In Proceedings of UIST ‘01, Orlando Florida, November 2001.

Parallel Bargrams for Consumer-based Information Exploration and Choice

Kent Wittenburg, Tom Lanning, Michael Heinrichs, Michael Stanton
Verizon Laboratories
40 Sylvan Rd.
Waltham, MA 02451 USA
kent.wittenburg@verizon.net,{tom.lanning,mike.heinrichs}@verizon.com,michael_stanton@brown.edu

ABSTRACT
In this paper we introduce multidimensional visualization and interaction techniques that are an extension to related work in parallel histograms and dynamic querying. Bargrams are, in effect, histograms whose bars have been tipped over and lined up end-to-end. We discuss affordances of parallel bargrams in the context of systems that support consumer-based information exploration and choice based on the attributes of the items in the choice set. Our tool called EZChooser has enabled a number of prototypes in such domains as Internet shopping, investment decisions, college choice, and so on, and a limited version has been deployed for car shopping. Evaluations of the techniques include an experiment indicating that trained users prefer EZChooser over static tables for choice tasks among sets of 50 items with 7-9 attributes.

Keywords
Information visualization, internet, electronic shopping, multidimensional visualization, e-commerce, decision support

INTRODUCTION

The task we are concerned with in this paper has been characterized as follows by Spence [13]:

> “Given a collection of objects, each described by the values associated with a set of attributes, find the most acceptable such object or, perhaps, a small number of candidate objects worthy of more detailed consideration.” — [13], p. 73

This generic task description applies to Internet shopping but also to many other related tasks such as making investment decisions or choosing colleges, towns, or real-estate. The typical set of UI techniques available to users today include database queries, static comparison tables, decision-trees, and (agent-based) recommender systems.

The limitations of classic database query techniques are well-known. Responses to queries all too often lead to zero hits or far too many hits, leading to complex follow-up dialogs. A fundamental issue is that in many decision contexts users may not be sure of what they are looking for before they start. Making a good choice may depend on the total set of choices available and how individual choices compare and interact across a number of dimensions. Users will improvise as a reaction to discovered knowledge [15], and, at best, classic querying provides only an indirect and tedious route towards formulating a mental model that can discover unanticipated solutions. For instance, in a car-buying context, if users discover the existence of better warranties or gas mileage than they had previously supposed, they may be willing to extend their previously assumed price limit. Classic querying approaches will not easily reveal such information to a user. The standard technique of deploying checkboxes in the UI to formulate queries in no way addresses these basic issues.

Static comparison tables are often made available in e-commerce contexts once users have whittled down their candidates to a sufficiently small number with classical querying techniques. Tables seem to be a good choice for comparing and contrasting small numbers of choices across small numbers of attributes. One might think that table-based attribute comparison would be equally useful for narrowing down large sets to small ones. However, large tables are difficult to process. Scrolling alone takes significant cognitive resources when the table layouts exceed screen real-estate. Labeling becomes difficult. Interactions necessary for the whittling-down process are not available with simple table renderings. Also, while sorting of table columns allows users to order a set of candidates along any single dimension, consideration of multiple attributes simultaneously cannot be accommodated within a single table view.

High-level decision-trees in the form of “guides” are another common approach to help users make complex choices. They incorporate dialogs asking the user to answer a series of questions. Even well-designed variants of the 20-questions approach falter when users can’t anticipate the consequences of taking a certain path through the decision space. In a travel domain, users don't want to fix the departure and arrival times before they know the consequences for connections or price [9]. Once they do go down a garden path, it's often the case that they are no more enlightened than when they started about the paths NOT taken. Many of their subsequent interactions may involve undoing previous choices in order to explore different paths in a depth-first tree traversal.

Agent-based recommender systems [11] provide a variety of techniques that support choice-making through similarity or criteria ranking of some kind. The system gathers various bits of direct or indirect preference information and then provides a solution. Users must trust the largely opaque system to do the job, appropriate in some highly complex situations (heavy-duty military or industrial applications) and also for some choices that are a matter of taste (films, music, food). However, in other cases the user may be left with an uncomfortable sense that they could make a better decision themselves if only the appropriate information relationships could be revealed by the system.

In contrast to these four approaches, the tack taken with such work as Dynamic Queries [12] and Attribute Explorer [16] is to investigate methods that allow users to form better cognitive models of the decision space—complex though it may be. Kirsh [6], among others, argues that computing environments can enable better decision-making by offering a sort of playground for users to manipulate the parameters of their problem.

> "The environment is not simply a reservoir of cues, constraints, and affordances for simplifying the decision process. The environment is also a realm where agents discover what they want."
> —Kirsch [6]

Environments that can be manipulated may make recall or decision tasks easier. Kirsh uses the example of manipulating the physical position of Tetris tiles in order to come to a decision about a next move. We believe that UIs for electronic shopping and other complex decision tasks should also allow users the opportunity to "play" with electronic artifacts of their decision parameters.

We now turn to the set of techniques that are the subject of this paper. Our work is in the tradition of what Tweedie has called interactive visualization artifacts [17]. We introduce the notion of a bargram, which as far as we know has not been named or precisely characterized in the literature. We then discuss the interactive affordances offered by parallel bargrams and how we have used them to build a system called EZChooser. EZChooser accepts a generic form of relational data or attribute tables, and has led to prototypes in many different choice domains. A limited version has been deployed on the Consumer Guide section of Verizon's SuperPages for new and used car shopping (http://cg.superpages.com). In later sections we discuss related work, followed by a report on evaluations of the system and, finally, our conclusion.

BARGRAMS

Bargrams, a name we have coined, are easily understood in the context of histograms. Figure 1a shows a histogram (or bargraph). A bargram (Figure 1b) is derived by 'tipping over' the columns (bars) of the histogram and laying them end-to-end, ignoring any null bins. Whereas the relative count in the histogram is reflected on the vertical axis, the bargram shows count through relative widths of the bars.

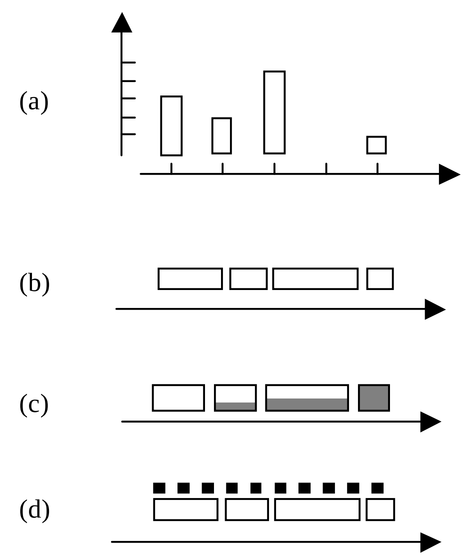

Figure 1: A histogram (a); the corresponding bargram (b); some value distribution information restored (c); and associated item vector (d).

Compared to histograms, bargrams carry less visual information about value distributions; for example, gaps are not shown (as they are in Figure 1a), and consequently outliers are not evident. Since items are simply ordered, there is no indication of whether adjacent items are close together or far apart with respect to a value scale. However, bargrams have the virtues of simplicity, and they take up relatively little vertical real estate.

A variant on the bargram restores some information regarding value distributions by reintroducing a second dimension as shown in Figure 1c: shading or some form of line graph can be added to indicate relative value distributions across the bins. The addition of shading restores some of the information lost from the original histogram.

Figure 1d shows the addition of graphical objects corresponding to the items themselves, a visual artifact we call an item vector.

AFFORDANCES OF PARALLEL BARGRAMS

As one might expect from their relationship to histograms, bargrams are quite generic. They naturally represent numeric values, but they can also be used for booleans, text, dates, and categories. Bargrams can be used in parallel to represent a potentially unlimited number of attributes

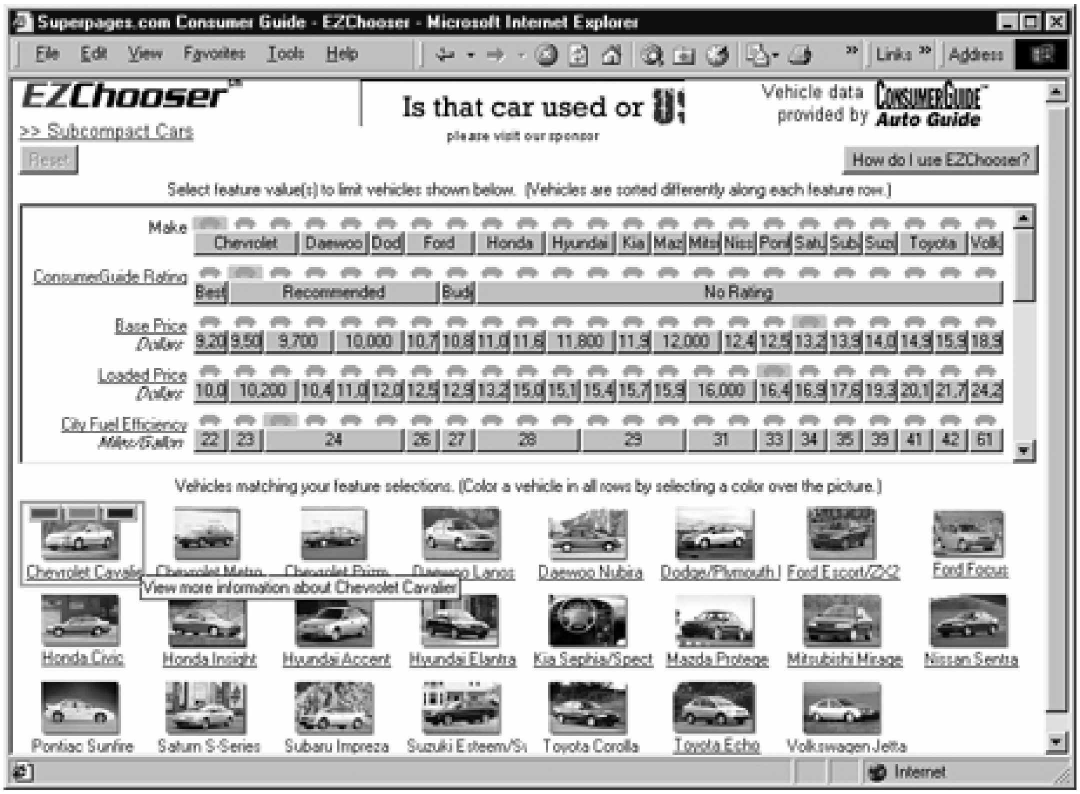

Figure 2: Screen shot of EZChooser. An upper frame contains a number of feature dimensions and the lower frame contains a presentation of the set of items that match restrictions.

(dimensions), affording brushing techniques—wherein a user selects items via their values in one dimension and those selected items' values are colored in all other dimensions [2].

Since bargrams are related directly with item counts, it is possible to associate a parallel item vector with each bargram. Each bin in a bargram holds some number of items related to its width, and those items (within limits) can be superimposed or drawn in parallel with the value bins. Figures 2 and 3 show examples of parallel bargrams with item vectors aligned just over each bargram. The items in this case are rendered as domain-specific glyphs (cars). Note then that each item vector imposes an ordering on the set of items. The same item appears in each row but typically at a different horizontal position.

An issue for this technique is that at times orderings determined by the attributes may only be partial, but the positioning may suggest—incorrectly—that something more is afoot. This issue is particularly apparent for nominal data. For example, in Figure 2 there are 15 vehicles in row 2 with "No Rating" as a value of "Consumer Guide Rating." The ordering within the set of 15 is arbitrary. However, some users may inappropriately infer a semantic difference in rank. On the flip side, it is obvious that attributes imposing a strict ranking of the set are particularly well suited to this type of display.

Value bins can be rendered naturally as buttons, suggesting an obvious selection interaction natural in the context of choice tasks. By selecting value bins through button-pressing, users can form queries whose results can be reflected back onto the display through the item vectors. We will detail these interactions in the next section in the context of our EZChooser tool.

EZCHOOSER

EZChooser is a tool designed to help general consumers solve the task of choosing one item from many based on attributes. Readers are encouraged to try the live system on SuperPages (follow links within the new or used car sections), bearing in mind that this version of EZChooser does not incorporate features appropriate for larger datasets that we will discuss here.

Layout

Figure 2 shows an example of a set of items and attributes in EZChooser. In this case, the set consists of 23 subcompact cars, whose images are listed in the bottom half of the display. In the top half of the display, feature dimensions are presented as parallel bargrams. The item vectors above each bargram render individuals as domain-specific glyphs (car shapes).

The entire presentation scales with the resizing of the hosting browser window. Dimensions rows, however, have a minimum height so a scroll bar is needed if the number of dimensions exceeds the screen real estate. The presentation

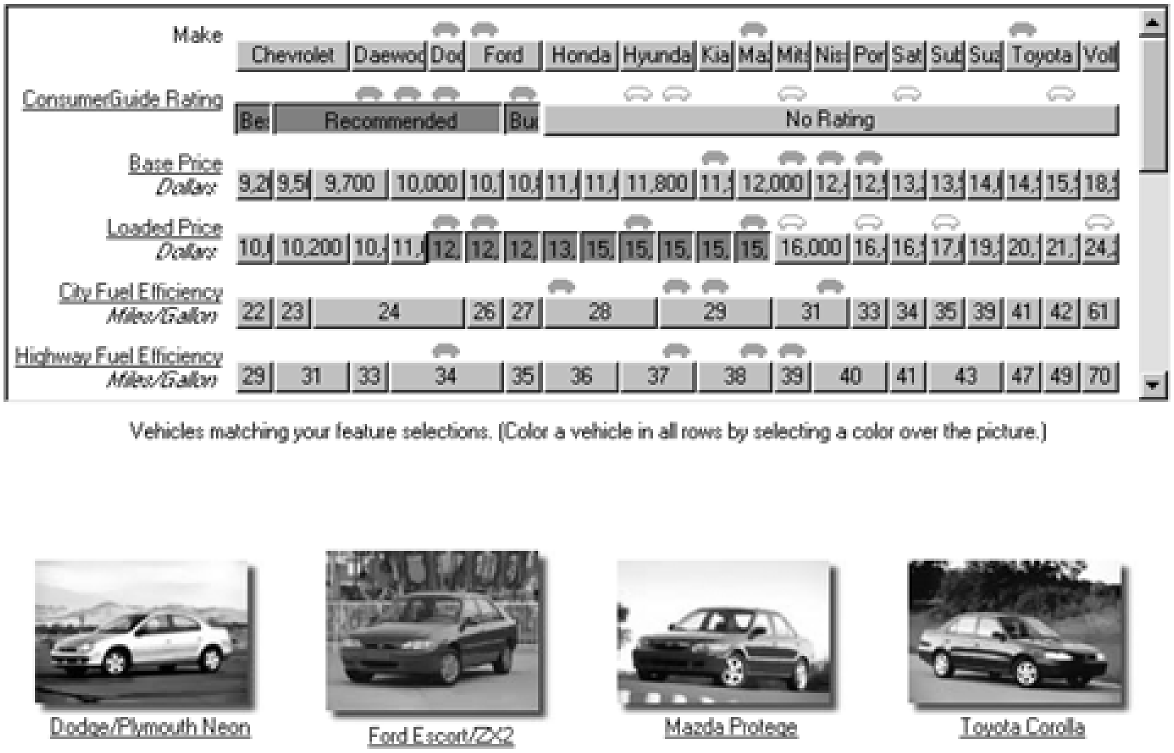

Figure 3: Querying with EZChooser. A user has selected some restrictions on the set. Glyphs above the rows indicate relationships among values and guide further restrictions.

of the matching item set, however, scales so that scrolling is not normally needed.

Brushing and Marking Interactions

Users may explore individuals in context through brushing, implemented through mousing over item glyphs or item images in the listing. In Figure 2, note that an item has been highlighted in each feature row and also in the vehicle list shown at the bottom. It is the first one in the top row of car images. (Highlighting is hard to see in the grayscale figure, but not in color.) From this interaction a user could determine where this vehicle ranks relative to others in the set. It lets users see at a glance (in color, at least) that, for instance, the car highlighted in Figure 2 is at the lower end of fuel efficiency compared to these other cars, the middle/upper range of price, and so on.

Users can scroll to see additional feature rows by using the scroll bar at the right of the dimension row frame. Since they may have to take their cursor off the highlighted item to do so, highlighting can be made sticky by clicking on a glyph or on one of the color swatches over an individual item presentation in the bottom of the screen (visible in the highlighted item in Figure 2). The system supports multiple markings with different colors. Users can, in fact, enter EZChooser from a Web page on a particular vehicle, and that vehicle will initially be marked with the color red. In this sense, users can start with an example at a detailed level, and then jump to an overview within a relevant set, a design that accommodates flexible exploration styles. We found also that users used marking as general memory cues as they explored the data sets.

Visual Queries

The value bins in each bargram row are, naturally, clickable. When users click buttons in each of the rows, they form dynamic queries [12] whose matching set is immediately shown as item presentations in the bottom half of the display. Value restrictions within dimension rows are treated as logical OR. Value restrictions across rows are logical AND.

Item vectors are utilized to represent fairly sophisticated information relative to a query. After clicking at least one value restriction button, it will be the case that some car glyphs above the value rows are drawn filled, others only outlined, and some not shown at all. The filled glyphs reflect the set of items shown at the bottom, i.e., those that match the query. If they appear over value buttons not yet selected, it is an indication that this button's additional value restriction can be invoked to restrict the set further without yielding a null result. The outlined glyphs indicate that users may choose value buttons under them to add to the set shown at the bottom. In other words, these are values which may be combined (logical OR) with selections already in those rows to add to the result set. If there are no glyphs at all over a button, that means that other value restrictions have already ruled out the items with these values. Selecting such value buttons would yield a null result.

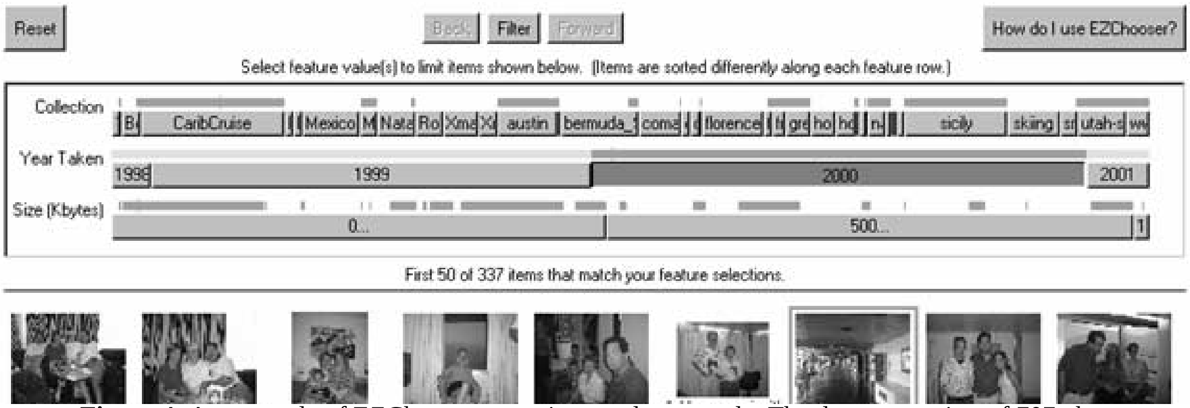

Figure 4: An example of EZChooser operating at a larger scale. The dataset consists of 707 photos of which 337 have been selected via the attribute restrictions. The first 50 of those are shown as thumbnail images. Items within vector rows are drawn at 1-pixel width.

Figure 3 shows an example. The user has selected three favorable Consumer Guide recommendation values (row 2), and also a loaded price range (row 4). Four vehicles, which are shown at the bottom, match these restrictions. The matching items' feature values are indicated in each row with filled glyphs. Outlined glyphs in both row 2 and 4 indicate that the user could include these other items if they select their associated value buttons.

One intention of these query visualization methods is to help guide users in subsequent query refinements, but an equally important goal is to visualize relationships among features and their values. Users can discern what features would be available should they choose certain value restrictions. The restrictions may be applied *in any order*. In the example in Figure 3, users can see the consequences were they to select subcompact cars with favorable Consumer Guide recommendations within a certain price range. For instance, only four car manufacturers have vehicles that match these restrictions (row 1), fuel economy would be in the mid-range (rows 5-6), and so on. Do they really want to narrow their choices with these restrictions? This tool lets them see the consequences within a larger context were they to do so, whereas classical querying interactions give no such indications. Users can easily and quickly try out these parameters and many others, allowing them to achieve the goals of flexible sequencing, discovery, and play mentioned in the introduction.

Larger Sets

Extensions have been made to a version of EZChooser available on our company's Intranet in order to accommodate larger sets. Figure 4 shows an example. With larger sets, users need to winnow down the size of the set under consideration as they proceed to explore the information relationships and hone in on their goal of choosing just one or a few items. We have thus incorporated a filtering action that has the effect of creating a new information space derived from the old. The visible effect is to change the display only in the parallel bargram area since filtering limits the set to what has already been selected (and rendered appropriately in the item listing area). As sets get larger or smaller, it is appropriate to form different aggregations of value bins for each bargram. Otherwise, labeling of bins is truncated and becomes unreadable. Also, value selections are made more difficult since they must consist of multiple button presses (or some sort of shift-click or dragging operation). We have experimented with a number of different approaches to clustering for that purpose. Our current approach for numeric attributes chooses ranges keyed to multiples of 10 (or 5) in an attempt to achieve 7 +/- 2 value bins. In the example shown Figure 4, the system starts out with three bins for image byte size—0 to 499, 500 to 999, and 1000 and above. As the user narrows down the set, bins would be shown in the 100s and eventually in actual bytes. Aggregation in the long run will entail many type-specific algorithms (dates, place names, user-specified hierarchies) that we have not yet implemented.

Our extended version of EZChooser renders the individuals in item vectors down to a minimum width of a single pixel, as seen in Figure 4. This in effect limits the sizes of the sets available to the tool to roughly 1000 items for large-resolution displays. (A horizontal resolution of 1280 pixels allows 1000 pixels to accommodate the bargrams themselves, with bargram labeling and framing requiring some additional horizontal real-estate.) As the set is filtered, the items are given progressively more space in which to render themselves, so an item that starts out as a single-pixel line at a large scales changes to a rectangle and then to a domain-specific glyph as more real estate becomes available.

As the set size of items under consideration increases, the ease of selecting individual items with brushing and marking techniques declines. At the scale evident in Figure 4, all but 50 of the items are off-screen in the item listing and the items within the dimension vectors are 1 pixel wide. Thus at this scale the tool operates by letting users select values in the aggregate and then filtering the set. As the candidate set size decreases, selecting and browsing of individuals in context becomes easier. However, once a candidate individual has been identified (through such progressive filtering or any other method), it may still be appropriate to reveal how this individual might relate to a very large context. For example, in a scenario involving college selection, a user might be interested in seeing how a candidate college choice looks relative to, say, all the colleges in a country or region. Our system allows marking the item in one of two ways—either by entering an EZChooser view from the context of a web page about that candidate college or else by starting with a large set and progressively filtering and then marking that item by hand. In either case, the marked item could then be viewed in the context of the original large candidate set. Marking at large scales should use, say, an arrow rather than just color, to point to the relative position (a rendering technique we have not yet implemented).

In sum, through the example of EZChooser, we can see that parallel bargrams and associated item vectors afford user interactions that scale across set sizes up to 1000 items with roughly 10-20 attributes. The interactions include selecting and marking individuals as well as selecting aggregate values to create Boolean queries. Item marking affords a quick visualization of the relative ranking of individuals within a larger multidimensional context. Querying is a simple matter of button pressing. The results of queries are reflected back into the same view so that users can understand the effects of those value restrictions in the larger context. Users can thus preview their queries prior to filtering their candidate choices and, perhaps more significantly, uncover relationships that would otherwise be hidden.

RELATED WORK

EZChooser falls within the spirit of dynamic querying [12]. It incorporates rapid adjustment of query parameters with instantaneous visual feedback, continuous reformulation of goals, and tight coupling of query parameters with result displays. As pointed out in [13], early dynamic querying systems [1] had the drawback that only results that satisfied the query were displayed. Thus it was not obvious how a user might have to modify a query in order to gain a different result. Attribute Explorer [17] incorporated feedback directly onto a certain type of attribute query widget—a feature that revealed new relational information in its own right as well as helped guide users to subsequent query modifications. Parallel bargrams are a further extension of this idea and can be considered an addition to the stable of techniques useful for dynamic querying.

The use of parallel dimensions in information visualization systems was pioneered by Inselberg [5]. The system called Parallel Coordinates revealed complex multidimensional patterns by drawing lines between all instances of items across parallel dimensions. Our work with parallel bargrams shares the property of making use of parallel item vectors, but the visualizations and interactions are different, as are the uses to which the systems are put. Parallel Coordinates was designed to reveal complex visual patterns for large datasets as a part of expert data detective work, whereas EZChooser was designed for supporting the choosing of individual objects by the general consumer.

We proceed with a discussion of other systems that use parallel bargrams, followed by dynamic histograms, and finally focus+context tables.

Other Bargram Systems

There are two other systems that we are aware of that use parallel bargrams. The first is MultiNav [7], which is the historical precursor to EZChooser, and the second is InfoZoom [8].

MultiNav research prototypes are described in [7]. The most advanced MultiNav prototype used a technique we called sliding rods as the primary interaction for multidimensional navigation. Users were able to drag any one of the dimension rows (we called them rods) from side to side, which had the effect of placing a succession of neighboring individuals in focus. Each of the other dimension rods would be positioned such that the single focused item was in the vertical center of the display. This type of multidimensional sliding interaction is another affordance of parallel item vectors coupled with bargrams.

Others in our lab ran a series of user tests with this MultiNav prototype with an eye towards deploying it on the Internet in consumer e-commerce settings. A number of usability issues were identified. One of them seemed to be the novelty of the sliding rods navigation interaction for the general consumer. Users seemed quite surprised at this form of interaction, and most did not discover it without coaching. While we still believe this technique has merit for showing correlations and an engaging style of feature-based browsing, we had to admit that it may be more appropriate in more specialized settings. We decided to revert to a more familiar form of interaction, namely, mouse button-pressing, in our revised system.

Another issue was the conflict between the browsing interactions (accomplished through sliding the rods) and querying (selecting value restrictions along the rods). When users selected a value restriction, they expected it to be reflected in the item display at the bottom of the screen as EZChooser now does. MultiNav painted the matched set of items back onto the item vectors on the bargrams, but it didn’t show a listing of query results. Instead, the bottom area of the screen was devoted to showing details of the single item in “browse focus”. Thus these two selection mechanisms were largely independent, causing confusion. We decided to meet most users’ expectations and devote the extra screen real estate to the querying aspects of the interface.

Independently from our work, a system formerly at the GMD with a focus+context table layout called FOCUS [14] evolved into the product named InfoZoom [8], which uses parallel bargrams for larger set sizes and focus+context tables for smaller ones. The primary interactions supported in InfoZoom are simple selection and filtering, accomplished with a single mouse click on a value bin. Interesting graphical renderings of values are included. Animation is used to transition between set sizes. The main difference between InfoZoom and EZChooser in the context of this paper is that InfoZoom does not include item vectors and the interactions they afford. Thus it is not possible to “preview” queries in order to reveal hidden relationships as EZChooser does. Nor can one view an individual in a larger multidimensional context with the marking technique we have mentioned. The rendering advantages of parallel bargrams over table layouts at larger scales seem to be the primary motivation for their use in InfoZoom.

Dynamic Histograms

Attribute Explorer [17] is the first prototype we know of that incorporated painting the results of queries back onto a set of parallel interface widgets that were responsible for forming the queries. Histograms were the basis of the interface widgets in question. Eick [4] proposed adapting a variety of widgets to incorporate such capability. Here we consider the merits of histograms vs. bargrams in the context of our choice task.

First, histograms are essentially unlimited with respect to scale. While this is true of simple bargrams as well (viz [8]), it is not true of the item vectors we have proposed to couple with bargrams. Parallel histograms are in fact proposed specifically by Doan et al. [3] for query previewing at large scales. If the focus is on aggregate information at large scales, histograms—or bargrams without item vectors—would be a good choice.

Another point follows from the coupling of item vectors with bargrams. Despite the fact that individuals can be assigned a position within histogram columns (and implementations of Attribute Explorer have done so), this visualization is not conducive to revealing fine-grained ranking information nor to certain types of browsing. When a restriction is made in the interface, all individuals within a given value bin that are so marked are aggregated and drawn contiguously with the same color. (Think of the analogy of a thermometer in which molecules change character and sink to become part of the mercury column.) The position of a given individual thus moves depending on the state of the restriction selection. We would argue that assigning a consistent position to an individual object on each dimension will be more likely to encourage interactions which depend on highlighting that individual. Also, mapping a two-dimensional widget to a one-dimensional one is better at revealing overall ranking information for individuals (although this can be a problem if the ordering is not strict, as mentioned earlier). Thus both the size of the set and the type of the attribute might be a factor in deciding which widget would be the better choice. Mixing and matching of visualization widgets is of course an option, and is in fact standard with dynamic querying systems.

Focus+context Tables

Another set of related work is based on extending conventional table layouts with focusing interaction techniques. One such system is TableLens [10], a general data analysis tool. Another is FOCUS, mentioned previously [14]. Both incorporate focus+context visualization techniques within table layouts so that areas of interest can be given relatively more visual real estate.

It is notable that FOCUS has evolved to the system InfoZoom that defaults to table layouts at smaller scales and bargrams at larger ones. Since same-valued cells are always contiguous in bargrams but not normally in tables—with the exception of a single column (the primary sort)—, labeling will be far more efficient since cells can be collapsed. Multidimensional distribution information can be revealed at a glance at large scales that would require a series of independent column sorts and a succession of views with tables. Flipping between table and *Uebersicht* (bargram) views in InfoZoom immediately reveals that bargrams condense information relative to tables since same-valued cells are always adjoined with bargrams (and the labeling can thus be combined), but not with tables.

This advantage of bargrams extends to query interactions. Since same-valued cells will be collapsed into "value bins" in every dimension in every view, it is easy to select values for the purpose of query restrictions. Using table layouts for such querying would require first a sort along a dimension, then a selection action, and finally the continuance of state across successive views to build a compound query.

On the other hand, table layouts have the virtues of familiarity, and they seem to work well at small scales. They allow the attribute values of a single individual to be scanned quickly along a row or column. The extensive set of visual exploration techniques incorporated into TableLens are built upon this solid base.

EVALUATION

In this section we will discuss an experiment that we conducted in order to test EZChooser vs. static tables for users engaged in choosing among sets of items through feature information. Our hypotheses were that the parallel bargrams and item vectors of EZChooser would be preferable for somewhat large sets (we tested sets of size 50), and that static tables would be preferable for small sets (we tested sets of size 3). We used Microsoft Excel in standard table display mode for the representative static table tool. The sort capabilities of Excel were ignored, and no user employed them.

We created data sets, large and small, from two content areas for the experiment--digital cameras and mutual funds. Data was extracted from commercial sources on the Web and thus accurately reflected the type of feature information users could expect to find published in these areas. The content in the parallel conditions was exactly the same. Images, for example, were included in Excel table cells whereas the corresponding EZChooser condition presented images in the listing view. Nine dimensions were used for digital cameras, including maximum resolution, self-timer, interface types, and so on. Seven dimensions were used for mutual funds, including investment objective, inception date, and rates of return over various time periods. We kept the number of dimensions relatively small so that scrolling was required in only one of two dimensions in the large Excel table condition.

Method Apparatus

16 subjects were recruited for the experiment. A few were summer students or nontechnical employees of our company, and thus not paid specifically for the experiment, but

the majority were paid $40.00 for approximately 60 minutes of their time. The paid subjects were recruited from outside the company, and many of them had previously been engaged for other user studies at our lab. The youngest subject was 12. The oldest was 67. All had some Internet experience.

Subjects were asked to perform tasks using 2 sets of tools (EZChooser and Excel Table), and 2 sets of data (digital cameras and mutual funds), each having 2 sizes of data (large set: 50 items, small set: 3 items) in different orders depending on a previously generated list. This list balanced the order in which subjects completed tasks controlled for tools, content, and size of data.

Procedure

From an opening web page, each subject was instructed on how to use the first of the tools. For example, one subject might be asked to click on the "data table demo" link. Upon seeing a sample of a table with non-experimental data (in this case, subcompact cars), the subject would be instructed about the features of tables, such as the fact that each row represents a different item, each column represents different features of the items, one has the ability to scroll for more items, etc. The subject would explore the sample on his own for a few minutes until comfortable with the tool.

The subject was then asked to view a page that displayed one of the experiment's datasets. The experimenter reviewed the relevant features orally and handed the subject a hardcopy sheet for reference. Trade-offs were mentioned where possible. For example, high resolution in a digital camera allows for higher quality pictures; however, high resolution requires more storage space and download time.

Once all features were explained, the subject was asked to choose one item from the set making use of the tool. The subject was instructed not to let information from external sources (comments from friends, other printed information sources, etc.) influence the final decision. How to weight the criteria was entirely up to the subject. If s/he had seen the same dataset before in a different condition (true for the latter two tasks users performed), instructions were to do his or her best to go through the decision process again using the alternative tool.

The experimenter timed each task, and noted the choice made. After each task was completed, the subject answered three written questions about the task just completed:

> 1. Please rate how satisfied you are with this tool in helping you with your choice task given the information provided. [1-7 scale]
>
> 2. Please list the criteria used to narrow down your choice: [fill in]
>
> 3. How confident are you that you made the best choice given the information presented?[1-7 scale]

After all tasks were completed, users answered the following question for large and small data sets:

> 4. How would you compare the two tools for helping you make choices with the LARGER/SMALLER data sets? [Choose one: Table vastly superior, Table a lot better, Table somewhat better, about the same, EZChooser somewhat better, EZChooser a lot better, EZChooser vastly superior]

They were then asked four variants of the following question about likelihood of use:

> 5. How likely would you be to use the EZChooser/Table tool in the future if it were available for making choices among sets of the LARGER/SMALLER size assuming tables/EZChooser were an alternative? [Choose one: Definitely never, Not likely, Somewhat unlikely, Maybe, Somewhat Likely, Probably, Definitely Yes]

Finally, users were asked to write down words they would use to describe EZChooser and the table tool, respectively.

Results

We observed no significant differences in time or questions 1-3 above. However, we did uncover differences with respect to the general evaluation questions (questions 4 and 5 and their variants). The findings are summarized as follows:

- EZChooser dominated Tables in terms of judged preference for larger data sets (question 4: larger). In the comparison of EZChooser and Table on the seven-point "final evaluation" scale for large data sets, the mean rating for EZChooser was significantly larger than the scale midpoint (and also necessarily larger than the rating for Tables); $t(15) = 3.31$, $p < 0.01$.
- EZChooser dominated Tables in terms of judged usage for larger data sets (question 5: larger). For each subject a difference score was computed for their ratings of likelihood of use of EZChooser and Tables for large data sets. The mean difference score was reliably greater than zero, $t(15) = 2.92$, $p < .02$.
- EZChooser and Tables were not judged to be significantly different on the preference scale for small data sets (question 4: smaller). In the comparison of EZChooser and Tables on the seven-point "final evaluation" scale for small data sets, the mean rating for EZChooser was not significantly larger than the scale midpoint (or Tables); $t(15) = 0.63$, N.S.
- EZChooser and Tables were not judged to be significantly different in likelihood of use for small data sets (question 5: smaller). For each subject a difference score was computed for their ratings of likelihood of use of EZ Chooser and Tables for small data sets. The mean difference score for rated usage for small data sets was not reliably different from zero, $t(15) = -0.27$, N.S.
- The size of the data set that subjects used when judging the two tools was important. EZChooser dominated Tables when the data set was large but not when the data set was small. An ANOVA was run separately for the comparison preference scores and for the difference scores of usage ratings. Independent variables were task (large, small) and subject (to control for subject means). Task was significant for usage $F(1, 14) = 6.85$, $p < 0.02$,

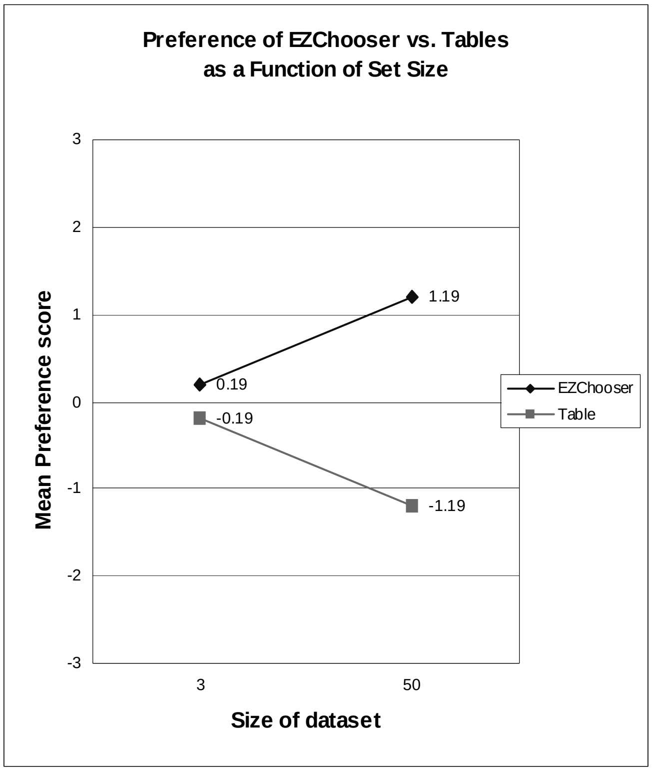

Figure 5: Mean scores for subjects' preferences for EZChooser vs. Tables (question 4).

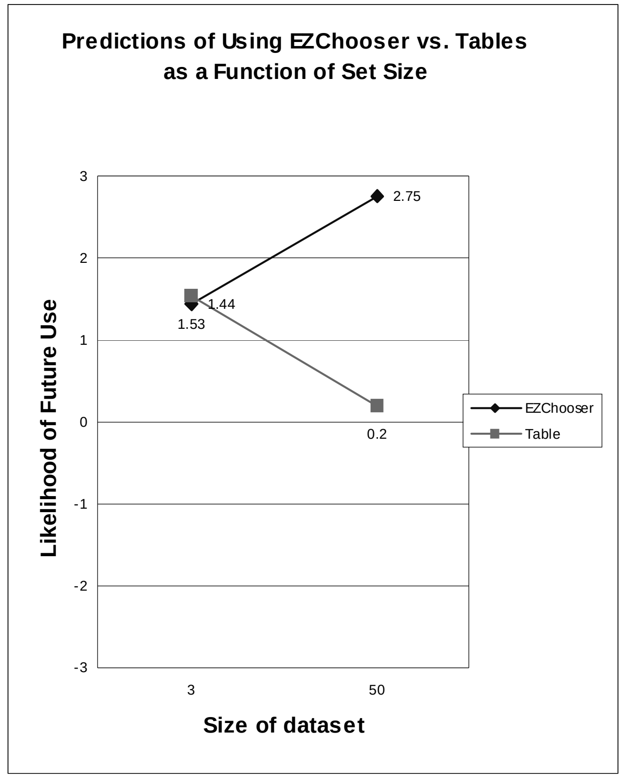

Figure 6: Mean scores for subjects' likelihood of use for EZChooser vs. Tables (question 5).

but task was not quite significant for preference $F(1, 15) = 2.79$, $p > 0.11$.

Figures 5 and 6 illustrate the difference in mean scores with respect to set size for the preference and likelihood of use questions, respectively.

Discussion

We were able to confirm our hypothesis that users would prefer EZChooser over static tables for larger sets, and also that they would be more likely to use EZChooser. In raw terms, 12 out of 16 subjects gave EZChooser a positive comparative preference score, and 6 users gave EZChooser the maximum rating of "vastly superior." However, we found no difference in preference when choosing within small sets. Tables were not found to be preferable, disconfirming our hypothesis that tables would be preferable for sets of size 3. Similarly, users said they would be no more likely to use tables over EZChooser for sets of the small size. EZChooser seems to have more appeal to users than we expected in the case of small set sizes.

The general preference results evident in these numbers were confirmed in the comment section of the questionnaire. Although these results are preliminary and in fact modest, it should be noted that achieving positive results for first-time users of novel visualization systems is rare.

What we have shown here are only two points along what we suppose is a continuum related to the size of the item set. We have presented the results in Figures 5 and 6 as line graphs although we are able to plot only two points on these lines. We hypothesize that these lines are curves continuing the upward or downward slopes that will eventually level off, but have no conclusive evidence that this is so.

Many other questions remain unanswered. Would users prefer tables if there were only two items? At what minimum number of items would EZChooser show statistically significant preference? What is the effect of the number of dimensions? Would the result have been different if users could sort on the columns of the tables? What are the effects of alternative orderings within the dimensions? Those questions will have to await further research.

There are easy criticisms to make of this experiment. We asked users to forget the results of a choice task and redo it with a different tool. We did not observe any subject failing to go through the process again, but of course we can't be sure of the absence of any effect. (However, we did control the order to tools presented so that this effect should have balanced out.) Our results are statistically significant only for preference information. Unfortunately, we were not able to show any actual performance differences. There is undoubtedly some effect on preference measures of subjects trying to please the evaluators. We attempted to remain neutral across the two tools, but that is naturally difficult. Finally, the most significant limitation of this experiment is that subjects were trained on the use of EZChooser before performing the task. Our goal was to produce a tool that does not require training. However, we felt it important first and foremost to establish the result that users would prefer the tool if they knew how to use it, and that result was achieved. Subsequent naturalistic user testing has in fact shown that a significant number of users are likely to have trouble understanding how to use the tool at first. We hope that it will be possible to improve on this through better help facilities and further tweaking of the design.

CONCLUSION

This paper has described some new multidimensional visualization techniques and their use in a system designed to help the general consumer make choices based on attribute information. The techniques are based on parallel bargrams associated with item vectors, an extension to related work in dynamic querying and attribute exploration using histograms. Through comparison to related work, we explained why these techniques work and for what set sizes they are appropriate. We evaluated the system through frequent informal user testing and also through an experiment that showed that users preferred EZChooser to static Excel tables for sets of 50 items with 7-9 attributes. Further systems work is needed to expand the set of clustering techniques to more attribute types. More empirical work is also called for to determine whether users can learn systems like EZChooser without training and also to explore the visual querying features specifically.

One of the most significant lessons we have learned over the course of this project is how challenging it is to alter the standard interaction paradigms of the general consumer on the Internet. As a result, the interactions we propose here are in fact less novel than our earlier prototype. The broader UI technology community might want to take note of this trade-off in the interest of having new ideas adopted, which is after all the goal for innovative user-interface systems and technologies.

ACKNOWLEDGMENTS

This project relied on contributions from many of our colleagues, to whom we are grateful. Christina Fyock, Glenn Li, and Dan LaLiberte were fellow project members who contributed in many ways. Chris Smith and Jayasudha Ayyagari worked with us on the SuperPages deployment. Greg Cermak did the statistical analysis in connection with the experiment. Demetrios Karis, Sumi Gupta, and Lori Eggert did user testing. Bob Spence and Mark Apperley greatly influenced the presentation through discussions related to an ongoing collaboration. And finally, John Vittal created the environment at GTE/Verizon Labs to make this research possible.

REFERENCES

1. Ahlberg, C. and Shneiderman, B. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In *Proceedings of CHI 1994* (Boston MA, USA, April 1994), ACM Press, 313-317.
2. Buja, A., Cook, D., and Swayne, D. F. Interactive High-Dimensional Data Visualization. *Journal of Computational and Graphical Statistics*, 5, 1 (1996), 78-99.
3. Doan, K., Plaisant, C., Shneiderman, B. Query Previews in Networked Information Systems, in *Proceedings of Advances in Digital Libraries, ADL '96* (Washington DC, USA, May 1996) IEEE CS Press, 120-129.
4. Eick, S. G. Data Visualization Sliders. In *Proceedings of UIST '94* (Marina del Rey CA, November 1994), ACM Press, 119-120.
5. Inselberg, A., and Dimsdale, B. Parallel Coordinates: A Tool For Visualizing Multi-dimensional Geometry, in *Visualization '90* (San Francisco CA, USA, 1990), IEEE Press, 361-370.
6. Kirsh, D. Interactivity and Multimedia Interfaces. *Instructional Science* 25,2 (1997) 79-96.
7. Lanning, T., Wittenburg, K., Heinrichs, M., Fyock, C., and Li, G. Multidimensional Information Visualization through Sliding Rods, in *Proceedings of Advanced Visual Interfaces--AVI 2000* (Palermo ITALY, May 2000), ACM Press, 173-180.
8. Lindner, H-G. Knowledge Reporting with InfoZoom, SAP Design Guild, Innovation Second Edition, 12/22/2000, http://www.sapdesignguild.org.
9. Pu, P. and Faltings, B. Enriching buyers' experiences: the SmartClient approach, in *Proceedings of CHI 2000* (The Hague NETHERLANDS, April 2000), ACM Press, 289-296.
10. Rao, R. and Card, S.K. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. In *Proceedings of CHI 1994* (Boston MA, USA, April 1994), ACM Press, 318-322.
11. Schafer, J. B., Konstan, J., and J. Riedl. Recommender Systems in E-Commerce, in *Proceedings of the ACM Conference on Electronic Commerce* (Denver CO, USA, 1999), ACM Press, 158-166.
12. Shneiderman, B. Dynamic Queries for Visual Information Seeking. *IEEE Software*, 11,6(1994), 70-77.
13. Spence, R. *Information Visualization.* ACM Press, 2000.
14. Spenke, M., Beilken, C., Berlage, T. FOCUS: The interactive table for product comparison and selection. In *Proceedings of UIST '96* (Seattle WA, USA, November 1996), ACM Press, 41-50.
15. Suchman, L. *Plans and Situated Actions: The Problem of human-machine communication*, Cambridge University Press, 1987.
16. Tweedie L.A., Spence, R., Dawkes, H., and Su, H. Externalizing Abstract Mathematical Models. In *Proceedings of CHI 1996* (Vancouver CANADA, April 1996), ACM Press, 406-412.
17. Tweedie, L.A., Spence, R., Williams, D., and Bhogal, R. The Attribute Explorer. In *Video Proceedings and Conference Companion of CHI 1994* (Boston MA, USA, April 1994), ACM Press, 435-436.