DOI: 10.2352/ISSN.2470-1173.2016.16HVEI-130

Optimizing for Visual Cognition in High Performance Scientific Computing

Colin Ware[1] David Rogers[2], Mark Petersen[2], James Ahrens[2] and Erol Aygar[1]

1. *Center for Coastal and Ocean Mapping, University of New Hampshire, Durham, NH, USA*
2. *Los Alamos National Laboratory*

Abstract

High performance scientific computing is undergoing radical changes as we move to Exascale (10^{18} FLOPS) and as a consequence products for visualization must increasingly be generated in-situ as opposed to after a model run. This changes both the nature of the data products and the overall cognitive work flow. Currently, data is saved in the form of model dumps, but these are both extremely large and not ideal for visualization. Instead, we need methods for saving model data in ways that are both compact and optimized for visualization. For example, our results show that animated representations are more perceptually efficient than static views even for steady flows, so we need ways of compressing vector field data for animated visualization. Another example, motion parallax is essential to perceive structures in dark matter simulations, so we need ways of saving large particle systems optimized for perception. Turning to the cognitive work flow, when scientists and engineers allocate their time to high performance computer simulations their effort is distributed between pre and post run work. To better understand the tradeoffs we created an analytics game to model the optimization of high performance computer codes simulating ocean dynamics. Visualization is a key part of this process. The results from two analytics game experiments suggest that simple changes can have a large impact on overall cognitive efficiency. Our first experiment showed that study participants continued to look at images for much longer than optimal. A second experiment revealed a large reduction in cognitive efficiency as working memory demands increased. We conclude with recommendations for systems design.

Introduction

High performance computers (HPCs) and the people who use them constitute a distributed cognitive system, with part of the work being done in the machine and part in the brains of developer scientists. HPCs are essential to applications ranging from forecasting climate change to understanding the structure of cosmic dark matter. These systems are undergoing dramatic changes in architecture with the advent of new computer hardware capable of performance at Exascale (10^{18} FLOPS), and in response the cognitive systems incorporating these machines must change.

To be efficient in producing insights, designers of HPCs must weigh the costs of code development, computer runs and the thought processes of individual scientists and engineers who later use the system. An overview of the cognitive system characteristic of high performance computer modeling is given in Fig. 1. This shows three distinct phases.

The first phase consists of building and tuning the computer codes designed to test scientific hypothesis. At this stage a large number of relatively low resolution model runs may be carried out per day. This is done in order to determine if the model is behaving as expected and also to compare it to other models.

The second phase consists of setting up and running a major computation. The first part of this is a collaborative effort among a number of scientists who must set the goals for the computation defined by the scientific questions it is intended to address. Once the setup has been completed and the computation started, there is little or no direct cognitive input from the developer scientists, but in-situ saved products can make it very much easier to gain insights in the final phase [1,19,20].

The third phase occurs following a major computation. This is where the results are interpreted. Occasionally, during the interpretation phase, new questions may arise and these may be addressed through additional computation; for example, Lagrangian pathline tracing may be done based on outputs of the model run. Most of the interpretation, however, is done by means of visualization products derived from model outputs using tools such as Paraview or VisIt.

Various time scales are involved in this distributed cognitive effort. The overall development of a major new model may take from two to three years; this is also true for the gestation of new ideas. Nested within large scale cognitive processes are many much smaller cognitive acts, as well as much smaller computations. For example, interpreting model outputs can involve scanning large numbers of images, sometimes only for a few seconds each. Only occasionally are new insights gained, but each act involved processing a large amount of computed data. By making individual acts of pattern perception more efficient, the entire cognitive process can be improved.

What will change with Exascale?

With exascale computer models complete pictures of model state will be saved far less frequently. Most visualization from HPCs is currently based on checkpoint-restart files. These contain all of the data needed to restart an application if the program should terminate, including all of the model parameters for a particular simulated time step. These files are large, often tens of gigabytes and will become larger as the resolution of models becomes finer and finer. For example the number of particles for a simulation of cosmic dark matter has increased from millions to tens of trillions [14]. It is also becoming much more costly to save checkpoint restart files because of changes in computer architecture. According to many experts, the only cost effective way of providing exascale performance is to move to light-weight processors, with many more on a single chip. Already there can be thousands of floating point units on a chip, but in the future there will be perhaps tens of thousands. These make it possible to carry out many more floating point operations but at the cost of a relative increase in getting, loading and saving data. So the ratio of calculations to model values saved will increase dramatically. Checkpoint saves will be much less frequent and there may be tens of thousands of model time steps between them.

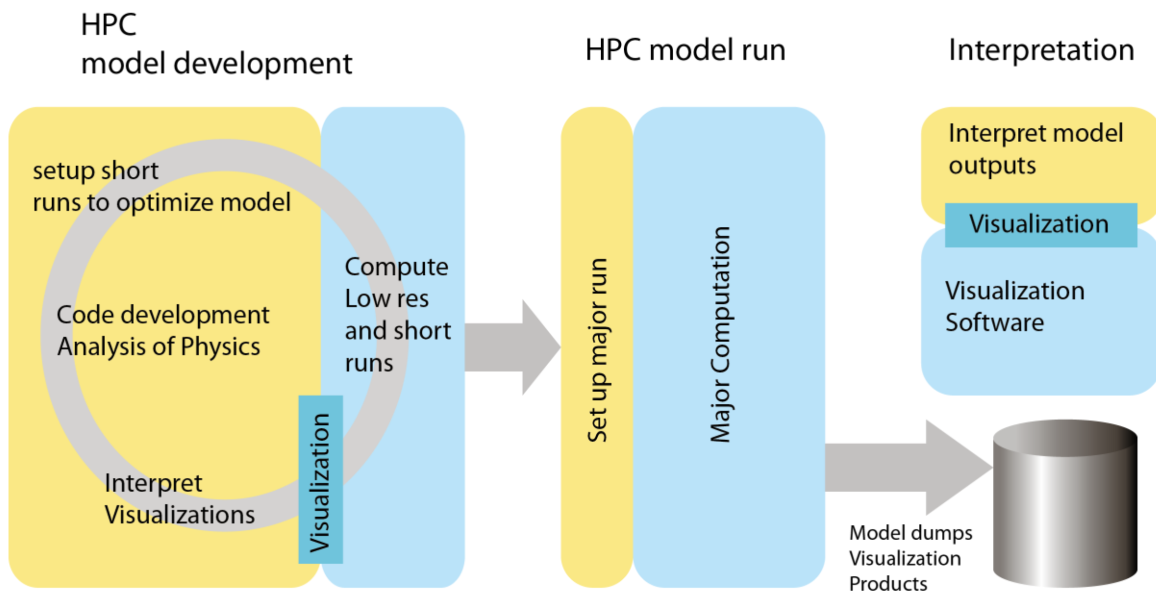

Fig. 1*. The architecture of the cognitive system that does model optimization for high performance computing. Yellow areas represent human cognition, blue areas represent computation.*

As a result, important patterns occurring at a fine temporal granularity will simple be lost due to infrequent saving of checkpoint files.

An additional problem is that checkpoint files are designed for saving model state and not for visualization. It can take significant time to construct a single transect through a large data set making interactive exploration impossible.

The in-situ solution and its implications

To deal with the problems of large models and infrequent saving of model data, in-situ methods are being developed to generate data products more suitable for perception and cognition [1,3]. One approach is to generate images of the computed model in-situ, as the computation proceeds. These are far smaller than checkpoint files by orders of magnitude even if thousands of images are saved. However, there are cases where images are insufficiently flexible, and we will make the case in this paper that we need more flexible intermediate representations in the form of compact storage formats designed to support a variety visualization methods.

In considering how to make HPC systems more cognitively effective it is useful to examine the problem at different scales. At the cognitively fine scale are individual acts of perception when meaningful patterns are perceived from an image. In order to make these fine scale cognitive events more effective it is critical that the data be presented in such a way that important patterns stand out. At present, we are only beginning to understand how best to do this, although saving images, as in Cinema [1] is a useful start. The main challenges have to do with visual cognition and designing systems as a whole to improve the rate at which insights can be gained.

The remainder of this paper is divided into three parts. The first two parts are examples where insights from human perception point to the need for innovation in the way data are saved. The third part addresses the bigger picture and ways in which information foraging theory may be adapted to study and optimize the cognitive system surrounding HPC.

Animated Traces even with Non Time-varying Vector Fields

There have been a number of studies into the representation of 2D vector fields supporting the idea that smooth, equally spaced streamlines may be the best way of showing vector fields that do not vary in time [7]. But, increasingly, people are using animated traces to show variables such as wind patterns [17]. Our observations of such visualizations suggest an advantage. Moving patterns use a different visual 'channel' to color or texture, meaning that if a flow pattern is to be layered over some other pattern it will interfere less. But are they actually better? We have been investigating the possibility that animate patterns may be better than static portrayals, even for patterns that do not vary in time.

Our experiment compared four different representations: animated streamlines, animated orthogonal particles, a conventional arrow grid, and equally spaced streamlines. Study participants were given a visual search task: to look for patterns like those shown in Fig. 2 embedded in a complex field of similar patterns. Fig 3 gives the results. It shows that animated visualizations enable people to find patterns with fewer than half the errors compared to the best static visualizations (A full report of this study is submitted elsewhere). A second experiment shows that people are also almost twice as accurate on an advection pathway tracing task compared to the alternatives.

These results show that it is important to save vector field data in ways that can support animated display. Saving images is unlikely to be the best way of doing this. Instead we are exploring ways of compressing vector fields in-situ that are perceptually lossless when rendered using the best methods.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.16HVEI-130

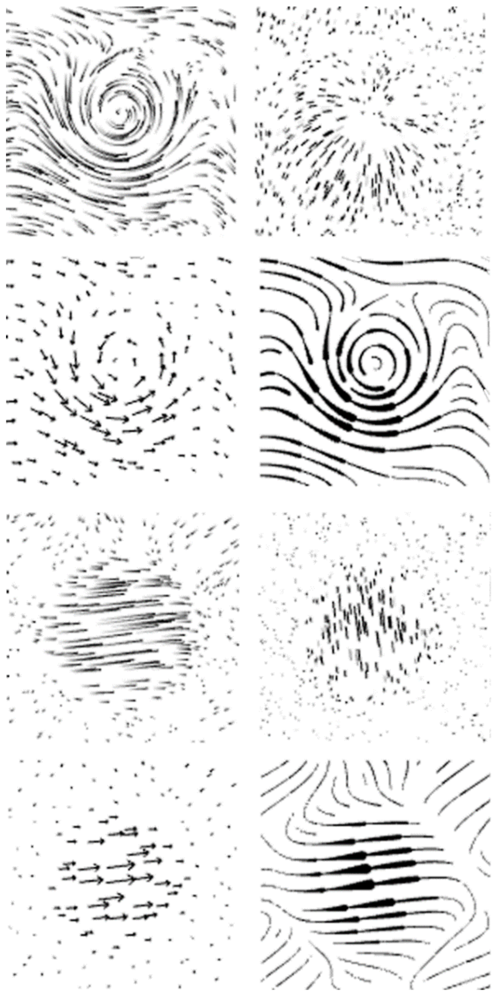

***Fig 2**. Alternative ways of showing vector field data. Row 1. Two methods for showing a circular pattern using animation: animated streamlines and animated orthogonal particles. Row 2. The circular pattern using arrows and equally spaced streamlines. Row 3. Left to right flow using animated streamlines and animated orthogonal particles. Row 4. Left to right flow using arrows and equally spaced streamlines.*

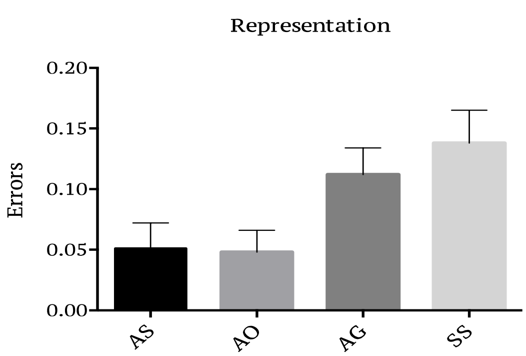

***Fig 3.** AS: Animated streamlines. AO: Animated orthogonal particles. AG: Arrow grid. SS: Equally spaced streamlines.*

For example, we may be able to show vector orientation using 11 bits of information and vector speed using 5 bits of information, thereby reducing the two double precision floats normally stored by a factor of eight. If such information can be stored in-situ a variety of portrayal methods can be used when a scientist wishes to review the results.

The Need for Animation When Viewing Large Particle Systems

Our second example of a perceptual challenge is the visualization of large particle systems. Halos are a theoretical construct representing the missing (dark) matter needed to make the observed universe behave according to the laws of gravity. To test and refine this theory the structure of halos is estimated by means of HPC simulations using very large systems of particles (see Fig. 4). The particle system is allowed to evolve over the life of the universe following physical laws. Simulated visible matter develops in parallel, influenced by the gravitational effects of dark matter.

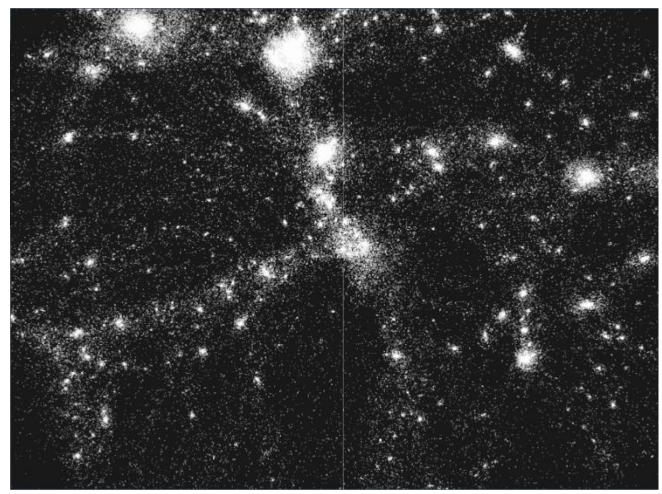

***Fig 4.** Halo simulation data.*

From a perceptual point of view, halo data consists of three-dimensional point clouds. Like all depth perception, the structure of point clouds can be perceived by means of depth cues. Depth cues consist of the way light is structured to provide information about distance from a particular viewpoint; examples are linear perspective, texture gradients and cast shadows. But for point cloud data most depth cues are not useful; for example, there is no perspective and only two depth cues really help for point clouds: stereoscopic depth (obtained by providing different views to the two eyes) and structure from motion, also called kinetic depth (obtained by rotating a scene). Although it has long been known that kinetic depth can help us perceive point cloud data [2], surprisingly the best way of doing this has not previously been studied and this research is needed before we can address the issue of the optimal compression schemes. We therefore carried out a study to investigate the optimal viewing conditions for point cloud data (Agar and Ware in preparation). The task was to detect 3D patterns within a simulated halos data set. Preliminary results, shown in Fig. 5 suggest that kinetic depth is more important than stereoscopic viewing in perceiving 3D structure. Still, many parameters still remain to be determined. For example, what is the optimal rate and angle of rotation? In addition, the best way of storing point cloud data so that features can be perceived remains to be determined.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.16HVEI-130

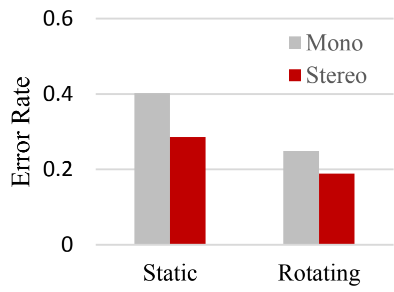

Fig 5. *Results from point cloud viewing study. Kinetic depth helps more than stereo in resolving 3D structures. The combination of stereo and motion is best.*

Cognitive Theory

We turn our attention now to the broader cognitive issues surrounding high performance computing and introduce a new methodology for experimentally understanding these systems. We propose that information foraging theory can be used to provide a framework for predicting how time allocation will be optimized.

Visual working memory has emerged as a central cognitive bottleneck in many tasks. Visual working memory has a capacity of only approximately three simple visual objects, where a simple object might be something like a yellow square or a green triangle [16] It is a critical resource for tasks such as visual comparisons between patterns. Information is typically held in visual working memory for between 200 ms and 3s. Understanding working memory capacity can provide a theoretical basis for user interface design decisions. For example, when is a zooming interface insufficient as a way of comparing patterns in large information spaces and when are extra windows needed The theoretical answer is that when patterns to be compared exceed the capacity of visual working memory, extra windows should be provided [12].

Adding more detail to the way people optimize cognitive tasks, Gray and Fu [4] developed the theory of soft constraints. This involves micro-strategies whereby individuals may make use of different cognitive resources. Working memory is implicated in soft constraints tradeoffs. For example, people will generally use more eye movements when comparing patterns as a way of avoiding burdening visual working memory [4,5,18]. Although most people have a visual working memory capacity of three or four simple patterns, when an external display is available only a single working memory slot will be used. This result suggests that making extra eye movements to compare patterns requires less cognitive effort than trying to more fully utilize visual working memory. In other words, the external display removes the need to burden short-term visual memory and it makes cognition more efficient as a result.

Although understanding detailed cognitive tradeoffs helps with the design of interaction methods, it does not tell us much about how people optimize strategies over longer time scales. Information foraging theory can help here [10,11]. Information foraging theory is derived from animal foraging theory developed in the field of behavioral ecology to account for how animals forage for food in patchy environments. When food is arranged in patches with large desert spaces between them, an animal must choose when to continue browsing a diminishing patch, and when to head off in search of a greener patch. Pirolli and collaborators adapted this model to form the basis of information foraging theory [11]. When foraging for knowledge, seekers must optimize their time in the similarly patchy information space of the World Wide Web. They can choose to continue studying a particular patch, such as a web site devoted to a relevant topic, or at any point in time they can break off to institute a new search, seeking new information.

There have been a number of approaches to the study of complex cognitive systems involving humans and various technologies, although none, that we are aware of, directly target high performance computing. One line of study has looked at the way people manage the trade-offs inherent in problem solving in real (as opposed to abstract) environments. Early researchers motivated by artificial intelligence suggested that people use formal logic to solve problems. But research later showed that human intelligence is mostly heuristic; people construct good-enough solutions given the time available, often using aspects of the immediate environment as cognitive props and by adapting the patterns of previous solutions. This approach is called satisficing [15].

Applying Information Foraging Theory: The Analytics Game

The central theorem of information foraging theory is Charnov's Marginal Value Theorem. This states that the optimal time to leave a patch occurs when the rate of gain from browsing that patch declines to the overall rate of gain (taking into account both patch feeding and searching for new patches). A major goal in the present research is to determine whether information foraging theory can help us understand the HPC cognitive system. We show that the marginal value theory can be applied to the process of the refinement of computer codes used in HPC simulations.

One of the challenges in studying the cognitive processes of science is that it is difficult or impossible to conduct controlled experiments using domain scientists working in their fields of expertise. Few scientists tolerate interference in the way they work and, in any case, scientific insights are too sparse and unpredictable to be a useful dependent variable. An approach to solving this kind of problem that has proven fruitful in domains such as behavioral economics has been to create games that provide simple modesl for the complexity of the target domain, and where the study participants can be undergraduate students, a resource readily available to most academics.

We developed the analytics game to explore how individuals adjust their behaviors in simple pattern detection and pattern comparison tasks. This game is designed to model, in simplified form, cognitively critical aspects of the HPC workflow. In order to understand how people optimize their behavior, we created a simplified task to approximate the cognitive process involved in the optimization of HPC codes simulating ocean dynamics. The task involved the detection of anomalous patterns. In the 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 at a cost.

The purpose of the particular game reported here is to gain insights into cognitive strategies employed by developer-scientists as they develop high performance computer simulations. A common cognitive process occurring in the optimization of supercomputing codes is a cycle wherein a developer-scientist creates 10-20 low resolution runs per day, often to test the effects of applying different small sets of parameter values. The results of each run are interpreted by means of a large number of 1D and 2D visualizations (there may be 30-40 of these). The 1D visualizations are plots representing variables such as vertical temperature profiles

DOI: 10.2352/ISSN.2470-1173.2016.16HVEI-130

at particular locations. The 2D visualizations are typically pseudo-colored horizontal or vertical slices or aggregated data such as Atlantic Meridional Overturning Circulation maps. A variety of insights may be gained from viewing these visualizations but one of the most common is the detection of numerical problems in the model. These can take the form of extreme values, implausible oscillations or other phenomena that do not match what is physically known to occur. To an experienced eye such anomalies will often be apparent at first glance.

The parallels between the cognitive system behavior we describe above and foraging theory are as follows: Examining a sequence of visualizations searching for insights is analogous to browsing a food patch; specifically, an insight, such as a detected anomaly is analogous to a gain in food calories. Domain scientists look at the visualization most likely to yield important insights first and progressively process visualization in decreasing order of expected value. This declining payoff is analogous to the depletion of a food patch. Finally, ceasing to study the output from a model run and switching attention to setting up a new run is analogous to quitting a particular patch of food and setting off in search of a new patch.

Both information foraging theory and soft constraints theory make clear predictions for the rate of gain of insights. According to information foraging theory, developer-scientists will cease looking at visualizations when the rate of gain insight value for a particular run falls below the overall rate (averaged over cycles consisting of setting up runs and interpreting the results). Also, the rate of gain of insights will be critically dependent on the time taken to load visualizations, so that developer-scientists will switch to new model runs sooner when there are long load times. Soft constraints theory incorporating visual working memory limitations predicts that where patterns must be compared, even very short delays between sequential pattern presentations will result in much lower detection rates. This can be expected to have an even greater impact on performance.

We carried out experiments using two versions of the game. In the first, the task was anomaly detection. Time delays in the presentation of images which might contain anomalies allowed us to determine the extent to which participants' behavior conformed to the predictions of information foraging theory. In the second experiment, the task involved comparisons between two patterns that shared a common feature, although represented in different ways. Pairs of patterns were presented either sequentially with short delays between successive images or side-by-side. Our hypothesis was that short lags would result in substantially lower difference detection rates because of the increases in working memory load.

Game One: Optimizing Payoffs for Anomaly Detection

Participants were told that their goal was to maximize their earnings on a game. They were rewarded with a $12.00 base amount plus an additional 10c per anomalous pattern detected.

The game was structured as follows. Participants had two kinds of tasks which alternated. The first corresponded to "setting up a run" and it was designed to simply occupy their attention for 30s. They looked at the screen waiting for a red rectangle to appear after a sequence of other colored rectangles. When the red rectangle appeared after 5s they pressed the left mouse button starting a new sequence. There were 6 cycles of this, occupying approximately 30s.

The anomaly detection task followed. In this task participants waited for an image to load, then they pressed one of two labeled keys on the keyboard depending on whether or not they saw an anomaly. This was repeated with a new pattern until they felt that they were exhausting the value of the visualizations in which case clicking the left mouse button initiated a new cycle. Participants gained one point from detecting a target. They lost one point when they missed a target or when they responded positively in the absence of a target.

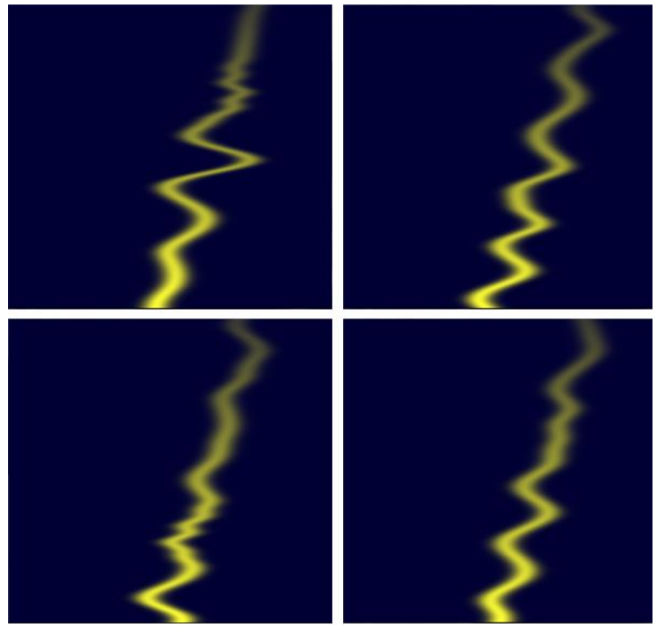

Fig.6. *The two images on the left are examples of wiggle "anomalies". The two on the right are examples of "normal".*

Conditions and procedure

There were two different types of pattern used in the trials.

Wiggles: The first was the wiggly curve shown in Figure 6. This was intended to roughly approximate the form of a western boundary current such as the Gulf Stream in the Atlantic or the Kiroshio in the Pacific. In this case an anomaly was defined as a higher spatial frequency component to the wiggles. Examples are shown in the two left hand curves.

Spirals: The second was the spiral patterns with colored cores shown in Fig.7. This was intended to roughly approximate ocean eddies. In this case an anomaly was a clockwise rotation with a red center. Only one of the spirals in Fig. 7 has these properties.

There were two image load latencies: ***Short***, 200 ms and ***Long***, 5.0s.

On each run, a random sequence of visualizations, such that the probability p of a target for the i^{th} visualization in a sequence, was given by:

$p = 1/(1+i/3)$

The experiment was a 2x2 design, the product of the two pattern types and the two delays.

The experiment started with training runs where the game was explained and participants were allowed to practice the task with both kinds of patterns. They were given feedback when they missed targets.

Following this, a trial run under a given condition (e.g. Long, Wiggles) lasted 5 minutes, during which the participants attempted to identify as many anomalies as they could. Following this they had 5 minutes sessions in the other conditions.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.16HVEI-130

In an approximately one-hour session, a study participant was given all four conditions twice in different random orders. About a week later they were brought back for another identical session. This resulted in four 5 min trial runs per participant per condition.

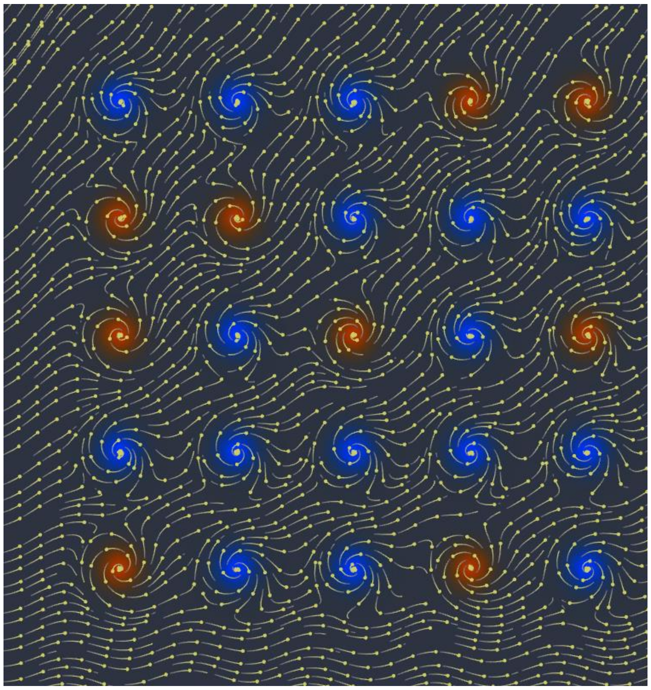

***Fig 7**. An example of a vortex "anomalies". The clockwise red-center spiral pattern is an anomaly.*

According to information foraging theory, in long image load conditions participants will look at fewer images than in the rapid load condition. Since the value of looking at visualizations declines as a function of the number of images viewed in a sequence, optimal performance will mean fewer images examined.

The time spent on a model run *(T_R)* is

$$T_R = ST_R + \sum_{i=1}^{n}(VT + VL)$$

ST_R: The run setup time
VT: Vis viewing and response time
VL: Vis loading time
n: number of visualizations viewed

The value of insights gained per run (I_R) is given by

$$I_R = \sum_{i=1}^{n} Cp(i)$$

C: The value of an insight
p(i): A function that models how the likelihood of insights declines with the number of images viewed. For example:
Cp(i) = C/(i+1)
Finally, the rate of return (insights per unit time) is given by I_R/T_R

Foraging theory predicts that people will start a new run when the rate of insights gained for a model run falls below the overall rate of insights. Fig. 8 illustrates the rate of return curves for the long load and short load conditions based on the assumption that once a visualization has loaded the time taken to view it and respond will be 1.2s

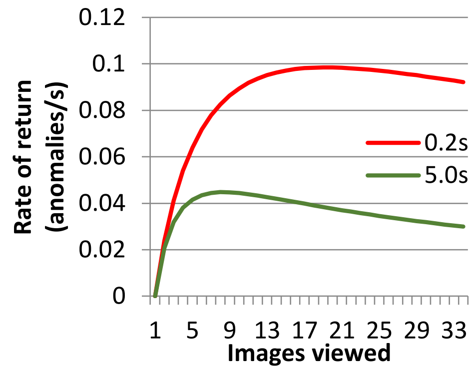

***Fig 8.** The theoretical optimal rate of return as a function of the number of images viewed before participants started a new run. The two curves are based on 0.2s and 5.0s image load times respectively. These simulations assume that the viewing and response time is 1.2s/image.*

Participants

There were 14 participants in Experiment 1. They were all undergraduate students paid for participating. For each of two sessions participants were given a base amount of $12.00 for the approximately one hour duration of the study. They earned an additional 10c for each anomalous pattern detected. This brought the average total remuneration to approximately $38.00.

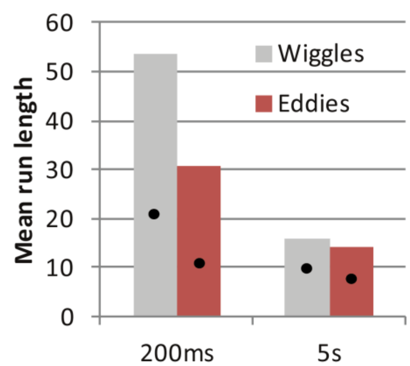

***Fig 9.** Mean run length under the difference conditions. The dots show optimal run lengths.*

Results from Experiment One

Data from one of the 14 participants was dropped from the analysis because that person never created new runs, simply processing the same sequence until the 5 minute time limit expired.

The main results for the remaining 13 participants are summarized in Figs 9 and 10. For the dependent variable run length, a two factor ANOVA (load time, pattern type) revealed highly significant main effects for both factors ($p<0.001$) and no significant interaction. As hypothesized, study participants were influence by the loading delay, as predicted by the model they shortening the run lengths in the long load conditions.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.16HVEI-130

Visual search was much more time consuming for the eddies pattern than for the wiggle pattern, taking approximately three times as long (2.7s vs 0.9s). The eddies processing time is considerably longer than the 1.2s per visualization used to generate the optimal behavior curves shown in Fig. 8, while the processing times for wiggles were somewhat shorter. In order to take this into account, optimal run lengths were recalculated from theory using the actual average visual processing times for the different conditions. Based on this, the black dots in Fig. 9 show optimal run lengths for the four conditions. These are substantially shorter than the runs generated by the study participants. One sample t-tests for each of the four conditions showed these differences to be significant (in all cases $p < 0.001$).

For the dependent variable earnings (Fig. 10), a two factor ANOVA (load time, pattern type) revealed highly significant main effects for both factors ($p<0.001$) and no significant interaction. The results show that earnings were 62% higher with the short load times. Since earnings are a proxy for cognitive work this demonstrates the importance of short load times in promoting cognitive efficiency.

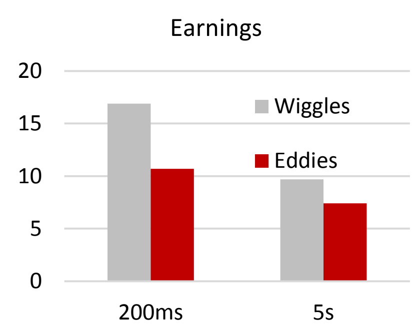

Fig. 10. *Mean earnings under the different conditions. Shorter latencies lead to higher earnings (a proxy for cognitive work).*

Game Two: Working Memory Involvement

The second experiment was designed to evaluate how people adjust their behavior to optimize return on cognitive effort when the burden on working memory is increased.

There were two key differences from the first experiment. First, in the second experiment the visual task involved *comparing* two visualizations that *contained common features* expressed in different ways, although the underlying features might or might not be identical (see Fig.11). The motivation for this was that often in computational models different variables may express the same underlying process. For example, the path of the Gulf Steam may be expressed as flow speed in one visualization and as a change in sea surface height in another (normally the Gulf Stream closely follows a steep gradient in sea surface height). In this case an anomaly would be present if the Gulf Stream failed to follow the gradient of sea surface height. For the patterns in Fig.11 the correct response is "different" because they contain different paths.

The second difference from the first experiment was that in half of the conditions pattern comparisons were made using alternating sequential views controlled by the user and in half they were made side-by-side. We predicted that comparing visualizations sequentially would result in shorter run lengths and lower earnings. In both cases working memory is involved, but in the side-by-side case rapid eye movements may allow three point to point comparisons a second. If there is even a short delay in the sequential case, people are likely to attempt to use working memory to remember more features of the pattern and errors will inevitably increase.

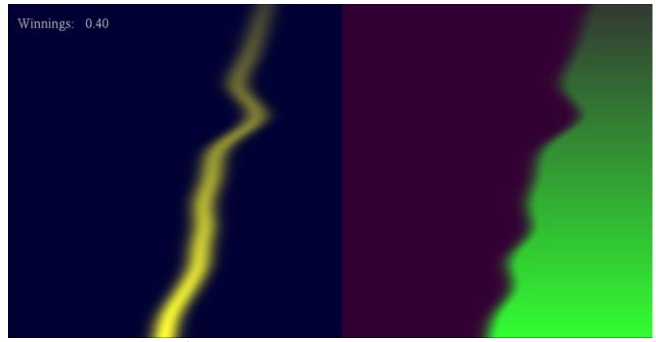

Fig. 11. The kinds of patterns used in the second experiment. The two figures on the left and right contain paths (wiggle patterns) which differ in the lower region.

Method

There were four conditions, the product of two variables:

- Two image load latencies: Short, 200 ms and Long, 5.0 s, and
- Side by side and sequential viewing: On half the trials images were compared side-by-side as shown in Fig. 11. On half the trials images were presented using sequential comparison with a 0.5s delay between successive images. In the sequential condition participants used the tab key to alternate between the pair of images before choosing.

The presentation of the conditions followed the same pattern as Experiment 1.

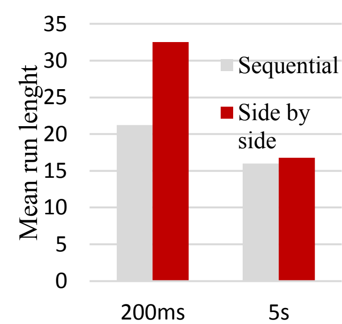

Fig. 12. Mean run length under the different conditions for Experiment 2.

Participants

There were 14 participants in Experiment 2. They were all undergraduate students paid for participating. For each of two sessions participants were given a base amount of $12.00 for the approximately one hour duration of a session. They earned an additional 10c for each anomalous pattern detected. This brought the average total remuneration to approximately $38.00.

Results from Experiment Two

The results are summarized in Figs12 and 13. ANOVAs (side-by side vs sequential) revealed a highly significant main effect ($p<0.001$) for the dependent variable run length. There was also a

DOI: 10.2352/ISSN.2470-1173.2016.16HVEI-130

significant interaction ($p< 0.01$). Surprisingly, in the long load time condition, the side-by-side vs sequential effect disappeared.

On average in the sequential condition there were 1.85 images swaps made. They most commonly looked at the first image, then the second then back at the first, before making a decision. If we take into account the image swap time of 0.5 seconds, participants in the sequential condition had a 0.925s penalty. That accounts for less than half of the extra time participants took in that condition.

Fig.13 shows that earnings were higher with side-by-side presentation as we predicted ($p <0.0001$). However the effect was less pronounced in the long load time condition.

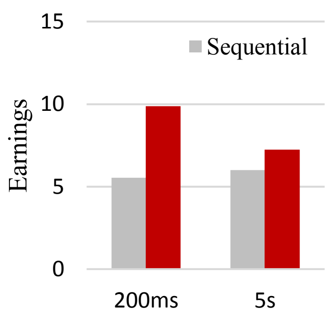

Fig 13. Earnings results for experiment 2. Earning were higher for side-by-side image comparisons.

Conclusion

It is useful to consider three cognitive time scales when thinking about HPC. At the finest scale are individual visual queries whereby a scientist comes to understand the resuls of a simulation by looking for patterns. At an intermediate scale is the process whereby a developer-scientist refines HPC simulation code by conducting many model runs, using visualizations to evaluate the results. At the largest scale is the three stage process illustrated in Fig. 1, encompassing the entire life cycle of HPC code development, major model runs and the interpretation of results.

This paper is mostly concerned with fine scale and intermediate scale cognitive processes. But there is a large scale cognitive implication arising from the transition to in-situ generation of visualization products that we will briefly discuss. In-situ products necessarily are designed to provide what are anticipated to be the most useful visualizations. If, after a major model run, it becomes apparent that a particular visualization is needed, it may no longer be possible to generate it, especially if it is needed to show phenomena at a fine temporal granularity. This means that there must be a major shifit in the cognitive work flow from the last stage of the process shown in Fig. 1 to the first stage.

To support this transition, software tools will be needed. HPC model users will require tools to show them what in-situ produces are available, and to insert code related to the generation of those products into the simulation code.

Returning now to cognitive activity on shorter time scales: Supporting fine scale visual thinking requires visualizations that show important data patterns portrayed in ways that makes them easy to perceive. We still do not know the best way of representing many kinds of data. The studies reported at the start of this paper illustrate this point. Both concern very common visualization problems, an both contain significant new insights suggesting that we are only beginning the process of identifying perceptually optimal data display methods. We need more research into the perceptually optimal ways of displaying patterns in data and many more discoveries surely await. Ultimately we should have perceptually validated guidelines for all major data types and data structures, and the common patterns found within them.

We have here suggested that information foraging theory can provide a framework for studying intermediate scale cognition that can be used to identify sub-optimal behavior patterns. In this paper we have shown how this theory can be applied to the domain of high performance computing with an analytics game designed to model, in simplified form, the human-computer system involved in the development of large scientific codes.

Results from our first game study revealed that participants adjusted their behavior according to the predictions of optimal foraging theory. The results showed that while participants optimized their behavior in some respects, there were also ways in which they behaved sub-optimally. As predicted by theory, there is a large cogntive cost involved in long image load times. The number of images examined under the different conditions varied according to the payoffs and costs; in the long latency conditions, participants look at fewer than half the images before starting a new run compared to the short latency conditions.

The study also revealed an overall sub-optimal behavior. Under all conditions participants looked at approximately twice as many images as they should have to maximize their rewards. One of the advantages of in-situ production of visualization products is that, with the right viewer, load times can be drastically reduced, and tools can be made to support highly interactive tools. These can greatly increase cognitive productivity. The studies we report here suggest that these methods should be availabe at the model development phase as well as at the final stage when large scale model runs are interpreted.

Experiment 2 showed that there can be an additional large cognitive cost if there is working memory burden. In the short load time conditions participants earned almost twice as much if the images were presented sequentially as opposed to side by side (recall that earnings are a proxy for cognitive productivity in these studies). We recommend that wherever possible visual comparisons should be made with side-by-side presentation of data. This points to a need to easily save images viewed during data analysis together with a viewer allowing them to be easily and rapidly juxtaposed in varios combinations.

We conclude this paper with some comments on the broader applicability of the analytics game methodology. Gaining insights into sub optimal behaviors and ultimately understanding and remediating them may be one of the most useful applications of analytic games. In particular, by providing perspective on the behavior of analysts, this research could reduce the amount of storage or energy used at extreme scale. We typically optimize large computing systems for technical reasons — cooling, reliability, etc. — and not for cognitive optimization, though the impacts on cost could be enormous.

The earliest use of simple games to understand problem solving strategies comes from the field behavioral economics. The work of Kahneman and Tversky [6] has transformed the field of economics by showing that people are not the rational actors that were axiomatic in prior economic models. Instead they avoid risk in a way that is strongly sub-optimal. These insights came from simple behavioral experiments with non-expert participants, such as undergraduate students in the present study.

A major criticism of the use of simple games in economics [13] has been that they may not apply to complex real world situations. This criticism can also be applied to the study we report here; we

DOI: 10.2352/ISSN.2470-1173.2016.16HVEI-130

cannot know for sure that simplified games will generalize. Nevertheless, we believe that they are useful tools, especially where there are few alternatives. In general it is not possible to conduct rigorous detailed evaluations of performance when the systems are complex because the number of execution paths in even quite simple analytics systems is typically vast. When this is combined with an even greater branching in the thought processes of an analyst, the result is unpredictable system behavior with few measurable outcomes. In addition, real world systems usually cannot be experimentally manipulated because of the costs that would be involved in diverting busy professionals from productive work.

We believe that the analytics game methodology may have broad application in uncovering dysfunctional cognition at a system level and evaluating the effects of remediation strategies. To be applicable, however, a number of criteria must be met. Variables of theoretical interest must be captured by the game. The rules and goals of the game should be clear to the player. It must be possible to play the games in a relatively short period of time. Having actual rewards is important; for these studies we departed from our usual practice of paying study participants a fixed amount, instead we made the reward contingent on performance. This was necessary because the games require a clear incentive to optimize behavior.

Games can be used to examine the effects of different cost/benefit tradeoffs. In the two games reported here, "runs" had a cost only in the sense that they prevented earnings. In actual high performance computing environments, however, there are very large costs — in energy, dollars, and person-hours — involved in operating the machines. Currently, the users of HPC usually have a fixed allocation or resources, in the form of millions of CPU hours. They typically optimize these CPU hours based on the "experiments" or "runs" they need to complete, to help with some analysis task. In future work we plan to develop games where the goal is to optimize the entire system, minimizing the ratio of overall costs to insights, where the overall costs include both human and machine components.

Acknowledgements

We gratefully acknowledge funding from the US Department of Energy DE-SC0012438.

References

[1] J. Ahrens, Jourdain, S., O'Leary, P., Patchett, J., Rogers, D. H., & Petersen, M. An image-based approach to extreme scale in situ visualization and analysis. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 424-434). IEEE Press. 2014.

[2] F P Brooks, Grasping reality through illusion—interactive graphics serving science. In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 1-11. ACM, 1988.

[3] N. Fabian, Moreland, K., Thompson, D., Bauer, A.C., Marion, P., Geveci, B. Rasquin, M. and Jan, K.E., October. The paraview coprocessing library: A scalable, general purpose in situ visualization library. In Large Data Analysis and Visualization (LDAV), 2011 IEEE Symposium on (pp. 89-96). IEEE. 2011.

[4] W.D. Gray, & Fu, W. (). Soft constraints in interactive behavior: The case of ignoring perfect knowledge in-the-world for imperfect knowledge in-the-head. Cognitive Science, 28(3), pp. 359-382. 2004.

[5] W.D. Gray, Simms, C. R., Fu, W. T., & Schoelles, M. J. The soft constraints hypothesis: A rational analysis approach to rource allocation for interactive behavior. Psychological Review, 113, 461–482. 2006.

[6] D. Kahneman, and Tversky, A., Prospect theory: An analysis of decision under risk. Econometrica: Journal of the Econometric Society, pp.263-291. 1979.

[7] D.H. Laidlaw, Kirby, R.M., Jackson, C.D., Davidson, J.S., Miller, T.S., Da Silva, M., Warren, W.H. and Tarr, M.J., Comparing 2D vector field visualization methods: A user study. Visualization and Computer Graphics, IEEE Transactions on, 11(1), pp.59-70. 2005.

[8] Z. Liu and Heer J. The Effects of Interactive Latency on Exploratory Visual Analysis IEEE Transactions on Visualization and Computer Graphics, 20 (12), 2014

[9] Z. Liu, B. Jiang, and J. Heer. imMens: real-time visual querying of big data. Computer Graphics Forum, 32(3pt4):421–430, 2013.

[10] P. Pirolli, & Card, S. Information foraging. Psychological review, 106(4), 643. 1999.

[11] P. Pirolli, Information Foraging Theory: Adaptive Interaction with Information, Oxford University Press. 2007.

[12] M. Plumlee and Ware, C. Cognitive costs of zooming versus using multiple windows. ACM Transactions on Applied Perception. 13(2) 1-31. 2006.

[13] M. Rabin. "Daniel Kahneman and Amos Tversky," in Warren J. Samuels, ed. American Economists of the Late Twentieth Century Edward Elger. 1996

[14] S. W Skillman., M S. Warren, M J. Turk, R H. Wechsler, D. E. Holz, and P. M. Sutter. "Dark Sky Simulations: Early Data Release." arXiv preprint arXiv:1407.2600 2014.

[15] H. Simon. The Sciences of the Artificial. MIT Press, Cambridge, Mass, 1st edition. 1969.

[16] E.K. Vogel, E. Woodman, G. F., & Luck, S. J. Storage of features, conjunctions, and objects in visual working memory. Journal of Experimental Psychology: Human Perception and Performance, 27, 92-114. 2001.

[17] C Ware, & Plumlee, M. D. Designing a better weather display. Information Visualization, 12(3-4), 221-239. 2013.

[18] C. Ware, and Arsenault, R. Target Finding with a Spatially Aware Hand-Held Chart Display, Human Factors. 54(6) 1040-1052. 2012.

[19] J. Woodring, Petersen, M.; Schmeisser, A.; Patchett, J.; Ahrens, J.; Hagen, H., In Situ Eddy Analysis in a High-Resolution Ocean Climate Model, Visualization and Computer Graphics, IEEE Transactions on , vol.22, no.1, pp.857-866, 2016

[20] H Yu, Wang, C., Grout, R.W., Chen, J.H. and Ma, K.L., In situ visualization for large-scale combustion simulations. IEEE Computer Graphics and Applications, (3), pp.45-57. 2010.

Author Biography

Colin Ware is the Director of the Data Visualization Research Lab, part of the Center for Coastal and Ocean Mapping at the University of New Hampshire. He has a PhD in Perceptual Psychology (Toronto, 1980) and an MMath in Computer Science (Waterloo, 1985).