Human Factors In Visualization Research

Melanie Tory and Torsten Möller

Abstract—Visualization can provide valuable assistance for data analysis and decision making tasks. However, how people perceive and interact with a visualization tool can strongly influence their understanding of the data as well as the system's usefulness. Human factors therefore contribute significantly to the visualization process and should play an important role in the design and evaluation of visualization tools. Several research initiatives have begun to explore human factors in visualization, particularly in perception-based design. Nonetheless, visualization work involving human factors is in its infancy, and many potentially promising areas have yet to be explored. Therefore, this paper aims to 1) review known methodology for doing human factors research, with specific emphasis on visualization, 2) review current human factors research in visualization to provide a basis for future investigation, and 3) identify promising areas for future research.

Index Terms—Human factors, visualization, perception, cognitive support, methodology.

✦

1 INTRODUCTION

MODERN technology provides access to large quantities of data in many application domains, such as medical imaging, fluid flow simulation, and geographic information systems (GIS). The complexity of the data can make analysis a challenging cognitive activity. Ware defines visualization as "a graphical representation of data or concepts," which is either an "internal construct of the mind" or an "external artifact supporting decision making."[1] In other words, visualizations assist humans with data analysis by representing information visually. This assistance may be called *cognitive support*. Visualizations can provide cognitive support through a number of mechanisms, as summarized in Table 1. These mechanisms can exploit advantages of human perception, such as parallel visual processing, and compensate for cognitive deficiencies, such as limited working memory.

1.1 Terminology: Continuous and Discrete Model Visualization

Visualization can be valuable in a wide variety of application domains. Visualization techniques have been traditionally categorized into two major areas:

- "scientific visualization," which involves scientific data with an inherent physical component, and
- "information visualization," which involves abstract, nonspatial data.

This terminology is somewhat ambiguous. For example, mathematical functions (e.g., $f(x) = x^3$) are scientific, meaning they should belong under "scientific visualization," but not necessarily physically based, so they also belong under

"information visualization." We introduce new terminology that is more precise. *Continuous model visualization* encompasses all visualization algorithms that use a continuous model of the data (i.e., the algorithm assumes that the phenomenon being studied is continuous, even if the data values are discrete) and is roughly analogous to "scientific visualization." *Discrete model visualization* includes visualization algorithms that use discrete data models and roughly corresponds to "information visualization." A complete description and justification of this terminology is beyond the scope of this paper. Refer to Tory and Möller [74] for details.

1.2 History of Human Factors in Visualization Research

Simply finding a graphic technique to display all the data may not provide adequate support for a user's task. Fig. 1 shows how users are an integral part of the visualization process, especially when the visualization tool is interactive.

Furthermore, Rheingans suggests that interaction should not be simply a "means to the end of finding a good representation" [62]. Interaction itself can be valuable since exploration may reveal insight that a set of fixed images cannot.

Human factors-based design involves designing artifacts to be usable and useful for the people who are intended to benefit from them. Unfortunately, this principle is sometimes neglected in visualization systems, particularly in continuous model visualization. The focus of most continuous model visualization research is on creating new and faster techniques for displaying data. We believe that more attention should be paid to users who must view and manipulate the data because how humans perceive, think about, and interact with images will affect their understanding of information presented visually. As a result, there is a strong need to study human factors as a basis for visualization design.

Over the last several years, interest in human factors within the visualization research community has been increasing. For example, Fig. 2 shows a slightly increasing trend within the *IEEE Transactions on Visualization and*

---

1. C. Ware, In [81, p. 1].

- *The authors are with the Computing Science Department, 8888 University Dr., Simon Fraser University, Burnaby, BC, Canada V5A 1S6. E-mail: {mktory, torsten}@cs.sfu.ca.*

TABLE 1
How Visualization Can Support Cognition

Method	Description
Increased Resources	
Parallel processing	Parallel processing by the visual system can increase the bandwidth of information extraction from the data.
Offload work to the perceptual system	With an appropriate visualization, some tasks can be done using simple perceptual operations.
External memory	Visualizations are external data representations that can reduce demands on human memory.
Increased storage and accessibility	Visualizations can store large amounts of information in an easily accessible form.
Reduced Search	
Grouping	Visualizations can group related information for easy search and access.
High data density	Visualizations can represent a large quantity of data in a small space.
Structure	Imposing structure on data and tasks can reduce task complexity.
Enhanced Recognition	
Recognition instead of recall	Recognizing information presented visually can be easier than recalling information.
Abstraction and aggregation	Selective omission and aggregation of data can allow higher level patterns to be recognized.
Perceptual Monitoring	Using pre-attentive visual characteristics allows monitoring of a large number of potential events.
Manipulable Medium	Visualizations can allow interactive exploration through manipulation of parameter values.
Organization	Manipulating the structural organization of data can allow different patterns to be recognized.

a. Modified from Card et al. [8], Table 1.3 (How information visualization amplifies cognition). Basic content of the table is the same as Card et al., but structural and wording changes have been made for conciseness, clarity, and to generalize the ideas to all types of visualization.

Computer Graphics (*TVCG*) journal to incorporate human factors ideas.

To generate this data, we reviewed all the titles and abstracts of articles over the journal's history and evaluated whether or not the article had a human factors component. Human factors contributions included utilizing theories of perception and cognition, designing a system to fit a particular task or human capability, and incorporating end users into the design and evaluation process.

Despite increasing interest in human factors, major contributions in this area are limited. Fig. 2 illustrates that, on average, only 23 percent of *TVCG* papers include a human factors component. In papers that include human factors, this component is often a small, minor part of the paper. Only six out of 217 abstracts (i.e., 2.8 percent) mentioned a user study. Studies involving humans are usually confined to the field of
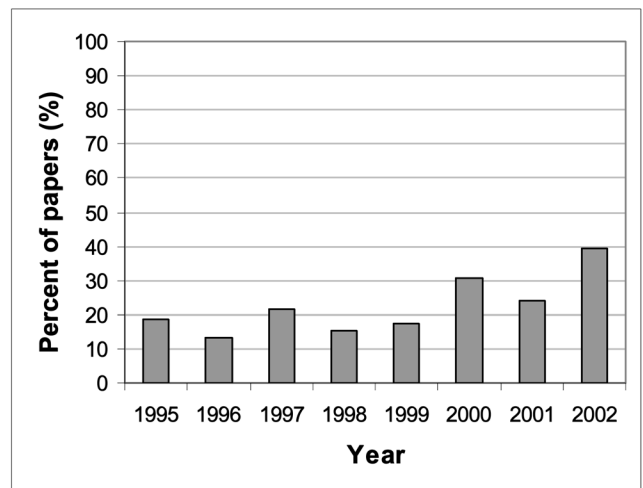

Fig. 1. The visualization process.

Fig. 2. Percent of *TVCG* papers judged to have a human factors component, plotted by year of publication.
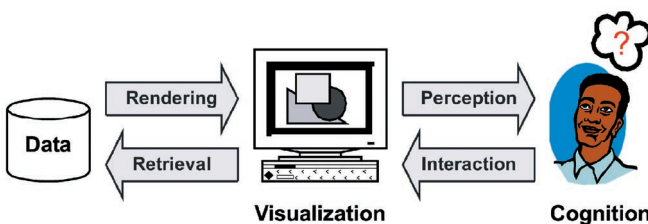
human-computer-interaction (HCI) and studies that consider specific problems of visual data presentation are rare.

This paper provides a basis for studying human factors-based design in visualization. We focus on continuous model visualization (e.g., volume and vector visualization), but ideas and examples will also be drawn from discrete model visualization and computer graphics. The remainder of the paper is organized as follows: Section 2 outlines HCI methodology to provide possible approaches for human factors-based design. Section 3 then summarizes current human factors research in visualization and Section 4 concludes by identifying research areas and methods that merit further investigation.

2 HUMAN FACTORS RESEARCH APPROACHES

The effectiveness of a visualization depends on perception, cognition, and the users' specific tasks and goals. How a viewer perceives an item in a visualization display depends on many factors, including lighting conditions, visual acuity, surrounding items, color scales, culture, and previous experience [81]. Important information may be overlooked if the user is in a hurry or cannot allocate their full attention to the visual display due to other task demands. Additionally, interactive systems will not achieve their full potential if users cannot easily interact with them.

How, then, can we determine what a user's goals are and whether a system meets them? What design guidelines should we follow? How can we evaluate whether a design succeeds? This section provides a brief summary of HCI methods that may be useful for designing and evaluating visualization systems.

2.1 User Motivated Design

At its most minimal level, human factors research is research that is motivated by users. For example, users may approach a visualization expert with a data set they want visualized; the visualization designer is then motivated to consider that specific problem. A second common example of user-motivated design involves speed issues. Visualization research frequently focuses on creating algorithms with faster frame rates so users can interactively explore visualizations.

Although the user-motivated approach can be useful for some research purposes and does consider users to some extent, the main research focus is on algorithm design, rather than human factors, and users are generally not consulted throughout the design process. For these reasons, user-motivated examples will not be considered in this review unless they additionally make use of other human factors knowledge or techniques.

2.2 User and Task-Based Design

User and task analysis determines system requirements based on tasks users need and want the system to support. For example, Springmeyer et al. conducted an extensive task analysis to find a common set of tasks in scientific data analysis [72]. Task analysis typically investigates a wide range of factors, as described by Hackos and Redish [23]:

- personal, social, and cultural characteristics of users,
- user preferences and values,
- goals (both general and specific) and how users achieve them,
- user knowledge, experience, and thought processes,
- physical environment,
- tasks to be performed with the system,
- problems users would like to see the system solve, and
- other tasks that must be performed while using the system (to give an idea of the user's cognitive load).

Task analysis allows designers to define detailed functional specifications and user interface limitations. For example, surgeons in an operating room may not be able to interact with a surgery planning tool using hand-based input devices since their hands may be busy doing other tasks. Furthermore, surgery likely imposes a large cognitive load, limiting the cognitive resources surgeons can devote to the visualization.

Structured task analysis methods include observation, interviews, and surveys of potential users. Because users may not notice what they do, may not know how to articulate what they do, and may misrepresent reality, user and task analysis methods are best carried out in the context of real work [23]. System designers can use the results of task analysis (guidelines, requirements, and constraints) to design useable and useful systems.

2.3 Perception and Cognition-Based Design

Cognition and perception theories can help designers find faults in current systems and develop new ideas that should be effective [80, Chapter 2]. An exhaustive list of these theories and design guidelines is beyond the scope of this review. Many perception-based design guidelines are described by Ware [81]. The following set of visualization-specific guidelines is based on Nigay and Vernier [54]. These generalized guidelines were developed mainly for discrete model visualization, but we believe they are also applicable to continuous model visualization:

- Since users' information needs are domain and task dependent, design must either

  1. be domain and task specific or
  2. look at domain-independent subtasks such as those defined by Shneiderman [71, Chapter 15]: overview, zoom, filter, details-on-demand, relate, history, and extract.

- To support users with different tasks and requirements, multiple visual representations of the data should be available. Several representations may be visible at once using multiple view windows. If it is not possible to render a global view of the data set in which every element is precisely represented, it is possible to combine detailed, partial representation(s) with vague, global representation(s). For example, in a medical imaging data set, slices and subvolumes of specific areas could be combined with a volume rendered overview of the entire volume.

- Changing between representations and views should be easy.
- Using multiple views is not always appropriate. Baldonado et al. describe a set of guidelines for when and how to utilize multiple views for visualization tasks [2].
- Continuity should be maintained so the user does not get lost when switching between representations. Woods provides several design guidelines to help provide such continuity or "visual momentum" [82]. (For example, use graceful transitions such as animation, maintain formatting consistency across views, and provide features that are easily discernible in all views and thus act as perceptual landmarks.)

- The following variables should always be visible:

  - The set of data elements (an overview). With volume or fluid flow data, the overview contains the entire object or space being visualized.
  - Relationships between data elements. Relationships may be either explicit (e.g., links between web pages) or implicit (e.g., relative positions of objects in a scene).
  - Method of locomotion. In other words, cues should be present to help the user understand how to navigate through the display and modify display parameters.
  - Details at the current location (e.g., the value of a voxel in volume data).
  - Details of the local neighborhood.
  - Navigation history. In other words, a list of previously explored display parameters, such as transfer functions in direct volume rendering (for details of this example, see Section 3.4).

- Data at the focus of interaction should be undistorted and represented at the highest possible resolution.
- Navigation tools should be reused to maintain consistent interaction metaphors throughout the system.

Design guidelines are not meant as absolute rules for all data types and applications and this list is not meant to be exhaustive. Nonetheless, design guidelines can provide guidance and suggestions for visualization developers in many application areas.

2.4 Prototype Implementation

Following system design, a prototype or complete system can be implemented and tested. Rapid prototyping techniques reduce implementation time so more designs can be tested when time and resources are limited. In HCI, rapid prototyping techniques include video prototypes, paper-based designs, story boards, and software mock-ups. For visualization, rapid prototyping techniques have not been greatly explored and certain challenges present themselves. For example, three-dimensional (3D) visualizations may be difficult to represent with paper drawings or video. Nonetheless, rapid prototyping techniques could allow users and designers to explore initial design ideas before investing time and resources to build a more complicated prototype. Furthermore, development of graphics and visualization libraries (e.g., the Visualization Toolkit [69]) is enabling designers to develop initial prototypes quicker and easier than before.

2.5 Testing

Both functionality and ease of interaction for visualization systems can be tested. Several evaluation methods are possible, as described in the following sections.

2.5.1 User Studies

User studies involve real users and allow designers to obtain both qualitative and quantitative data. Quantitative data typically measures task performance (e.g., time to complete a specific task) or accuracy (e.g., number of mistakes). User ratings on questions such as task difficulty or preference also provide quantitative data. Qualitative data may be obtained through questionnaires, interviews, or observation of subjects using the system.

Walenstein describes several challenges with formal user studies [80, Chapter 2]. They can be time-consuming, expensive, and difficult to design. Although they quickly highlight problems in interfaces (e.g., it is easy to see whether a user can find the button to perform a task), user studies do not always identify problems and benefits of visualization ideas. Benefits of the tool may be useful only to experts (who can be difficult to find or may not have time to participate in lengthy studies) or following a long practice period. Comparison of tools may produce results confounded by the many differences between the tools. Missing or inappropriate features in the test tool or problems in the interface can easily dominate the results and hide benefits of the ideas we really want to test. Thus, it seems that user studies can only be useful with an extremely polished tool so that huge amounts of time must be invested to test simple ideas that may not turn out to be useful.

One solution to this problem is to have user studies focus on design *ideas* rather than complete visualization tools and to test specific hypotheses [80, Chapter 2]. We should first use perceptual and cognitive theories to develop a design idea that is predicted to have a specific benefit. For example, we might predict that ordering data values by time will decrease time to find time trends. This translates easily into a hypothesis that can be tested. We can then develop a simple tool designed to test only this hypothesis. Our test should attempt to validate 1) whether the idea is effective and 2) why it is or is not effective. Of course, this may not be as easy as it sounds. Taking the idea out of context may render it useless or may limit our ability to generalize the results. Moreover, choosing an appropriate level of tool complexity may be a difficult decision involving many trade offs.

2.5.2 Usability Inspections

Additional evaluation methods established in HCI include *cognitive walk-throughs* (where an expert "walks through" a specific task using a prototype system, thinking carefully about potential problems that could occur at each step) and *heuristic evaluations* (where an expert evaluates an interface with respect to several predefined heuristics) [50]. Similarly, Blackwell et al. describe *cognitive dimensions*, a set of heuristics for evaluating cognitive aspects of a system [5], and Baldonado et al. designed a set of heuristics specific to
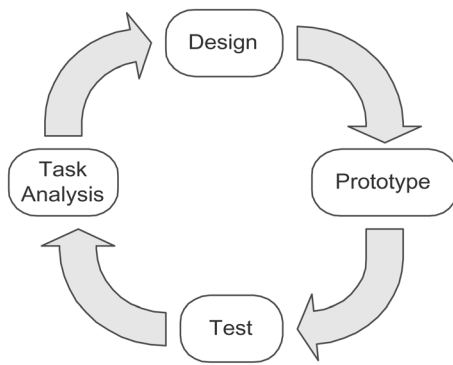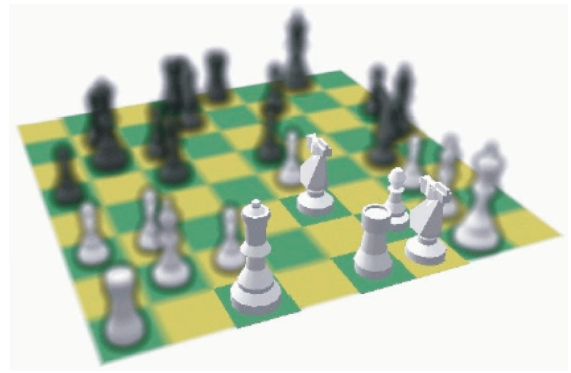
Fig. 3. Design cycle.

Fig. 4. Focusing effects can highlight information. In this example, a chess tutoring system highlights pieces that cover the knight on E3 [42]. ©2001 IEEE.

multiple view visualizations [2]. These *usability inspection* methods avoid many of the problems with user studies and may be beneficial for evaluating visualizations. However, because these techniques are (for the most part) designed for user interface testing, it is not clear how well they will evaluate visualization ideas. For example, many visualization tasks are ill-defined. Walking through a complex cognitive task is very different from walking through a well-defined interface manipulation task. Furthermore, by leaving end users out of the evaluation process, usability inspection methods limit our ability to find unexpected errors.

2.5.3 Summary of Evaluation Methods

Each evaluation method will find different types of problems in a visualization system and has different benefits and drawbacks. Different tests may be appropriate at different stages of the design/development process. Hence, a good evaluation process will likely include a variety of tests and incorporate testing throughout the design cycle.

For example, usability inspections may be useful early in the design process to evaluate whether design ideas adhere to general visualization and user interface design principles. Usability inspections may also be useful prior to a user study to identify interface problems that could affect the study results. A pilot user study could then be run to test study procedures and the system interface. Finally, a user study could be run to formally compare two or more visualization ideas, or to test specific hypotheses.

2.6 User-Centered Design

User-centered design is an iterative process involving task analysis, design, prototype implementation, and testing, as illustrated in Fig. 3. Users are involved as much as possible at each design phase. Development may start at any position in the cycle, but would typically start with an analysis of the tasks the system should perform or testing of an existing system to determine its faults and limitations.

User-centered design is more a philosophy than a specific method. Although it is generally accepted in human computer interaction, we believe this approach is not currently well-known in visualization and could support better visualization design.

Various aspects of human factors-based design have been incorporated into visualization research and development. We provide examples of these contributions throughout the next section.

3 RESEARCH EXAMPLES

Adoption of human factors methodology and stringent evaluation techniques by the visualization community is in its infancy. A number of research groups have begun to consider these ideas and incorporate them into the design process to greater or lesser extents. This section will summarize these human factors contributions.

3.1 Improving Perception in Visualization Systems

Several papers have looked at how our knowledge of perception can be used to improve visualization designs. For example, *depth of focus* is the range of distances in which objects appear sharp for a particular position of the eye's lens. Objects outside this range will appear blurry. Focusing effects can be used to highlight information by blurring everything except the highlighted objects [42]. For example, in a GIS application, all routes between two cities except for the shortest one could be blurred to highlight the best route. Similarly, Fig. 4 illustrates a chess tutoring system that can highlight pieces that threaten or cover a target. Here, the goal of blurring is to highlight information, not to focus on objects in the center of a user's field of view. Hence, the blurred objects are not necessarily at similar depths, a difference from traditional "depth of focus" effects.

3.1.1 Preattentive Processing in Visualization

Certain visual features (color, orientation, lightness, position, length, etc.) "pop-out" of an image, so that searching for them is very fast [7, Chapter 6], [81, Chapter 5]. Healey et al. take advantage of this phenomenon, called preattentive processing, to improve glyph-based multivariate data displays [26], [27], as illustrated in Fig. 5. Each data variable is mapped to a preattentive feature of a glyph so that it may be processed using preattentive visual search. Care is taken to ensure there is no interference between different preattentive features in the same display. Similarly, Ebert et al. developed procedural methods for designing preattentively distinct shapes [15] and Interrante considered how visual textures can be used for multivariate visualization [32]. An interesting future study could directly compare these approaches (either empirically or through case studies) to provide additional insight into when each is most valuable.
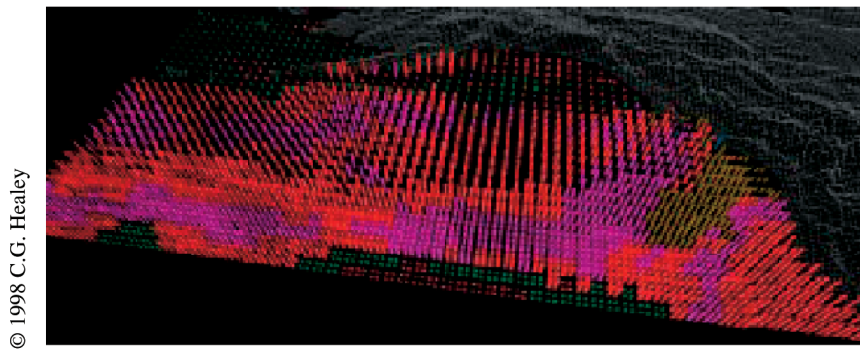
Fig. 5. Glyph-based visualization of an oceanography data set [26]. Color represents plankton density, height represents current strength, and texture density represents sea surface temperature.

Dense multivariate data displays such as Healey's glyphs represent a large amount of data in limited space. With careful design, they may allow independent features in the data to be perceived preattentively. However, in most cases, we are interested in combinations of features. For example, an ore prospector may wish to find regions where ore is plentiful, has high quality, and is easily accessible. Searches like this are usually *conjunctive*, that is, the target has no unique features. For example, we could represent ore quality by color (red for high quality, blue for low quality) and ore quantity by shape (circle for high quantity, square for low quantity). If we are looking for both high quality and high quantity, we must search for red circles among blue circles, blue squares, and red squares. This search is conjunctive since neither the color nor the shape is unique to the target. Research has shown that conjunctive search is rarely preattentive [77]; thus, the utility of dense multivariate data encoding may be limited.

3.1.2 Encoding Data With Color

Visualization systems often encode ordinal and quantitative data using intensity or color gradients. For example, topographic maps often represent elevation using a color scale and medical images use a gray-level or color gradient to distinguish tissues with different properties. However, not all mathematically linear gradients are perceptually linear (e.g., neither the mathematically linear grayscale nor the rainbow (hue) scale are perceptually linear). For this reason, several perceptually linear gradients have been developed, as described by Levkowitz and Herman [46] and Rheingans [64]. Most of these gradients are based on variations in color value and/or saturation.

Similarly, many visualizations use colors to segregate or highlight objects. For example, a medical visualization may show different organs in different colors and an air traffic control display may use color to highlight potential collisions between aircraft. Choosing colors for such displays is not easy because not all colors are equally distinguishable by observers. For this reason, Healey has developed a procedure for designing sets of easily distinguishable colors [25].

Bergman et al. generalized these ideas in a taxonomy based on principles of perception, visualization tasks, and data types [3]. Their taxonomy can be used to develop and choose effective color scales for specific data types and goals.

3.1.3 Shape Perception

For visualization with isosurfaces, understanding the 3D shape of the surface can be difficult. Surfaces are often irregularly shaped, so we have no similar shapes to compare to and depth cues from parallel lines and right angles are lacking. Three-dimensional shape perception is particularly difficult when the isosurface is semitransparent. Semitransparent isosurfaces are useful when we want to show one isosurface inside another (e.g., a physician may want to determine whether a planned radiation treatment area encloses a cancerous region while minimally affecting surrounding tissue). Texturing the surface with partially transparent textures (especially those that highlight curvature) can address this problem [31], [63] (see Fig. 6). In addition, Lum et al. show that moving particles on a surface provide similar shape enhancements [49]. An interesting future study could compare these two methods empirically to determine when each is most appropriate.

Irani et al. considered how to improve interpretation of node-link diagrams used in software design (i.e., Unified Modeling Language or UML diagrams) [33]. They replaced nodes and links with 3D primitives (geons) to take advantage of humans' ability to remember and distinguish 3D shapes and hypothesized that 3D shape diagrams would be easier to interpret. Fig. 7 compares a geon diagram to the equivalent UML diagram.

Irani et al. also tested how to best represent object types and relationships such as dependency and aggregation. In a user study testing identification of relationship types, error rates were much lower for geon diagrams than for UML diagrams. Because this idea was motivated by a real-world problem, the design was based on perceptual and cognitive theories, and the idea was tested through a carefully designed study with real users, this work provides an excellent example of how human factors methodology and ideas can be incorporated into visualization research.

Since contours play an important role in shape perception [58], 3D shape may be more easily interpreted from cartoon-style drawings than from photorealistic images. For this reason, there is current interest in developing non-photorealistic rendering (NPR) styles for continuous model visualization. For volume data, contour rendering [13], pen-and-ink style rendering [75], and volume rendering with NPR effects [48], [65] have been explored. For example, Fig. 8 illustrates how the addition of NPR effects can enhance edges in direct volume rendering. Note that the concept of photorealism here is a little obscure. Since some
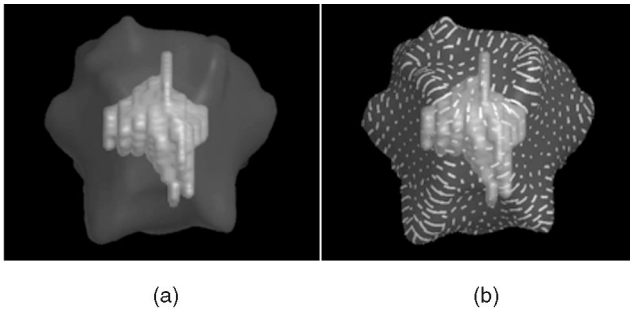
Fig. 6. Surface shape perception enhanced by texture [31]. Shape of a semitransparent outer surface (a) is more difficult to perceive than shape of a textured outer surface (b). ©1997 IEEE.

of the data we visualize cannot be seen in reality (e.g., flow vectors), it is impossible for this data to appear "realistic." However, the field of NPR also encompasses other ideas, such as specific artistic styles, the process of production being mimicked, and "the freedom not to have to reproduce the appearance of objects precisely as they are" [73].

NPR effects have also been applied to vector and tensor data, with two purposes in mind: 1) Images rendered with NPR effects, such as line drawings, may be "simpler" than photorealistic images and, thus, easier to interpret and 2) NPR effects may be able to render complicated multivariate data better than existing techniques. Salisbury et al. use oriented line-drawn textures to illustrate the direction of 2D flow vectors [67]. Healey and Enns use "paint strokes" to visualize multivalued data by mapping each data value to a property of the stroke (e.g., size, color, orientation, etc.) [28]. Similarly, Kirby et al. use layered "painting" strokes to visualize multivalued data [39]. For example, in a flow data set, a scientist may want to visualize velocity, vorticity, rate of a strain tensor, etc. We can break the data into components, map each component to a stroke, and then layer the strokes in a "painting," as shown in Fig. 8. Laidlaw et al. use a similar approach to visualize tensor data from diffusion weighted magnetic resonance imaging [44].

Using NPR styles to improve perception may hold promise, but it is still unclear how to best apply these ideas. Are all NPR styles equally effective? Are NPR styles perceived the way we expect? When are they more effective than other rendering styles? How should we choose a rendering style for a particular data set and task? Careful evaluation, possibly through user studies and heuristic evaluations, must be done before we can answer these questions and make effective use of NPR ideas for data analysis.
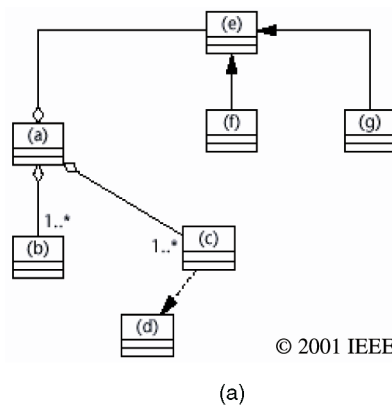
3.2 Interaction Metaphors

Interacting with 3D visualizations can be challenging because mapping movements of a 2D mouse to actions in 3D space is not straightforward. Research has shown that manipulating objects relative to each other is easier than using absolute coordinates [29]. In addition, interaction may be easier when the interface is directly related to the task through task-specific props. Examples of task-specific props for visualization are: a physical model head and clip plane that aid interaction with volumetric brain data [30] and the "Cubic Mouse," a 3D input device for volume data that allows users to navigate along major axes by moving three perpendicular rods in a physical box [19]. Development of task-specific input devices for other visualization applications (e.g., flow visualization) could make interaction easier and thereby enhance data analysis.

In addition, visualization tasks often require significant maneuvering (i.e., manipulating windows and widgets, navigating around interfaces, and managing data). For example, a scientist examining gas flow within a fuel cell may begin by examining several visual images. Generating these images may require manipulation of several windows and widgets within the visualization tool. If the scientist then decides to examine the data quantitatively, he or she may need to return to the original data files to look up values and/or switch to a different computer program in order to perform a mathematical analysis or generate statistics. Maneuvering operations can be time consuming and distract users from their ultimate goals; thus, Spring-meyer et al. suggest minimizing unnecessary navigation and integrating all the necessary tools for a task [72].
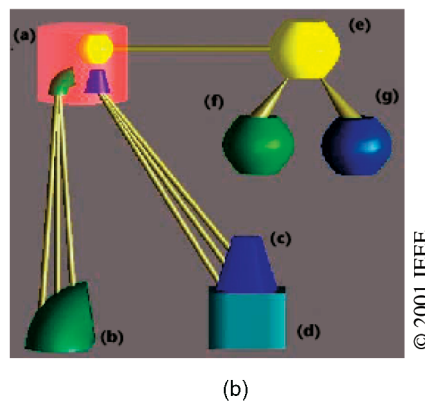
3.3 Perceptual Models for Computer Graphics

Numerous mathematical models of visual perception exist. Typical models approximate contrast sensitivity, amplitude nonlinearity (sensitivity changes with varying light level), and masking effects of human vision. Two examples are the Daly Visual Differences Predictor [14] and the Sarnoff Visual Discrimination Model [47].

Variations on these models have been used for realistic image synthesis. Researchers have developed mappings between the range of luminances in the real world and

Fig. 7. UML diagram (a) and the equivalent geon diagram (b) [33].

CT volume   CT volume with silhouette and boundary enhancements

(a)

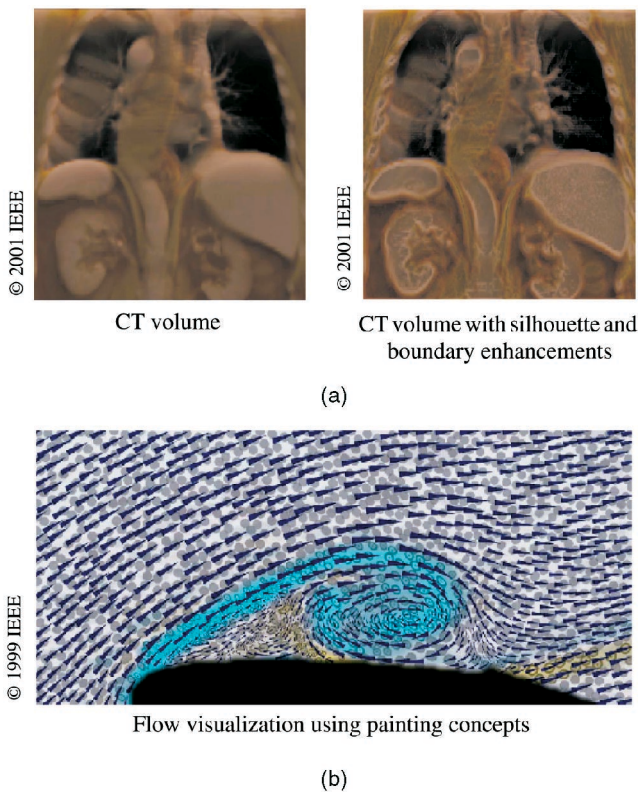Flow visualization using painting concepts

(b)

Fig. 8. NPR effects in visualization. (a) NPR enhancements highlight contours in volume data [65]. (b) Multivariate data visualization using concepts from painting [39] (Key: velocity = arrow direction, speed = arrow area, vorticity = underpainting/ellipse color (blue = clockwise, yellow = counterclockwise) and ellipse texture contrast, rate of strain = log (ellipse radii), divergence = ellipse area, shear = ellipse eccentricity).

those that can be replicated in computer displays and have used perception models to determine the mapping quality. (For example, see Greenberg et al. [22]. Other references are provided by Bolin and Meyer [6] and Pattanaik et al. [55].) Similarly, Pellacini et al. developed a perceptually meaningful model of surface gloss [57] and Ferwerda et al. [17] and Pattanaik et al. [55] developed models that simulate visual adaptation to changing light levels.

Improving realism is not too relevant to visualization because the goal is to represent data, not to display a realistic image of the world. Applications more relevant to visualization include increasing rendering speed (to enable interactive data exploration) and reducing image artifacts (to enhance perception and prevent incorrect interpretations of data). Reddy removed imperceptible details to reduce scene complexity and improve rendering speed [60]. Two possibilities were: 1) reduce detail in the periphery where vision is less sharp and 2) track the user's gaze and modify the resolution accordingly. Ferwerda et al. used perceptual knowledge to visually mask graphics artifacts (e.g., texture patterns can mask flat shading artifacts) [18]. Bolin and Meyer combined both ideas, using an adaptive rendering process that finds and fixes visible artifacts but stops increasing image quality when improvements are no longer expected to be visible [6].

3.4 Transfer Functions

In direct volume rendering, each voxel (sample in a 3D volume grid) is first classified as belonging to a particular category based on its intensity and/or spatial gradient value(s). Voxels are then assigned a color and transparency level based on this classification. The function that does this is called a *transfer function*. One example in Computed Tomography (CT) data would be to make skin semitransparent and bones opaque so the bones could be seen beneath the skin. In this case, transfer function design is quite easy since bones and skin have very different intensity values in CT data and can be easily distinguished. However, in general, finding good transfer functions is difficult and is therefore a major research area in volume visualization.

3.4.1 User Exploration of Transfer Functions

One option for finding transfer functions is to leave it up to the user. This requires an effective interface for the user to specify the function and the user must either 1) know what function is useful or 2) be able to easily experiment with different parameters. Searching for a good set of parameters can be time-consuming and frustrating because the parameter space is large, rerendering a scene after specifying new parameters can take several seconds, and user interfaces to specify parameters are not always carefully designed or tested. For example, Kniss et al. attempted to create intuitive user interface widgets for transfer function specification [40]; however, they do no user testing and do not include any users in their design process. Providing users with more information about the data, such as histograms and similar metrics [1], [40], may help them to choose good transfer functions, but no user studies have been done to verify this.

One promising idea is to provide a history tool so that users do not have to remember which transfer functions they have tried. Patten and Ma describe an example of this approach [56]. They use a graph-based display to show transfer function relationships between thumbnail views of previously rendered images. Similarly, the spreadsheet-style interface developed by Jankun-Kelly and Ma [34] allows users to explore a range of parameter combinations at the same time and compare the resulting images (see Fig. 9). These approaches satisfy several of the criteria described in Section 2.3, including 1) ease of switching between representations, 2) continuity between representations, and 3) visibility of the current location, the local neighborhood, links between elements, and navigation history. However, additional screen space is required. Also, in the graph case, the graph could grow very large in a short amount of time, making navigation more difficult and decreasing continuity when switching between representations. By contrast, the spreadsheet will not grow excessively large, but navigation history is not well represented.

3.4.2 Transfer Function Parameters

Part of the difficulty in defining transfer functions is deciding what parameters should be involved in the first place. Most systems assign opacity and color based on voxel intensity values alone. In some cases, the first derivative (also known as the spatial gradient) [38], and sometimes the second derivative [38], are used to locate and highlight boundaries. Occasionally, transfer functions also consider spatial location, allowing users to specify spatial regions of interest (ROIs) [11], [40]. For example, Cohen et al. allow users to increase the opacity around ROIs (e.g., around a
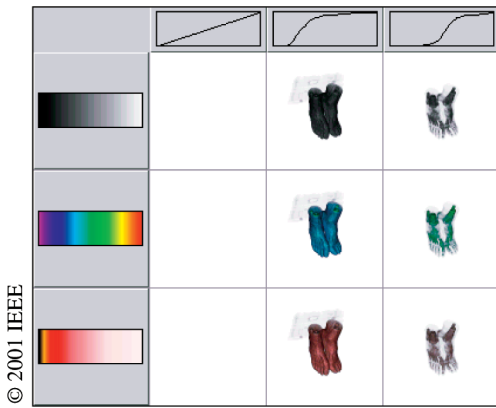
Fig. 9. Spreadsheet-like interface for exploring transfer function parameters [34].

vortex in a flow data set). In general, adding more parameters increases transfer function flexibility, but also increases the size of the search space, making it harder for users to find good parameter combinations.

3.4.3 Automatic and Semi-Automatic Transfer Function Generation

Transfer functions can be generated automatically or with minimal user input. These algorithms typically find and highlight boundaries in the data set [16], [20], [38]. Fully automatic methods are extremely simple for the user, but the transfer functions they can generate are limited because they can only utilize 1) the original data and 2) user knowledge that can be communicated to the program prior to rendering. They cannot take advantage of user intuition and insight during the rendering process or any user knowledge that cannot be captured through predefined interaction metaphors. It is also potentially dangerous to remove the user entirely since the data exploration process may play an important role in understanding the images. Semi-automatic methods address these issues by allowing limited user control. For example, Kindlmann and Durkin [38] constrain transfer functions to those that display boundaries, but allow users to modify a "boundary emphasis function" to explore the data set. Thus, user input is easier because of constraints on parameters, but exploring the entire parameter space is no longer possible.

3.4.4 Visual Search for Transfer Functions

Another alternative is to have the computer generate many different transfer functions and sample images and allow the user to select useful ones [52] or direct future searches based on images close to the desired image [24]. This effectively changes the user's search from an abstract mathematical one to a visual one. However, this strategy depends heavily on human visual search strategies and thus requires substantial user testing that has not yet been done. It is also possible to automatically choose good results based on user-specified evaluation criteria [24]. However, like fully automatic methods, this removal of the user from the data exploration process may be counterproductive.

3.4.5 Input Constraints

Bergman et al. use rules based on the data structure, the visualization goals, and perceptual knowledge to help users

define good transfer functions, specifically color maps [3]. As the user selects parameters, the system imposes constraints on other parameters to avoid useless combinations. For example, once the user selects a color map, colors of contour lines are restricted to those that have sufficient luminance contrast to be visible on top of the color map. The rule-based approach also considers a user's visualization goals, as advocated in Section 2.2, by suggesting color scales that match the current task. For example, tasks that require faithful representation of values of a continuous variable (e.g., viewing a medical image to gain an impression of the overall structure) are best performed with perceptually linear color gradients such as those described in Section 3.1.2. By contrast, tasks that require separating data into distinct categories (e.g., dividing a landscape into promising and nonpromising areas for crop growth) are best suited by a discrete set of perceptually distinguishable colors (also mentioned in Section 3.1.2).

3.5 Detail and Context Displays

Data sets are often very large; for example, a typical medical scan might generate a volume with 512^3 voxels. However, only a limited number of graphic items with limited resolution can be concurrently displayed on a computer monitor. Thus, displaying more items often means displaying less detail about each item. If all items are displayed, few details can be read, but if only a few items are shown, we can lose track of their global location. This requires users to retain large amounts of either detail or context information in working memory, producing an extra cognitive load that may affect performance.

One way to increase the amount of information displayed is to make the display larger. However, large displays still have limited space. Furthermore, imagine creating a display large enough to show every detail in a company's sales database. Such a display would be too large for us to see all parts of it concurrently and information overload could result from the sheer amount of visible information. Thus, increasing the screen size will not always solve this problem. Although some studies have compared display sizes for visualization tasks (e.g., Kasik et al. [35]), additional work could investigate the topic in more detail. Specifically, studies could consider when, if, and how increasing display size and resolution (independently and together) affects visualization task performance.

3.5.1 Detail and Context for 2D Graphics

For 2D displays, fitting large amounts of information into a small space is called the screen real-estate problem. Several techniques have addressed this problem by providing details for areas of interest while still showing where they fit into a global context. *Detail + overview* techniques show both a global overview and details of a selected area, but in separate windows. An icon in the global view indicates the location of details currently shown in the detail view [8, p. 634]. In a GIS system, the overview might consist of a simplified city map with a box around the currently displayed neighborhood.

For some users and applications, integrating the detail and overview displays may impose a cognitive overhead. For this reason, *focus + context* (also known as *detail-in-context*) methods keep the focus view spatially located within the global overview or context. This increases
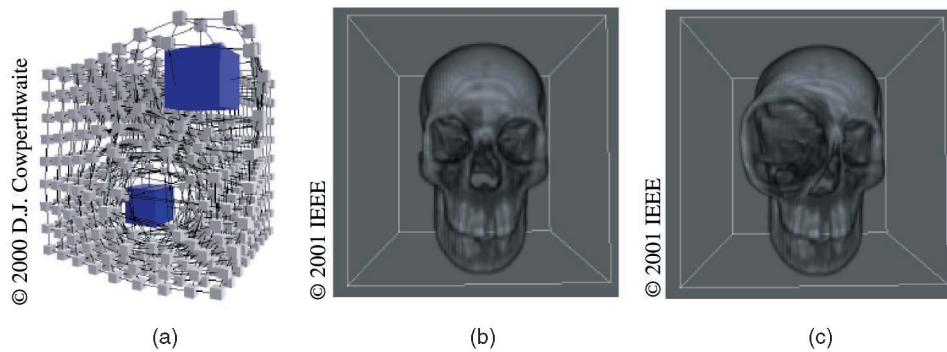
Fig. 10. 3D distortion lenses. (a) The regular grid structure of the graph provides perceptual cues to help users understand the distortion applied to the data set [12]. (b) Volume data set of a skull with no distortion [45]. (c) Skull data set with a spherical distortion lens [45].

continuity between the global and local representations, as suggested in Section 2.3. For example, image sequences (e.g., consecutive slices of volume data) can be arranged in a 2D pattern and then can shift around and change their relative sizes depending on which images are most interesting to the viewer [4], [78].

Most other focus + context displays use image distortion, such as perspective distortion (e.g., the "perspective wall" [51]) or fish-eye lenses. Fish-eye lenses magnify the center of the field of view, with a continuous fall-off in magnification toward the edges. Degree-of-interest values determine the level of detail to be displayed for each item and are assigned through user interaction [36]. Lens techniques were originally proposed for text [21], but have also been applied to images (and other 2D visualizations), where the image is modeled as a pliable surface that can be stretched in specific areas [10], [36], [68].

3.5.2 Detail and Context for 3D Graphics

In 2.5D displays (where 3D graphics are projected onto a 2D screen), the screen real-estate problem is just as prevalent, with the added complication that objects can occlude each other. To address this problem, Ritter et al. developed an overview + detail method to integrate a 3D graphic overview with textual detail [66]. Several researchers have used a 3D extension of the fish-eye lens distortion to display discrete 3D data sets [12], [37], [59]. This distortion method is most effective when the data set has a regular structure and contains objects with straight lines, but can be effective for other data types by drawing a 3D grid along with the data set. Bends in the grid provide perceptual cues to help the user understand what type of distortion has occurred (see Fig. 10a).

Volume data sets (e.g., medical images) often contain irregular shapes. In addition, it can be difficult to draw a 3D grid that will not be occluded by the volume data; thus, distortion is more difficult to understand. In general, detail and context displays for volume data have not been explored in much detail. LaMar et al. integrated a 3D distortion lens with a texture-based volume renderer [45] (see Fig. 10b and Fig. 10c). Kurzion and Yagel developed a 3D distortion method that could be used to provide focus + context [43]; however, it was not evaluated for this purpose. Interacting with distortion lenses may help make the distortion clearer to the viewer; a future user study could verify or refute this idea.

Clip planes show slice detail within a 3D spatial context, but remove data between the clip plane and the viewer. This violates the criteria (from Section 2.3) that the set of all data elements and the local neighborhood should always be visible. An alternative is to open the volume up along a clip plane, using a book or cutting metaphor [9], [12], [43] so that context information is pushed aside but not removed. In the "Corner Cube" [61], three orthogonal slices of a medical data set are projected to the sides of a cube, allowing them to be viewed in their relative orientations. Volumes of interest (VOI) are drawn as isosurfaces in the center of the cube, with outlines projected to the cube walls; thus, the walls provide anatomical context for the VOIs. However, context information in the corner cube is limited to three slices in fixed orientations and few details of the VOIs are displayed.

In direct volume rendering, transfer functions reduce occlusion by making voxels semitransparent so that voxels behind them may be seen. Thus, a transfer function determines how much each voxel contributes to the image. Cohen et al. developed a method to highlight spatial areas of interest by increasing the opacity [11]; this makes the area more visible, but does not increase the screen space allotted to areas of interest.

A number of groups propose to increase context information by projecting a volume from multiple directions at once. Treavett et al. warp the view plane to produce an image containing two points of view [76]. Similarly, in the "magic mirrors" approach, virtual mirrors are placed around a volume so the viewer can see parts of the volume that would be otherwise occluded [41]. Magic mirrors can also display the data using different transfer functions or rendering styles (e.g., a flow vector data set could be displayed using volume rendering, streamlines, and glyphs, each on a different mirror). In similar work by Shaw et al., users can render regions of interest in various styles by moving 2D lenses or "plates" through the data set [70]. Data under the plate is rendered with that plate's style, while the remainder of the data is rendered normally. Although many of these ideas show promise, work in this area is in its infancy and (like transfer functions) these techniques only change the rendering style of a region of interest, not its relative magnification.

3.6 Human-Computer Cooperation

Most visualization systems are designed so that humans and computers can cooperate, each performing the tasks

they do best. Computers can easily store and display data, but humans are better at interpreting data and making decisions. Although this idea is very useful, it is possible for computers to play a more active role in the visualization process than simply presenting data and providing an interface for data manipulation.

As viewers look at images, they compare the image with their existing mental model (a collection of hypotheses about a system's properties and functions [80, p. 284]) of the data and presentation method and adjust either their mental model or their understanding of the image if the two conflict [79], [80, p. 284]. For example, an operator trying to identify a problem in a nuclear power plant may begin with a particular suspicion about the cause of the problem and then change that diagnosis to several other possibilities as features in the visualization provide contradictory evidence. For complex data, constructing a mental model requires interaction and time since all the data cannot be seen in a single view. This process can be aided or inhibited by perceptual and cognitive factors such as those described in Section 2.3.

Visualization systems need not be limited to concrete data sets. Allowing users to write down and manipulate their mental models, ideas, and insight (e.g., as mind maps) could reduce demands on human memory and help users identify new patterns or relationships, as described in Table 1. Going one step further, visualization tools could automatically recognize errors and limitations in mental models and help users to update and refine their mental models to more closely match the real world. This idea has been explored for software visualization; for example, Murphy and Notkin developed a tool that allows software developers to visually describe how they believe software modules are related. The system then identifies errors in the user's model and visually communicates those errors to the user [53].

Murphy and Notkin's tool has two advantages over most visualization systems: 1) It knows the answers to the questions asked by its users because the questions involve only simple queries about code structure and 2) users can communicate their knowledge about the code structure through a strict, well-defined language. In many other cases, these two criteria will not be met and the system is likely to fail if it tries to assume too much knowledge about the data or the user. For instance, a surgeon might want to determine how much tissue should be removed during surgery. This is a more complicated question than a simple query, no well-defined language exists for surgeons to communicate their expert knowledge of similar cases and a patient's particular history, and complex expert systems would be required for the computer to be able to effectively evaluate a physician's decision.

Similar ideas for cognitive support were introduced by Springmeyer et al. [72], who observed scientists doing data analysis in several disciplines. (Note also that this is one of the few visualization papers devoted to task analysis.) One of their conclusions was that current computer-based tools did not support "integration of insight," an important data analysis task involving taking notes, recording and organizing ideas and images, keeping track of the data analysis history, and sharing ideas with others.

Overall, visualization systems could play several roles:

1. visually represent data to enhance data analysis,
2. visually display users' mental models, interpretations of the data, ideas, hypotheses, and insight,
3. help users to improve their mental models by finding supporting and contradictory evidence for their hypotheses, and
4. help users organize and share ideas.

Current research in visualization is almost exclusively devoted to the first objective. Research into the others has not been greatly explored and could make a valuable addition to data analysis tools.

4 CONCLUSION

Recent work in perception and cognition-based design has produced interesting new ideas for data visualization. Specific examples include ways to improve perception of 3D shape, techniques to more easily distinguish and highlight objects, new interaction methods and input devices (e.g., real-world props), faster rendering for better interactivity, interfaces to make transfer function specification easier, and methods of reducing memory load (e.g., detail and context displays).

Since many areas of perception and cognition research are likely not utilized to their full potential, further work in this area is promising. Thus, we should continue to develop and apply the techniques mentioned throughout the paper. Examples include finding new applications for depth-of-focus techniques and further exploring NPR ideas. In addition, in Section 3, we identified several specific directions for future work, including:

- Determining when, if, and how increasing display size and resolution (independently and together) affects performance at visualization tasks,
- Empirically comparing techniques (detail and context methods, transfer function specification methods, multivariate data visualization techniques, shape enhancement methods, NPR methods, etc.) to determine when each method should be chosen over comparable methods,
- Performing user studies to consider whether histograms (and similar data) support transfer function specification and which data is most useful,
- Developing and evaluating task-specific input devices to aid interaction,
- Reducing unnecessary navigation within and between tools (through better display design and integration of tools based on task requirements),
- Developing tools that provide cognitive support for insight and organization of ideas, and
- Exploiting perception and cognition theories that have not yet been considered in visualization design.

Furthermore, many topics in human factors-based design have not been explored by the visualization community in much depth. Much of the current methodology for designing visualization tools and interfaces is ad hoc and informal. Only a few visualization designs utilize perceptual and cognitive theories. Even fewer research groups do iterative user-centered design or structured task analysis. Rapid prototyping is not widely adopted and could facilitate this process. Developing rapid prototyping methods specifically for

visualization could greatly decrease the time and effort invested in ineffective designs and thus speed up progress in visualization research. Stringent evaluation is lacking in many visualization research programs, making it difficult to choose promising ideas for further study. Empirical comparisons of visualization and interaction techniques could provide valuable insight into whether, when, and why techniques provide effective cognitive support. Of course, since many of these methods were designed for HCI research rather than visualization, they may not be ideal. New methodology must be developed specifically for visual data presentation, but this will only happen once current methods have been adopted and evaluated to determine their inadequacies.

ACKNOWLEDGMENTS

REFERENCES

[1] C.L. Bajaj, V. Pascucci, and D.R. Schikore, "The Contour Spectrum," *Proc. IEEE Visualization,* pp.167-173, 1997.

[2] M.Q.W. Baldonado, A. Woodruff, and A. Kuchinsky, "Guidelines for Using Multiple Views in Information Visualization," *Proc. Working Conf. Advanced Visual Interfaces,* pp. 110-119, 2000.

[3] L.D. Bergman, B.E. Rogowitz, and L.A. Treinish, "A Rule-Based Tool for Assisting Colormap Selection," *Proc. IEEE Visualization,* pp. 118-125, 1995.

[4] S. Björk and J. Redström, "Redefining the Focus and Context of Focus + Context Visualization," *Proc. IEEE Symp. Information Visualization,* pp. 85-89, 2000.

[5] A.F. Blackwell et al., "Cognitive Dimensions of Notations: Design Tools for Cognitive Technology," *Proc. Cognitive Technology,* pp. 325-341, 2001.

[6] M.R. Bolin and G.W. Meyer, "A Perceptually Based Adaptive Sampling Algorithm," *Proc. SIGGRAPH,* pp. 299-309, 1998.

[7] V. Bruce, P.R. Green, and M.A. Georgeson, *Visual Perception: Physiology, Psychology, and Ecology,* third ed. Exeter, U.K.: Psychology Press, 1996.

[8] S.K. Card, J.D. Mackinlay, and B. Shneiderman, *Readings in Information Visualization: Using Vision to Think.* San Francisco: Morgan Kaufmann Publishers, 1999.

[9] M.S.T. Carpendale et al., "The tardis: A Visual Exploration Environment for Landscape Dynamics," *Proc. Visual Data Exploration and Analysis VI,* 1999.

[10] M.S.T. Carpendale, D.J. Cowperthwaite, and F.D. Fracchia, "3-Dimensional Pliable Surfaces: For the Effective Presentation of Visual Information," *Proc. ACM Symp. User Interface Software and Technology,* pp. 217-226, 1995.

[11] M.F. Cohen et al., "Volume Seedlings," *Proc. Symp. Interactive 3D Graphics,* pp. 139-145, 1992.

[12] D.J. Cowperthwaite, "Occlusion Resolution Operators for Three-Dimensional Detail-in-Context," doctoral dissertation, Computing Science Dept., Simon Fraser Univ., Burnaby, B.C., 2000.

[13] B. Csébfalvi et al., "Fast Visualization of Object Contours by Non-Photorealistic Volume Rendering," *Proc. Eurographics,* vol. 20, no. 3, 2001.

[14] S. Daly, "The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity," *Digital Images and Human Vision,* A.B. Watson, ed., pp. 179-206, Cambridge, Mass.: MIT Press, 1993.

[15] D.S. Ebert et al., "Procedural Shape Generation for Multi-Dimensional Data Visualization," *Computers & Graphics,* vol. 24, no. 3, pp. 375-384, June 2000.

[16] S. Fang, T. Biddlecome, and M. Tuceryan, "Image-Based Transfer Function Design for Data Exploration in Volume Visualization," *Proc. IEEE Visualization,* pp. 319-326, 1998.

[17] J.A. Ferwerda et al., "A Model of Visual Adaptation for Realistic Image Synthesis," *Proc. SIGGRAPH,* pp. 249-258, 1996.

[18] J.A. Ferwerda et al., "A Model of Visual Masking for Computer Graphics," *Proc. SIGGRAPH,* pp. 143-152, 1997.

[19] B. Fröhlich et al., "Cubic-Mouse-Based Interaction in Virtual Environments," *IEEE Computer Graphics and Applications,* vol. 20, no. 4, pp. 12-15, July/Aug. 2000.

[20] I. Fujishiro, T. Azuma, and Y. Takeshima, "Automating Transfer Function Design for Comprehensible Volume Rendering Based on 3D Field Topology Analysis," *Proc. IEEE Visualization,* pp. 467-470, 1999.

[21] G.W. Furnas, "Generalized Fisheye Views," *Proc. Conf. Human Factors in Computing Systems,* pp. 16-23, 1986.

[22] D.P. Greenberg et al., "A Framework for Realistic Image Synthesis," *Proc. SIGGRAPH,* pp. 477-494, 1997.

[23] J.T. Hackos and J.C. Redish, *User and Task Analysis for Interface Design,* pp. 7-8. Toronto: John Wiley & Sons, 1998.

[24] T. He et al., "Generation of Transfer Functions with Stochastic Search Techniques," *Proc. IEEE Visualization,* pp. 227-234, 489, 1996.

[25] C.G. Healey, "Choosing Effective Colours for Data Visualization," *Proc. IEEE Visualization,* pp. 263-270, 493, 1996.

[26] C.G. Healey, "On the Use of Perceptual Cues and Data Mining for Effective Visualization of Scientific Datasets," *Proc. Graphics Interface,* pp. 177-184, 1998.

[27] C.G. Healey, K.S. Booth, and J.T. Enns, "High-Speed Visual Estimation Using Preattentive Processing," *ACM Trans. Computer-Human Interaction,* vol. 3, no. 2, pp. 107-135, 1996.

[28] C.G. Healey and J.T. Enns, "Perception and Painting: A Search for Effective, Engaging Visualizations," *IEEE Computer Graphics and Applications,* pp. 10-15, Mar./Apr. 2002.

[29] K. Hinckley et al., "A Survey of Design Issues in Spatial Input," *Proc. ACM Symp. User Interface Software and Technology,* pp. 213-222, 1994.

[30] K. Hinckley et al., "Passive Real-World Interface Props for Neurosurgical Visualization," *Proc. Conf. Human Factors in Computing Systems,* pp. 452-458, 1994.

[31] V. Interrante, H. Fuchs, and S.M. Pizer, "Conveying the 3D Shape of Smoothly Curving Transparent Surfaces via Texture," *IEEE Trans. Visualization and Computer Graphics,* vol. 3, no. 2, pp. 98-117, Apr.-June 1997.

[32] V. Interrante, "Harnessing Natural Textures for Multivariate Visualization," *IEEE Computer Graphics and Applications,* vol. 20, no. 6, pp. 6-10, Nov./Dec. 2000.

[33] P. Irani, M. Tingley, and C. Ware, "Using Perceptual Syntax to Enhance Semantic Content in Diagrams," *IEEE Computer Graphics and Applications,* vol. 21, no. 5, pp. 76-84, Sept./Oct. 2001.

[34] T.J. Jankun-Kelly and K.-L. Ma, "Visualization Exploration and Encapsulation via a Spreadsheet-Like Interface," *IEEE Trans. Visualization and Computer Graphics,* vol. 7, no. 3, pp. 275 -287, July/Sept. 2001.

[35] D.J. Kasik et al., "Evaluating Graphics Displays for Complex 3D Models," *IEEE Computer Graphics and Applications,* vol. 22, no. 3, pp. 56-64, May/June 2002.

[36] T.A. Keahey, "The Generalized Detail In-Context Problem," *Proc. IEEE Symp. Information Visualization,* pp. 44-51, 152, 1998.

[37] T.A. Keahey, "Visualization of High-Dimensional Clusters Using Nonlinear Magnification," Technical Report LA-UR-98-2776, Los Alamos Nat'l Laboratory, 1998.

[38] G. Kindlmann and J.W. Durkin, "Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering," *Proc. IEEE Symp. Volume Visualization,* pp. 79-86, 1998.

[39] R.M. Kirby, H. Marmanis, and D.H. Laidlaw, "Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting," *Proc. IEEE Visualization,* pp. 333-340, 540, 1999.

[40] J. Kniss, G. Kindlmann, and C. Hansen, "Interactive Volume Rendering Using Multi-Dimensional Transfer Functions and Direct Manipulation Widgets," *Proc. IEEE Visualization,* pp. 255-262, 562, 2001.

[41] A. König, H. Doleisch, and E. Gröller, "Multiple Views and Magic Mirrors—fMRI Visualization of the Human Brain," Technical Report TR-186-2-99-08, Inst. of Computer Graphics and Algorithms, Vienna Univ. of Technology, Feb. 1999.

[42] R. Kosara, S. Miksch, and H. Hauser, "Semantic Depth of Field," *Proc. IEEE Symp. Information Visualization,* pp. 97-104, 2001.

[43] Y. Kurzion and R. Yagel, "Interactive Space Deformation with Hardware-Assisted Rendering," *IEEE Computer Graphics and Applications,* vol. 17, no. 5, pp. 66-77, Sept./Oct. 1997.

[44] D.H. Laidlaw et al., "Visualizing Diffusion Tensor Images of the Mouse Spinal Cord," *Proc. IEEE Visualization,* pp. 127-134, 527, 1998.

[45] E. LaMar, B. Hamann, and K.I. Joy, "A Magnification Lens for Interactive Volume Visualization," *Proc. Ninth Pacific Conf. Computer Graphics and Applications,* pp. 223-232, 2001.

[46] H. Levkowitz and G.T. Herman, "Color Scales for Image Data," *IEEE Computer Graphics and Applications,* vol. 12, no. 1, pp. 72-80, 1992.

[47] J. Lubin, "A Visual Discrimination Model for Imaging System Design and Evaluation," *Vision Models for Target Detection and Recognition,* E. Peli, ed., pp. 245-283, World Scientific, 1995.

[48] E.B. Lum and K.-L. Ma, "Hardware-Accelerated Parallel Non-Photorealistic Volume Rendering," *Proc. Non-Photorealistic Animation and Rendering,* pp. 67-74, 158, 2002.

[49] E.B. Lum, A. Stompel, and K.-L. Ma, "Kinetic Visualization: A Technique for Illustrating 3D Shape and Structure," *Proc. IEEE Visualization 2002,* pp. 435-442, 2002.

[50] R.L. Mack and J. Nielsen, "Usability Inspection Methods: Executive Summary," *Readings in Human-Computer Interaction: Toward the Year 2000,* second ed., R.M. Baecker et al., eds., pp. 170-181, San Francisco: Morgan Kaufmann, 1995.

[51] J.D. Mackinlay, G.G. Robertson, and S.K. Card, "The Perspective Wall: Detail and Context Smoothly Integrated," *Proc. Conf. Human Factors in Computing Systems,* pp. 173-179, 1991.

[52] J. Marks et al., "Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation," *Proc. SIGGRAPH,* pp. 389-400, 1997.

[53] G.C. Murphy and D. Notkin, "Reengineering with Reflexion Models: A Case Study," *Computer,* vol. 30, no. 8, pp. 29-36, Aug. 1997.

[54] L. Nigay and F. Vernier, "Design Method of Interaction Techniques for Large Information Spaces," *Proc. Advanced Visualization Interfaces,* pp. 37-46, 1998.

[55] S.N. Pattanaik et al., "A Multiscale Model of Adaptation and Spatial Vision for Realistic Image Display," *Proc. SIGGRAPH,* pp. 287-298, 1998.

[56] J.M. Patten and K.-L. Ma, "A Graph Based Interface for Representing Volume Visualization Results," *Proc. Graphics Interface,* pp. 117-124, 1998.

[57] F. Pellacini, J.A. Ferwerda, and D.P. Greenberg, "Toward a Psychophysically-Based Light Reflection Model for Image Synthesis," *Proc. SIGGRAPH,* pp. 55-64, 2000.

[58] G. Peters, "Theories of Three-Dimensional Object Perception: A Survey," *Recent Research Developments in Pattern Recognition,* Transworld Research Network, 2000.

[59] A. Raab and M. Rüger, "3D-Zoom: Interactive Visualisation of Structures and Relations in Complex Graphics," *3D Image Analysis and Synthesis,* B. Girod, H. Niemann, and H.P. Seidel, eds., pp. 125-132, 1996.

[60] M. Reddy, "Perceptually Optimized 3D Graphics," *IEEE Computer Graphics and Applications,* vol. 21, no. 5, pp. 68-75, Sept./Oct. 2001.

[61] K. Rehm et al., "A Symbolic Environment for Visualizing Activated Foci in Functional Neuroimaging Datasets," *Medical Image Analysis,* vol. 2, no. 3, pp. 215-226, 1998.

[62] P. Rheingans, "Are We There Yet? Exploring with Dynamic Visualization," *IEEE Computer Graphics and Applications,* vol. 22, no. 1, pp. 6-10, Jan./Feb. 2002.

[63] P. Rheingans, "Opacity-Modulating Triangular Textures for Irregular Surfaces," *Proc. IEEE Visualization,* pp. 219-225, 1996.

[64] P. Rheingans, "Task-Based Color Scale Design," *Proc. Applied Image and Pattern Recognition,* pp. 35-43, 1999.

[65] P. Rheingans and D. Ebert, "Volume Illustration: Nonphotorealistic Rendering of Volume Models," *IEEE Trans. Visualization and Computer Graphics,* vol. 7, no. 3, pp. 253-264, July-Sept. 2001.

[66] F. Ritter et al., "Virtual 3D Puzzles: A New Method for Exploring Geometric Models in VR," *IEEE Computer Graphics and Applications,* vol. 21, no. 5, pp. 11-13, Sept./Oct. 2001.

[67] M.P. Salisbury et al., "Orientable Textures for Image-Based Pen-and-Ink Illustration," *Proc. SIGGRAPH,* pp. 401-406, 1997.

[68] M. Sarkar et al., "Stretching the Rubber Sheet: A Metaphor for Viewing Large Layouts on Small Screens," *Proc. ACM Symp. User Interface Software and Technology,* pp. 81-91, 1993.

[69] W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics,* second ed. Upper Saddle River, N.J.: Prentice Hall, 1998.

[70] C.D. Shaw et al., "Interactive Lens Visualization Techniques," *Proc. IEEE Visualization,* pp. 155-160, 1999.

[71] B. Shneiderman, *Designing the User Interface,* third ed. Reading, Mass.: Addison-Wesley 1997.

[72] R.R. Springmeyer, M.M. Blattner, and N.L. Max, "A Characterization of the Scientific Data Analysis Process," *Proc. IEEE Visualization,* pp. 235-242, 1992.

[73] T. Strothotte and S. Schlechtweg, *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation,* pp. 10-11. San Francisco: Morgan Kaufmann, 2002.

[74] M. Tory and T. Möller, "A Model-Based Visualization Taxonomy," Technical Report CMPT-TR2002-06, Computing Science Dept., Simon Fraser Univ., 2002.

[75] S.M.F. Treavett and M. Chen, "Pen-and-Ink Rendering in Volume Visualization," *Proc. IEEE Visualization,* pp. 203-210, 557, 2000.

[76] S.M.F. Treavett et al., "Volumes of Expression: Artistic Modelling and Rendering of Volume Datasets," *Proc. Computer Graphics Int'l,* 2001.

[77] A. Treisman and S. Gormican, "Search, Similarity and Integration of Features between and within Dimensions," *J. Experimental Psychology: Human Perception and Performance,* vol. 17, pp. 652-676, 1991.

[78] J.E. Van der Heyden et al., "Visual Presentation of Magnetic Resonance Images," *Proc. IEEE Visualization,* pp. 423-426, 1998.

[79] Y. Waern, "On the Dynamics of Mental Models," *Mental Models and Human-Computer Interaction 1,* D. Ackermann and M.J. Tauber, eds., pp. 73-93, New York: Elsevier Science, 1990.

[80] A. Walenstein, "Cognitive Support in Software Engineering Tools: A Distributed Cognition Framework," doctoral dissertation, Computing Science Dept., Simon Fraser Univ., Burnaby, B.C., Canada, 2002.

[81] C. Ware, *Information Visualization: Perception for Design.* San Francisco: Morgan Kaufmann (Academic Press), 2000.

[82] D.D. Woods, "Visual Momentum: A Concept to Improve the Cognitive Coupling of Person and Computer," *Int'l J. Man-Machine Studies,* vol. 21, pp. 229-244, 1984.

Melanie Tory is a PhD candidate in the Graphics, Usability, and Visualization Lab at Simon Fraser University. She received the BSc degree from the University of British Columbia in 1999. Her research objectives are to maximize the value of visualization tools by developing and evaluating effective user interfaces. She is currently exploring ways to combine 2D and 3D views for visualization of spatial data.

Torsten Möller received the PhD degree in computer and information science from Ohio State University in 1999 and a Vordiplom (BSc) in mathematical computer science from Humboldt University of Berlin, Germany. He is an assistant professor inthe School Of Computing Science at Simon Fraser University. His research interests include the fields of scientific visualization and computer graphics, especially the mathematical foundations of visualization and graphics. He is codirector of the Graphics, Usability and Visualization Lab and serves on the Board of Advisors for the Centre for Scientific Computing at Simon Fraser University.

▷ **For more information on this or any computing topic, please visit our Digital Library at** http://computer.org/publications/dlib.