ViA: A Perceptual Visualization Assistant

Christopher G. Healey[* a], Robert St. Amant[a], Mahmoud S. Elhaddad[a]

[a]Department of Computer Science, North Carolina State University

ABSTRACT

This paper describes an automated visualization assistant called ViA. ViA is designed to help users construct perceptually optimal visualizations to represent, explore, and analyze large, complex, multidimensional datasets. We have approached this problem by studying what is known about the control of human visual attention. By harnessing the low-level human visual system, we can support our dual goals of *rapid* and *accurate* visualization. Perceptual guidelines that we have built using psychophysical experiments form the basis for ViA. ViA uses modified mixed-initiative planning algorithms from artificial intelligence to search for perceptually optimal data attribute to visual feature (data-feature) mappings. Our perceptual guidelines are integrated into evaluation engines that provide evaluation weights for a given data-feature mapping, and hints on how that mapping might be improved. ViA begins by asking users a set of simple questions about their dataset and the analysis tasks they want to perform. Answers to these questions are used in combination with the evaluation engines to identify and intelligently pursue promising data-feature mappings. The result is an automatically-generated set of mappings that are perceptually salient, but that also respect the context of the dataset and users' preferences about how they want to visualize their data.

Keywords: artifical intelligence, color, computer graphics, human vision, mixed-initiative planning, multidimensional dataset, preattentive processing, LRTA*, scientific visualization, texture

1. INTRODUCTION

Scientific visualization is the conversion of collections of strings and numbers (often called datasets) into images that are used for visual exploration and analysis. One of the fundamental research issues in this area is the visualization of "multidimensional datasets". A multidimensional dataset contains data elements that encode many separate attributes. For example, consider a dataset of environmental conditions for the continental United States. Each data element contains a corresponding weather station's latitude and longitude, the time and day that measurements were recorded, and weather conditions like temperature, pressure, humidity, wind speed and direction, and precipitation. Our challenge is to find a way to display these values simultaneously without overwhelming the viewer's ability to understand what they are seeing.

Recent work in our laboratory has studied ways to harness the low-level human visual system during multidimensional visualization. When we look at an image, certain visual features can be identified very quickly, without the need for search. These features are often called preattentive, because their detection precedes focused attention in the low-level human visual system.[1–6] Preattentive features include visual properties like color, brightness, orientation, size, and motion. When applied properly, these features can be used to perform exploratory data analysis. Examples include searching for data elements with a unique feature, identifying the boundaries between groups of elements with common features, tracking groups of elements as they move in time and space, and estimating the number of elements with a specific visual feature. Preattentive tasks are performed very rapidly and accurately; they can often be completed in a "single glance" of 200 milliseconds (msec) or less. Our human vision studies focus on identifying relevant results in the vision and psychophysical literature, then extending these results and integrating them into a visualization environment. Tools that make use of preattentive vision offer a number of important advantages for multidimensional visualization:

1. Visual analysis is rapid, accurate, and relatively effortless since preattentive tasks can be completed in 200 msec or less; tasks performed on static displays extend to a dynamic environment where data frames are shown one after another in a movie-like fashion.[7]

2. The time required for task completion is independent of display size; users can increase the number of data elements in a display with little or no increase in the time required to analyze the display.[1,4–6,8–10]

[*] Correspondence: Christopher G. Healey, CB #7534, 1010 Main Campus Drive, NC State University, Raleigh, NC, 27695-7534; Email: healey@csc.ncsu.edu

3. Certain combinations of visual features cause interference patterns that mask information in the low-level visual system; by identifying these situations, our visualization tools can be built to avoid data-feature mappings that might interfere with the analysis task.[4,11–14]

To date, we have compiled an interlocking collection of results on the use of color (hue and luminance)[15] and texture (size, density, and regularity)[16,17] for multidimensional visualization. These results have been used to visualize a number of real-world applications including medical scans,[18] weather tracking,[16,17] and scientific simulations.[19]

Our perceptual guidelines support the construction of perceptually salient visualizations for an underlying dataset. Unfortunately, these "visual mappings" (the assignment of a visual feature to represent each attribute in the dataset) must still be built by hand. This process is often difficult and time consuming, since users are asked to:

- apply a collection of interrelated guidelines,
- make various trade-offs to ensure none of the guidelines are violated, and
- decide at each step in this process how to move towards the best possible mapping for the given dataset and analysis tasks.

In short, users must still act as "visualization experts" when applying our guidelines to view their data. In order to alleviate this requirement, we are constructing an automated visualization assistant called ViA. ViA will free users from having to understand and manipulate the guidelines needed to build effective visualizations. They will only be responsible for questions that draw on their expert knowledge about their data, specifically: "What is the logical structure of the dataset?" and "What types of exploration and analysis tasks need to be performed?" Answers to these questions will be combined with our perceptual guidelines using an artifical intelligence (AI) mixed-initiative search algorithm. This algorithm will intelligently search the space of all possible mappings to find only those mappings that are perceptually optimal, and that visualize the underlying dataset in a way that effectively supports the users' analysis goals.

1.1. Previous Work

Previous work in visualization has addressed the goal of automating the selection of a visual mapping M. This work provided much of the insight and inspiration that we are using to construct ViA. Wehrend and Lewis built a classification system to describe visualization techniques in a domain-independent manner[20]; these classifications are used to try to suggest an appropriate M. A similar technique was described by Lohse et al.[21] Robertson used a natural scene paradigm to guide the choice of visual representations for data[22–24]; this methodology identifies the types of information conveyed by a particular representation, then attempts to match these to the underlying characteristics of a dataset during the selection of M. Mackinlay described automated methods that used measures of expressiveness and effectiveness to develop 2D graphical presentations.[25] Beshers and Feiner used similar rules to build graph-based "worlds within worlds" to visualize multidimensional data.[26,27] Senay and Ignatius extended Mackinlay's work to 3D using a visualization system (Vista) built on heuristic rules.[28] Gallop proposed data models and structures to classify visualizations.[29] Rogowitz and Treinish described a rule-based visualization architecture designed to represent continuous surfaces.[30] A simple table lookup scheme is used to match a dataset with certain properties to an appropriate M. The paper advocated the use of perceptual rules to guarantee that n-fold increases in an attribute's value resulted in a perceptual n-fold increase in the visual feature used to represent that attribute. Unfortunately, the construction of these perceptual rules was left as work-in-progress. Later studies by Bergman et al.[31] described a colormap tool that used system-generated and user-provided information about a dataset to limit a viewer's choice of color scales during visualization.

1.2. Research Goals

Although each of the existing automated visualization techniques perform well in their target environments, none of them completely address six important goals that we feel are critical for a generalized visualization assistant:

- *multidimensional:* visualization of multidimensional datasets must be considered; this is significantly more difficult than mapping a single visual feature to a single data attribute,
- *robust rules:* visual features should be selected using a flexible collection of perceptual guidelines, rather than relying only on anecdotal rules and experience-based intuition,

- *dataset terminology:* a standard method is needed to describe the important properties of a dataset D and the exploration tasks a user wants to perform,
- *intelligent search:* an intelligent search technique should be used to find the most appropriate visualizations for a particular dataset and analysis task,
- *multiple mappings:* multiple data-feature mappings should be offered to the user, allowing them to view their dataset in various ways; the use of different mappings will often highlight different areas of interest that no single mapping can completely capture, and
- *user control:* users should be able to control certain properties of the data-feature mappings ViA produced; this includes applying constraints (*e.g.,* the ability to force the choice of a particular visual feature for a particular attribute) and stating preferences (*e.g.,* the ability to mark certain mappings as "preferable" to others based on context or other user-chosen criteria).

ViA will address these goals in the following manner:

1. We will apply a collection of perceptual rules built using psychophysical experiments. We believe this provides a flexible framework that will support a more general collection of application environments, moreover, our investigations are specifically designed to support multidimensional data visualization.
2. Part of our design of ViA will include the construction of a formal taxonomy to classify the important properties of a dataset D for use during the selection of a visual mapping M.
3. ViA will use modified mixed-initiative search algorithms to find collections of perceptually optimal data-feature mappings.
4. ViA will allow users to impose constraints during search, or to tag certain mappings as "more desirable" than others when results are returned.

The use of AI search algorithms will allow us to move intelligently through the entire space of all possible mappings, returning the top k mappings found during the search. Perceptual guidelines from our human vision studies will be integrated into evaluation engines that provide the search algorithms with evaluation weights for a given M, and hints on how M might be improved (Figure 1). Although a completely automated assistant might seem to be an appropriate goal, we do not believe this is feasible. The evaluation engines cannot be perfect, and specific datasets may have unique properties that cannot be addressed in a general way by ViA. The strength of ViA is its ability to produce a collection of mappings that satisfy the rules of human perception. Any one of these mappings can then be extended by the user to include context or to highlight dataset-specific properties.

2. ARCHITECTURE

ViA's architecture is based on conventional principles for building knowledge-based systems in artificial intelligence. Its function can be divided into a general-purpose search engine and a set of independent knowledge sources (evaluation engines) that guide the search effectively through the visualization domain. The search engine is based on LRTA*, an incremental, real-time search algorithm (described in more detail in the Heuristic Search section below). Each decision made by the search engine is guided in part by an evaluation of its current results, as returned by some specific set of the evaluation knowledge sources. Each of these knowledge sources encodes a collection of visualization rules based on robust, experimentally validated information about perception. These rules explicitly consider the potential interactions between different visual features when they are used to visualize multiple data attributes simultaneously in a single display.

The two conceptual components of ViA meet all four of our design requirements for an automated visualization process. The knowledge sources represent robust evaluation rules that produce multidimensional data-feature mappings. A standard taxonomy will be used to describe the dataset properties and analysis tasks used to evaluate an individual mapping. Instead of choosing a single mapping for a given set of data attributes and stopping, the search process will continue to consider all the mappings that meet the criteria of its evaluation knowledge sources. Some of the multiple mappings will be more effective for visualization than others; the incremental search process can be directed by the user, with mapping results evaluated and pruned based both on objective information about the dataset and subjective preferences defined by the user.

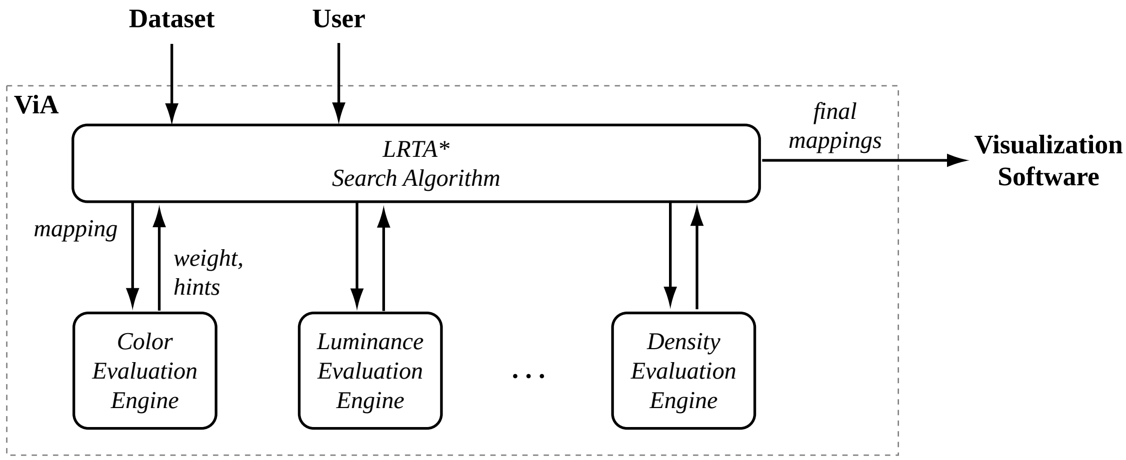

Figure 1. An overview of ViA's architecture, including an LRTA* mixed-initiative search algorithm and a collection of evaluation engines, one for each visual feature available for use during visualization; information about the data attributes and analysis tasks come from both the user and the dataset itself; the final data-feature mappings chosen by ViA are used by an external visualization software system to control the display of the dataset

The search begins by asking several questions about the dataset to be visualized. First, users rank data attributes with regard to their importance, producing either a total or partial ordering. Users must then indicate whether each variable is continuous or discrete, ordinal or nominal. ViA will use this information to take advantage of an existing ordering on attribute values in the visualizations it generates. Users can also associate an analysis task with any attribute (*e.g.,* whether the goal of the visualization is search, boundary detection, tracking, estimation, and so on). This will further constrain the types of visual features used to display this attribute. Finally the system itself determines various properties of each attribute (*e.g.,* its spatial frequency, number of unique values, and domain).

Once the search begins, it is guided in two ways. First, the search engine passes its incrementally constructed mappings to the evaluation engines for examination. On reaching a maximum in its evaluation process, the engine can pause to present the user with its results up to that point. The user can review the selected mappings and indicated which components should be retained and which should be revised. Alternatively, the user can select one or more of the lower-ranked mappings and instruct the search engine to continue in a new direction. At this intermediate stopping point the user can also tune various dataset properties like the importance ranking of attributes, to guide the search toward different combinations in the resulting visualization. This approach reflects our past work on automated assistance for statistical reasoning.[32,33]

3. EVALUATION ENGINES

The mixed-initiative search algorithm needs a way to evaluate visual mappings; these evaluations are used to identify visualizations that are most appropriate for the given dataset and analysis tasks. Each mapping M selected by the search algorithm will be examined by a collection of "evaluation engines". A mapping M is a relationship that associates a visual feature v_j with each attribute A_j in the dataset. A separate engine will be built to evaluate each visual feature available during the construction of M (*e.g.,* one engine for color, one for luminance, one for density, and so on, see also Figure 1). ViA will use these engines to obtain an evaluation weight for every attribute to visual feature pairing $A_j \rightarrow v_j \in M$.

The design of our evaluation engines is built on results from our human vision studies. Our initial work in this area found that viewers can perform high-speed estimation using both color and orientation (*i.e.,* estimating the percentage of elements colored blue, or the percentage of elements rotated $45°$ off horizontal). Our results also demonstrated that neither feature interferes with the other during estimation.

We then studied methods for choosing collections of colors that can be rapidly distinguished from one another.[15] Our findings showed that three selection criteria must be considered: the distance between colors in a perceptually balanced color space, the ability to separate each color from all the others with a straight line, and the named color region each color occupies. Proper application of these criteria produces up to seven isoluminant colors that can be easily distinguished from one another, even when all the colors are shown simultaneously in a single display.

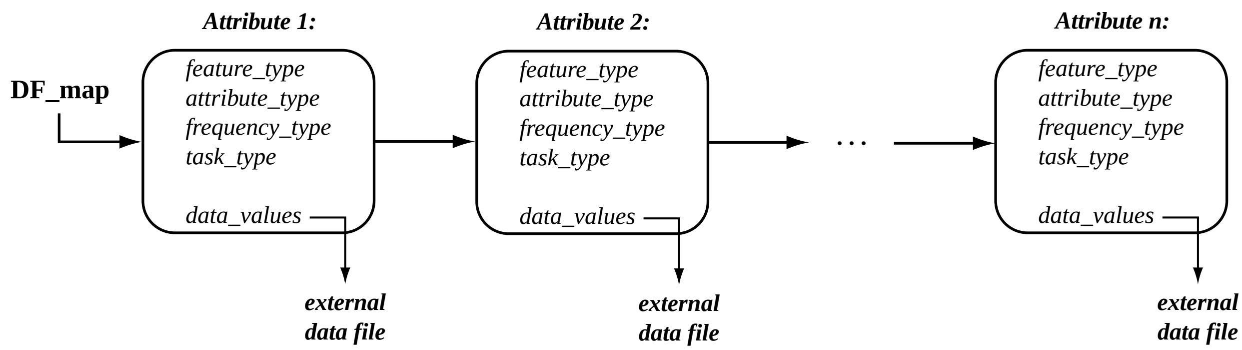

Figure 2. The *DF_map* structure, a linked list of nodes, each node contains information for a single attribute, including: attribute type (discrete or continuous, ordinal or nominal), frequency type (low or high), the visual feature currently being used to represent the attribute, the analysis task users want to perform on the attribute, and a link to a binary file that contains the actual data values

We have also conducted studies to identify the perceptual dimensions that combine to produce visual texture patterns.[16] Our results allowed us to construct pexels (perceptual texture elements) that visualize information in a dataset through the variation of each pexel's size (or height), density (or contrast), and regularity of placement. A final set of experiments showed the presence of color-on-texture interference: random variations in color can mask underlying texture patterns, but variations in perceptual texture dimensions have no effect on a viewer's ability to identify individual colors.[17] Our current research is extending this perceptual foundation through the study of 2D and 3D orientation, and the perceptual components of motion (*e.g.,* flicker, direction, velocity, and coherence).

Our collection of guidelines on the use of color, texture, and motion during visualization form a solid foundation on which to construct our evaluation engines. As mentioned in the Architecture section, we must also identify the dataset properties needed to correctly apply our guidelines during the evaluation of M. Previous work[24,29–31] suggests these could include an attribute importance ordering, the spatial frequency of each $A_j \in D$, the domain type of A_j (continuous or discrete, ordinal or nominal), and the exploration and analysis tasks the viewer wants to perform. Some of these properties (*e.g.,* spatial frequency) can be inferred from the dataset. Others (*e.g.,* exploration and analysis tasks) need to be specified by the viewer. ViA will test each attribute-feature pair $A_j \rightarrow v_j \in M$ for:

- *interference:* can other attribute-feature mappings $A_k \rightarrow v_k \in M$ visually interfere with $A_j \rightarrow v_j$?
- *task applicability:* does v_j support the exploration and analysis tasks the viewer wants to perform on A_j?
- *attribute domain type:* does v_j support the domain type of A_j (*e.g.,* can v_j produce continuous scales if A_j is continuous, or can v_j produce an appropriate number of distinguishable values if A_j is discrete?)
- *spatial frequency:* is v_j appropriate for representing the spatial frequency of A_j?

Information about a particular M is stored in a *DF_map* data structure (Figure 2). This data structure is used to pass mappings between the LRTA* search algorithm and the evaluation engines. A *DF_map* is a linked list of nodes, one for each attribute-feature pair $A_j \rightarrow v_j \in M$. A node encodes the visual feature being used to display the attribute, the analysis task users want to perform on the attribute (if any), the attribute's spatial frequency, and its type (continuous or discrete, ordinal or nominal). A reference is also stored to the binary file that contains the raw data values.

Each node in the *DF_map* is examined separately. The visual feature v_j being used to display the node's attribute determines which evaluation engine to invoke. The evaluation engine returns a weight between 0.0 (a completely flawed attribute-feature pairing) and 1.0 (a perfect attribute-feature pairing). Any violation of our perceptual guidelines will produce a decrease in the weight returned. The engine will also try to provide "hints" to fix the problem, thereby improving M. For example, suppose $A_j \rightarrow color$. The color evaluation engine would check to see if $A_k \rightarrow luminance \in M$. If so, and if $A_j > A_k$ in terms of attribute importance, there may exist a luminance interference effect (*i.e.,* background luminance patterns used to display A_k may mask color patterns attached to the more important attribute A_j). The engine would also check to see if A_j had a high spatial frequency, since color (particularly isoluminant color) is not well-suited for representing fine spatial detail. Either case would cause the evaluation engine to return a low weight for $A_j \rightarrow color$. In both cases it would also "hint" to use luminance rather than color to represent A_j.

Hints are returned as a linked list of nodes that contain two values: *feature_type* and *improvement_weight*. *feature_type* is a new visual feature recommended by the evaluation engine to represent the given attribute (in our previous example, *feature_type* would be luminance). *improvement_weight* is the engine's estimated improvement in the evaluation weight if *feature_type* were adopted.

The search algorithm will collect all the evaluation weights and corresponding hints, then use the hints to direct its search to a new set of mappings. Hints from different evaluation engines can conflict with one another. The search algorithm must decide which hints to apply and which to ignore; this will often produce multiple search paths to further test the improvement in M offered by different collections of hints. If any of the hints are valid, some of the new mappings should produce better evaluation weights. In this way, ViA can restrict its search to small groups of mappings with a strong potential for improvement.

4. MIXED-INITIATIVE SEARCH

We take a search perspective on the selection of an appropriate mapping between data attributes and visual features. Classical problem-solving algorithms require the specification of a state space (including an initial state and a set of goal states), a set of operators that transform one state into another, constraints on the application of the operators, and appropriate knowledge to guide the selection of operators in a given state.

We can formalize the problem of generating a mapping between data attributes and visual features as follows. Let $D = \{d_1, \ldots, d_n\}$ be the set of all attributes for a given dataset, $F = \{f1, \ \ldots \ , f_m\}$ the set of all available visual features. A point p in the state space is an ordered pair (D_p, F_p) where $D_p \subseteq D$ and $F_p \subseteq F$. The initial state is characterized by (D, F), while a solution point is characterized by $(D_s \subset D, F_s \subset F)$. The set of operations $O = \{o_{ij} | i \in [1, 2, \ldots n], j \in [1, 2, \ldots, m]\} \subseteq D \times F$ is a set of all possible assignments $A_i \rightarrow v_j$ of dataset attributes to visual features. In other words, a search algorithm will move through the search space by removing a dataset attribute and a visual feature from (D_p, F_p) and associating them. As the search progresses a partial mapping is constructed. At some point a mapping is selected as a solution.

In the remainder of this section we discuss the characteristics of this problem domain, such as the requirements for responsiveness and user involvement. We then present the rationale for using a real-time heuristic search technique, LRTA*, which provides a good match for the properties of the domain.

4.1. Problem Domain Characteristics

The number of visual features available for assignment to dataset attributes in the current system is limited to a relatively small number: color, luminance, height, density, and regularity. If we assume that the user provides a complete preference ordering on dataset attributes, a simple solution might be as follows:

1. Select the top N dataset attributes, in order, where N is the number of visual features we wish to display.
2. Treat every permutation of visual features as a mapping to the fixed ordering of dataset attributes.
3. Evaluate each permutation and select the best one.

Because we have set an upper bound on the number of dataset attributes that can be simultaneously visualized, and we have fixed their ordering in terms of importance, the number of nodes in the largest mapping problem is $T(n, n) = n! = 120$ nodes — a small search space by most measures. Our tentative solution formulates the mapping problem as one of optimization, and solves it by exhaustively searching the bounded problem space for the optimal solution.

This would be acceptable only if the application domain requirements provide no additional constraints. This is not the case. First, the running time of this algorithm is exponential in the size of the set of visual features, and we have no reason to believe that this set will remain at its current tractable size. Once some number of novel visual features (*e.g.,* the addition of flicker, direction and velocity of motion, and motion coherence) have been incorporated, a brute force approach will break down. Second, we cannot expect that users will always provide a total ordering on dataset attributes; the user may be interested in more attributes than can be visualized at once, which leads to a combinatorial explosion in the number of visualizations that must be generated in a brute force search.

Third, and most important, the automated evaluation provided by the evaluation engines for a mapping is heuristic. That is, the search guidance may lead to the best possible mapping, but this is not the only plausible result. Evaluation knowledge sources have been developed for general, domain-independent and user-independent applications. They may not match the specific patterns in a dataset; they may not meet the individual preferences of a user; for novel visual features, they may require tuning for their integration into the more comprehensive evaluation process. An algorithm that addresses this uncertainty must be interactive, to take advantage of user control knowledge in selecting a path through the search space. If the algorithm reaches a specific solution and the user decides that it is inappropriate, the algorithm must be able to return to its search to produce alternative suggestions.

The interactive generation of visualization mappings gives rise to three requirements that move beyond simply allowing steering by the user. The system must also be incremental, responsive, and adaptive. Equipped with plausible but uncertain heuristics, the search engine should be able to move the search process into regions of the state space populated by good mappings, but in the end this determination may need to be made by a user. If fine tuning (and sometimes even the broader search process) is open to user direction, then the process must be incremental (rather than a black box search) in order to allow for this guidance. This implies that the system must be responsive, taking into account the standard requirements of interactive computing. We can divide the visualization process into several stages, including the selection of a mapping, the refinement of the mapping into a visualization specification (*e.g.,* discretizing a variable for color selection), and the actual rendering of the visualization. If the generation of good mappings grows too time-consuming, this can reduce user satisfaction with the entire process. Finally, the system must be adaptive. Some of the earliest, relatively fragile expert systems produced a single solution to a given problem and presented it to the user. If the user judged the solution inadequate but found no difficulties with the problem representation, the system could not recover from the failure. An effective system in the visualization domain must learn about the search space as it is traversed, to eliminate the possibility of following paths found unfruitful in previous episodes and to adapt its evaluation to the user's choices.

4.2. Heuristic Search

Real-time search techniques fall into the class of "anytime" algorithms, algorithms that deliberate for some variable period of time in search of an optimal solution.[34] An anytime algorithm halts when the cost of continuing is perceived to be greater than the benefit of the eventual solution. At this point the algorithm returns with the action it evaluates as the current best. After committing to an action, real-time search techniques are often required to start a new search episode, taking the state after the action as the new initial state. These algorithms can thus incorporate techniques for learning the topology of the search space over the multiple search trials. We can make a direct analogy between interactive visualization and real-time computing; in fact, some argue that all forms of interactive computing belong in the real-time arena.

In the class of real-time heuristic search techniques, Learning Real-Time A* (LRTA*), which is guaranteed not to over-estimate the promise of previously visited nodes, is well-suited to learning over multiple trials.[35] We base our development of the heuristic function for this problem on the independent evaluation knowledge sources discussed above. These knowledge sources implement functional mappings, $e_{i,j} : (A_i \to v_j) \to [0,1] \in \mathcal{R}$, that depend on the characteristic of the input dataset and the proposed associated visual feature. An evaluation $e_{i,j}$ is a value between 0 and 1 that reflects the merit of the assignment $A_i \to v_j$. We define the intrinsic promise or heuristic estimation of a point s in the state space to be

$$Q(s) = 1 + 1/n \sum_{j=1}^{n} x_j (e_{S(j),j} - 1) \tag{1}$$

where x_j is a Boolean variable that is assigned a truth value that reflects whether the state s provides an assignment for dataset j, and $S(j)$ is a function that computes the feature assigned to dataset j under state s.

This function satisfies the admissibility requirement for optimality, an important property for LRTA*. Informally, admissibility in a search heuristic ensures that the search algorithm will not halt with a solution that is less than optimal. The above heuristic estimate can be viewed as representing an overestimate of user satisfaction with the mappings implied by s. If we think of the heuristic estimate as a cost function, then $1 - Q(s)$ is an underestimate for the user satisfaction of the given mapping. The intrinsic estimate of dissatisfaction, however, does not capture the cost of deliberation time to compute a mapping; it assumes that the user is only concerned about visualization quality and not with the time spent to achieve it. One simple approach to this problem is using the intrinsic dissatisfaction function to compute the aggregate cost of moving to a certain point in the solution space through the incorporation of a time variable.

Let $Q(s)$ be the intrinsic cost estimate. Then the aggregate cost function $f(s)$ can be modeled as

$$f(s) = A(1 - Q(s)) + (1 - A)(1 - Q(s))[1 - e^{-Bt}] \quad (2)$$

where $A \in [0, 1]$ is a tuning parameter to reflect the user's appreciation of visualization quality over deliberation time, B is time constant that denotes the rate of depreciation of intrinsic quality over time, and t is the total number of node expansions to reach the point s in the state space. It directly follows that as deliberation time increases the search technique develops a tendency toward delivering acceptable solutions rather than pursuing optimality.

5. CONCLUSIONS

To summarize, ViA will advance research on rule-based visualization and the automated selection of data-feature mappings by:

- building a formal taxonomy to enumerate both D's properties and the exploration and analysis tasks to be performed,
- constructing a set of evaluation engines to rank the perceptual salience of any data-feature mapping M for an underlying dataset D,
- testing the applicability of mixed-initiative search algorithms that allow external hinting for the selection of a collection of mappings, and
- comparing automatically constructed mappings to those designed by domain experts, or to those built by hand from guidelines in our perceptual foundation.

The result is a software system viewers can use to build data-feature mappings for their multidimensional datasets. ViA will ask the viewer a limited set of questions, then suggest a collection of mappings most appropriate for their exploration and analysis tasks. ViA will be integrated into our perceptual display software, so any mappings it suggests can be immediately applied to visualize an underlying dataset.

The current ViA system is now working in a prototype environment. The LRTA* search algorithm has been implemented; several evaluation engines (including those for color, luminance, size, density, and regularity) have been constructed. Techniques have been built to allow the Lisp-based LRTA* and C++-based evaluation engines to communicate with one another using *DF_map* and *hint* data structures. The combined system is capable of suggesting collections of basic data-feature mappings for an underlying multidimensional dataset.

Our future work will take several directions from the foundation we have established. First, we will continue development on knowledge sources for visual features. These will include the refinement of existing knowledge sources and the exploration of novel visual features. Second, we will examine in more detail the complementary contributions of the user and the system to the mapping search. Considerable research has been devoted to the concept of interactive statistical strategies, formal descriptions of the actions and decisions involved in applying statistical tools to a problem.[36] We believe that comparable interactive visualization strategies can be defined. These will benefit users of differing expertise in a complex, knowledge-intensive process. Third, ViA will be evaluated by testing the effectiveness of the mappings it produces. This will be done using a set of real-world datasets that include results from CT medical scans,[18] storm and typhoon tracking,[16,17] and marine biology simulations.[19] Data-feature mappings produced by ViA will be compared to mappings designed by domain experts in each area, and to mappings built by hand using our perceptual guidelines. We will conduct validation studies to test a viewer's ability to perform application-specific exploration and analysis using the ViA-based, domain expert, and hand-built mappings. Results from these studies will highlight the strengths and weaknesses of each category of mapping. It will also allow us to improve ViA by identifying situations where it fails to locate the most effective mappings.

REFERENCES

1. J. Duncan and G. W. Humphreys, "Visual search and stimulus similarity," *Psychological Review* **96**(3), pp. 433–458, 1989.
2. H. E. Egeth and S. Yantis, "Visual attention: Control, representation, and time course," *Annual Review of Psychology* **48**, pp. 269–297, 1997.
3. B. Julész and J. R. Bergen, "Textons, the fundamental elements in preattentive vision and perception of textures," *The Bell System Technical Journal* **62**(6), pp. 1619–1645, 1983.
4. A. Triesman, "Preattentive processing in vision," *Computer Vision, Graphics and Image Processing* **31**, pp. 156–177, 1985.

5. A. Triesman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology* **12**, pp. 97–136, 1980.
6. J. M. Wolfe, "Guided Search 2.0: A revised model of visual search," *Psychonomic Bulletin & Review* **1**(2), pp. 202–238, 1994.
7. C. G. Healey, K. S. Booth, and J. T. Enns, "Real-time multivariate data visualization using preattentive processing," *ACM Transactions on Modeling and Computer Simulation* **5**(3), pp. 190–221, 1995.
8. A. Triesman, "Search, similarity, and integration of features between and within dimensions," *Journal of Experimental Psychology: Human Perception & Performance* **17**(3), pp. 652–676, 1991.
9. J. M. Wolfe, S. R. Friedman-Hill, M. I. Stewart, and K. M. O'Connell, "The role of categorization in visual search for orientation," *Journal of Experimental Psychology: Human Perception & Performance* **18**(1), pp. 34–49, 1992.
10. J. M. Wolfe, K. P. Yu, M. I. Stewart, A. D. Shorter, S. R. Friedman-Hill, and K. R. Cave, "Limitations on the parallel guidance of visual search: Color $\times$ color and orientation $\times$ orientation conjunctions," *Journal of Experimental Psychology: Human Perception & Performance* **16**(4), pp. 879–892, 1990.
11. T. C. Callaghan, "Dimensional interaction of hue and brightness in preattentive field segregation," *Perception & Psychophysics* **36**(1), pp. 25–34, 1984.
12. T. C. Callaghan, "Interference and domination in texture segregation: Hue, geometric form, and line orientation," *Perception & Psychophysics* **46**(4), pp. 299–311, 1989.
13. T. C. Callaghan, "Interference and dominance in texture segregation," in *Visual Search*, D. Brogan, ed., pp. 81–87, Taylor & Francis, New York, New York, 1990.
14. R. J. Snowden, "Texture segregation and visual search: A comparison of the effects of random variations along irrelevant dimensions," *Journal of Experimental Psychology: Human Perception and Performance* **24**(5), pp. 1354–1367, 1998.
15. C. G. Healey, "Choosing effective colours for data visualization," in *Proceedings Visualization '96*, pp. 263–270, (San Francisco, California), 1996.
16. C. G. Healey and J. T. Enns, "Building perceptual textures to visualize multidimensional datasets," in *Proceedings Visualization '98*, pp. 111–118, (Research Triangle Park, North Carolina), 1998.
17. C. G. Healey and J. T. Enns, "Large datasets at a glance: Combining textures and colors in scientific visualization," *IEEE Transactions on Visualization and Computer Graphics* **5**(2), pp. 145–167, 1999.
18. R. Tam, C. G. Healey, and B. Flak, "Volume visualization of abdominal aortic aneurysms," in *Proceedings Visualization '97*, pp. 43–50, (Phoenix, Arizona), 1997.
19. C. G. Healey, "One the use of perceptual cues and data mining for effective visualization of scientific datasets," in *Proceedings Graphics Interface '98*, pp. 177–184, (Vancouver, Canada), 1998.
20. S. Wehrend and C. Lewis, "A problem-oriented classification of visualization techniques," in *Proceedings Visualization '90*, pp. 139–143, (San Francisco, California), 1990.
21. J. Lohse, H. Rueter, K. Biolsi, and N. Walker, "Classifying visual knowledge representations: A foundation for visualization research," in *Proceedings Visualization '90*, pp. 131–138, (San Francisco, California), 1990.
22. P. K. Robertson, "A methodology for scientific data visualisation: Choosing representations based on a natural scene paradigm," in *Proceedings Visualization '90*, pp. 114–123, (San Francisco, California), 1990.
23. P. K. Robertson, "A methodology for choosing data representations," *IEEE Computer Graphics & Applications* **11**(3), pp. 56–67, 1991.
24. P. K. Robertson and L. De Ferrari, "Systematic approaches to visualization: Is a reference model needed?," in *Scientific Visualization: Advances and Challenges*, L. Rosenblum, ed., pp. 239–250, Academic Press, San Diego, California, 1994.
25. J. Mackinlay, "Automating the design of graphical presentations of relational information," *ACM Transactions on Graphics* **5**(2), pp. 110–141, 1986.
26. C. Beshers and S. Feiner, "AutoVisual: Rule-based design of interactive multivariate visualizations," *IEEE Computer Graphics & Applications* **13**, pp. 41–49, July 1993.
27. C. Beshers and S. Feiner, "Automated design of data visualizations," in *Scientific Visualization: Advances and Challenges*, L. Rosenblum, ed., pp. 87–102, Academic Press, San Diego, California, 1994.
28. H. Senay and E. Ignatius, "A knowledge-based system for visualization design," *IEEE Computer Graphics & Applications* **14**(6), pp. 36–47, 1994.
29. J. Gallop, "Underlying data models and structures for visualization," in *Scientific Visualization: Advances and Challenges*, L. Rosenblum, ed., pp. 87–102, Academic Press, San Diego, California, 1994.
30. B. E. Rogowitz and L. A. Treinish, "An architecture for rule-based visualization," in *Proceedings Visualization '93*, pp. 236–243, (San Jose, California), 1993.

31. L. D. Bergman, B. E. Rogowitz, and L. A. Treinish, “A rule-based tool for assisting colormap selection,” in *Proceedings Visualization ’95*, pp. 118–125, (Atlanta, Georgia), 1995.
32. R. St. Amant and P. R. Cohen, “Intelligent support for exploratory data analysis,” *Journal of Computational and Graphical Statistics* **7**(4), pp. 545–558, 1998.
33. R. St. Amant and P. R. Cohen, “Interaction with a mixed-initiative system for exploratory data analysis,” *Knowledge-Based Systems* **10**(5), pp. 265–273, 1998.
34. S. Zilberstein, *Operational Rationality through Compilation of Anytime Algorithms*. PhD thesis, University of California at Berkeley, 1993. Department of Computer Science.
35. R. E. Korf, “Real-time heuristic search,” *Artificial Intelligence* **42**, pp. 189–211, 1990.
36. D. Hand, “Patterns in statistical strategy,” in *Artificial Intelligence and Statistics*, W. Gale, ed., pp. 355–387, Addison-Wesley Publishing Company, 1986.