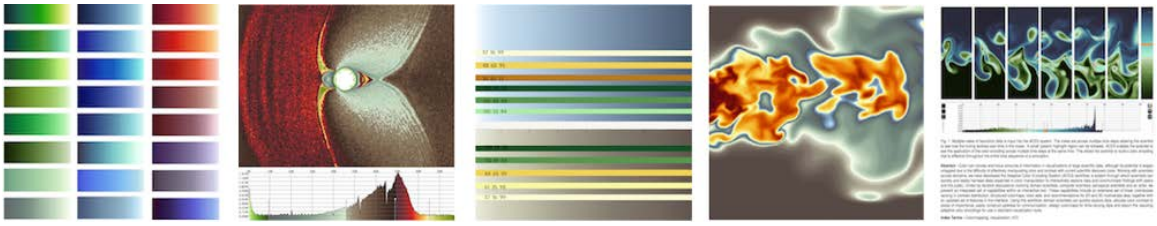


Figure 17: `http://www.sciviscolor.org`has five sections from left to right: the Colormap Section containing linear colormaps, divergent options, structured colormaps and discrete colormaps; ColorMoves, our interactive colormap construction tool; Color sets, combinations of color scales and discrete colors that can be combined harmoniously in one visualization; Color strategies, a few easy to follow suggestions for making colormaps tuned to the data and visualization needs; and Publications and Projects. Not represented here but also on the website are examples of 3D color usage as well as a gallery from which to draw color encoding solutions.

2.3.1 Colormaps that Improve Perception of High-Resolution Ocean Data

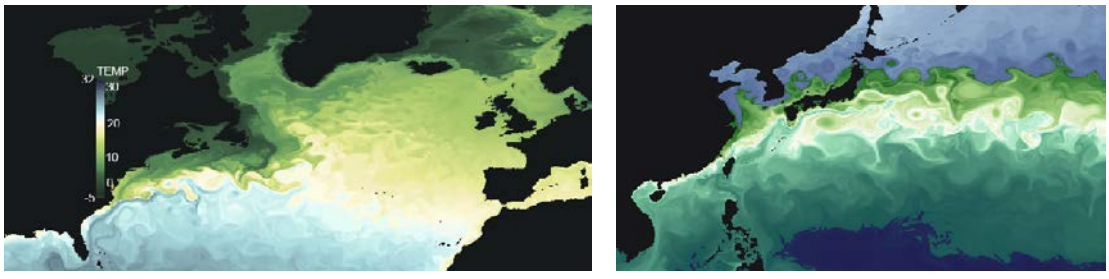


Figure 18: Left: The Gulf Stream current rendered in the ECX blue/green divergent colormap. Right: a nested colormap is used to highlight the Kuroshio current off the coast of Japan.

In an early collaboration with MPAS-Ocean scientists [26], Samsel exploited color contrast theory to create divergent colormaps with greater perceptual depth. This paper also pioneered the use of structured colormaps to place color directly where it was needed to emphasize features in the data, in this case nesting a colormap within the data range covered by a current of interest. A color counting user study in that paper showed that participants were able to see higher number of colors in the

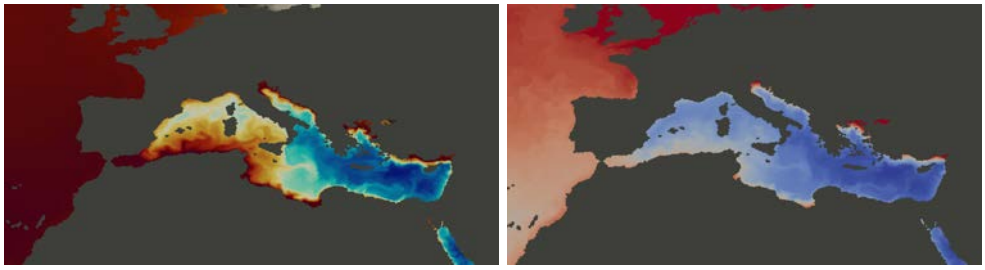


Figure 19: Left: a colormap showing greater detail in ocean salinity; this colormap was created using ColorMoves to place color directly into the most important region of the data. Right: a colormap created within ParaView by restricting the data range.

new ECX colormaps, in particular, a blue/green divergent and an extended cool/warm. Figure 18 illustrates the use of the blue/green divergent colormap to highlight the Gulf Stream current (left) and the use of a nested colormap to bring out the Kuroshio current (right). The crowdsourced user evaluations within this paper validated the scientist's conclusions that these colormaps provided greater discriminative power than the commonly used (Moreland) cool/warm divergent or rainbow colormaps. The same color counting approach (Section 2.4.5) was later used to validate a subset of ECX colormaps designed for environmental sciences (Section 3.5.2).

Domain convention in the climate sciences, which gives precedence to a cool versus warm approach in divergent colormaps, led to the development of a blue/orange divergent (BOD), a cool/warm colormap taking a longer line through color space. The blue/orange and the extended cool/warm (ECW) were recently tested using a new methodology described in Section 2.2.4. These two colormaps were found to have greater discriminative power for resolving small features than three commonly used colormaps: rainbow, Viridis, and the Moreland cool/warm (Figure 31) [41]. The blue/orange divergent and a yellow/green/blue (YGB) ECX colormap were also tested in a key reading study (Section 2.4.1) and found to be more effective at a statistically significant level at this quantitative task as shown in Figure 30.

2.3.2 ColorMoves

First introduced at SIGCHI'16, ColorMoves [23] is a browser-based tool to enable scientists to quickly and easily develop colormaps tailored to their data and task. Its interactive capability allows scientists to see changes in the visualization in real time. Interactive exploration of the data provides the ability to identify specific ranges of interest within the data and apply the hues and degrees of contrast to optimally represent the data.

This ability was exploited in this paper to provide greater insight into ocean salinity. Salinity is a variable that only varies over narrow geographic ranges, making it difficult to effectively visualize using traditional colormaps and visualization tools. ColorMoves allowed the scientist to place color contrast within very small data ranges, allowing greater detail to be seen as illustrated in Figure 19. A crowdsourced user study comparing colormaps developed within ParaView by using default colormaps over restricted ranges was compared to multiple colormaps created using ColorMoves to place color directly in an optimal data range. This study, along with expert opinion and usability assessments, validated the efficacy of ColorMoves to create data-specific color encodings.

The power of ColorMoves is in how it enables scientists to allocate color contrast to areas of importance, to easily construct palettes for communication. or design colormaps for time-varying data. An example of using ColorMoves to develop a colormap that is effective over many time steps can be seen in Figure 20. ColorMoves is described in detail in Section 3.5.1.

2.3.3 Colormaps: Structured, Linear, Divergent

In order to meet the needs of scientists, a broad range of linear colormaps have been developed, and incorporated into ColorMoves. These are shown in Figure 21. These linear colormaps provide a

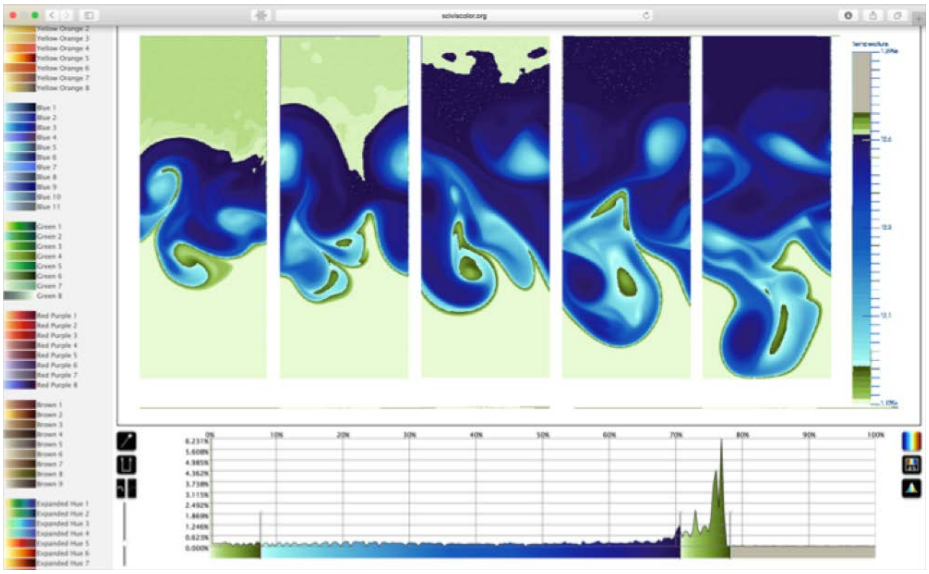


Figure 20: One of the most popular features in ColorMoves is the ability to upload multiple time steps into one window. This allows one to assess the efficacy of a colormap to show the data across time. Here, five time steps of MPAS-Ocean baroclinic data are shown side by side allowing one to see how the structure of the data unfolds in time with a single colormap applied.

range of luminance and color contrast distributions that serve as building blocks for more complex data-driven color encodings.

The recent perceptual evaluation methodology, described in Sections 2.2.4 and 2.4.3, showed that the luminance distribution has the most significant impact on the discriminatory power of a colormap. However, drawing on additional concepts of color contrast theory allows a colormap to attain comparable discriminative power while going through narrower hue ranges (see Figure 31). Figure 22 walks through the design process. This linear green colormap goes beyond a simple dark to light contrast. The development of such a colormap combines additional shifts of cool/warm greens. The greens move from warm to cool to warm. These complementary shifts of multiple types and distributions of contrast create higher discriminative power than simply monotonically increasing in luminance.

Since many domain scientists are primarily concerned with perceptual depth into the data and/or the communication task, we have focused on providing the community with a range of linear, divergent, and structured colormaps that contain multiple types of luminance distributions. The structured colormaps, Figure 23, consist of spans that cover different percentages of the colormap enabling scientists to highlight specific sections of data without building a colormap from scratch.

A set of the early ECX colormaps were incorporated into ParaView [22]. Since then, subsets of the SciVisColor colormaps have been incorporated into a range of visualization tools including ViSIT and VisUS. Additionally, the ColorMoves section of the website has a script to create a matplotlib compatible color object.

2.3.4 Color Sets and Color Strategies

Selecting a colormap for specific data involves aligning areas of importance within the data with sufficient contrast within the colormap. The complexity of the statistical distribution within the data and the goals of the particular visualization impact the difficulty of determining an optimal colormap. A far more difficult problem arises when there are multiple variables and the interaction of colors impacts the contrast balance. Scientists frequently requested a “starting point" for constructing colormaps. Many visualizations call for layering scalar fields and variables. For these situations we developed sets of colormaps and discrete colors able to exist with one another with minimum

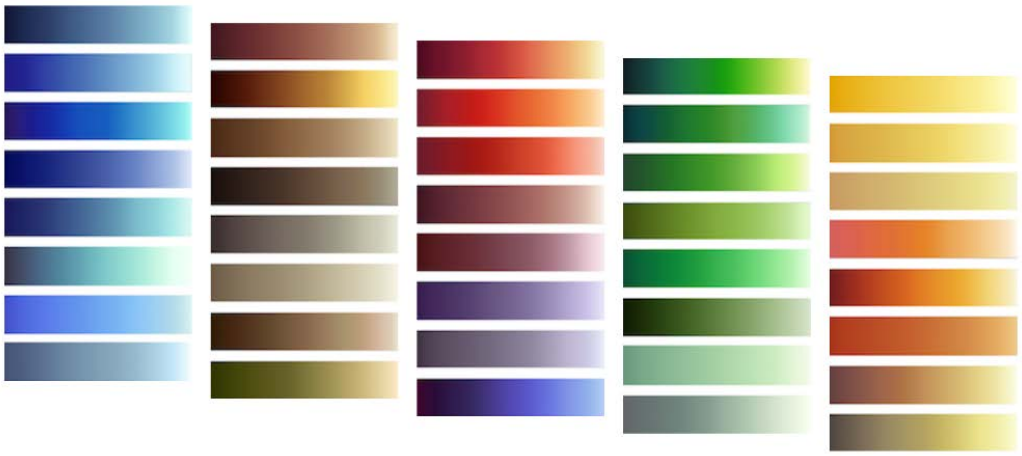


Figure 21: The 40 colorscales developed to meet the range of needs within scientific visualization. The scales are divided by hue groups containing scales that differ in the range of the hue span, level of saturation, distribution of luminance and more.

Figure 22: Construction of a green colorscale that moves from a *warm* dark yellow-green through a mid-range *cool* blue-green and back to a *warm* light yellow-green. This allows the colorscale to maintain intuitive order while moving between warm and cool greens – shifts that create additional contrast.

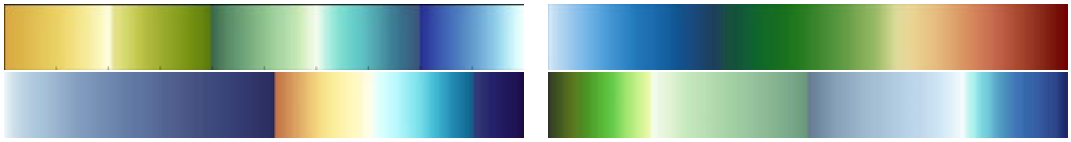


Figure 23: Four of the structured colormap options available at SciVisColor.org.

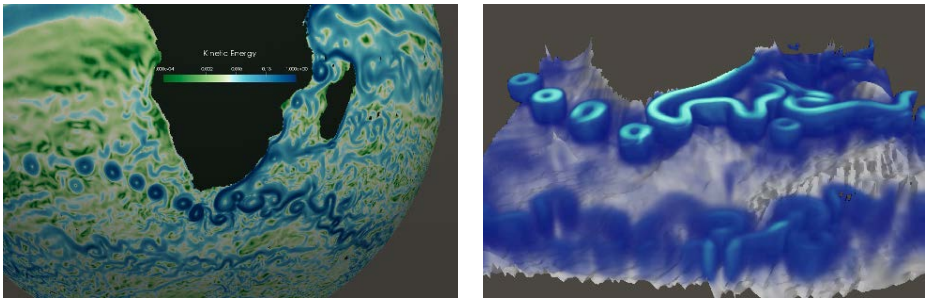


Figure 24: Eddies and currents around the Agulhas Retroflection in the Southern Ocean are visualized on the left using the surface kinetic energy variable in an ECX blue/green divergent colormap. On the right, the same variable is visualized in three dimensions utilized VolViewer, the ray casting tool described in Section 3.7.

interaction and thus distraction from the data itself. Also provided are sets of color scales that can be combined for use on 3D data with multiple scalar fields, on both surfaces and/or volumes.

2.3.5 Case Studies in Color Application

The above approaches to color and colormapping were applied to specific data and visualization challenges resulting in multiple ECX publications and awards. Figure 24 shows two images from a video of ocean currents and eddies [25], which won the Best Award for the Scientific Visualization & Data Analytics Showcase at the 2015 ACM/IEEE International Conference for High Performance Computing, Networking, Storage, and Analysis (SC). The colormap used to visualize the eddies and currents in the surface kinetic energy variable (left) was an early version of the ECX blue/green divergent colormap now available in ColorMoves. The development of this colormap provided significant improvement over then-available colormaps in the visualization of the eddies around the Agulhas Retroflection. The right-hand image depicts the innovative ray casting technique of VolViewer (Section 3.7) to view the structural depth of the same ocean eddies and currents in 3D from ocean floor to ocean surface.

Another application of these techniques can be seen in a recent SIGCHI publication [24] that highlighted a multi-disciplinary collaboration to improve visualization of multi-variate ensemble data of asteroid impacts. Figure 25 depicts three variables of interest: water vapor, asteroid ablation material, and temperature. This visualization exploited the use of innovative application of opacity together with optimized color triads for representing multiple variables in a single visualization with minimal visual interference between the variables.

This particular collaboration is an excellent example of the ECX approach which brings together a multi-disciplinary team of scientists, HPC experts, visualization specialists, and an artist in an iterative process to facilitate scientific discovery.

A video based on the ensemble asteroid data also won the Best Award for the Scientific Visualization & Data Analytics Showcase at the 2016 ACM/IEEE International Conference for High Performance Computing, Networking, Storage, and Analysis (SC) [20].

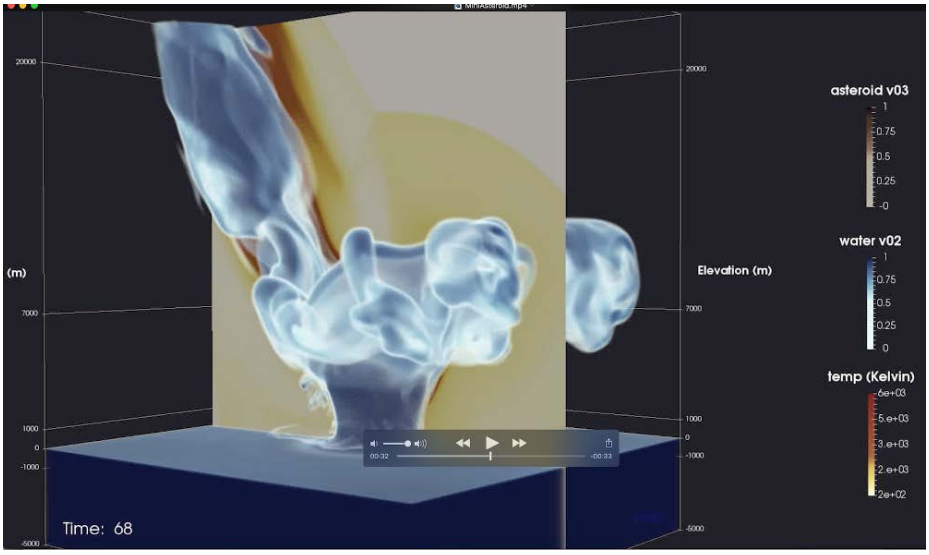


Figure 25: Ocean asteroid impact simulation with LANL Senior Scientist, Galen Gisler. Three variables are represented – water vapor, the asteroid and its ablating material as well as temperature on a center slice of the data. A video based this work won the Best Data Visualization and Analytics Showcase at SuperComputing 2016.

2.3.6 Trajectory Mapper: Interactive Widgets and Artist-Designed Encodings for Visualizing Multivariate Trajectory Data

Trajectory Mapper (Figure 26) represents a collaboration between Samsel and the University of Minnesota Interactive Visualization Lab. A system of novel interactive widgets and artist-designed visual encodings to support exploratory multivariate visualization of spatial trajectories, Trajectory Mapper was presented as a short paper at EuroVis2017 [18]. Trajectories are rendered using a three-way multi-texturing algorithm so that the color, texture, and shape of each mark can be manipulated separately in response to data. Visual encodings designed by artists and arranged in categories (e.g., divergent, linear, structured) are utilized as strong starting points for visual exploration. Interactive widgets including linked parallel coordinates plots, 3D camera controls, and projection to arbitrary 3D planes facilitate data exploration. An innovative visual mapper menu enables rapid experimentation with alternative data mappings using the artist-designed or custom encodings that can be created with no programming using image editing software.

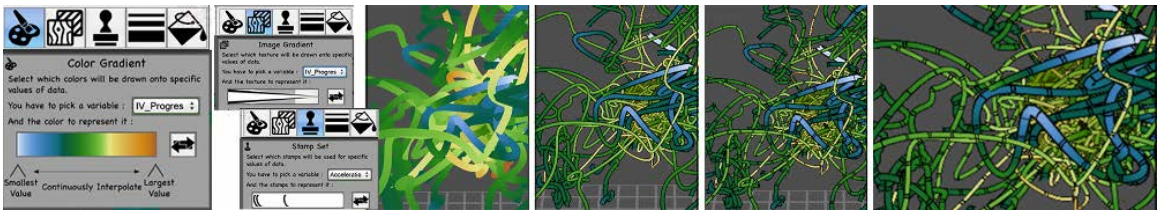


Figure 26: The Trajectory Mapper interface and example encodings.

2.3.7 A Theoretical Framework for Colormap Assessment

The multi-disciplinary approach of ECX includes cognitive and perceptual expertise, a visual artist, computer scientists, and researchers with backgrounds in mathematics and physics. This unique blend of viewpoints led us to the realization that there is no common framework for understanding colormap design and assessment rules. The same term, e.g. linearity, may be interpreted differently by a mathematician than it is by an artist or a perceptual scientist. These ambiguities and differences

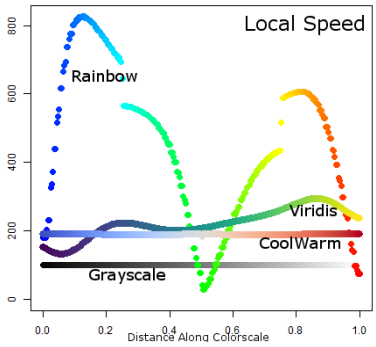


Figure 27: Speed can be used to measure uniformity, discriminative power, or order. In this figure, the local speed of four colormaps, mapped according to the specific colormap, is plotted assuming the metric distance in CIELAB76. The rainbow colormap is highly non-uniform compared to grayscale, cool/warm, or Viridis.

exist throughout the literature on colormap design. Such inconsistencies make designing meaningful experiments to assess colormaps difficult and likewise makes it difficult to automate colormap improvement or design. The effort to understand and resolve these ambiguities led to a paper to be presented at IEEE VIS 2017 [14].

Perceptual Rule	**Mathematical Rule**	**Evaluation Measure**
local discriminative power	longest path	average local speed
local uniformity	constant speed	deviation of local speed
local legend-based order	local invertibility	minimal local speed
local intuitive order	–	local triangle side difference
global discriminative power	–	average global speed
global uniformity	constant speed/linearity	deviation of global speed
global legend-based order	global invertibility	minimal global speed
global intuitive order	–	global triangle side difference

Table 4: Summary of the relationship between the perceptual design rule and the mathematical rules, which we identified in our literature research, and the available measures that can be used for their evaluation in any metric color space.

Beginning with an extensive literature review, we gathered together similar design rules, proposing a concise unified taxonomy for colormap design rules using unambiguous nomenclature. We then developed a mathematical framework for color that is independent of task, data, and color space and requires only the assumption of a metric distance that exists in some color space and that mimics human perception. From the distance measure, we derive a metric speed and resultant statistical properties such as average, standard deviation, minimum, and maximum.

Our framework associates a mathematical definition with a perceptual concept for three of the most common colormap design concepts: uniformity, discriminative power, and order. We clarify and separate global and local interpretations and provide measures for each of the these perceptual concepts. The associated perceptual rules, mathematical rules, and evaluation measures are listed in Table 4. This framework thus provides straightforward measures to meaningfully assess colormap properties. Examples of some of the measures can be seen in Figure 27 where the local speed is compared for four commonly used colormaps and in Figure 28, a matrix visualization of the global speed.

The development of the mathematical measures enables both user evaluation and assessment approaches. A set of pilot user studies (Section 2.4.6) looked at global versus local order in common colormaps. Additionally, the collaboration on the mathematical framework motivated the experimental

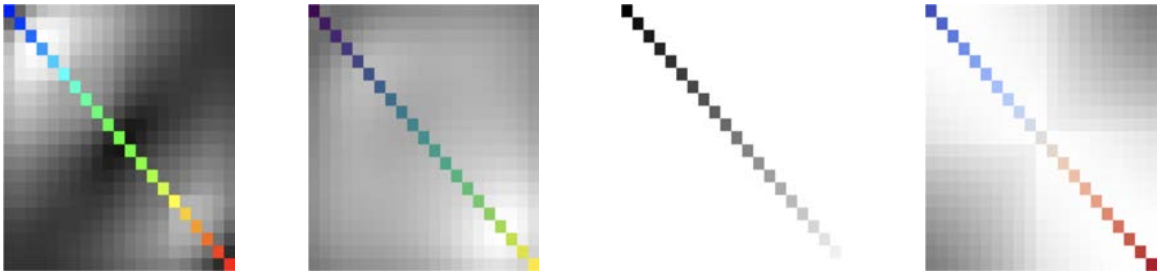


Figure 28: Matrix visualization of global speed for (left to right): rainbow, viridis, grayscale, and cool/warm. The luminance of each entry (i, k) represents how high the speed $V_{i,k}^{\Delta E^{76}} = \Delta E^{76}(x_i, x_k)/(t_k - t_i)$ is from the color $x(t_i)$ in the same row i to the color $x(t_k)$ in the same column k on the diagonal using the ΔE^{76} metric. Note again the non-uniformity of the rainbow in comparison to the overall uniformity of the other three colormaps.

approach to directly measure discriminative power [40, 41]. The resulting papers are described in Section 2.2 and additional details on the related user studies are found in Sections 2.4.3 and 2.4.4.

In order to facilitate comparing colormaps and informing decisions on choice of colormaps, the measures for colormap assessment as derived in this paper have been implemented in an online tool at `http://www.colormeasures.org` [13], discussed in more detail in Section 3.6.

This theoretical framework has already served as a catalyst to develop experimental methodologies to directly measure perceptual concepts relevant to colormap assessment and design. In future research, we hope to build upon this work by extending our theoretical framework to find mathematical descriptions for additional perceptual qualities of colormaps, such as smoothness, and associated measures to evaluate these additional properties. This paper lays the groundwork for an extensive user evaluation program to experimentally test the derived measures in order to determine which measures most accurately model human perception. This research is foundational for the development of an automated approach to colormap assessment, improvement, and design.

2.4 Perceptual User Studies

The ECX collaboration has included a robust program of cognitive and perceptual evaluation carried out by our collaboration partners at UNH, UT and LANL. The perceptual evaluations of color and flow visualization have been one aspect of that program. From this research, ECX has garnered a wide range of publications (see Section 1.6 for the full list of publications). This section details recent user studies contributing to ECX publications.

2.4.1 Validating a Crowdsourced Approach to Colormap Assessment

Over the past year, we carried out a study to validate Amazon's Mechanical Turk (Mturk) as a research platform for color-based user evaluation. Mturk and other crowdsourcing sites can provide an inexpensive and easy approach to participant recruitment for user studies. Within the behavioral sciences such as psychology and linguistics, validation of Mturk was a critical issue accomplished through replication of classic experiments previously done in a laboratory setting. Likewise, researchers in information visualization used replication of previous work to validate and understand Mturk and its workers as a platform for research. This research resulted in a paper presented at the 2017 EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization [35].

From the perspective of visualization studies, the wide range of subjects and monitors offers a significant benefit in its ecological validity, at the cost of losing some measure of control over those same features. For evaluations involving color, one of the most significant issues is color vision deficiency (CVD). In order to understand the effect of CVD, Turton and Ware led the effort to quasi-replicate an experiment from Ware's 1988 paper, *Color sequences for univariate maps: Theory, experiments and principles* [37]. The *key task* in that paper provided an experiment that could be easily crowdsourced and provided a measurable quantity, the mean absolute error, for statistical comparisons. In this experiment, participants were shown a set of crosshairs on a data set and asked to choose the color from a set of 16 color keys that best matched the color at the center of the crosshairs. Our 21st century version of this can only be considered a quasi-replication as the original data was lost in the winds of time. For this updated version of the experiment, Ware created a synthetic scalar field based on Gabor functions as the data set, rendering the data in a variety of colormaps. Two examples of the rendered data set are shown in Figure 29.

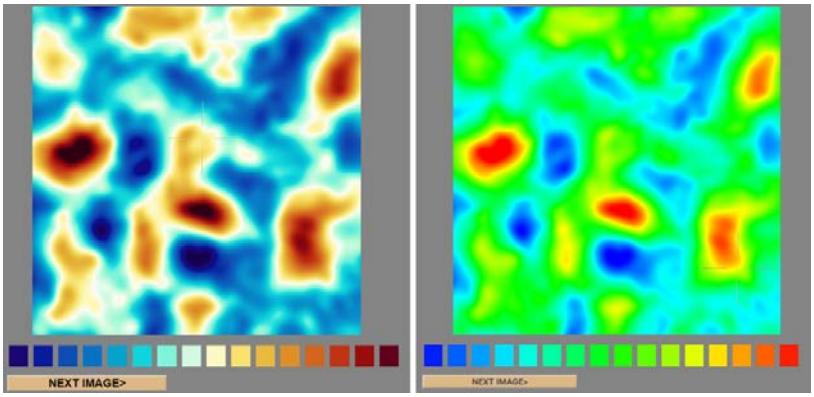


Figure 29: Two examples of the key task stimuli. On the left, the synthetic data set is rendered in the blue/orange divergent colormap, on the right, in the rainbow colormap.

In order to compare results between the previous and current experiment, we chose four colormaps similar to those in the Ware 1988 paper: a rainbow (from ParaView), a red-green, a greyscale and a grey to red saturation colormap. In order to update the experiment and to assess ECX-developed colormaps, two common standards were included: the Moreland cool/warm and Viridis; and two ECX colormaps were included: blue/orange divergent and a yellow/green/blue colormap. The colormaps are shown in Figure 30.

We defined three participant pools: (1) a CVD group of participants self-identifying as having CVD while taking a Farnworth D15 color cap arrangement test for colorblindness; (2) a women-only group that was effectively CVD-free; and (3) a usual Mturker group, typical of the participant pool we select when doing user studies on Mturk. Further details on how these groups were populated can be found in the paper. In all, we collected over 8000 trials across the eight colormaps and three participant groups on Mturk.

2.4.2 Results

We found statistically significant differences between CVD and non-CVD participants across many of these colormaps with the exceptions of the luminance-based greyscale, the standard cool/warm, and the ECX blue/orange divergent. The mean absolute errors can be seen in Figure 30 for all colormaps and participant groups. When comparing the non-CVD group with a more typical group of participants, the only significant difference we found between the usual group and the non-CVD group was for the saturation grey-red saturation colormap – a colormap demonstrably poor at conveying metric information. From these results, we conclude that, with reasonable precautions to minimize potential colorblind issues, Mturk can provide a valid research platform for color-based studies.

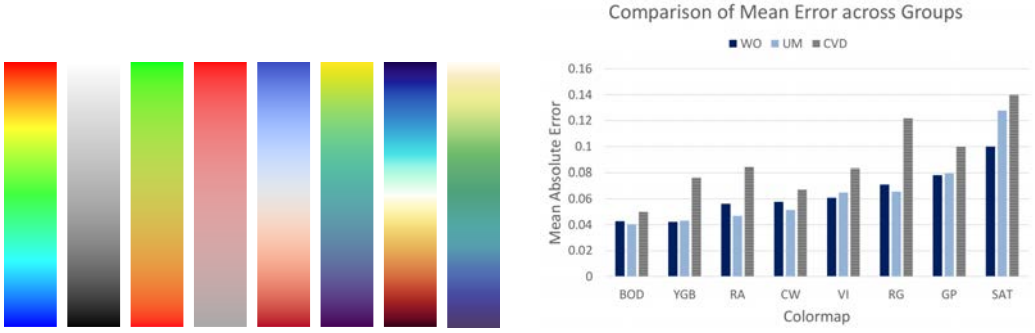


Figure 30: Left: The eight colormaps used in the crowdsourcing experiment. From left to right: the rainbow, greyscale, red/green and grey to red saturation scale were similar to those in the original Ware 1988 experiment; the cool/warm and Viridis are current standards while the blue/orange divergent and yellow/green/blue are from ECX (developed by F. Samsel.) Right: Mean absolute errors for three participant groups and eight colormaps from the key task study. Note the differences between the CVD group, a group highly populated by Mturkers with color vision deficiencies, and the other two participant groups. The WO (women-only) group is effectively CVD-free while the UM (usual Mturker) group is the one typically used in ECX color user studies. The ECX blue/orange divergent and yellow/green/blue were particularly effective at the key reading task – a measure of a colormap's effectiveness for the quantitative task. Note that the blue/orange divergent was also found to be colorblind-safe.

We also assessed the ability of a colormap to carry metric information. A comparison across colormaps found that the ECX blue/orange divergent and yellow/green/blue colormaps both provide an improved ability to impart metric information compared to some common standards. While the rainbow colormap performs well for qualitative tasks, its well-known flaws argue against its common use. The results from these studies provide tested alternatives to the rainbow when choosing a colormap for a quantitative task.

Note that the JavaScript/HTML/CSS code used to generate the keys for each image was generalized into the Key Task module for the Evaluation Toolkit (Section 3.1).

2.4.3 Discriminative Power and Uniformity User Studies

Stemming from discussions key to the theoretical framework paper (Section 2.3.7), Ware and Turton developed a methodology to test the uniformity and overall discriminative power of a colormap through feature detection (Section 2.2.4). The stimuli consisted of a set of images with small features embedded within the colormap. An example stimuli is shown in Figure 15. The participant task

was to click on the highest feature visible in each column. Five variations of the stimuli, each with six columns of features equally spaced but shifted along the colorscale, measured the discrimination threshold across the colormap.

These studies resulted in a paper presented at the 2017 EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization [41].

The experiment gathered 147 Mturk participants and minimized potential CVD contamination using the CVD exclusion group developed above (Section 2.4.1). For this initial paper, data was collected for a feature size of 15 pixels. Three of the ECX colormaps, the blue/orange divergent (BOD), the extended cool/warm (ECW) and the yellow/green/blue (YGB) were part of the initial set of eight colormaps tested in order to validate the methodology although the YGB was not included in the publication due to its non-uniformity. Overall the results revealed clear differences between different colormaps with quite small inter-subject variability, (Figure 16), suggesting that this method is indeed an effective and easily applied test of the feature resolving power of colormaps. The ECX blue/orange divergent and extended cool/warm colormaps provided the best resolving power over most of their extents while maintaining good uniformity.

This same methodology was extended to nine base colormaps, shown in Figure 15. The yellow-blue (YB) and green-red (GR) colormaps were added to explicitly probe the color channels and the thermal (TH) was added for its extreme luminance changes. Additional Mturk studies were run for features at 10 and 45 pixel sizes along with studies at 15 pixel for the newly added YB, GR, and TH base colormaps. These user studies were part of the recently submitted TVCG publication [40].

The mean discriminative power for each colormap and at each feature size is shown in Figure 31. The error bars are 95% confidence intervals. These studies and the resultant analysis demonstrate the relative importance of luminance over the colors channels in creating discriminative power in a colormap.

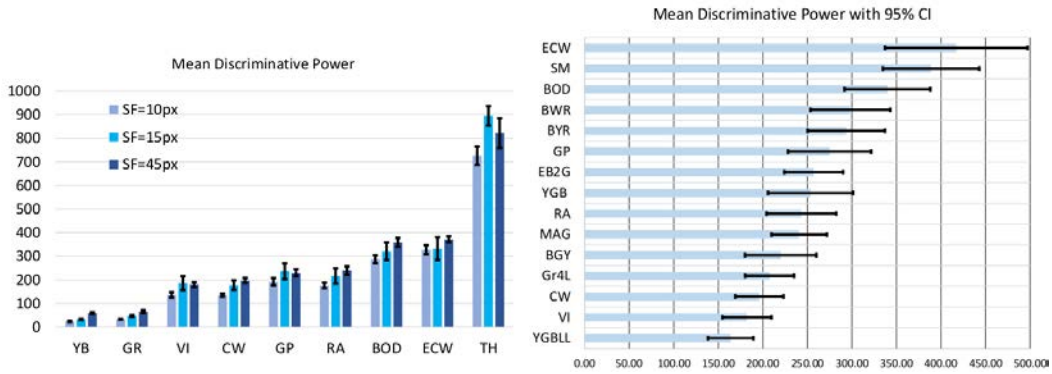


Figure 31: Left: The mean discriminative power for the nine base colormaps tested at three spatial frequencies: 10, 15 and 45 pixels. Error bars correspond to 95% CIs. Note how the thermal colormap, with many luminance shifts has the highest discriminative power, followed by the two ECX divergent colormaps, also with multiple luminance traversals. Right: The mean discriminative power for the set of tested ECX colormaps and the comparison colormaps drawn from ParaView and matplotlib. Error bars correspond to 95% CIs. Note how the ECX Gr4L colormap, which goes through a narrow range of green hues, has comparable discriminative power to colormaps such as Viridis or Magma that go through wider ranges of hue.

2.4.4 Discriminative Power of ECX Colormaps

In order to assess the discriminative power of ECX colormaps, this methodology was extended to features at a fourth size: 30 pixels. User studies were run across the same nine base colormaps, along with a set of ECX colormaps, and a wide range of similar colormaps from ParaView and matplotlib. The colormaps that were tested include the ECX yellow/green/blue (YGB), a muted yellow/green/blue (YGBLL), a linear green (Gr4L), blue/orange divergent (BOD), and the extended cool/warm (ECW). The comparison colormaps are all open-source colormaps available either from

ParaView or matplotlib. The tested colormaps include: BuGrYl (BGYm), erdc_blue2green (EB2G), Magma (MAG), Viridis (VI), BuYlRd (BYR), BlWhRd (BWR), Moreland cool/warm (CW), and seismic (SM). All colormaps can be seen in Figure 32. In all, 18 colormaps were tested at the 30 pixel size.

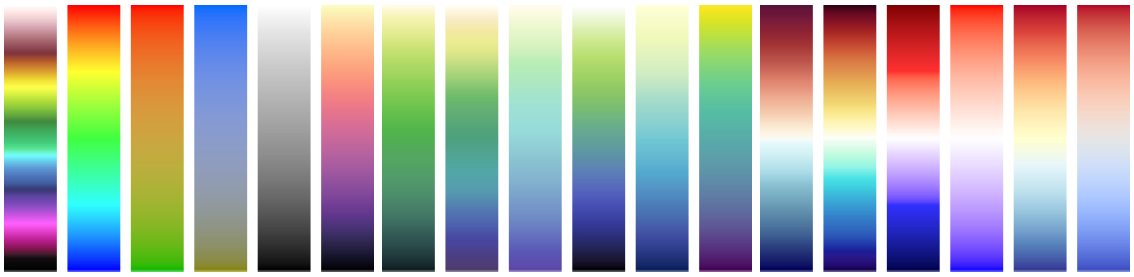


Figure 32: The 18 colormaps tested in the 30pixel feature size study. From left to right: TH, RA, GR, YB, GP, MAG, Gr4L, YGB, YGBLL, EB2G, BGY, VI, ECW, BOD, SM, BWR, BYR, CW.

A plot of the mean discriminative power for each colormap is shown in Figure 31. Overall the divergent colormaps have greater discriminative power than linear colormaps. This is consistent with the previous finding of the importance of luminance in feature discrimination. Note that the TH, YB, and GR colormaps have been excluded from this comparison.

2.4.5 Color Counting Studies

A recent paper [28] details a set of intuitive colormaps designed to enhance visualization of environmental data. Using the color counting technique developed in Samsel et al. [26], a small subset of the colormaps developed for this environmental visualization paper were compared to current standard colormaps. The colormaps tested included a linear set and a divergent set. The colormaps were rendered in both a synthetic dataset and an MPAS-Ocean dataset as the experimental stimuli.

The colormaps used in this study are shown in Figure 33 as rendered in the synthetic data set. The first is a set of divergent colormaps, the second is a set of linear colormaps. The first set consisted of a blue/green divergent (BGD), a brown/blue divergent (Br4), an alternate brown/blue divergent (Br6), a green/brown divergent (GrBr). The blue/orange divergent was also included in this group. The standard cool/warm was used as the comparison colormap. The second set consisted of a linear blue (BL17), a linear green (Gr4L), a green for environmental purposes (GrEnviro), and YGB. These linear colormaps were compared to Viridis.

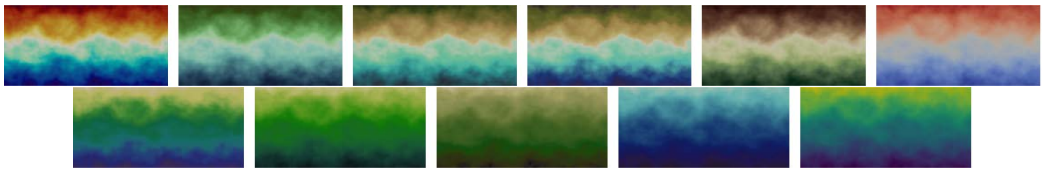


Figure 33: The sets of colormaps tested from the intuitive environmental visualization paper. Top row of divergent (left to right): BOD, BGD, Br4, Br6, GrBr, and the standard CW. Bottom row of linear colormaps: YGB, Gr4L, GrEnviro, Bl17, and the standard VI.

As in the analysis of [26], the non-parametric *sign* test was used to compare the number of distinct colors seen by a crowdsourced group of participants. For the divergent comparisons, BOD again proved to have the highest discriminative power against all of the other options, with p values ranging from 0.001 to 0.05, across both the synthetic and ocean data sets. The other ECX divergents also had statistically higher numbers of colors identified than the standard cool/warm, with p values ranging from 0.01 to 0.05. For the linear colormaps, YGB had statistically higher numbers of colors identified versus Viridis ($p < 0.05$) and all the other colormaps showed comparable levels of colors identified.

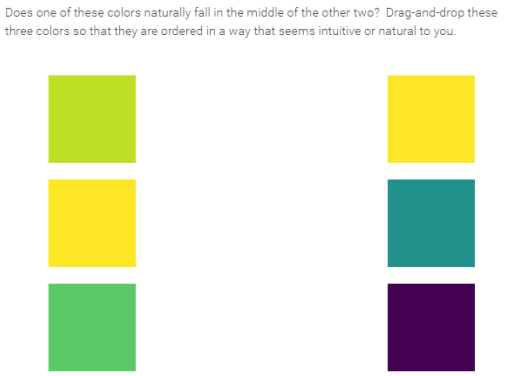


Figure 34: A screenshot from a user study on order. Participants were asked to order the three patches in some way that seemed intuitive or natural to them. The set of patches on the left were near neighbors (local) within the Viridis colormap. The three patches on the right were drawn from distant or global sections of the colormap. Note that a participant only saw one set of three patches in any given question.

These tested colormaps, along with a complete set of other colormaps relevant to environmental visualization, were incorporated into *ColorMoves: The Environment* [27,28] with environment-specific nomenclature (see Section 2.3). All colormaps for *ColorMoves: The Environment* are available from a GitHub repository [21]. While the user studies results were not included in this short workshop paper, they were useful in understanding the characteristics of these colormaps.

Note that the JavaScript/HTML/CSS code used to run the color counting study was generalized into the Click Counting module for the Evaluation Toolkit (Section 3.1).

2.4.6 Color Ordering Studies

As part of the collaborative work on the theoretical framework for color assessment paper [14], described in Section 2.3.7, a series of user studies were carried out to study the concepts of local and global order. In these crowdsourced studies, participants were given three color patches and asked to order them in some way that seemed intuitive to them. A study screenshot is shown in Figure 34. In some cases, the color patches were selected from neighboring points in the colormap while in others, they were selected from distant points. This allowed us to validate that colormaps might satisfy local but not global order.

While these user studies were not included in the publication, they provided useful insights into how people view perceptual concepts involving color and are an example of studies that can be done to test the mathematical measures from [14] and how those measures align with human perception.

2.4.7 Vector Field Visualization and Compression Studies

In the new initiative described in Section 2.2.3, Ware and Turton are collaborating on a series of experiments to study vector field visualization and compression. An early experiment studied the smallest feature size that could be represented using different rendering methods: standard arrows (where the tail of the arrow is at the data point), centered arrows (where the center of the arrow is at the data point), particle tracers, and a Jobard-Lefer streamline algorithm. As shown in Figure 35, synthetic features were rendered in the four visualization approaches. The spatial frequency of the features was varied. A two alternative forced choice study asked Mturk participants to identify which image had the non-circular pattern. Results from the 2AFC experiment, Figure 36, indicated that both tracers and streamlines allow smaller features in the data to be detected than do the two types of arrow renderings.

Another set of ongoing studies are exploring what levels of compression can be detected by observers, again for different types of flow visualization renderings. Compression artifacts are difficult to detect, both for experts and non-experts alike. A series of pilot studies led to the approach shown in

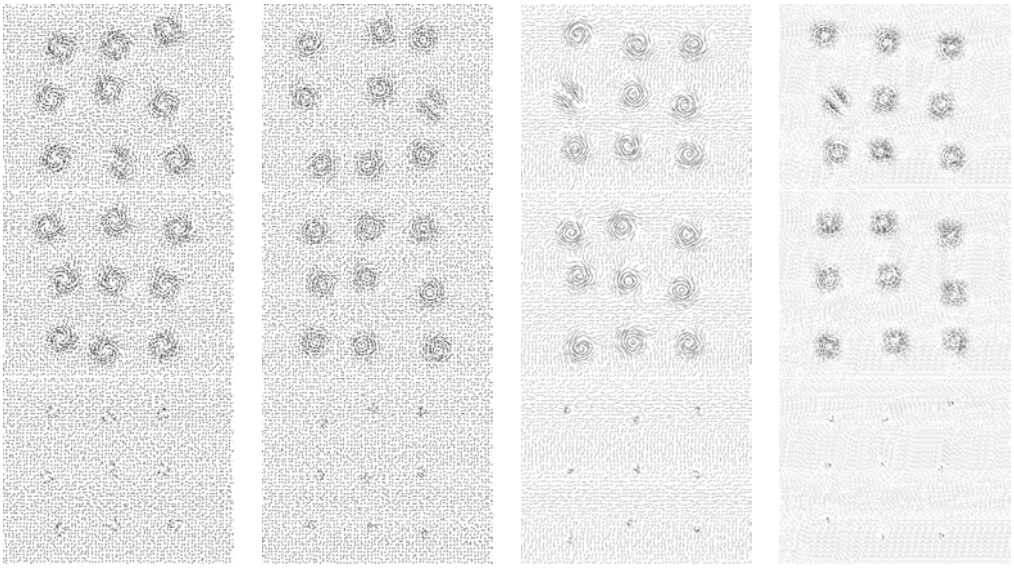


Figure 35: Examples of the vector field visualization renderings (from left to right): arrows, centered arrows, particle tracers, and Jobard-Lefer streamlines. The top row has one of the nine features modified so that it is not a full circle (for the largest spatial frequency studied). The middle row has all nine circles. The bottom row shows the eight-circle stimuli images at the discrimination threshold specific to each of the four rendering methods.

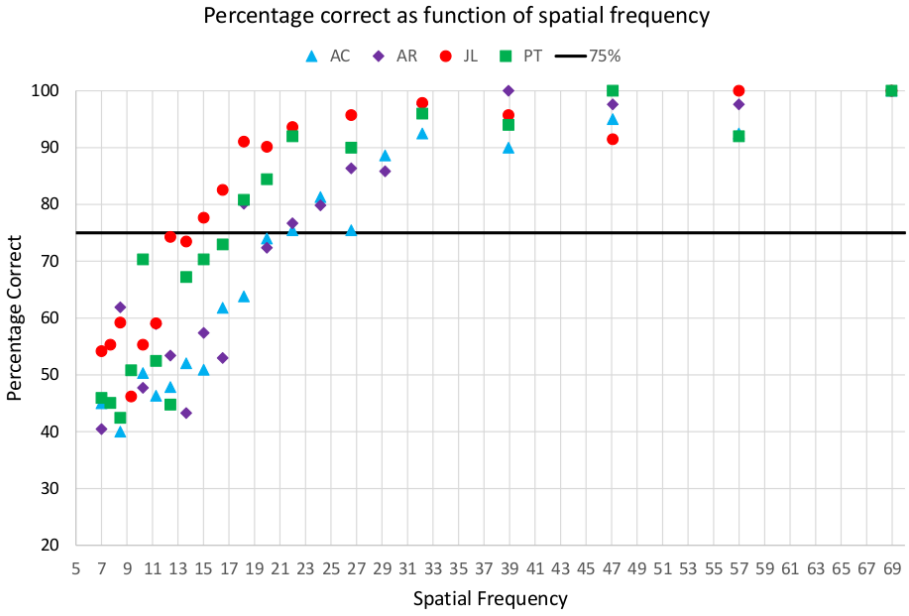


Figure 36: Results from the 2AFC experiments for flow visualization renderings. The percentage of participants correctly choosing the image with only eight circles is plotted as a function of spatial frequency for the four types of flow visualization: arrows (AR), centered arrows (AC), particle tracers (PT) and Jobard-Lefer streamlines (JL). Note how the JL and PT renderings allow much smaller features to be identified.

Figure 14. In this approach, uncompressed MPAS-Ocean (shown in red) is overlaid with compressed data. The JPEG compression level is varied. The experiment was initially crowdsourced in a 2AFC approach, asking participants to identify which of the overlaid patterns showed compression artifacts.

A comparable experiment is underway using a method of adjustment approach to determine the discrimination threshold using participants familiar with flow visualization.

Note that several of the Evaluation Toolkit modules were used in these vector field visualization and compression studies (Section 3.1).

3 ECX Toolkit

As a result of this project, we have created a set of open source tools that can be used to explore and understand the high dimensional space embodied in this project. Details about the tools are found below.

3.1 ETK – The Evaluation Toolkit

ETK, the Evaluation Toolkit, is a JavaScript/HTML/CSS toolkit that embodies standard psychophysical techniques to run perceptual experiments based on images as the visualization artifacts. Using this toolkit, visualization researchers can leverage the psychophysical approaches embodied in the modules. Embedded into Qualtrics survey software, ETK modules streamline the user study implementation process. The resultant study can be used either in a laboratory setting or launched in a crowdsourced approach. ETK has a website, `http://www.etklab.org` [34] with documentation and demos and an online GitHub repository for the ETK modules [33].

Initial development of ETK was in 2016. During this past year, three new modules have been added to ETK along with expanded functionality within many of the modules. The ETK system was presented as a short paper at EuroVis 2017 [32]. There are currently seven modules in ETK:

2AFC 2-Alternative Forced Choice: a module that presents the stimuli images for a 2AFC experiment, each compared to a baseline image.

MoA Method of Adjustment: a module that allows a subject to cycle back and forth through a set of stimuli images and make a choice of a specific image, e.g., at a threshold.

sMoA Sliding Method of Adjustment: a variant on the MoA module, this module instead shows two sequential images from the carousel at a time. This allows the participant to search for a threshold by, for instance, finding the effect under study in one image but not in the other.

RRC Round Robin Comparison module: a module that presents each possible pair of images in an A/B choice experiment.

C2A Compare 2 Arrays: a module that compares each image in an array to its counterpart in a second array. This can be used to present specific A versus B comparisons or as a 2AFC approach where the baseline image varies as a function of stimuli level.

CC Click Counting: a module that counts the number of times the displayed image is clicked.

KT Key Task: a flexible module that displays a series of individual stimuli images along with a set of answer keys; keys can be coded by color or by name/text.

The ETK modules handle any randomizations required by the experimental approach such as image (or image pair) order and left/right appearance. As a check for potentially bad participants, a flag is set when a participant only chooses one side of all comparison pairs. ETK functionality has been extended to allow multiple baselines and multiple images for each stimuli level for relevant modules, appropriately randomized. This allows each participant to see a unique version of the experiment. Where relevant, a user flag determines if a randomly chosen subset of images is shown. An optional time delay flag creates a delay between image presentation – useful to prevent jitter between images from influencing participant choices.

Within ECX publications, ETK's 2AFC module has been utilized in perceptual evaluation of video compression [10, 11]. The Key Task module, shown in Figure 29, was used in validating ECX colormaps while also validating Mechanical Turk as a viable research platform for doing color-based user studies [35]. The 2AFC, C2A, and sMoA modules have also been utilized in ongoing flow visualization research within ECX (Section 2.4.7. Figure 14 is an example of one set of comparison images using the sMoA module. Recently, researchers at the National Center for Atmospheric Research, in a (separate) collaboration with Turton, developed an ETK-based experiment to evaluate perceptual artifacts in compressed climate data.

The Evaluation Toolkit has recently been implemented into a stand-alone browser-based evaluation application, BREF, developed at the Technical University of Kaiserslautern [29]. Moving ETK out of

Qualtrics survey software will hopefully make ETK more accessible to a wider range of visualization researchers. We are continuing to expand the range of psychophysical approaches and modules available within ETK.

3.2 ETH – The Experimental Test Harness

The Experimental Test Harness (ETH) is a lightweight, open-source testing harness that promotes exploration of a variety of analysis and visualization pipelines in many different operations, work distributions, and mappings onto hardware. The toolkit is based on VTK [30], which is a core capability for the analysis and visualization community. VTK implements a data-centric pipeline of operators, filters and rendering operations that operate on data, then pass it along to the next element in the pipeline. There are several important capabilities of the test harness, which are discussed below. These capabilities are the classes of parameters that are varied in the tests run for this paper.

ETH operates on data and does not require code coupling. ETH loads test data from disk, and this has several advantages over other systems. First, we can run the framework on many different science domains, without changing the framework. Because the toolkit is based on VTK, any dataset that VTK reads can be used in tests. Existing toolkits tend to be based on *proxy apps*, which contain simplified versions of the physics of their related simulation. Operating on real data is critical to testing the visualization and analysis operators as simulated data does not generally contain enough complexity to test and exercise interesting analysis and visualization operations.

ETH has easily configurable visualization operations. A critical feature of a useful testing system is a way to modify analysis and visualization operations so that they approximate those that are to be tested. Again, since ETH is based on VTK, many operations can be easily added to the pipelines tested, and they can be specific to the data and visualizations that are of interest. This means that ETH can generate specific visualization products that serve as good proxies for perceptual and cognitive test products.

ETH can run with different process-couplings. Given the number of different hardware and process options available today, it is useful to study different configurations of parallel jobs. ETH can be configured to run pairs of sim proxy and vis proxy processes in three ways: 1) coupled into a single process, or 2) as communicating processes, which enables the processes to run on the same nodes, thereby avoiding any I/O operations, or 3) on different nodes, enabling testing using differing numbers of nodes and on heterogeneous systems.

ETH can run different rendering pipelines. Traditional visualization workflows use algorithms that iterate over the input data to extract intermediate geometrical representations of the data that can then be rendered using OpenGL, which then iterates over the intermediate data to determine each element's contribution to the output image. Recent technical advances [17, 19, 36] make it practical to support raycasting renderers that operate directly on data, avoiding the need for intermediate representations and the memory space they require. Further, in many cases, raycasting can produce significantly better images at lower cost, particularly as datasets get large. Since the choice of visualization back-ends significantly affects a system's overall performance and the effectiveness of the results, ETH can be configured for alternative rendering back-ends as shown in Figure 38, enabling this dimension to be explored as it affects the overall system performance.

3.2.1 Design of the Framework

ETH is a parallel execution framework, but the basic unit of the toolkit is a pair of processes, each of which can perform analysis or visualization operations on real simulation data as shown in Figure 37. The toolkit reads data from the disk and then operates on the data in parallel. An experiment can then be run over different process or node configurations, and different rendering pipelines to produce

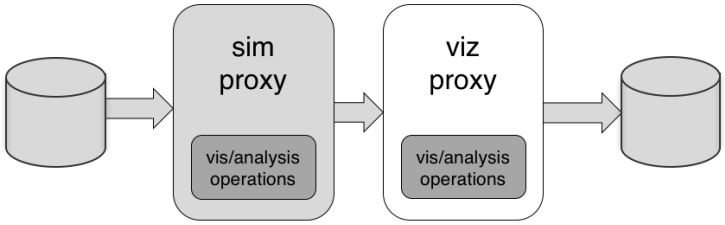


Figure 37: Diagram showing the basic unit of ETH, a sim proxy, vis proxy pairing. Data is loaded from the disk, sent to the sim proxy, and then passed from there to the vis proxy. From there it is written to disk. Visualization and analysis operations can be performed in either place, depending upon the pipeline that is being tested. This means that ETH can compare sets of operations weighted towards either.

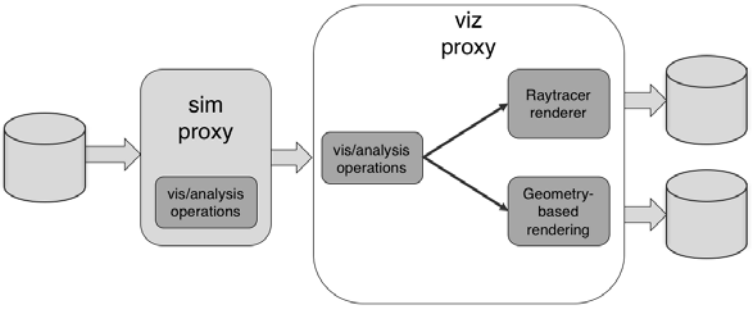


Figure 38: Diagram showing options for pipeline execution inside the vis process. We can send the result of some set of operations to different rendering pipelines. In this case, we can compare a traditional geometry-based rendering approach with a raycasting approach that operates directly on the data, and takes advantage of new software rendering libraries that are optimized for this.

artifact on disk. The system can accommodate a wide range of analysis and visualization operations in either a simulation process or an analysis process, so the toolkit can be customized to operate on a user's specific data, and perform visualizations that are simplified or specifically tuned to the user's work flow.

Modes include unified (single process), core-to-core, and node-to-node. There are two rendering pipelines supported as well – raycasting (non-geometric) and geometry-based rendering.

3.2.2 Execution Details

As described above, ETH runs with pairs of simulation and visualization proxies coupled together into one process, or running in separate processes and communicating via the socket layer. In the first case, experiments are easily run using the standard batch scheduler.

When the simulation and visualization proxies run in *separate* processes, the two sets of proxy processes must exchange information that enables the visualization proxy processes to connect to their paired simulation proxy processes. This is done by starting the parallel application in two steps: first, the simulation proxy application is started. Each process of the application then adds its assigned IP address and port number to a globally accessible layout file, then opens its port and waits

for connection. The visualization proxy application is then started. Each process of the application then references the global layout file, determines the location of the simulation proxy(s) it will receive data from, waits for the corresponding port to open, and then establishes the connection.

When the simulation and visualization proxy applications run on the same nodes, this is again easy to start: the job script simply begins the simulation proxy application first, in the background, and then starts the visualization proxy application in the foreground.

When the simulation and visualization proxy applications run on separate sets of nodes, several alternatives exist depending on the mix of nodes required; on homogeneous systems, a single job allocating the total number of nodes required, and MPI arguments can be used to start the two parallel processes offset from one another. When heterogeneous collections of nodes are desired, it will be up to the scheduling system to arrange for two separate jobs to be started concurrently.

3.3 ECX Estimator

In this section, we present a simplified version of the model we use to estimate the performance, energy, and storage of the visualization pipelines at different sampling rates and application configurations. The actual model itself is codified and available at https://github.com/ascr-ecx/ecx-estimator.

Table 5: Summary of symbols used in our model

E	Total energy consumed by the visualization pipeline
P	Average power consumption for the visualization pipeline
t	Total execution time for a visualization pipeline
t_{sim}	Time taken by the simulation phase
$t_{i/o}$	Time taken by the I/O phase
t_{viz}	Time taken by the visualization phase
$S_{i/o}$	Size of output (in GB) produced by the simulation
N_{viz}	Number of images produced by the simulation
α	Time cost to write 1 GB of raw data; value estimated by linear solver
β	Time cost to generate one image corresponding to one timestep; value estimated by linear solver
$iter_{ref}$	Number of iterations or timesteps in the reference configuration
$iter_{any}$	Number of iterations or timesteps performed
$rate_{ref}$	Output sampling rate used in the reference configuration
$rate_{any}$	Output sampling rate for which performance metrics must be estimated
$t_{sim.ref}$	Total execution time of the reference configuration
$S_{i/o.ref}$	Size of output produced for the reference configuration
$N_{viz.ref}$	Number of images produced for the reference configuration
$S_{i/o.any}$	Estimated size of output produced by any given configuration
$N_{viz.any}$	Estimated number of images produced by any given configuration

The energy consumed by a visualization pipeline can be expressed as the product of its average power and total execution time.

$$E = P \cdot t \tag{1}$$

Our experiments showed that the average power across all sampling rates can be considered constant. We need only model the execution time of the application, which can be expressed as the time taken for the simulation, I/O, and visualization phases:

$$t = t_{sim} + t_{i/o} + t_{viz} \tag{2}$$

Here, the time take for the simulation phase, t_{sim}, is a constant for a given number of timesteps or iterations of the simulation. The time taken for I/O and visualization phases are dependent on the amount of data written and the number of images visualized, which in turn are dependent on the

sampling rate. Mathematically, we can express the time taken for an application as

$$t = t_{sim} + \alpha S_{i/o} + \beta N_{viz} \tag{3}$$

in which α and β denote, respectively, the time taken to write 1 GB of output and to produce one set of images corresponding to one timestep. $S_{i/o}$ and N_{viz} are, respectively, the total output size and number of images. These values can be estimated easily for any sampling rate given a reference point as they are linearly dependent on the sampling rate.

We can express Equation 3 in a more generic form:

$$t = \frac{iter_{any}}{iter_{ref}} \times t_{sim.ref} + \alpha S_{i/o} + \beta N_{viz} \tag{4}$$

That is, the simulation time will scale with the number of iterations or timesteps in the simulation.

We use a linear solver to estimate the values of α and β. The data collected from three different configuration points, namely, (i) in-situ, once every 8 hours, (ii) in-situ, once every 72 hours, and (iii) post-processing, once every 24 hours, was used to solve for α and β. Alternatively, regression techniques may be used to solve these equations using the following system of equations:

$$\begin{aligned} t_{sim} + 0.1\alpha + 60\beta &= 676 \\ t_{sim} + 0.6\alpha + 540\beta &= 1261 \\ t_{sim} + 80\alpha + 180\beta &= 1322 \end{aligned} \tag{5}$$

Solving this system of equations gives the following values: t_{sim}=603 , α=1.2, and β=6.3. That is, it takes 603 s to perform the simulation (for six simulated months), 1.2 s to produce one image and 6.3 s to read/write 1 GB of data.

For the actual implementation, we used a similar empirical approach and increased the parameters used in the model. The parameters covered includes rendering algorithm, number of time steps, number of cameras per time step, temporal sampling rate, spatial sampling rate, work distribution/job layout scheme. While the model presented in this section is specific for MPAS-O, in the actual implementation we can study different applications by simply including a configuration file for different applications with different equation co-efficients.

3.4 ECX Inspector

The ECX Inspector allows users to explore the results of experiments in ETH, and to estimate how parameter settings could impact an upcoming run. Visualizing this is best achieved as an interactive exploration of high dimensional spaces.

3.4.1 Visualizaiton of High Dimensional Spaces

The ECX project experiments create a high dimensional design space that can be difficult to understand and explore. Dependencies between input variables, as embodied in the ECX Estimator (see Section 3.3), can be difficult to discover. One option for visualizing this is traditional methods for looking at high dimensional data, such as the scatter plot matrix, as shown in Figure 39. However, as we are sampling these parameters at regular intervals, there is little of interest in such a plot, except for power and energy values, as shown in the lower right corner of the visualization. In addition, early feedback from users on visualizations such as this led us to pursue a different, more interactive approach to visualizing the data. We turn to our research on Cinema as a method of viewing large image databases, and interactively exploring high dimensional data spaces [43].

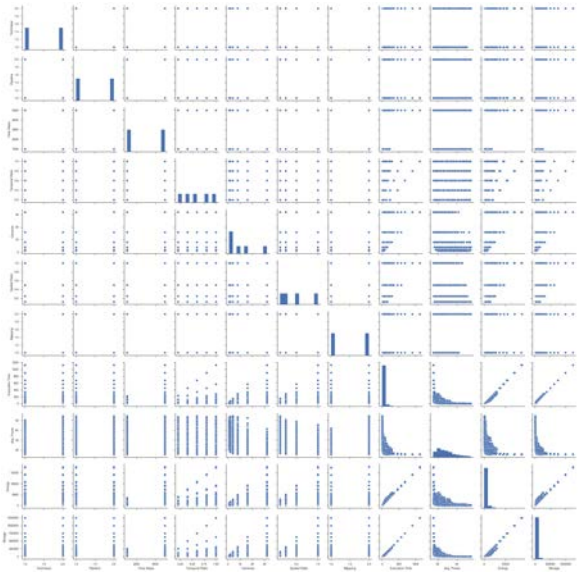
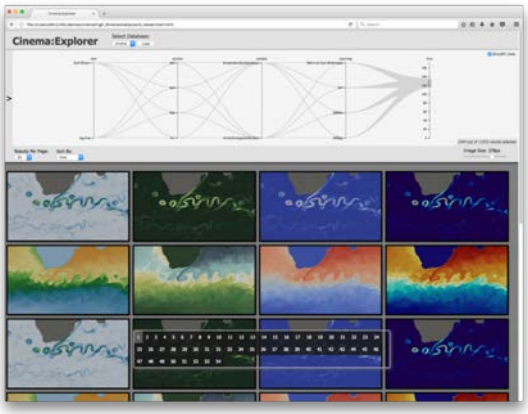


Figure 39: Scatter plot of a range inputs and outputs for the ECX estimator script. The estimator was run over a wide range of values, in order to study the behavior of and entire high dimensional space. We note that such traditional visualizations of ECX data do not afford good inspection of the high dimensional data, in part because we are sampling over fairly regular samples of each variable. Instead, we pursued a Cinema-based approach to high dimensional visualization, incorporated into the ECX Inspector, to promote interactive exploration of the entire space.

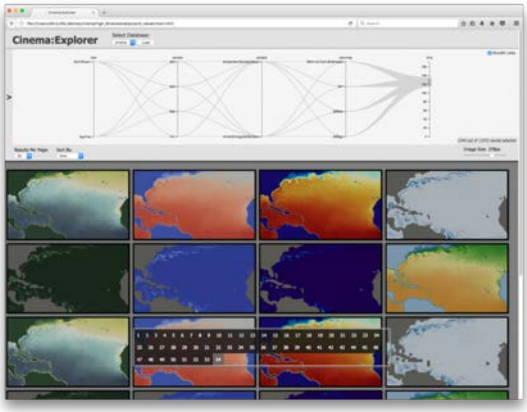
3.4.2 Utilizing Cinema-based Viewers

If our ECX parameter space is a vector of input values paired with an image (showing and example of a rendered visualization), we can cast that data into a Cinema Spec D database [44], and explore it using current viewers. We see the results of this in Figure 40. The Spec D viewer promotes high dimensional data exploration through the use of an interactive parallel coordinates viewer. The user can select ranges of variables to explore on each axis, and the viewer shows all results that match the set of parameters. Thus, the user can examine many different results simultaneously, and compare the impact of different parameters on the end results. This is a useful way to explore high dimensional parameter spaces quickly, and is useful in exploring the results of ECX experiments.

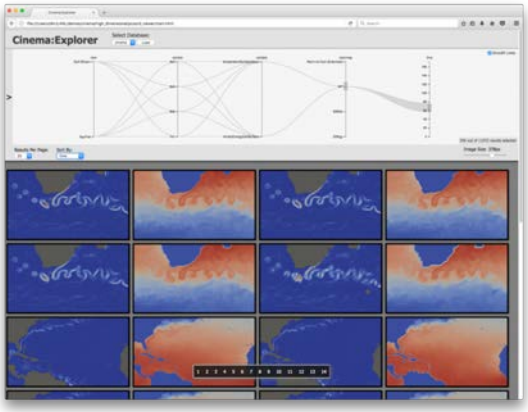
(a) Small images comparing a small set of timesteps, and concentrating on the Agulhas flow.



(b) Larger images, with different subset of timesteps chose than those in (a).



(c) Comparing a different view, this time of the Gulf Stream.



(d) Restricting colormaps and comparing them on the Agulhas flow.

Figure 40: Several views of a high dimensional ECX Cinema database, showing how a user can easily compare sets of results, and interact with the parallel coordinates selector to understand how parameters impact results. These results show a variety of visualizations that can be achieved with different parameters for MPAS Ocean simulations.

However, we can do better by providing a method of conducting *what-if* experiments comparing more than one result.

3.4.3 Exploring What-if Scenarios

Figure 41: Screen capture of one prototype of the ECX Inspector.

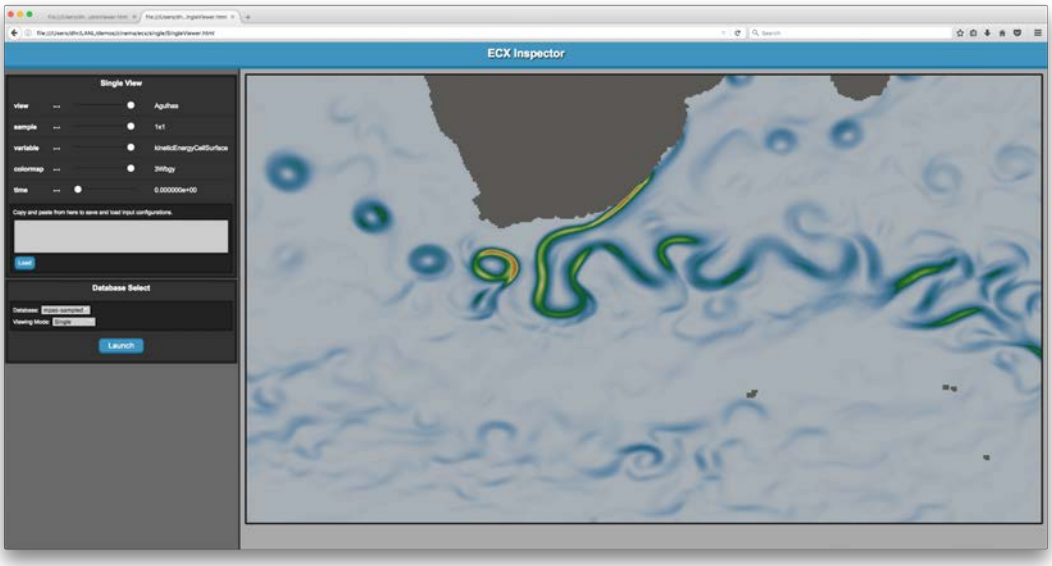


Figure 42: Screen capture of one prototype of the ECX Inspector.

We developed a Cinema-based viewer that promotes exploration of the ECX parameter space, in conjunction with the output of the *ECX Estimator*. This provides a scientist with a method of comparing the visual results from a set of parameters, in the context of an estimated measure of the power and disk space that will be used in creating the final result. Because this is an image-based approach (through Cinema), this is a general method of exploring visualization and analysis spaces that can be used across science domains, data types, and visualizations.

We developed two viewers to explore these data spaces. A single database viewer, shown in Figures 42 and 41, and a double database viewer, shown in Figure 43. The double database viewer includes an implementation of the ECX Estimator, so it also includes a visual comparison of estimates

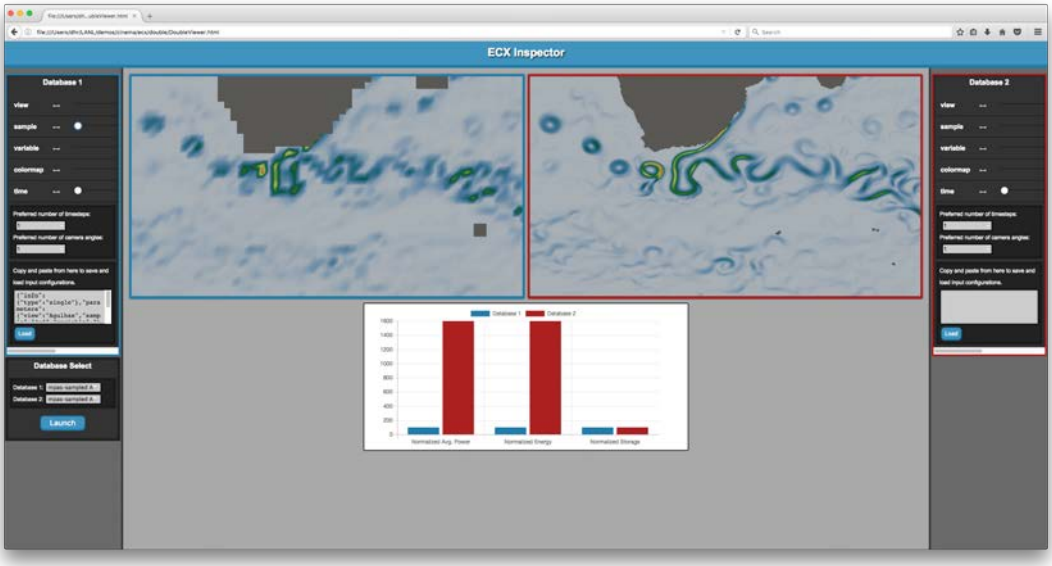


Figure 43: Screen capture of one prototype of the ECX Inspector. This viewer allows the user to explore two different databases, or different views of the same database, and change parameters such as sampling strategy, view, colormaps, and varibles - all parameters in ETH experiments. The viewer includes a visualization comparing the power and disk usage for the two approaches.

for the power and disk usage for each approach. Each viewer has several different modes that allow comparison of the data in different ways. Examples of modes supported are:

1. Comparison of small multiples. The single inspector allows the user to show groups of results, allowing comparison of many different examples that differ by only one parameter. For example, the user can compare several sampling strategies at once.

2. Search by query. One mode of the single viewer allows search of the entire parameter space with a simple query language.

3. Side-by-side interactive comparison. The double viewer supports side-by-side comparison of the same database, or two different databases.

4. Interactive visualization of data subsets with a parallel coordinates selector. The Cinema Spec D viewer promotes direct interactive visualization of the high dimensional space.

The ease of developing different viewers for Cinema databases means that this is a rich area for further study, and that viewers can be customized to address specific needs of specific communities. We feel this is a rich area for further study. This software is currently under review in LANL's open source release process, and will be released when that is completed. Further information about the work is available in the user documentation for the software.

3.5 SciVisColor & ColorMoves

The SciVisColor website [27] brings together many aspects of color developed through the ECX project. The colormaps found on `http://www.sciviscolor.org`are available in an open source repository [21]. The features of SciVisColor and associated ECX publications are detailed below.

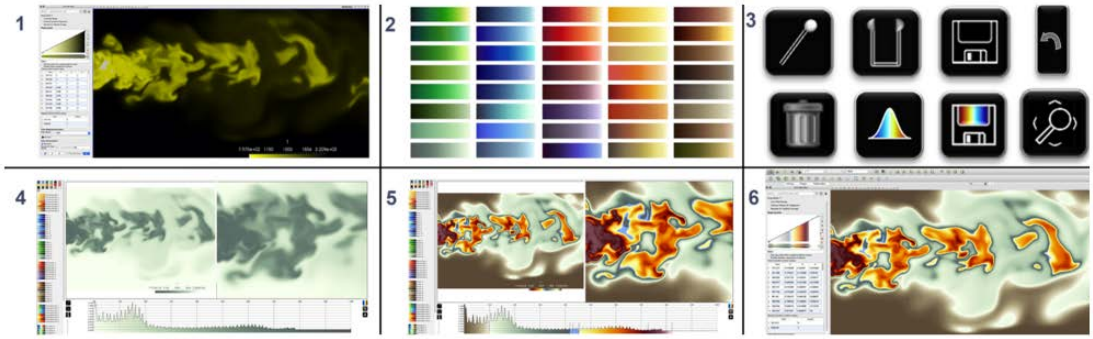


Figure 44: Steps for constructing a colormap in ColorMoves: (1) a data float file is created in ParaView; (2) hue families; (3) control buttons; (4) shows two images in the viewer using the horizontal multiple view option. (5) illustrates the development of a data-driven color encoding: four pins and one nest (in periwinkle) are added with two muted saturation colorscales. The high color contrast yellow to red colorscale highlights the data of interest to the scientist. An additional pin at the right end of the data histogram allows a discrete white to be inserted to customize the data range. (6) This adaptive color encoding can be exported for import back into ParaView.

3.5.1 ColorMoves

ColorMoves is a browser-based colormap development tool that allows a scientist to interactively build detailed color encodings that align with the statistical distribution of the data, data ranges of interest, and tasks of the visualization. The intuitive and immediate interaction enables scientist to explore and communicate their data with a higher level of effectiveness, flexibility, and ease. First introduced at SIGCHI'15 [26], ColorMoves has been exploited in a range of publications since then and added functionality has improved its usefulness to the scientist.

Figure 44 provides a visual overview of the ColorMoves user interface. Using a data float PNG file as input, ColorMoves employs a drag-and-drop approach starting with a suite of predefined colormaps. There are three distinct sections of the UI: the input section, the output section, and the colormap picker section. The lower section of the interface is the input section and is the workspace for colormap assembly. The upper section of the interface is the output section; it shows the visualization with the current colormap applied. The colormap picker section acts as a toolbox for the input section, and a toggle button brings it into view.

A colormap is constructed by defining areas of interest within the data with two different insert tools: pins or nests. Individual colormaps can be selected and dropped into the data ranges defined by the insert tools. Colormaps can be flipped and cropped as needed (e.g., to avoid multiple dark or light regions across the full color encoding). Other useful features include the multiple view options. Multiple data files can be added to the output section and can either be viewed sequentially or viewed together through splitting the screen, as in Figure 20. Either approach allows the user to assess how a constructed colormap will perform across multiple time steps or varying spatial regions. The resultant color encoding can be exported either in JSON or XML format for input into common visualization tools. Additionally, a python script is available under the ColorMoves tab to convert the color encoding to a matplotlib compatible object.

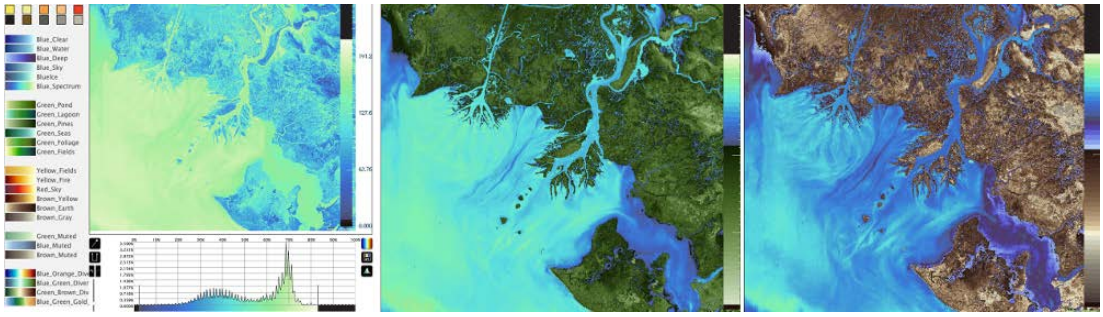


Figure 45: An example use of *ColorMoves: The Environment*. Sentinel2 CIR satellite data of the Wax Lake Delta is pseudo-colored using colormaps from the sets of intuitive colormaps geared towards environmental visualizations. By controlling the data ranges and a wide selection of intuitively colored colormaps, ColorMoves:The Environment enables the scientist to precisely define the features of interest in the visualization.

3.5.2 ColorMoves: The Environment

Visualizations benefit from the use of intuitive colors, enabling an observer to make use of more automatic, subconscious channels. In this research, we applied the concept of intuitive color to the generation of thematic colormaps for the environmental sciences. Custom sets of colormaps for water, atmosphere, land, and vegetation were integrated into a domain-specific version of ColorMoves, *ColorMoves: The Environment*, to enable the environmental scientist to develop colormaps geared toward their data. An example use of this version of ColorMoves is shown in Figure 45. Full details can be found in the paper presented at the EuroVis Workshop on Visualisation in Environmental Sciences [28].

3.5.3 Color Resources

Resources to aid the scientist in developing data-specific colormaps can also be found in the other sections of SciVisColor.org. These include:

Color Sets Sets of analogous colormaps and/or discrete colors that work well together.

Color Strategies Recommendations for specific types of data and task.

3D Color Strategies Recommendations for 3D data.

Gallery Images of many different visualizations developed through the ECX project that provide ideas and inspirations for other users.

3.6 ColorMeasures

The theoretical framework paper [14], described in Section 2.3.7, develops a set of mathematical definitions and corresponding mathematical measures for three of the most common colormap design rules: uniformity, discriminative power, and order. These definitions and measures are given for both local and global interpretations of the design rules. These measures can be used to directly assess and compare colormaps.

In order to facilitate comparing colormaps and informing decisions on choice of colormaps, the measures for colormap assessment and the visualizations from the paper have been implemented in an online tool at `http://www.colormeasures.org` [13].

The ColorMeasures viewer, shown in Figure 46, takes a JSON colormap file (with RGB values) as input and reproduces the measures and visualizations developed in the theoretical framework paper. These are shown in three common color spaces: CIELAB76 (ΔE_{76}), CIEDE2000 (ΔE_{00}), and CIECAM02 (ΔE_{02}). Additionally, the measure values are output into an HTML form where they can be easily copied for further offline study and comparison.

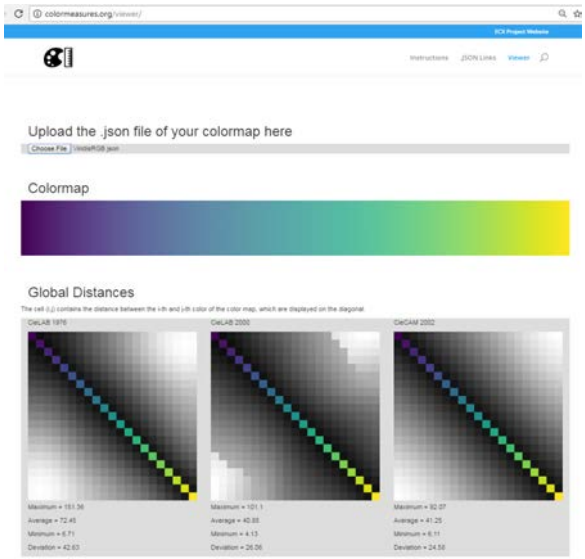


Figure 46: A screenshot of the ColorMeasures colormap assessment viewer. In this example, Viridis has been loaded via the "Choose File" button. The viewer starts with a image of the full colormap and then presents the various global and local measures along with an explanation for each measure. Note the differences between the three color spaces for the global distance measure. Since Viridis was designed in CIECAM02, the global distance deviation is lowest in that color space.

Output from the tool consists of: An image of the colormap and a set of local and global measures for the colormap (see paper for full details). The measures are calculated using one of the metrics ΔE1976, ΔE2000, or the Euclidean distance in CIECAM02-UCS. The measures are the following.

- The **average speed** correlates with the **discriminative power** – higher local/global speeds correspond to higher local/global discriminative power.
- The **standard deviation** of the speed anti-correlates with the **uniformity** – lower deviation in local/global speed correspond to higher local/global uniformity.
- The **minimal speed** indicates the **invertibility** and therefore the **legend-based order** – the colormap suffices local/global legend-based order if the minimal local/global speed is strictly positive.
- The **minimal triangle side difference** indicates the **intuitive order** – the colormap suffices local/global intuitive order if the minimal local/global triangle side difference is strictly positive.

By understanding the properties of a colormap, the user can make a more appropriate colormap choice for specific data and tasks.

3.7 Parallel Volume Visualization

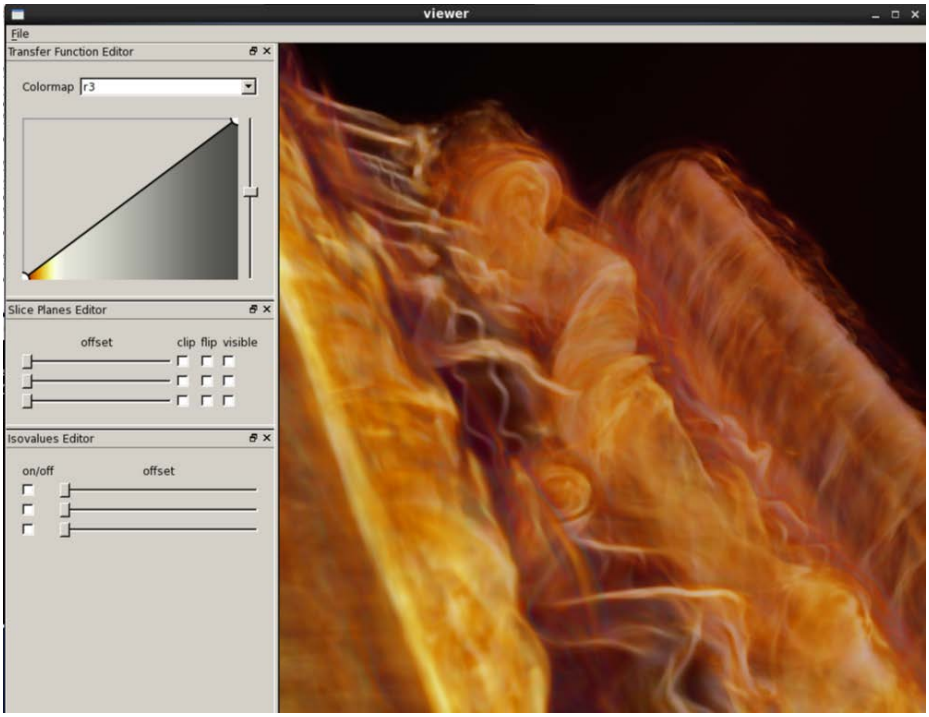


Figure 47: VolViewer Vierer showing real-time visualization of data from a simulation of magnetic reconnection in relativistic plasmas. The data is courtesy of Bill Daughton (Los Alamos National Lab) and Berk Geveci (Kitware) [16].

A core aspect of the ECX project is to evaluate different visualization algorithms for use in exascale computational science in terms of their cost in both time and energy. We have identified *direct volume visualization* as an alternative to traditional geometry-pipeline based, sort-last visualization systems for use in-situ in large-scale simulation. To do so, we have developed **PVOL** to visualize large distributed data and compared its performance in time and energy utilization against traditional methods.

In the earlier phases of this grant, we developed **VolViewer** – a visualization tool based on **OSPRay**, an Intel raytracer. VolViewer was designed to run on a single node, and demonstrated that doing visualization of structured data directly, without the creation of intermediate datasets, had advantages in performance and memory footprint which would impact the energy utilization of energy in the visualization of large datasets.

In the latter phase of the project we have extended VolViewer to run on distributed systems so that it can be evaluated as a tool for in-situ visualization of very large simulations. Our system, now known as **PVOL**, takes advantage of the locality of data in distributed-memory system as a basis for parallel processing; since data for subregions of the computational space will reside on each node, we assume that *all* data for each subregion (including both raw and derived data) will reside on the same node and thus the intersection a ray with the computational space can be traced forward, traversing each encountered subregion in turn independently, without regard for other subregions until the ray (or secondary rays arising from the intersection of primary rays with surfaces) reaches subregion boundaries, at which time rays can be communicated to the neighboring subregion.

PVOL is further specialized for in-situ visualization of large-scale simulations by minimizing the need for expensive system synchronization. Traditional systems are constrained by OpenGL-based sort-first or sort-last parallel architectures that necessitate synchronization for each individual frame of a timestep (which may be on the order of thousands). Instead, PVOL is designed to render all the necessary frames simultaneously, synchronizing only to when all results are complete enabling

control to return to the simulation. This capability is made possible by minimizing derived data – isosurfaces generated for a range of isovalues, for example – that are required by a Cinema database.

In the remainder of this section we will present an overview of the PVol architecture.

3.7.1 Raytracing on Distributed-Memory Parallel Systems

There are several approaches to computing the intersection of a ray with data distributed across multiple nodes of a distributed-memory parallel system. Perhaps the simplest approach is to utilize a sort-last algorithm that first renders data locally on the parallel nodes followed by a depth-based distributed compositing step to resolve visibility among results to produce a final image. In effect, this approach replaces simple rasterization on each node with *local* raytracing – that is, raytracing with regard to only the data present on the local node. This approach forgoes full global illumination effects since data rendered on a node will not be affected by data on other nodes (*other* than simple visibility, but precluding shadowing, ambient occlusion, reflection, and refraction effects).

Alternatively, approximating full global illumination requires that every ray – both primary "camera" rays as well as secondary illumination rays for shadowing, ambient occlusion, reflection, and refraction – be traced with respect to *all* the data. This is apparent in both the canonical recursive algorithm for ray tracing first devised by Whitted [42] in which the energy returned to the viewpoint from a point on a surface is computed by replacing constant terms in the Phong lighting model with results derived by tracing rays recursively in the direction of the light source (for shadowing), in the reflective direction for for reflection, and from behind (potentially bent by a refractive index) for transparency/translucency and in the Rendering Equation of Kajiya, in which lighting from a point on a surface is computed as an integral of light arriving at the point from all directions.

Tracing a ray through a distributed-memory model can be done by making any data required for a ray available to processor responsible for tracing the ray, and the rendering of a frame can be parallelized by dividing the image plane into patches that are the responsibility of different processes ref Carson/Ingo. A caching strategy can take advantage of both spatial and temporal *coherence*, the fact that rays traced for nearby samples in the image plane or from nearby viewpoints in successive frames will follow similar paths through the data space, so while retrieving data for one ray may be expensive, the cost is likely to be amortized across multiple rays and, further, that successive frames in an interactive or movie sequence will likely involve the same data.

This strategy is particularly effective for interactive rendering when there is sufficient memory available to support a large resident set on each rendering node; in the extreme, over time, it devolves to copying the entire dataset to every node that responsible for rendering a screen patch. However, the proportion of a data set that might be visible in an arbitrary view is unknown, and the overall data size of a large simulation is likely to be both substantially larger than the memory available on any single node and very expensive to move.

Alternatively, we can move rays from process to process in the distributed system as they traverse the data space. As the ray is traced, global knowledge of the data layout is consulted to determine which node contains the first data that the ray is likely to intersect, and the ray is passed to that node for processing. If the ray is not retired by the data on that node, a new destination node is determined and the ray passed on. Some systems (for example, ref Paul?) further optimize this by passing rays to the nodes containing the next several datasets that this ray *might* intersect, comparing the results to choose the correct result.

This algorithm alleviates the memory pressure on the system nodes, in that the additional memory required on each node is capped by the number of rays required for the rendering process, known from a) the resolution of the image plane, b) the number of images being rendered, and c) the recursive depth of the ray-tree required by the lighting model.

This strategy is particularly effective in cases of limited coherence. In a highly coherent case, the many rays necessary for rendering an image will follow a very similar path through the data space, burdening the same set of processors at the same time and in the same order. However, when rays diverge substantially the work will be distributed more evenly across the system nodes.

3.7.2 Raytracing for In-situ Visualization

In-situ visualization in large data-parallel simulation systems has several properties that differentiate it from typical model-based raytracing.

- **Locality** The spatial subdivision of the computational space used to parallelize the simulation ensures that the data products that a visualization renders for a particular partition derive solely from data from that partition, so rays can be traced through the partition's space without concern that data on other nodes will intercede.
- **Volume Rendering** An increasingly important visualization technique is *volume rendering*, in which opacity and emissitivity, derived by applying a transfer function to data along a ray, is integrated along viewing rays. Unlike simple surface raytracing in which we search for intersected surfaces geometrically, volume rendering requires iterating along the ray evaluating highly coherent samples of the data.
- **Implicit Surfaces** Several of the most important visualization techniques performed locally using *implicit surfaces* rather than explicit geometrical models describing, for example, isosurfaces and cut surfaces. This can significantly reduce the memory needed for intermediate datasets.
- **Parametric Models** Many important models can easily be represented parametrically, rather than explicitly as sets of geometrical elements (triangles). In particular, high-quality spheres can be rendered at minimal cost from a specification of a centerpoint and a radius rather than as sets of triangles sufficiently large to avoid polygonal artifacts. Similarly, streamlines can be represented as paths and radii.
- **Cinema** In-situ visualization requires that the set of of visualization products to be produced at each time step be pre-defined. To mitigate the possibility that a desired result is missed, *many* products are generated for each timestep of the simulation. In particular, a timestep may be viewed from many individual viewpoints, using many different isovalues, slicing planes and transfer fucntions. The number of images for each timestep may easily range into the tens of thousands. Unlike simple frame-by-frame rasterization algorithms, all these images may be launched simultaneously, resulting in a workload that is globally incoherent for load balancing yet locally coherent for optimal utilization of data-parallel hardware such as vector-processing units.

3.7.3 Additive Rendering for Highly Parallel Raytracing

The results of a lighting model (such as Phong) determined once the necessary parameters are available – the calculation of the energy returned from a point on a translucent surface depends on whether the light source is occluded and what is transmitted through the surface from behind. In the case of recursive ray tracing, this involves the results of forward tracing of rays from the surface to the light source and onward into the scene before the results of the lighting model can be computed. This could be performed in a distributed fashion by casting rays (potentially to other nodes) with return addresses, waiting for the results, then evaluating the lighting model and passing the result back up the ray tree.

Instead, we can formulate the lighting model as an additive combination of the results of the secondary rays. In particular, a simple lighting model for a point p on a surface including an approximation with local occlusion plus diffuse energy from a set of potentially occluded infinitely-distant light sources would return energy from a point on a surface as:

$$I_{total}(p) = I_{ambient}(p) + I_{diffuse}(p)$$

where

$$I_{ambient}(p) = \sum_{i=1}^{n} \frac{K_a}{n} Op(p, r, d)$$

for K_a, the total ambient light energy and $Op(r, d)$ is the opacity of an interval of length d in a random direction r, and

$$I_{diffuse}(p) = \sum_{i=0}^{n} K_i Op(p, L_i)(N_p \cdot (L_i - p))$$

for K_i the intensity of light source i, $Op(p, L_i)$ the opacity along the ray segment from point p to light source i, and N_p the surface normal at point p.

Adding more terms to this simple model, we can add include more advanced techniques such as reflectivity, surface translucency and gradient-based volume shading. Further, this model can be recursive; a contribution from behind a translucent surface or from a reflection can simply be added in. In this form, the correct result for any primary ray is simply the sum of the contributions of the primary ray itself and the properly attenuated secondary rays.

This approach enables us to devise an algorithm that does not require return addresses for the results of ray evaluations, nor the bookkeeping necessary to determine when all the necessary results are available before the lighting model can be computed and the results passed up the ray-tree for use in higher-up lighting calculations. Instead, we attach a pix While this does not support all the effects required for full-featured photorealistic rendering, it supports the simple Phong or Torrance-Sparrow models, shadowing, translucency etc. that predominate in scientific visualization.

3.7.4 Conclusions

Based on these observations, we have developed a distributed architecture for raytracing for scientific visualization. Our basic principles are:

Pass Rays, Not Data PVol strictly passes rays from node to node as the ray traverses the data space.

Raytrace Locally We assume that rays can be processed independently within data partitions, and involve other partition's data only when the ray passes from partition to partition.

Accumulate Pixels All rays (primary and secondary) are tagged with the pixel that gave rise to the initial primary ray, and carry the necessary information to determine an additive contribution to that pixel as they traverse the global data space.

Parallel Frames Many independent visualizations should be computed simultaneously.

3.7.5 PVol Architecture

Application Generally speaking, PVol is a distributed application based on the ability of any node of the system to request that work be performed on any other node or, in fact, all nodes. This capability is provided by an instance of class *Application* on each node.

The Application architecture is pure peer-to-peer: any process in a distributed system can request that arbitrary work be performed on any other process or, in fact, on *all* the other processes. Nonetheless, it does not preclude a master/slave utilization in which one process acts as the primary conduit for communications with the outside world. One possible use would be a simple viewer, in which one node manages the user interface whle all nodes participate in rendering. In this section we will present an overview of the simple viewer application.

We assign the role of managing the user iterface to the root node, which receives interactive input from the user, determines changes in the camera (or other parameters, such as isovalues), requests the distributed system to render a new visualization, participates in the rendering and, when the result is available, posts the result to the viewer window. The remaining processes simply wait for requests to do something.

In this example, the root node will respond to user-interface requests to load data. This may be done in several ways, notably by instructing all the participating nodes to load their portion of the dataset from the file system, or by passing instructions enabling each node to attach to data made available by a concurrently running simulation. Once data is available, the user interface enters an interactive loop updating the view window based on user input.

Work-Request Framework PVol is built on top of a framework that enables any node to request that one or more other node perform work. Such work requests are instances of subclasses of class *Work* that wraps the task to be performed and any accompanying data. Objects inheriting from Work virtualize data serialization and deserialization and have the ability to send themselves to other processes where they will be deserialized and their action performed.

Distributed Data Objects PVol incorporates a distributed data name space to represent logically single objects across the distributed system. These distributed objects are linked by a unique object identifier that enables each node to locate local data corresponding to a global name. Examples of such objects include cameras, which are identical on every node, and volume data, which is a distributed partitioned data object with one constituent partition on each node. We also represent partitioned volumetric datasets in this manner; each node will have its subset of the dataset available through its common identifier.

Renderings Each individual rendered frame requires parameterization, including both the camera to be used as well as visualization parameters such as isovalues and transfer functions, and a destination for pixels. These are referred to as *renderings*. Renderings are distributed objects and are known to all processes.

RenderingSets In order to support the parallel rendering of many frames, we group *renderings* into *rendering sets*. A rendering set groups a (potentially large) set of renderings into a single task; the rendering process actually is performed on rendering sets and synchronization (in effect, notification of completion) is done at the rendering set level. Rendering sets are distributed data objects, known to all processes, and refer to renderings using their global names.

3.7.6 The Rendering Algorithm

Primary Ray Generation Rendering takes place when a request is broadcast to render a rendering set. Upon receipt, each process iterates through the rendering set's renderings, generating primary rays whenever it determines that its partition is the frontmost for some subset of the image plane, seen from the rendering's camera location. These primary rays are then aggregated into *ray lists* that are initially traced locally.

Local Ray Processing Ray lists are processed locally by tracing them through local data. The action of processing a ray list modifies the rays based on what occurred as they traversed the local data; rays may encounter opaque surfaces and terminate, accumulate opacity and emissivity when volume rendering and/or exit the partition through the far side from the entry point.

Local ray processing proceeds in two steps: by first determining whether each ray intersects any surfaces within the local data space (either implicit, such as spheres, tubes, cutting planes etc that are described parametrically or as explicit geometry). If so, the potential depth range f the ray is truncated to by that surface, and further local processing addresses only the portion of the ray lying in front of the surface intersection.

Once this active range is determined, subsequent processing varies by whether isosurface or volume rendering is to be performed. If so, the active range is traversed in small steps (determined by the sampling rate of the underlying data set), integrating for volume rendering and/or searching for isovalue crossings for isosurface rendering.

Once the input rays are traced locally, the system must determined what to do with the results. Several possibilities exist, based on the type of ray and the local result. The following provide examples of further ray processing.

Note that while these may result in many logical messages being transmitted, PVol aggregates these messages by destination to minimize actual message traffic.

Surface Strikes When a *primary* ray strikes a surface, the ray's action depends on the opacity of the surface and on the lighting model being used. If shadows are enabled, secondary rays

toward the light sources are cast, carrying the potential diffuse contribution that the light may add to the resulting pixel; otherwise, a fixed diffuse component is added locally based on the light source. Similarly, if ambient occlusion is enabled, a set of AO rays are cast in random directions; otherwise a fixed ambient contribution is added locally. If a non-zero local contribution to the final pixel is present a message is sent to the destination pixel, residing on wichever node is responsible for aggregating contributions for the current rendering. Finally, if the surface is not opaque, an attenuated ray is cast forward to determine what comes from behind.

If a *secondary* ray strikes a surface, it is attenuated by the opacity of the surface. If its attenuation reaches zero (for example, a shadow ray striking an opaque surface) it is simply dropped. If the attenuation is still non-zero, the ray is re-queued for local processing.

Accumulation of Volume-Rendering Opacity If a primary ray accumulates enough opacity to completely occlude anything farther along the ray, a message is sent to the appropriate destination from the buffer to accumulate the result.

Exiting the Partition If a ray exits the process' partition space, then it is passed to the neighboring partition (if there is one). If not, and if the ray is a *secondary* ray (eg. a shadow or AO ray that doesn't get occluded), its current contribution is sent to the destination frme buffer.

3.7.7 Completion

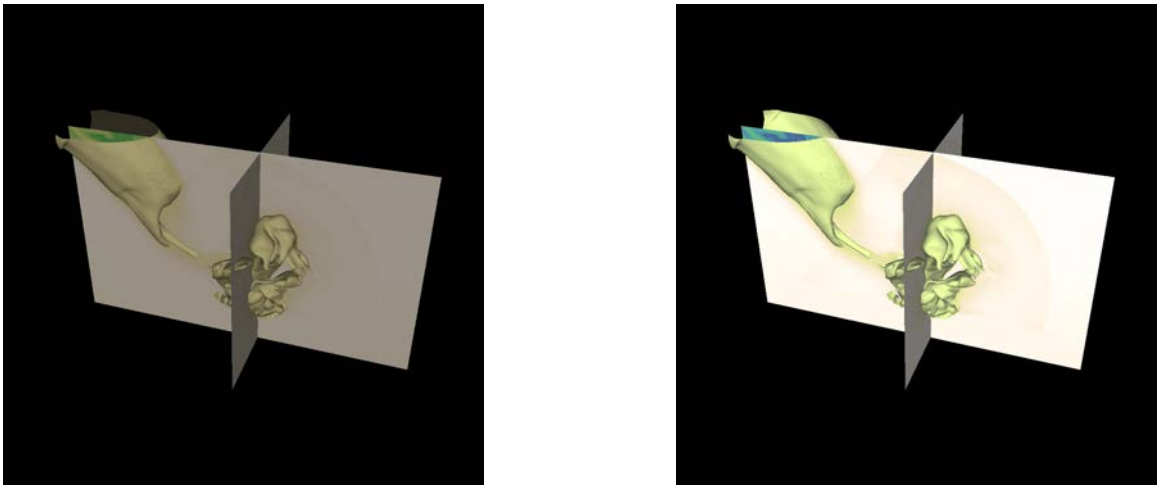
In most cases, it is necessary to determine when the processing of a rendering set is complete so that results may be passed to consumers (for example, posted to a viewer window or written to a Cinema database) and so that resources can be released (for example, allowing an attached simulation to proceed, thereby overwriting shared data space).

Completion occurs when all rays (primary and secondary) for a rendering set have been retired. An approximate completion test is performed using an asynchronous tree-structured algorithm imposed on the cooperating nodes. Each node is aware of the whether any rays remain active in its subtree and, when its subtree state changes between idle and active (either way) a message is sent up the tree informing its parent of the change, potentially changing the state of the parent. This eventually reaches the root of the tree, and when the root's state changes from active to idle, a synchronous test is made to test the system's overall state, necessitated due to the asynchronicity of the tree-based test.

3.7.8 Results

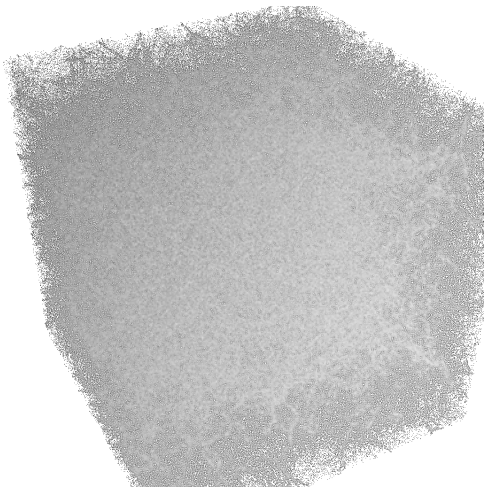
PVol was compared against traditional geometry-based rendering using the ETH test harness ref. Two cases were evaluated: the visualization of an xRage simulation, comparing PVol's direct volumetric capabilities to perform implicit isosurfacing and volume slicing with VTK's explicit operators (figure 48), and of a HACC simulation, comparing PVol's rendering of implicit procedural spheres against VTK's representations of spheres as simple points and as gaussian splats (figure 49).

While extensive results are available in [1], two in particular are of interest. Figure 50 shows the normalized execution time of the raycast and geometric algorithms on the xRage data after resampling onto structured grids of varying resolution, and demonstrates the scaling benefits of the raycasting algorithm. Figure 51 shows the normalized execution time of the raycast algorithm vs. the two geometry-based algorithms over datasets of varying size. Again, the raycast approach demonstrates significant scaling advantages.

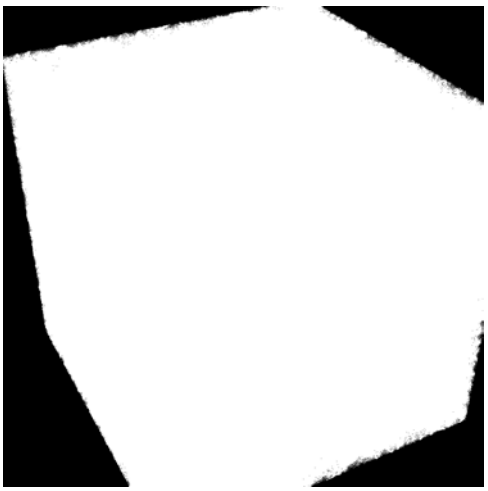


(a) Raycast image. **(b)** Geometry-based image.

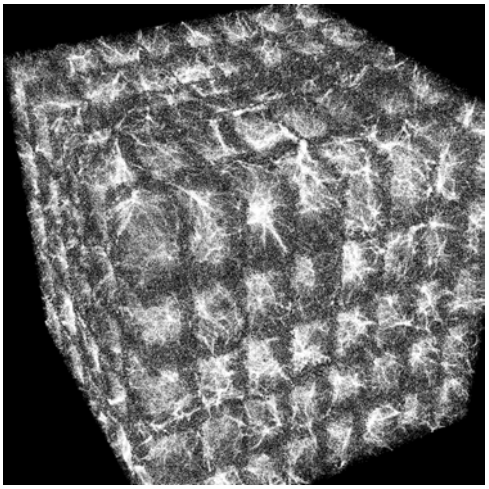
Figure 48: Visualizations of xRage data rendered using PVol (a) and VTK (b).



(a) PVol implicit spheres.



(b) VTK simple points.



(c) VTK gaussian splats.

Figure 49: Visualizations of HACC data rendered using PVol's implicit spheres (a), VTK's simple points(b) and VTK's gaussian splats.

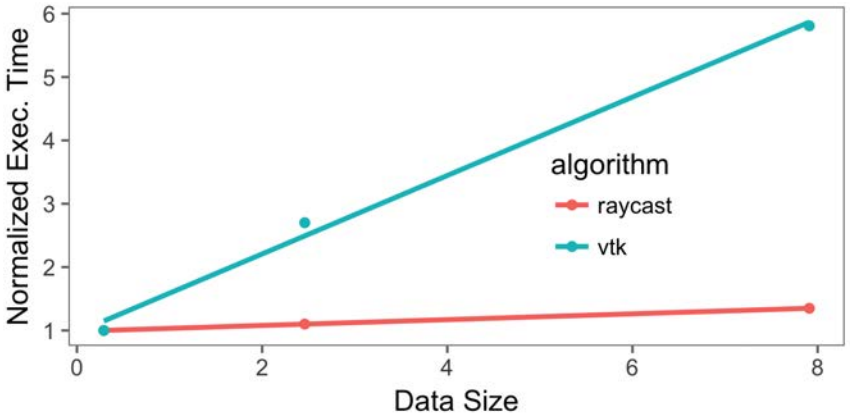


Figure 50: Scalability of raycasting and geometry-based (vtk) methods with respect to problem size. Test cases are results are normalized to the smallest problem size for each algorithm.

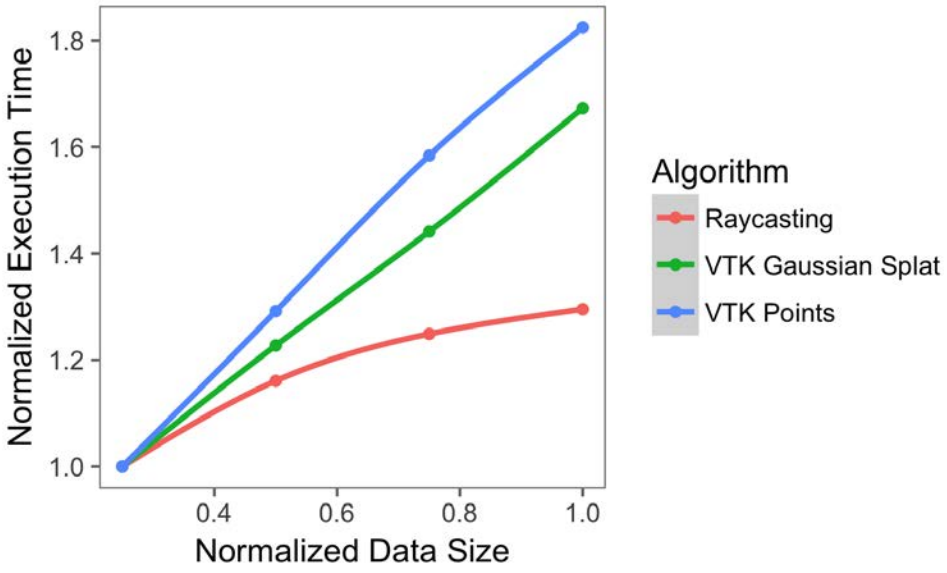


Figure 51: Scalability of raycasting, gaussian splat, and VTK points with respect to problem size. Test cases are 1/4, 1/2, 3/4 and the full HACC particle data, normalized to the smallest problem size. size for each algorithm.

3.8 Discovery Jam

Figure 52: Discovery Jam workshop participants working with Phillip Wolfram, Ocean-modeling Scientist with LANL's Climate, Ocean, Sea-Ice Modeling team, working on alternatives for visualizing and analyzing ocean mixing patterns in the Antarctic.

It is clear that modern scientific discovery requires a new data-intensive approach. Vast stores of data available – from experiments, observations and simulations – may enable countless new discoveries, yet scientists continue to struggle to make sense of their data. In keeping with many IEEE VIS attendees, we believe visualization holds an answer to this problem, but we also believe that a new, perhaps even disruptive, approach is needed in order to realize this potential.

We believe interactive analysis and visualization holds an answer to this problem, but we also believe that a new, perhaps even disruptive, approach is needed in order to realize this potential. We created the Discovery Jam as an exciting, vibrant, inspirational mechanism for creating interdisciplinary teams to address specific challenge problems. A Discovery Jam demonstrates – through example – a culture of collaboration and a method of training a new cohort of scientists, technologists, and artists to work together toward discovery across disciplines.

The first Discovery Jam was held at the 2016 VisWeek. Two domain scientists brought their data challenges. Phillip Wolfram, with the LANL Climate, Ocean, Sea-Ice Modeling team, joined Discovery Jam 2016 with the goal of visualizing oceanic eddy-mean flows. David Mascareñas, with the LANL Engineering Institute, was focused on visualization of full-field coherence functions for structural dynamics characterization. Building on the success of our team's half-day workshop at VisWeek 2016, we proposed a full-day Discovery Jam workshop that includes a hands-on data hack-a-thon in collaboration with domain scientists.

The Discovery Jam demonstrates – through example – a culture of collaboration and training a new cohort of scientists, technologists, and artists to work together toward discovery across disciplines. The Discovery Jam will bring scientists to VisWeek, to present their difficult, high-impact science discovery problems to experts in the vis community. The results from the workshop will be added to the online community being hosted at `http://www.discoveryjam.com`.

4 ECX Team

Gregory Abram PI, University of Texas. Dr. Greg Abram is a visualization researcher at the Texas Advanced Computing Center, a research division of the University of Texas at Austin. Prior to joining TACC, he was at the IBM TJ Watson Research Center. He received his Ph.D. from the University of North Carolina at Chapel Hill in 1986.

Vignesh Adhinarayanan Doctoral Graduate Student, Virginia Tech University. Vignesh Adhinarayanan is a Ph.D. student in the Department of Computer Science at Virginia Tech with Prof. Wu-chun Feng. He is a member of the SyNeRGy Lab working on Green Computing. He is working on developing power models and optimization techniques for Graphics Processing Units. Previously, he worked at Waran Research Foundation (WARFT) as an undergraduate research trainee in the area of heterogeneous computing. He is interested in the broad areas of green computing (energy-efficient computing), parallel computing, and hetergeneous computer architectures.

James P. Ahrens PI, Los Alamos National Laboratory. Dr. James Ahrens (LANL) is the founder and design lead of ParaView, a widely adopted visualization and data analysis package for large-scale scientific simulation data. ParaView has had an extremely positive impact on the large-scale data analytic capabilities available to simulation scientists around the world. Dr. Ahrens graduated in 1996 with a Ph.D. in computer science from the University of Washington. Following his graduate studies, he joined LANL as a technical staff member. At Los Alamos, he is the leader of an awesome data analysis and visualization team of twenty staff scientists, postdoctoral fellows, and students of all levels, as well as a national leader of programmatic initiatives important to the Department of Energy's (DOE) National Nuclear Security Administration (NNSA) Advanced Simulation and Computing (ASC) program and the Office of Science (SC) Advanced Scientific Computing Research (ASCR) programs.

Anne S. Berres Dr. Anne Berres received a Ph.D. in Computer Science from the University of Kaiserslautern, Germany in 2015. Her core expertise lies in various aspects of visual computing, most notably in-situ visualization, image processing, and geometry. As a Postdoctoral Researcher at Los Alamos National Laboratory (2015-2017), she conducted sampling and compression of large-scale simulation data, and developed climate simulations in an HPC environment. She is currently a Postdoctoral Researcher at Oak Ridge National Laboratory, where she applies her skills to image-based deep learning on architecturally diverse HPC machines, and web-based visualization and analysis for geographic data.

Roxana Bujack Los Alamos National Laboratory. Dr. Roxana Bujack is a staff scientist in the Data Science at Scale Team at Los Alamos National Laboratory, joining in July 2016. She graduated in mathematics and computer science and received her Ph.D. in the Image and Signal Processing Group at Leipzig University. She worked as a postdoctoral researcher at IDAV at the University of California, Davis and at the Computer Graphics and HCI Group at the Technical University Kaiserslautern. Her research interests include visualization, pattern recognition, vector fields, moment invariants, high performance computing, massive data analysis, Lagrangian flow representations, and Clifford analysis.

Wu-chun Feng PI, Virginia Tech University. Dr. Wu Feng is the Elizabeth & James E. Turner Fellow & Professor in the Department of Computer Science, Department of Electrical & Computer Engineering, and Virginia Bioinformatics Institute at Virginia Tech. His interests lie broadly at the synergistic intersection of computer architecture, systems software and middleware, and applications software. Most recently, his research has dealt with high-performance networking protocols, dynamic multicore scheduling, accelerator-based computing for bioinformatics, virtual computing, power-aware computing, and bioinformatics in general. He joined Virginia Tech in 2006 after seven years at Los Alamos National Laboratory. He is the recipient of three Best Paper Awards in human-computer interaction, high-performance networking, and bioinformatics, respectively, and three R&D 100 Awards in green supercomputing, high-speed networking, and

bioinformatics, respectively. He leads the SyNeRGy Laboratory as well as four grass roots projects: The Green500 List, mpiBLAST, Supercomputing in Small Spaces, and MyVICE.

David H. Rogers Co-PI, Los Alamos National Laboratory. David Honegger Rogers joined LANL in 2013, after a decade of leading the Scalable Analysis and Visualization Team at Sandia National Laboratory, where he was instrumental in bringing in-situ analysis and visualization into production. He now focuses on interactive web-based analysis tools that integrate design, scalable analytics and principles of cognitive science to promote scientific discovery. Prior to working on large scale data analysis, he worked at DreamWorks Feature animation, writing and managing production software. He has degrees in Computer Science, Architecture, and an MFA in Writing for Children.

Francesca Samsel PI, University of Texas. Francesca Samsel is a Research Associate at the University of Texas, in the Center for Agile Technology. Her work balances on the fulcrum between art and science. Activity immersed in collaborations with scientists and computer visualization specialists, she seeks to meld scientific factual understanding with artistic distillation and metaphor. Using the combined language, she seeks to present the environmental issues of our time in an approachable form to a wide range of audiences. The bulk of her work day is spent with the Research Visualization Team at LANL. Using her expertise in color and communication, the non-numerical kind, they are involved in a long-term collaboration to improve the visualization tools to help scientists gain deeper understanding from data.

Terece L. Turton University of Texas. Dr. Terece Turton is a Research Associate at the University of Texas at Austin, in the Center for Agile Technology. She received her Ph.D. in Physics from the University of Michigan and did postdoctoral work at the Superconducting Super Collider, Michigan State University, and the University of Cincinnati. Her current work focuses on perceptual user evaluation in scientific visualization and includes the Evaluation Toolkit, www.etklab.org, an open-source repository of modules to automate image-based perceptual experiments.

Colin Ware PI, University of New Hampshire. Dr. Colin Ware is the Director of the Data Visualization Research Lab at the Center for Coastal and Ocean Mapping at the University of New Hampshire. Cross-appointed between the Departments of Ocean Engineering and Computer Science, he specializes in advanced data visualization with a special interest in applications of visualization to Ocean Mapping. He combines interests in both basic and applied research and has advanced degrees in both computer science (MMath, Waterloo) and in the psychology of perception (Ph.D., Toronto). Many of his over 150 publications relate to the use of color, texture, motion and 3D displays in information visualization. His approach is always to combine theory with practice. A founding member of the Ocean Mapping Group at the University of New Brunswick and now at the Ocean Mapping Center at UNH, he has been designing 3D interactive visualization systems for ocean mapping for 25 years. He and his graduate students created the first version of the Fledermaus visualization system, now widely used in Oceanographic applications. Ware has also contributed to software system visualization. He directed the development of NestedVision3D, a system for visualizing very large networks of information. Ware has been instrumental in the creation of two spinoff visualization companies based initially on his research. Interactive Visualization Systems, Inc. makes visualization software for advanced ocean mapping applications. NVision Software Systems, Inc. provided visualization tools to enhance the understanding of large highly interconnected datasets. He is currently working on a variety of projects including ocean current visualization, the visualization of marine mammal tracks, and more theoretically, visual thinking algorithms. Ware has written two books: *Visual Thinking for Design* is an up to date account of the psychology of how we think using graphic displays as tools. *Information Visualization: Perception for Design*, now in its 3rd edition, is a comprehensive survey of what human perception tells us about how to display information.

5 References

[1] Abram, G., Adhinarayanan, V., Feng, W.-c., Rogers, D., Ahrens, J., and Wilson, L. ETH: A framework for the design-space exploration of extreme-scale scientific visualizations. Tech. rep., Virginia Tech University and Los Alamos National Laboratory, 2017. LA-UR-17-26715.

[2] Adhinarayanan, V. Performance, power, and energy of in-situ and post-processing visualization. Tech. rep., Los Alamos National Laboratory, 2015. LA-UR-15-27749.

[3] Adhinarayanan, V., Feng, W.-c., Rogers, D., Ahrens, J., and Pakin, S. Characterizing and modeling power and energy for extreme-scale in-situ visualization. In *Parallel and Distributed Processing Symposium (IPDPS), 2017 IEEE International* (2017), IEEE, pp. 978–987. LA-UR-16-22435.

[4] Adhinarayanan, V., Feng, W.-c., Woodring, J., Rogers, D., and Ahrens, J. On the greenness of in-situ and post-processing visualization pipelines. In *Proceedings of the 2015 IEEE Parallel and Distributed Processing Symposium Workshop (IPDPSW)* (2015), IEEE, pp. 880–887. LA-UR-15-21414.

[5] Adhinarayanan, V., Pakin, S., Rogers, D., chun Feng, W., and Ahrens, J. Performance, power, and energy of in-situ and post-processing visualization: A case study in climate simulation. Best Research Poster Finalist.

[6] Ashton, Z. C., Wendelberger, J. R., Ticknor, L. O., Turton, T. L., and Samsel, F. Analyzing task-based user study data to determine colormap efficiency. Tech. rep., Los Alamos National Laboratory, 2015. LA-UR-15-25715.

[7] Aygar, E., Ware, C., and Rogers, D. H. The contribution of stereoscopic and motion depth cues to the perception of structures in 3d point clouds, 2017. In press *Transactions on Applied Perception*.

[8] Banesh, D., Schoonover, J. A., Ahrens, J. P., and Hamann, B. Extracting, Visualizing and Tracking Mesoscale Ocean Eddies in Two-dimensional Image Sequences Using Contours and Moments. In *Workshop on Visualisation in Environmental Sciences (EnvirVis)* (2017), K. Rink, A. Middel, D. Zeckzer, and R. Bujack, Eds., The Eurographics Association.

[9] Berres, A., Adhinarayanan, V., Turton, T. L., Feng, W., and Rogers, D. H. A pipeline for large data processing using regular sampling for unstructured grids. Tech. rep., Los Alamos National Laboratory, 2016. LA-UR-16-24358.

[10] Berres, A., Turton, T. L., Rogers, D., and Ahrens, J. VideoDB: Reducing large image databases through video encoding and video compression. Tech. rep., Los Alamos National Laboratory, 2016. LA-UR-16-24358.

[11] Berres, A. S., Turton, T. L., Petersen, M., Rogers, D. H., and Ahrens, J. P. Video Compression for Ocean Simulation Image Databases. In *Workshop on Visualisation in Environmental Sciences (EnvirVis)* (2017), K. Rink, A. Middel, D. Zeckzer, and R. Bujack, Eds., The Eurographics Association. LA-UR-17-21590.

[12] Bishwajit Dutta, Vignesh Adhinarayanan, W.-c. F. A comparative study of cpus and integrated gpus for green scientific visualization. In submission to In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization (ISAV).

[13] Bujack, R. Colormeasures Website. https://colormeasures.org.

[14] Bujack, R., Turton, T. L., Samsel, F., Ware, C., Rogers, D. H., and Ahrens, J. The good, the bad, and the ugly: A theoretical framework for the assessment of continuous colormaps. In *To appear at VIS2017, IEEE Conference on Scientific Visualization* (2017), IEEE.

[15] Gholkar, N., Mueller, F., and Rountree, B. Power tuning hpc jobs on power-constrained systems. In *Proceedings of the 2016 International Conference on Parallel Architectures and Compilation* (2016), ACM, pp. 179–191.

[16] Guo, F., Li, H., Daughton, W., and Liu, Y.-H. Formation of Hard Power-laws in the Energetic Particle Spectra Resulting from Relativistic Magnetic Reconnection. *Phys. Rev. Lett. 113* (2014), 155005.

[17] Knoll, A., Wald, I., Navrátil, P. A., Papka, M. E., and Gaither, K. P. Ray tracing and volume rendering large molecular data on multi-core and many-core architectures. In *Proceedings of the 8th International Workshop on Ultrascale Visualization* (New York, NY, USA, 2013), UltraVis '13, ACM, pp. 5:1–5:8.

[18] Lange, D., Samsel, F., Karamouzas, I., Guy, S. J., Dockter, R., Kowalewski, T., and Keefe, D. F. Trajectory Mapper: Interactive Widgets and Artist-Designed Encodings for Visualizing Multivariate Trajectory Data. In *EuroVis 2017 - Short Papers* (2017), B. Kozlikova, T. Schreck, and T. Wischgoll, Eds., The Eurographics Association.

[19] Parker, S., Parker, M., Livnat, Y., Sloan, P. P., Hansen, C., and Shirley, P. Interactive ray tracing for volume visualization. *IEEE Transactions on Visualization and Computer Graphics 5*, 3 (Jul 1999), 238–250.

[20] Patchett, J. M., Samsel, F., Tsai, K. C., Gisler, G. R., Rogers, D., Abram, G., and Turton, T. L. Visualization and analysis of threats from asteroid ocean impacts. LA-UR-17-20419.

[21] Rogers, D. H., Turton, T. L., and Samsel, F. SciVisColor GitHub. https://github.com/ascr-ecx/sciviscolor.

[22] Samsel, F., and Ayachit, U. http://www.kitware.com/blog/home/post/886, May 2015. Kitware Blog.

[23] Samsel, F., Klaassen, S., Petersen, M., Turton, T. L., Abram, G., Rogers, D. H., and Ahrens, J. Interactive colormapping: Enabling multiple data range and detailed views of ocean salinity. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2016), CHI EA '16, ACM, pp. 700–709.

[24] Samsel, F., Patchett, J. M., Rogers, D. H., and Tsai, K. Employing color theory to visualize volume-rendered multivariate ensembles of asteroid impact simulations. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2017), CHI EA '17, ACM, pp. 1126–1134.

[25] Samsel, F., Petersen, M., Abram, G., Turton, T. L., Rogers, D., and Ahrens, J. Visualization of ocean currents and eddies in a high-resolution ocean model. Winner, Best Scientific Visualization & Data Analytics Showcase.

[26] Samsel, F., Petersen, M., Geld, T., Abram, G., Wendelberger, J., and Ahrens, J. Colormaps that improve perception of high-resolution ocean data. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (2015), CHI EA '15, pp. 703–710. LA-UR-15-20105.

[27] Samsel, F., Turton, T. L., Bujack, R., Rogers, D. H., Ahrens, J., Abram, G., and Ware, C. Data Science at Scale SciVis Color Website. https://sciviscolor.org.

[28] Samsel, F., Turton, T. L., Wolfram, P., and Bujack, R. Intuitive Colormaps for Environmental Visualization. In *Workshop on Visualisation in Environmental Sciences (EnvirVis)* (2017), K. Rink, A. Middel, D. Zeckzer, and R. Bujack, Eds., The Eurographics Association.

[29] Schöffel, S., Schwank, J., Stärz, J., Ebert, A., and van der Veer, G. Bref: Browser-based evaluation framework. In *Proceedings of the 2017 ACM Conference Companion Publication on Designing Interactive Systems* (New York, NY, USA, 2017), DIS '17 Companion, ACM, pp. 313–316.

[30] Schroeder, W., Martin, K., and Lorensen, B. The Visualization Toolkit: An Object-Oriented Approach to 3-D Graphics, 2005.

[31] Shalf, J., Dosanjh, S., and Morrison, J. Exascale computing technology challenges. In *International Conference on High Performance Computing for Computational Science* (2010), Springer, pp. 1–25.

[32] Turton, T. L., Berres, A. S., Rogers, D. H., and Ahrens, J. ETK: An Evaluation Toolkit for Visualization User Studies. In *EuroVis 2017 - Short Papers* (2017), B. Kozlikova, T. Schreck, and T. Wischgoll, Eds., The Eurographics Association.

[33] Turton, T. L., and Rogers, D. H. Evaluation Toolkit GitHub. https://github.com/ascr-ecx/etk.

[34] Turton, T. L., and Rogers, D. H. Evaluation Toolkit Website. https://etklab.org.

[35] Turton, T. L., Ware, C., Samsel, F., and Rogers, D. H. A crowdsourced approach to colormap assessment. In *EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization (EuroRV3)* (2017), K. Lawonn, N. Smit, and D. Cunningham, Eds., The Eurographics Association.

[36] Wald, I., Johnson, G., Amstutz, J., Brownlee, C., Knoll, A., Jeffers, J., Günther, J., and Navratil, P. Ospray - a cpu ray tracing framework for scientific visualization. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (Jan 2017), 931–940.

[37] Ware, C. Color sequences for univariate maps: Theory, experiments and principles. *IEEE Computer Graphics and Applications 8*, 5 (1988), 41–49.

[38] Ware, C., Bolan, D., Miller, R., Rogers, D. H., and Ahrens, J. P. Animated versus static views of steady flow patterns. In *Proceedings of the ACM Symposium on Applied Perception* (New York, NY, USA, 2016), SAP '16, ACM, pp. 77–84.

[39] Ware, C., Rogers, D., Petersen, M., Ahrens, J., and Aygar, E. Optimizing for visual cognition in high performance scientific computing. *Electronic Imaging 2016*, 16 (2016), 1–9.

[40] Ware, C., Turton, T. L., Bujack, R., Samsel, F., Shrivastava, P., and Rogers, D. H. Measuring and modeling the feature discrimination threshold functions of colormaps, 2017. In submission to *Transactions on Visualization and Computer Graphics*.

[41] Ware, C., Turton, T. L., Samsel, F., Bujack, R., and Rogers, D. H. Evaluating the perceptual uniformity of color sequences for feature discrimination. In *EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization (EuroRV3)* (2017), K. Lawonn, N. Smit, and D. Cunningham, Eds., The Eurographics Association.

[42] Whitted, T. An improved illumination model for shaded display. *Commun. ACM 23*, 6 (June 1980), 343–349.

[43] Woodring, J., Ahrens, J., Patchett, J., Tauxe, C., and Rogers, D. High dimensional scientific data exploration via cinema. in publication.

[44] Woodring, J., Rogers, D., Ahrens, J., and Patchett, J. Cinema database specification dietrich release v1.0. Tech. Rep. LA-UR-17-25072, Los Alamos National Laboratory, 2017.