Visualization Viewpoints

Editor: Theresa-Marie Rhyne

Designing a Visualization Framework for Multidimensional Data

Brent Dennis, Sarat Kocherlakota, Amit Sawant, Laura Tateosian, and Christopher G. Healey

North Carolina State University

Interest in visualization has grown in recent years, producing rapid advances in the diversity of research and in the scope of proposed techniques. Much of the initial focus in computer-based visualization concentrated on display algorithms, often for specific domains. For example, volume, flow, and terrain visualization techniques have generated significant insights into fundamental graphics and visualization theory, aiding the application experts who use these techniques to advance their own research. More recent work has extended visualization to abstract data sets like network intrusion detection, recommender systems, and database query results.

Although display algorithms are a critical component in the visualization process, they are not the only issue to consider. More and more, we see visualization as a path from data to understanding. From this perspective, two obvious questions arise:

- What should we do before we display the data?
- What can we do after the user views the data?

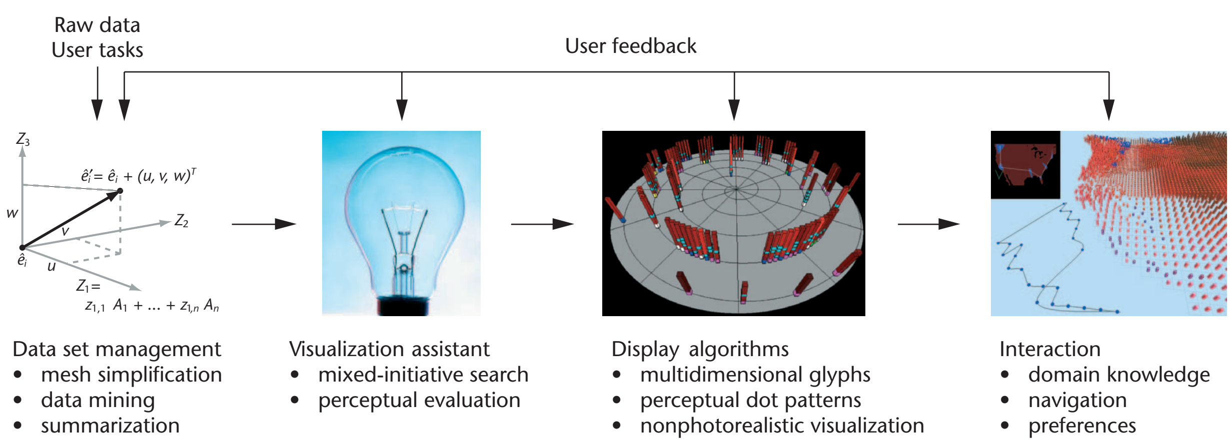
This is not a new idea, of course. Our work is motivated by others in the community, including methods to integrate data management into visualization, metadata generation and management, techniques to preprocess data to extract and display critical details, and intelligent systems that help users design effective visualizations. This article describes our initial end-to-end system that starts with data management and continues through assisted visualization design, display, navigation, and user interaction (see Figure 1). The purposes of this discussion are to

- promote a more comprehensive visualization framework;
- describe how to apply expertise from human psychophysics, databases, rational logic, and artificial intelligence to visualization; and
- illustrate the benefits of a more complete framework using examples from our own experiences.

The goal of this article is not to advocate our particular framework or its components over existing systems (such as SCIRun, GraphViz, or vtk). The examples we describe could be modified or extended within the framework, or replaced entirely with methods better designed to handle specific data of interest to a user. What we are proposing is that promising techniques beyond the display of information be studied in the context of visualization.

Data management

Visualization begins by considering the data and the tasks users want to perform on the data. The need to manage raw data prior to visualization is recognized as an



1 One possible visualization framework, including data set management, assisted visualization design, display, and interaction.

Published by the IEEE Computer Society

important problem. Even the most sophisticated display technique can be overwhelmed by a sufficiently large data set. Researchers might need basic data structures, database management, and even optimized networking protocols, for example, in situations where the data set will not fit in main memory. Properties of real-world data like missing values, noise, uncertainty, and duplicates often exist and must also be handled efficiently.[1]

Initial suggestions for managing data included the use of consistent data models like the Common Data Format, Hierarchical Data Format, ArcInfo shape files, and so on. Using this approach would let us store, query, and filter data using a common library of routines prior to visualization. Researchers proposed different methods for performing these operations, including directed graphs where edges represent data flow and nodes represent data processing, and spreadsheet-based systems that apply different operators to the data. A separate set of investigations studied ways to combine relational database engines and visualization systems (such as Tioga-2[2]). This allowed users to harness the storage and query power of a relational database during visualization.

We are conducting numerous studies to try to manage data prior to visualization. One project uses data mining classification algorithms to compress a data set.[1] These algorithms analyze a training set to build rules that subdivide multidimensional data elements into different categories. The algorithms can then apply the rules to assign membership to unclassified elements. Users can select subsets of the data by category to reduce the number of elements to visualize. The rules can also help compress dimensionality, for example, by identifying and combining dependent attributes into a single composite value. Experimental results show that different classification algorithms perform differently in the presence of errors, incomplete information, and noise in the training set. These findings act as guidelines to indicate when data mining classification might help, and to identify which algorithms are best-suited to the particular type of data being analyzed.

A separate investigation is studying the use of feature-preserving mesh simplification for data compression.[3] Multidimensional data is converted into a triangular mesh where each vertex corresponds to a data element. Attribute values are encoded as surface features stored at each vertex. The system can then apply feature-preserving simplification algorithms to reduce the mesh's size. Spatial regions with little variation in their attribute values reduce to a few vertices, or data elements. Areas with high variation remain intact (see Figure 2).

Results from a multidimensional weather data set showed that we could reduce the number of elements by 90 percent or more, while still guaranteeing less than a 1 percent error in the reconstructed results. Our data management technique is independent of the particular simplification algorithm we used. This means that we can take advantage of improved algorithms as they are proposed.

2 Multidimensional weather data over Europe and Asia reduced with our feature-preserving simplification system. In this example we are applying Hoppe's quadric mesh simplification algorithm. Color represents temperature (blue for cold to red for hot), size represents pressure (larger for higher), and orientation represents precipitation (vertical for light to horizontal for heavy). Dense glyph regions show areas with rapid spatial variation in attribute or terrain values, sparse regions show areas of slow variation.

Assisted visualization design

Constructing effective visualizations is nontrivial, even for visualization experts. We must consider the data to be visualized, the user's analysis tasks, and the strengths and limitations of different display techniques to arrive at a satisfactory result. Moreover, we often want to build multiple visualizations for a given data set because seeing the data from different perspectives can highlight different aspects of its structure. We can ease this burden by building a system that helps both designers and users construct effective visualizations.

Researchers have proposed different ways to choose visualizations automatically or semiautomatically, for example, rule-based colormap selection[4] or design galleries.[5] Our current solution to this problem is a mixed-initiative software tool called ViA (short for Visualization Assistant).[6] ViA uses rules of perception derived from psychophysical experiments to build visualizations that map individual data attributes to different visual features (for example, hue, luminance, or size). This data-feature mapping can then be applied to visualize multidimensional data elements.

ViA uses artificial intelligence search algorithms to rapidly identify high-quality data-feature mappings. It designs and evaluates each mapping based on

- data characteristics like attribute domain (continuous or discrete, ordinal or nominal) and spatial frequency;
- viewer characteristics like the relative importance of each attribute and the basic analysis tasks to be performed on the different attributes; and
- guidelines from human perception that define how we perceive different properties of color, texture, and motion in a visualization display.

The first two constraints represent application-independent input about the data and the viewer's analysis needs. The third constraint ensures that the resulting visualizations are perceptually sound.

Researchers are compiling a large body of knowledge on how the low-level human visual system sees basic image features. These include properties of color like hue and luminance; properties of texture like size, orientation, and density; and properties of motion like flicker, direction, and velocity.[7] Results come from numerous disciplines including cognitive psychology, computer vision, and visualization.

We use controlled experiments to study these features, first in isolation to determine their fundamental ability to encode information, and then in combination to identify visual interference patterns. Results form guidelines for each feature, including the type of data and tasks it can best support, its perceptual salience relative to other features, and so on.

ViA stores each feature's strengths and weaknesses in an evaluation engine. A data-feature mapping is decomposed into a set of data attribute-to-feature pairs (one attribute mapped to hue, one attribute mapped to size, and so on). Each evaluation engine can then determine how well its visual feature supports the particular attribute it maps to, returning a normalized evaluation weight to summarize the analysis. For poor pairings, an engine will also suggest hints on how its attribute could be better visualized.

ViA's search algorithm collects results for each attribute-feature pair in a data-feature mapping. Nonconflicting hints are chained together to produce strategies on how to improve the current mapping. The algorithm assigns each chain an expected improvement weight, then places the chain on a priority queue. The chain with the highest weight is removed from the queue and applied to produce a new mapping. This mapping is similarly evaluated. Prioritized hint chains let ViA restrict

Success Stories

A critical step in our visualization design is the application of a technique to real data. We have collaborated with colleagues from areas like oceanography, e-commerce, aviation, and astrophysics. Results range from anecdotal reports to statistical analyses of different performance metrics during controlled validation experiments. We briefly highlight two examples.

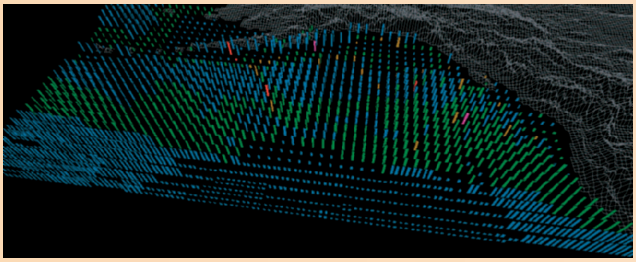
Salmon migration simulations

Marine biologists at the University of British Columbia are studying the movement, feeding patterns, and migration routes of salmon in the northern Pacific Ocean. Salmon begin their life as fry in freshwater streams and tributaries, spending approximately one year feeding before swimming downstream to the ocean. They spend two to four years in the ocean, then return back to their exact location of birth to spawn and die. Salmon use smell to identify which rivers and tributaries to follow to return to their spawning grounds. What they do in the open ocean is less clear, however. Temperature, salinity, currents, and the availability of plankton food sources impact where the fish are located. Readings for these properties are critical when designing and testing migration models.

We can derive ocean temperatures and current directions from satellite images. Plankton densities are more difficult to obtain. The normal method for recording plankton counts is through physical probes set from ships. This makes it difficult to cover the entire region of interest, particularly during winter months when conditions are dangerous over much of the northern Pacific Ocean. Previously, scientists used bilinear interpolation of known plankton densities to estimate missing values. Unfortunately, this often produces poor results, particularly for months where few known values are available.

We applied our data mining algorithms to estimate unknown plankton densities. Locations with known values formed a training set with independent variables of month, year, latitude, longitude, sea surface temperature (SST), current direction, current strength, and classification-variable plankton density. Based on feedback from our biology colleagues, we discretized continuous values into interval ranges.

Our data mining algorithms identified month as the most important classification attribute, followed by current strength and SST. The other attributes were not significant to determining plankton density. The biologists concurred with these results; plankton densities display a seasonal variability, large current upwellings produce larger plankton blooms and higher ocean temperatures cause faster plankton production and higher densities. The classification rules allowed us to use a sample point's month, current strength, and SST to estimate a plankton density range. We fed these results into our perceptual visualization system to display real and estimated plankton densities, current strengths, and SSTs for any month and year of interest (see Figure A). Experimental validation and domain expert feedback



A **Visualization of ocean conditions for February 1956. Color represents plankton density (blue, green, brown, red, and purple for lowest to highest), height represents current strength, and density represents the sea surface temperature.**

its search to paths that have the highest probability of producing better mappings. Search continues until a predefined stopping condition, at which time the top *k* mappings are presented. Viewers can select from the different mappings to visualize their data in different ways. They can modify the input conditions and ask ViA to continue searching. Viewers can also lock attribute-feature pairs in a mapping, then ask ViA to identify new results based on these additional constraints. This lets viewers collaborate with ViA to iterate a final set of high-quality, perceptually salient visualizations.

Display

Display algorithms for visualization is an area of significant research (a comprehensive overview of visualization display techniques is beyond the scope of this article; see elsewhere[7] for more information). The methods studied in our group are based on perception, either through suggestions from ViA or from other models of how we perceive visual information. The resulting visualizations are optimized for a user's data and tasks. They also harness the capabilities of a low-level visual system to guarantee rapid and accurate visual analysis. Data exploration tasks can usually be completed in 200 milliseconds per image or less. We present three separate examples to demonstrate how we are applying this knowledge to construct effective visualizations.

The first technique is common in visualization: the use of multidimensional glyphs—simple 2D or 3D geometric objects that can vary their visual appearance to encode information. The visual features our glyphs support are exactly the ones that we have studied during our psychophysical experiments: properties of color, texture, and motion. For example, a square patch (a 2D glyph)

confirmed that the data mining classification produced better results than bilinear interpolation. Salmon migration patterns simulated with the new plankton fields were more accurate when compared to historical records of migration distributions across different approaches to the river entrance.

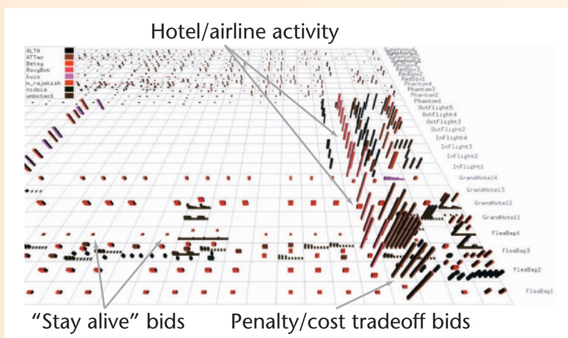
E-commerce auction agents

Researchers in artificial intelligence and e-commerce are designing intelligent agents capable of purchasing items for a human client through online auctions. To bid effectively against human and software competitors, an agent must adapt to the specific rules that characterize different auctions.

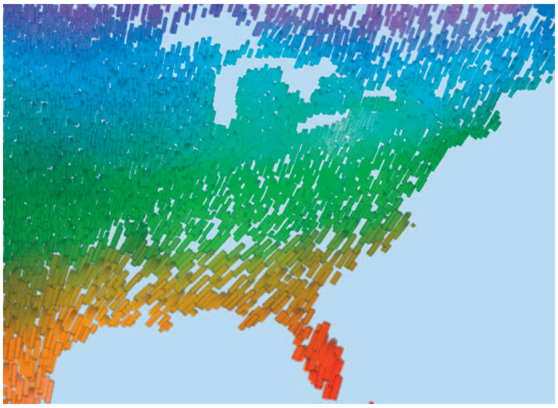
The University of Michigan runs the Trading Agent Competition (TAC), a simulated auction environment that lets agents test their strategies and compete against one another. Each agent is tasked with building vacation packages of airline tickets, hotel rooms, and entertainment tickets. Different auctions have different rules. Airline auctions have an unlimited number of seats for each day, with prices fluctuating randomly. Hotel auctions have a fixed number of rooms for each day. The daily room price is set in a Dutch auction fashion, where all winning bids pay the lowest winning price. Hotel auctions close early if a predetermined period of inactivity occurs. Each agent is allocated a random selection of entertainment tickets. Agents trade with one another in stock market-type auctions to secure the entertainment tickets they need.

We used ViA to search for TAC visualizations. We chose *x*-position for a bid's time, *y*-position for its auction, color for the agent ID, height for price, and width for quantity. A simulation forms a rectangular grid of cells, with rows for different auctions and columns for individual time steps. Rectangular towers within a cell visualize a bid made in the given auction at the given time, with the tower's color, height, and width defining the agent making the bid, bid price, and bid quantity (see Figure B). The visualization can run in real time as the simulation unfolds, or it can be used to postprocess simulation logs.

Each year an online competition is held to identify the top TAC agents, who then compete at an annual conference. Our visualizations were first used during TAC competitions at the 4th International Conference on Multi-Agent Systems (ICMAS 2000). Although researchers were already graphing activity and analyzing simulation logs, we immediately highlighted a number of new insights into agents' strategies. Most agents deferred purchasing airline and hotel tickets until the end of the simulation, implying there is no advantage to an early purchase. Agents ping hotel auctions with low-cost bids to keep them alive. Finally, some agents took advantage of the Dutch auction nature of hotel auctions to bid in ways that guaranteed securing hotel rooms, but with a high probability of paying much less than their actual bid price. We continue to update our visualizations for new auction types being run through the TAC.



B Visualizing e-commerce agent activity: *x*-axis represents the time, *y*-axis represents the auction, color represents the agent ID, height represents the price, and width represents the quantity.



3 Weather conditions over the eastern US visualized with simulated paintbrush strokes. Color represents temperature (purple/blue for cold to orange/red for hot), density represents wind speed (denser for stronger), orientation represents precipitation (tilted for heavier), size represents pressure (larger for higher), and luminance represents clouds (brighter for heavier).

can change its hue, luminance, size, orientation, and position. It can flicker on or off and move with a particular direction and velocity. The spatial density of patches can vary throughout the display. We have used this technique to visualize both physical and abstract data(for example, the weather conditions shown in Figure 3). Another system animates glyphs to visualize correspondence between queries on an underlying database (for example, movie suggestions from the MovieLens recommender system, third image in Figure 1).

A second algorithm uses our perception of orientation to visualize 2D flow fields. A small collection of dots is seeded throughout the flow field. The dots are carefully positioned so that their locations relative to one another generate perceived orientations that match the flow directions in the flow field (see Figure 4). Ken Stevens proposed the model that positioned the dots.[8] He hypothesized that for any target dot, the dominant orientation formed from pairs of neighboring dots will be the orientation perceived at the target dot. Stevens conducted experiments and designed a software system to validate his ideas. Our visualization algorithm reverses this process. Rather than determining the perceived orientations for a fixed dot pattern, we generate a collection of dots to produce a set of orientations that match an underlying flow field.

A third technique, motivated by nonphotorealism in computer graphics, involves building aesthetic visualizations.[9] We wanted to determine whether aesthetic representations improve a visualization's effectiveness. Our interest is based in part on investigations of orientation and engagement in human perception. Orientation automatically attracts a viewer's focus of attention to a particular location in an image. Guidelines on the use of low-level vision can help to control orientation, allowing us to direct a viewer's gaze to important areas in a visualization. Following orientation, a viewer's attention can engage at a location, examining various details of the image in more depth. We hypothesize that an aesthetic representation will increase engagement, encouraging the viewer to remain at the chosen focus of attention. This might increase the amount of detail viewers are able to remember and recall.

We designed a simple correspondence model to match visual features in a glyph-based visualization to brushstroke properties in a nonphotorealistic image. This allows us to use a data-feature mapping to render a set of brushstrokes that produce a painting of the data elements stored in a multidimensional data set (see Figure 3). We are conducting experiments to identify the emotional and compositional image properties that affect aesthetic judgment. We can then investigate whether we can vary these properties to produce perceptually effective and visually appealing visualizations.

Interaction and navigation

Most visualization systems let users perform standard types of interaction with the data as it is displayed, for example, dynamic camera positioning and the ability to change the data-feature mapping as the data is rendered. We hope to move beyond these basic operations so that users can collaborate with our visualization systems. This allows the computer and the user to combine their different strengths: the computer's ability to rapidly analyze, search, apply rules, and render data, and the user's ability to interpret patterns and relationships, understand external details about the data being studied, and apply specialized domain knowledge.

Components in our framework include interfaces that enable many default options to be changed. The individual systems understand how to communicate with one another to share information and results. The ability to program or script the framework has been proposed as a useful option (such as in SCIRun or vtk), and one that we plan to study further.

A related issue is the need to include application context in a visualization. ViA tries to support this by allowing a user to lock subsets of a data-feature mapping. For example, the user could tell ViA, "In our visualizations this attribute must be displayed with color." ViA will search for visualizations that satisfy the user's constraints. ViA can also identify what penalties, if any, are incurred by reporting differences in evaluation weights between the best mappings both with and without the constraints. This lets users make an informed decision about how they want to proceed.

When visualizing a large data set, we can often display only a small fraction of its content in detail. This problem is the focus of significant research in information visualization. Researchers have proposed many powerful methods to display both a high-level overview and low-level detail simultaneously. Examples include focus+context and overview+detail algorithms like the fisheye lens or the hyperbolic tree, and hierarchical level-of-detail techniques like treemaps. Our own dot-based flow visualization algorithm uses a hierarchical level-of-detail decomposition to visualize flow fields that are too large to fit in a single screen.

Another project studies this issue in the context of navigation.[10] Users are often interested in a dynamically changing subset of the elements in a data set. These elements are identified with Boolean logic rules provided by the user. Spatially neighboring elements of interest are clustered into local groups, then connected with a minimum-length graph. A global spanning tree is constructed to bind the local graphs together. The graph structure is presented in an inset map, allowing users to decide where in the data set they want to explore (fourth image in Figure 1). Animated camera paths can be built and run on the local graphs and the global spanning tree to tour through an area of interest, or to see an overview of locations that users may want to investigate in more detail.

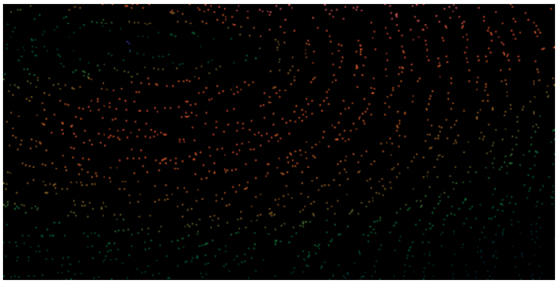
Although effective, requiring users to explicitly define the rules that identify elements of interest has its drawbacks. It's time consuming, and the rules are static. If a user's interests change during visualization, the rules must be updated by hand to reflect these new discoveries. It might also be difficult for a user to characterize the exact properties of an element that make it interesting. We hope to apply preference elicitation algorithms to augment the user's input by implicitly determining additional rules of interest. Preference elicitation is an area of artificial intelligence that studies how to structure and order different scenarios based on a user's preferences or interests. Actions performed by the user provide evidence to order different scenarios. Questions might be posed to the user to clarify ambiguous situations, or to significantly improve the current ordering. In our research, scenarios correspond to different data-feature mappings (that is, different visualizations). Possible sources of evidence include watching where users move in the visualization, tracking the data-feature mappings they choose, and allowing elements to be interactively selected on-screen, but without specifying what makes the element interesting.

Future directions

Methods to compress and summarize data, to generate perceptually salient visualizations, and to navigate within the data have helped our domain experts make new discoveries that were not found with existing analysis and display tools. The use of perceptual rules at various stages of the framework is particularly helpful for presenting large amounts of data in ways that support rapid and accurate comprehension.

Techniques from areas like data management, assisted design, and user interaction can lead to more effective visualizations. We encourage the visualization community to pursue research that further enhances our knowledge and understanding of how these ideas fit within a visualization context.

We continue to focus on preparing data for visualization, displaying data effectively, and assisting users in their explorations and interactions. We are currently studying ways to mathematically compress, classify, and summarize data prior to its visualization. We are investigating how to apply guidelines from art theory and art history to produce better visualizations. Finally, we are extending our visualization knowledge to new application areas like gene and transcript exploration, analyzing astronomical scans, terrain simplification, and bioterrorism prevention. ■



4 Visualizing flow in a simulated supernova collapse, relative dot positions represent direction and color represents magnitude (blue for low to red for high).

Acknowledgment

This work is funded in part by the National Science Foundation grants IIS-9988507, ACI-0083421, and ACI-0092308.

References

1. C.G. Healey, "On the Use of Perceptual Cues and Data Mining for Effective Visualization of Scientific Datasets," *Proc. Graphics Interface 98*, Morgan Kaufmann, 1998, pp. 177-184.
2. A. Aiken et al., "Tioga-2: A Direct Manipulation Database Visualization Environment," *Proc. 12th Int'l Conf. Data Engineering*, IEEE CS Press, 1996, pp. 208-217.
3. J.D. Walter and C.G. Healey, "Attribute Preserving Dataset Simplification," *Proc. IEEE Visualization*, IEEE CS Press, 2001, pp. 113-120.
4. L.D. Bergman, B.E. Rogowitz, and L.A. Treinish, "A Rule-Based Tool for Assisting Colormap Selection," *Proc. IEEE Visualization*, IEEE CS Press, 1995, pp. 118-125.
5. J. Marks et al., "Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation," *Proc. Siggraph*, ACM Press, 1997, pp. 389-400.
6. C.G. Healey, R. St. Amant, and J. Chang, "Assisted Visualization of E-Commerce Auction Agents," *Proc. Graphics Interface*, Morgan Kaufmann, 2001, pp. 201-208.
7. C. Ware, *Information Visualization: Perception for Design*, Morgan Kaufmann, 2004.
8. K.A. Stevens, "Computation of Locally Parallel Structure," *Biological Cybernetics*, vol. 29, 1978, pp. 19-28.
9. C.G. Healey et al., "Perceptually-Based Brush Strokes for Nonphotorealistic Visualization," *ACM Trans. Graphics*, vol. 23, no. 1, 2004, pp. 64-96.
10. B.M. Dennis and C.G. Healey, "Assisted Navigation for Large Information Spaces," *Proc. IEEE Visualization*, IEEE CS Press, 2002, pp. 419-426.

Readers may contact Christopher G. Healey at healey@csc.ncsu.edu.

Readers may contact Theresa-Marie Rhyne at tmrhyne@ncsu.edu.