Towards a Visual Interface for Information Visualization

Daryl H. Hepting

Department of Computer Science, University of Regina
Regina, Saskatchewan, S4S 0A2, Canada
E-mail: `dhh@cs.uregina.ca`

Abstract

Information visualization, aided by ever more accessible computational resources, continues to grow in popularity and significance. The capability to generate complex imagery by computer is often necessary but not always sufficient to gain the desired insight. The success of a visual representation in a given context may be affected by many variables, not the least of which is the individual user's experience. Even if a precise relationship could be found between context and "best" visual representation, the complete articulation of a context is practically impossible. In other fields, this is known as sensitive dependence to initial conditions. A more feasible alternative is to begin with an incomplete articulation of a context and allow the user to interactively develop and refine it. Although most computer interfaces for information visualization tools are predominantly verbal, a predominantly visual interface can have significant advantages. Such an interface allows users to avoid the usual translations between visual and verbal modes and it removes users' need for a specialized visualization vocabulary. A visual interface can also shift the focus of the visualization process from the data towards the user. These ideas are discussed in the context of a prototype tool, the design of which is illustrated with an example, and the evaluation of which has provided many positive results.

1. Introduction

In 1992, Jessup [22] contended that scientific visualization had the promise to democratize visual thinking. The application of computer technology to the generation of visual representations has provided an important capability to scientists and researchers. By emphasizing only the capability to produce computer-generated visual representations, this promise has yet to be realized. Visual thinking will remain inaccessible to those who lack the necessary vocabulary, unless other means of access are provided.

Consider that any visual representation can be decomposed into *components*, each with their own *elements*. A component could be "graph type", with elements including "bar chart", "pie chart", "line chart", "scatterplot", and so on. Each visual representation can be denoted as an N-tuple, where e_i is an element of component C_i. In practice, not all N-tuples will correspond to valid visual representations because of incompatibilities between elements of different components. The Cartesian product of the elements from all the components forms the N-dimensional space of available visual representations.

$$\langle e_1, e_2, \ldots, e_N \rangle \in C_1 \times C_2 \times \ldots \times C_N$$

The space of available visual representations can be very large and it can be difficult to grasp the implications of all available combinations of elements. This fact only exacerbates the problem of selecting and specifying the individual elements in a visual representation. In the midst of so many combinations, it can be difficult to find a visual representation which is apposite.

Even for this predominantly visual task, computer interfaces have remained predominantly verbal. For example, various forms of textual menus are an important part of many Graphical User Interfaces (GUI's). These verbal interfaces create a need to translate to and from verbal representations of the visual. Visual programming systems, like those available in Modular Visualization Environments (MVE's) [13], make programming more accessible, but do not change the nature of the task: a user without a specific vocabulary for visualization would need to work with programmers to find visual expression for their ideas.

Verbal processing requires more sequential processing, both for comprehension and production, than does visual processing [17]. However, in terms of perception of input, audition is necessarily sequential whereas vision is simultaneous [3].

The issues of translation would not be so great a concern if one could completely articulate all aspects of the problem. However, as Winograd and Flores [12] point out, this is not possible. Furthermore, traditional visualization software either limits the choice of users and constructs a visual representation automatically or it leaves users without support in dealing with a myriad of choices. This paper addresses those concerns and presents work towards a predominantly visual interface for information visualization.

Section 2 introduces relevant background from a wide range of sources touching on graphic communication, computer-aided visualization, and human-computer interaction. Section 3 develops the details of the design for the new software. Section 4 describes an evaluation of the implemented prototype software, along with some results. Finally, Section 5 presents conclusions and future work.

2. Background

The use of components and elements to describe particular visual representations is an adaptation of Bertin's [9] retinal variables which he used to systematically explore marks on a plane and how those marks could be used to construct diagrams, networks, and maps. Graphic communication in two dimensions has been thoroughly studied and the construction of visual representations within this realm is fairly well understood. For this reason, the example chosen to illustrate this presentation and to evaluate the prototype software is a two-dimensional graphing problem based on a small dataset from Bertin [9][page 100]. It provides a view of the French economy from the early 1960's. For each département in France, the data provides the workforce (in thousands of workers) for each of the three sectors (primary, secondary, and tertiary) in the economy; the total workforce (the sum of the three sectors); and the percentage of the workforce in each sector. Figure 1 presents a sample visual representation of this data.

Bertin [9] remarked that "to construct 100 DIFFERENT FIGURES from the same information requires less imagination than patience. However, certain choices become compelling due to their greater 'efficiency."' But the question of efficiency is closely linked to the task at hand and the user's experience with the elements of a visual representation, as Casner [19] describes. Although Bertin contends that meaning can be communicated fully through a graphic and its legend, the more widely accepted view is that communication and interpretation occur, or are influenced by things, outside this realm.

Sicard and Marck [20] distinguish cognitive, didactic, and aesthetic logics in scientific pictures, which are not separable without knowledge of the author's intent. For them, scientific pictures are "imbued with the 'view' of the author which claims to be objective. But, in fact, it is attached to 'thought history', technological history, scientific history and is marked by aesthetic choices, cultural bias, and perceptional practices." For Winograd and Flores [12], this means that "the ideal of an objectively knowledgeable expert must be replaced with a recognition of the importance of background. This can lead to the design of tools that facilitate a dialog of evolving understanding among a knowledgeable community." This type of computer-based tool can be hard to construct.

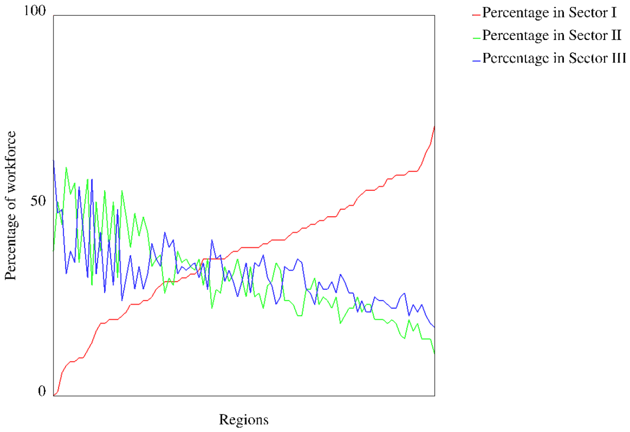


Figure 1: One possible visual representation of Bertin's data, constructed from components and elements that specify features including the graph type, the annotation, the sorting of the data, and the colours.

In their introduction to *Readings in Information Visualization: Using Vision to Think*, Card, Mackinlay, and Shneiderman define a user-centered view of visualization as "the use of computer-supported, interactive, visual representations of data to amplify cognition" [23]. However, a data-centric perspective is far more common in the modern era, where visualization is the "use of computer imaging technology as a tool for comprehending data obtained by simulation or physical measurement" [24]. Visualization naturally has both data-centered and user-centered aspects. Springmeyer *et al.* [25] analysed scientists' visualization research process and identified user-centric "integration of insight" activities in addition to the better-understood and better-supported data-centric "investigation" activities.

Adapting the classification of Kochhar *et al.* [26], it is possible to distinguish manual, automatic, and augmented visualization systems based on their relationship of human and computer.

Manual systems require the user to completely describe and control the operation of the visualization application. The space of alternatives available for exploration in these schemes is implicitly limited by the user's own experience. Systems exemplified apE (animation production Environment) [5] and AVS (Application Visualization System) [27], are collectively known as Modular Visualization Environments (MVE's). MVE's have come to prominence because they allow users to create complete visualizations from components connected using a visual dataflow model. DataDesk, the statistical graphics package first described by Velleman and Pratt [10] in 1989, provides a direct-manipulation interface to statistics and a good example of Tukey's Exploratory Data Analysis [2]. It builds on the idea that multiple, connected views of data can greatly enhance the power of data analysis. Graphical interfaces are seen as ways to specify "like this one, only different in the following ways." Insight is acknowledged as important. The Spreadsheet for Information Visualization (SIV) [11], based on work presented by Levoy [1], is a novel use of the spreadsheet programming paradigm that allows the user to explore the effect of the same operation on several related images.

Automated systems appear to the user as black boxes which are given input and produce output. The rationale behind them is that the number of alternative visual representations is so large that the user would be overwhelmed if he or she had to deal with the space in its entirety. In accepting this guidance from the computer, the user relies more on the computer for its application of design rules and gives up more freedom to exercise personal choices about what the visual representations will contain. In 1986, APT (A Presentation Tool) by Mackinlay [21] contributed a formalization of the design rules for two-dimensional static graphs, based on Bertin [9] and others. It was a prescriptive system because it chose graphics on the basis of expressiveness and effectiveness criteria. With BOZ in 1991, Casner [19] added information about the task to his presentation system and this resulted in a noticeable improvement in user performance with the graphs that his system generated.

Augmented systems aid the user by allowing certain well-defined tasks to be performed primarily by the computer, with the effect of increasing the capabilities of people to tackle complex problems. Because any articulation of a design is an ongoing process which is necessarily incomplete, it is important for the user to maintain some control. Rogowitz and Treinish [15] described a visualization architecture that allowed the user to choose a higher-level interaction with the visualization process, based on the invocation of appropriate rules. The VISTA (VISualization Tool Assistant) environment described by Senay and Ignatius [14] would analyse, as much as possible, the input data and suggest a visual representation to which the user could make modifications. The SageTools [28] system allowed users to work in the context of past graphics with the option to modify what had already been done. The Integrated Visualization Environment (IVE) [29] implemented the cooperative computer-aided design (CCAD) paradigm. It used a generative approach, in which the user could intercede after each iteration to select promising designs for further development. Design Galleries [30] worked to provide a good sampling of the range of alternatives. The user specified the means for comparison and the system worked offline to generate and evaluate the images based on the user's specification and then displayed the results.

Rather than focus on the results produced by these visualization systems and attempt to answer whether a "best" visual representation can be decided for any context or any group of users, it is productive to look at the process by which these representations can be developed.

According to Winograd and Flores [12], we can "create computer systems whose use leads to better domains of interpretation. The machine can convey a kind of coaching in which new possibilities for interpretation and action emerge." Even proverbial wisdom [1] suggests that involving users in producing visual representations may help them with their understanding.

Norman [4] describes the twin gulfs of execution and evaluation. With a goal in mind, a user experiences the gulf of execution in deciding which commands to execute in order to move from his or her present state to the goal state. Similarly, the gulf of evaluation is encountered when a user tries to reconcile an intermediate result state with the original goal state. An effective interface will minimize these gulfs, and for visualization tasks a visual interface is required.

There is increasing evidence, in part reported by Schooler and Engstler-Schooler [31], which indicates that attempts to verbalize descriptions of non-reportable phenomena may overshadow the original information. This is particularly important for people who do not have a strong graphical vocabulary. For example, verbalizing the appearance of a previously seen face interfered with the ability to later recognize that face from a group of similar ones. Likewise, the verbal specification of a desired visual representation may ultimately impede access to the original mental imagery. Remaining in the perceptual domain of images may alleviate this problem. A successful visual interface must also provide non-verbal means of navigation, a feature which is lacking in MVE's for example.

Even if users would choose the same visual representation in a given context, Sicard and Marck [20] suggest that it still may be of value for each user to go through the process of creating that representation for him or herself. Involving the user in this process significantly reduces the reliance on an accurate user model, which could be quite difficult to develop.

In 1991, Sims [7] presented a method for the use of artificial evolution in computer graphics which employed both a

---

[1] The actual proverb, possibly of either Chinese or Native American origins is: *Tell me and I will forget, show me and I may remember, involve me and I will understand.*

genetic algorithm [18] and genetic programming [32]. Both of these "genetic" methods work by simulating the cellular-level processes of cross-over and mutation. The former does this as means to search a space whereas the latter works to transform the space itself. For Sims, the goal was to evolve images and textures. However, because it can be surprising to see images from different generations with no apparent connection between them, it can work to defeat the user's control. In 1992, Todd and Latham [8] also discussed a genetic approach to the generation of new images, theirs being more restrictive and controllable by not including genetic programming.

Even for small problems with relatively few alternatives, an exhaustive evaluation is almost always completely impractical. Instead, humans rely on heuristic search methods which are likely to find acceptable solutions in a reasonable amount of time. These search heuristics can be of two sorts, in general. If the problem is well-understood, local search techniques may be employed effectively. If the problem is new, a global search may be better suited to the exploration of alternatives.

3. Design

A software system, called *cogito* [2], was designed to address the shortcomings of traditional visualization tools. In particular, the system deals with the problem of articulation with a visual interface that provides non-verbal access to alternatives. With an incomplete articulation of the context, the iteration performed in selecting and evaluating candidate visual representations is crucial to the visualization process. The evaluation of visual representations can be done more effectively if the available alternatives are understood, and interaction is essential to accomplish this. The *cogito* system supports "combinatory play" by considering every visual representation to be the product of elements from each of several components and it relies on the user to choose these elements. Not only is this conception of components and elements familiar from Bertin, it also occurs in MVE systems like AVS [27] (Application Visualization System), and the toolkit philosophy of the Visualization ToolKit [33]. But, in *cogito*, the user does not choose these elements in isolation. Rather, he or she chooses between whole visual representations, each of which comprise particular elements.

The computer is well-suited to provide such external memory to support this decision-making process. Placed between manual and automatic systems, the design of *cogito* uses the computer to perform bookkeeping functions and allows the user evaluate and select. A traditional visualization system, with its need for expertise in programming, can separate the user from this important function. Whereas programming support is also required for *cogito*, the user and the programmer may work together to create the notion of the space of available representations and the user is still able to interact directly with the computer. Figure 2 illustrates this difference.

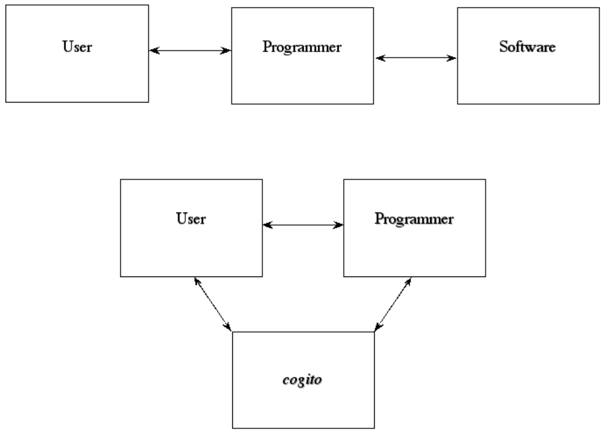


Figure 2: In the traditional model of interaction with visualization systems, the programmer mediates the user's experience with the software. The new model embodied in *cogito* allows the user to work directly with the software.

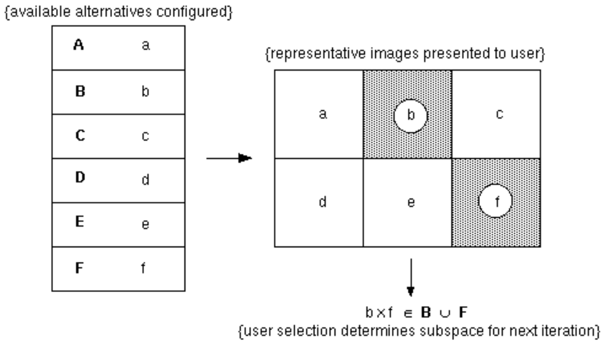


Figure 3: Schematic look at the interface: the space of available alternatives is grouped according to user-specified criteria. Each group (A – F) has a representative element (a – f) which is displayed to the user. The subspace for the next search iteration is based on the user selection (b and f).

---

[2] The name of the system, *cogito* is taken from the Latin verb "to think", which etymologically means "to shake together". This is done to acknowledge the role of the combination of ideas in various models of human inventive thought.

The *cogito* system provides, through views, the means to structure and examine the space according to a range of criteria. The user sees the current space, with the current organizational view, one screen at a time. Cells, which display individual visual representations and permit certain operations on them, comprise each screen. A schematic of one of these screens is shown in Figure 3. As the programmer and user define the space, it is also possible to use different organizational methods for the space of alternatives. In Figure 4, for example, one sees 3 different ways to organize a space with three dimensions. Using the terminology of Figure 3, the representatives $x_1 \ldots x_4$ in Figure 4(b) are formed by choosing sequentially from X and randomly from Y and Z.

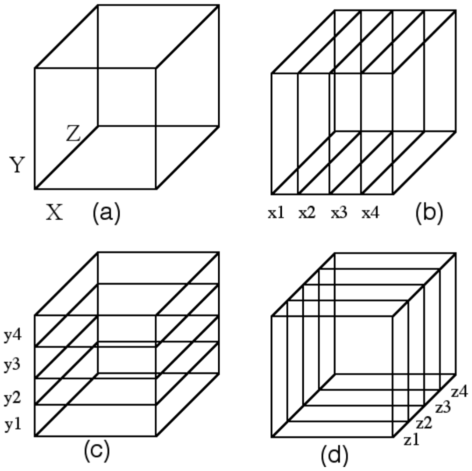


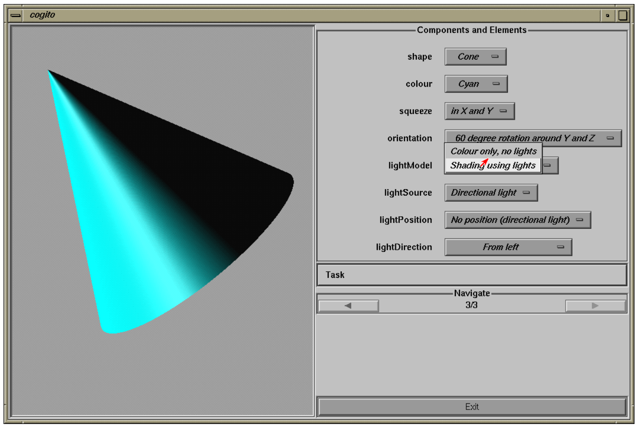
Figure 4: Consider a three-dimensional space, depicted in the top left, with axes X, Y, and Z. Organizing the space in terms of any of those 3 axes leads to the other states depicted. If elements in component X are chosen sequentially, those in Y and Z can be selected randomly to give a sense available options.

The user indicates desirable elements or complete visual representations by non-verbal selection (done by clicking directly on the desired cell). Once the user is satisfied with the selections made on a particular space, a new space consistent with those selections is generated by a genetic approach which performs crossover operations amongst selected combinations. Successive generations can be used to either narrow or expand the search space (up to the size of the original), depending on the needs of the user. Additionally, an "image editor" is provided to directly make small changes. In this way, the space of all available visual representations can be navigated.

Although the user may be aware of all the individual elements, combinations may be encountered from outside the user's experience. It is therefore much more likely that the space of available representations can be thoroughly explored using this interface.

4. Evaluation

In order to assess the effectiveness of the new software design, a user study was developed and executed [16]. From the full software that was implemented, two different interfaces to the space of available visual representations were created. Both used the same underlying engine. Respectively, these were predominantly verbal and visual interfaces. Interface A, shown in Figure 5, provided an unstructured view of the available alternatives. It allowed the user to make incremental changes to the image and see the results immediately. The user did this by selecting elements for each component from text-based option lists. Interface B, shown in Figure 6, provided a structured, hierarchical view of the available alternatives following directly from the design illustrated in Figure 3. The user is made aware of options by showing a collection of visual representations from a regular sample of the whole space. From this displayed collection, the user could choose visual representations of interest and based on that information, the system would compose a new space consistent with those selections. This process could continue, with user operations able to both narrow and expand the search space, until the user decided to end the process.



shape Cone
colour Cyan
squeeze in X and Y
orientation 60 degree rotation around Y and Z
lightModel Shading using lights
lightSource Directional light
lightPosition No position (directional light)
lightDirection From left

Figure 5: A sample of Interface A, showing material from the user study's training task. Elements from the different components are selected directly from the option menus.

Two hypotheses were tested, expressed by the following null hypotheses:

- $\mathtt{H}_0$`-activity`:there is no preference for involvement and exploration

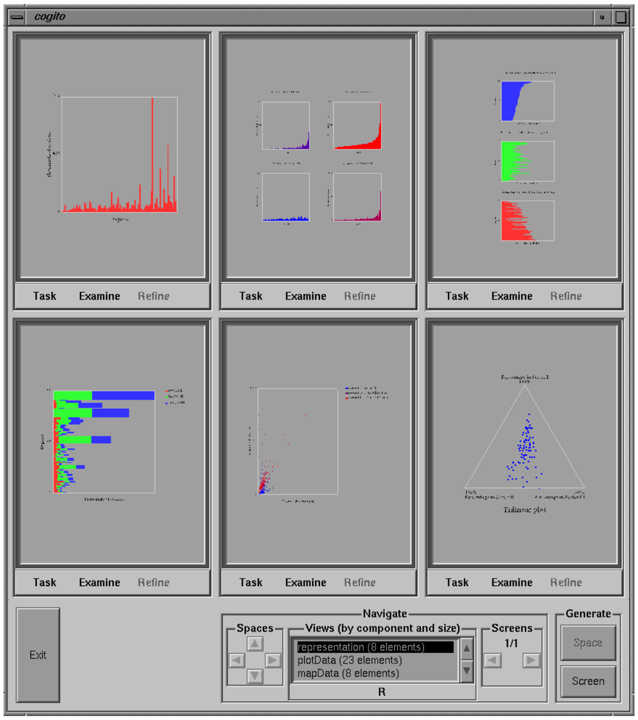


Figure 6: A sample of Interface B, showing material from the user study's graphing task. Rather than requiring direct specification, the system generates sample visual representations from the space of available visual representations.

- H_0`-interface`: there is no preference for either interface

Subjects completed a pre-task questionnaire, a training task, the graphing task, and a post-task questionnaire during a session that lasted approximately 1 hour. A between-subjects experiment design was used, so each subject saw either Interface A or Interface B. This choice was made because of the similarity of the two interfaces which might have introduced learning effects and because of the amount of time needed to test both interfaces would have placed excessive demands on the subjects, who had volunteered their time.

The pre-task questionnaire was designed to gain background information about the subjects, which could be used for subsequent analysis. In addition to information about gender, age, years of education, and main area of interest, subjects were asked to provide subjective judgements about their use of graphs, their creation of graphs, and their level of expertise with graph-generating software and computer graphics generally.

The requirement for the training task was to train the subjects in the software without training them in the main task problem. In order to minimize the task-specific learning effects, the training task was designed to use a completely different application with a different character of questions.

Exploration	Involvement	
	Negative	Positive
Negative	2	2
Positive	1	29

Table 1: Contingency table for exploration and involvement: the relationship between exploration and involvement is significant ($p = 0.031$), so H_0`-activity` can be rejected.

In the graphing task, subjects were asked to answer some questions about the data, described in Section 2, by creating and interpreting some visual representations. The questions were chosen to emphasize the three types of reading one might use with a graph: elementary (Is there a region whose workforce is evenly distributed amongst the three sectors?), intermediate (What is the size of Sector I in regions where Sector III predominates?), and overall (What is a discernable relationship between the 3 sectors?) [9].

The post-task questionnaire used a total of 17 Likert scale questions (requesting a rating on a scale of the form: "Strongly Disagree", "Disagree", "Agree", "Strongly Agree") and 4 open-ended questions. The questionnaire was designed to assess subjects' feelings about their completion of the task; the degree to which the software helped them; and their impressions of the task. The written questions were meant to solicit which features were liked and disliked, which features were missing, and any other comments about the system.

The experiment was not designed to collect performance data in a meaningful way. Instead, questionnaire responses were analysed, selected visual representations were logged, and written comments were catalogued. A chi-square test in Version 9.0 of the SPSS (Statistical Package for the Social Sciences) [6] software was used to analyse both the questionnaire responses and the selections of visual representations because both sets of data were categorical and the test allowed comparison of the proportions between categories.

The 34 volunteers for the study were predominantly males between the ages of 18 and 25 and undergraduate students in the Faculty of Applied Science at Simon Fraser Unversity. Despite this uniformity in the background of the subjects, there was considerable variability in the way that they approached the questions. The task seemed to be very difficult for some, which underscored the importance of task knowledge.

With respect to H_0`-activity`, whether exploration and involvement are helpful, subjects were classified based on their responses to questions about whether each was helpful. Responses of "Very unhelpful" or "Unhelpful" were classified as "Negative" and responses of "Helpful" or "Very helpful" were classified as "Positive." For this analysis, a contingency table was built (see Table 1) to analyse the four combinations of responses to the two questions. Because of the small sample size, Fisher's Exact test was used to compute the significance. It indicated that $\mathtt{H}_0$`-activity` could be rejected, $p < 0.05$.

There was insufficient evidence to reject $\mathtt{H}_0$`-interface`, whether one interface was preferred. Although some factors did indicate a trend in preference for Interface B, it seems that Interface A's relatively limited features for exploration and involvement were appreciated. Interface B had a steeper learning curve likely because of its conceptual novelty, so this may have negatively affected its ratings. The distribution of selection frequencies for elements in the components indicates that users of interface B tried more varied alternatives, which agreed with a hypothesized benefit of this approach.

5. Conclusions

Although the evaluation results obtained to date are not conclusive, they are encouraging and this study represents an important contribution to the research. Many new questions, for which the original study was not designed, have been raised and subsequent studies are presently being prepared in order to address them.

This study was not designed to explore this question of when people might choose to adopt a representation which is novel to them. As Casner [19] wrote, an efficient representation may come at the price of learning time, and some people are not willing nor able to make this commitment.

From several user comments, an integration of the two styles of interaction will lead to a more effective tool, giving people a choice of how they would rather work. In fact, this suggested combination of global exploration and local refinement of alternatives is not surprising. Also, both interfaces to the system made alternative visual representations available but they did not provide the user with any suggestions about which representation to use. It might be a useful feature to begin the exploration at a reasonable position in the search space.

The importance of involvement and exploration have been underscored by the results of the user study. For the future, a more complete implementation and more thorough testing are planned. Specifically, better support for navigation and collaboration are indicated. The study examined one piece of what is a larger visualization process. Therefore, future work will have an expanded scope. For example, many comments from the user study are application-specific, and the means for a user to tailor his or her view of the application will doubtless contribute to improved satisfaction. This test concentrated on a single user at a single session whereas these ideas but these ideas are applicable to use by a single user over an extended period and by several users who wish to collaborate.

The system represents an important step towards providing the access to computer-aided visual thinking indicated by Jessup [22].

Acknowledgements

I thank Bob Russell, John Hart, Katherine Arbuthnott and Jennifer Hepting for many discussions which have improved the presentation of this material. I also thank the anonymous referees for their comments.

References

[1] M. Levoy. Spreadsheets for images. In A. Glassner, editor, *Computer Graphics: SIGGRAPH 94 Proceedings*, pages 139–146, 1994.

[2] J. W. Tukey. *Exploratory data analysis*. Addison-Wesley, 1977.

[3] E. B. Goldstein. *Sensation and perception*. Brooks/Cole, fifth edition, 1999.

[4] D. A. Norman. *The Psychology of Everyday Things*. Basic Books, New York, 1988.

[5] D. S. Dyer. A dataflow toolkit for visualization. *IEEE Computer Graphics and Applications*, pages 60–69, July 1990.

[6] SPSS Incorporated. *SPSS Base 9.0 : Applications Guide*. Prentice Hall, 1999.

[7] K. Sims. Artificial evolution in computer graphics. In R. J. Beach, editor, *Computer Graphics: SIGGRAPH '91 Conference Proceedings*, pages 319–328. ACM Press, 1991.

[8] S. Todd and W. Latham. *Evolutionary art and computers*. Academic Press, London, 1992.

[9] J. Bertin. *Semiology of graphics : diagrams, networks, maps*. University of Wisconsin Press, 1983. Translated by W. J. Berg.

[10] P. F. Velleman and P. Pratt. A graphical interface for data analysis. *Journal of Statistical Computation and Simulation*, 32:223–228, 1989.

[11] E. H. Chi. Principles for information visualization spreadsheets. *IEEE Computer Graphics and Applications*, pages 30–38, July/August 1998.

[12] T. Winograd and C. F. Flores. *Understanding Computers and Cognition*. Ablex, Norwood, New Jersey, USA, 1985.

[13] G. Cameron. Modular visualization environments: Past, present, and future. *Computer Graphics*, pages 3–4, November 1995.

[14] H. Senay and E. Ignatius. A knowledge-based system for visualization design. *IEEE Computer Graphics and Applications*, 14(6):36, 1994.

[15] B. E. Rogowitz and L. A. Treinish. Data structures and perceptual structures. *SPIE*, 1913:600–612, 1993.

[16] D. H. Hepting. *A New Paradigm for Exploration in Computer-Aided Visualization*. PhD thesis, Simon Fraser University, 1999.

[17] G. S. Dell and P. G. O-Seaghdha. Stages of lexical access in language production. *Cognition*, 42:287–314, 1992.

[18] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MA, 1989.

[19] S. M. Casner. A task-analytic approach to the automated design of graphic presentations. *ACM Transactions on Graphics*, 10(2):111–151, April 1991.

[20] M. Sicard and J.-A. Marck. The importance of mental perception when creating research pictures. In R. Sutherland and J. Mason, editors, *Exploiting Mental Imagery with Computers in Mathematics Education*, pages 252–262. Springer-Verlag, New York, 1995.

[21] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141, 1986.

[22] M. E. Jessup. Scientific visualization: Viewpoint on collaborations of art, science, and engineering. *SIGBIO Newsletter*, pages 1–9, February 1992.

[23] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan-Kaufman, 1999.

[24] R. B. Haber and D. A. McNabb. Visualization idioms: A conceptual model for scientific visualization systems. In *Visualization in Scientific Computing*, pages 74–93. IEEE Computer Society Press, 1990.

[25] R. R. Springmeyer, M. M. Blattner, and N. L. Max. A characterization of the scientific data analysis process. In A. E. Kaufman and G. M. Nielson, editors, *Proceedings of Visualization '92*, pages 235–242, 1992.

[26] S. Kochhar, J. Marks, and M. Friedell. Interaction paradigms for human-computer cooperation in graphical-object modelling. In S. MacKay and E. M. Kidd, editors, *Proceedings of Graphics Interface '91*, pages 180–189, 1991.

[27] C. Upson et al. The application visualization system: A computational environment for scientific visualization. *IEEE Computer Graphics and Applications*, 9(4):30–42, 1989.

[28] S. F. Roth, J. Kolojejchick, J. Mattis, and M. Chuah. Sagetools: An intelligent environment for sketching, browsing, and customizing data-graphics. In *Proceedings CHI'95 Human Factors in Computing Systems*, pages 409–410. ACM Press, 1995.

[29] S. Kochhar, M. Friedell, and M. LaPolla. Cooperative, computer-aided design of scientific visualizations. In *Proceedings of Visualization '91*, pages 306–313, 1991.

[30] J. Marks et al. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Computer Graphics: SIGGRAPH '97 Conference Proceedings*, 1997.

[31] J. W. Schooler and T. Y. Engstler-Schooler. Verbal overshadowing of visual memories: Some things are better left unsaid. *Cognitive Pyschology*, 22:36–71, 1990.

[32] J. R. Koza. *Genetic Programming : On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)*. MIT Press, 1992.

[33] W. J. Schroeder, K. M. Martin, and W. E. Lorensen. The design and implementation of an object-oriented toolkit for 3D graphics and visualization. In *Proceedings of Visualization 96*, pages 93–100, 1996.