Carnegie Mellon University
Research Showcase @ CMU

Human-Computer Interaction Institute | School of Computer Science

2005

The Persectives Browser: Exploratory Data Analysis for Everyone

Mark Derthick
Carnegie Mellon University

John Zimmerman
Carnegie Mellon University

Follow this and additional works at: http://repository.cmu.edu/hcii

This Conference Proceeding is brought to you for free and open access by the School of Computer Science at Research Showcase @ CMU. It has been accepted for inclusion in Human-Computer Interaction Institute by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

Submitted to the 2005 IEEE Symposium on Information Visualization

The Perspectives Browser: Exploratory Data Analysis for Everyone

Mark Derthick and John Zimmerman
Carnegie-Mellon University • Human-Computer Interaction Institute
{mad, johnz}@cs.cmu.edu

ABSTRACT
Web search engines have gained tremendous audiences for information retrieval from unstructured documents. The number of structured and semi-structured documents available on the web is also huge, and collections of these are more amenable to data mining. Yet there has been no similar explosion of interest in this kind of exploration. Finding patterns in databases of political contributions, pollution and environmental data, or hospital and school performance would surely interest many citizens. The Perspectives Browser is intended to support this kind of exploration for users with little or no training in statistics or programming. Given an "advanced search" type query, it visualizes dependencies on the query of up to 30 variables. In preliminary studies, participants found interesting three-variable dependencies in an art collection. We concentrate on image databases because the content can be concisely summarized, but the dependency visualization applies to any hierarchically organized nominal or ordinal variables.

Categories and subject descriptors
H.5.2 [Information Interfaces and Presentation]: User Interfaces---Graphical user interfaces, Interaction styles, Screen design; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval---Information filtering, Query formulation, Selection process; H.3.7 [Information Storage and Retrieval]: Digital Libraries---User issues; H.2.8 [Database Management]: Database Applications---Data mining, Image databases;

General Terms
Design, Human Factors.

Keywords
Exploratory Data Analysis, Information Visualization, Browse, Focus + Context Techniques, Dynamic Query, Interface Design

1 INTRODUCTION

Currently there are two ends of a spectrum in visual approaches to knowledge discovery within large datasets: Visual Data Mining (VDM) and Exploratory Data Analysis (EDA). VDM seeks to combine peoples' semantic knowledge and innate visual pattern recognition ability with the computers' ability to systematically search and explicitly compute expected and observed frequencies [9]. VDM's main challenge has been in both finding efficient algorithms for searching large databases and visualization techniques for displaying large databases. EDA, on the other hand, focuses on data visualizations where users can interactively manipulate display characteristics in order to discover underlying structures in the data. While both of these methods provide insight into the underlying data, they require users to have both an understanding of statistics and an ability to computationally manipulate the data and the visualization display characteristics.

Many people have a need to engage in knowledge discovery tasks, such as parents who want to explore school performance records in order to select a good school; patients who wish to learn more about hospitals before selecting a healthcare provider or choosing a facility for a medical procedure; and voters who wish to better understand the relationships between political contributions and voting records when deciding who to vote for. However, because of the high skill bar required for both VDM and EDA, most people can't attempt this task. This was once the case for information retrieval tasks but research and commercial development have helped produce tools such as Google that open up information retrieval tasks to almost anyone who can type.

In order to address this issue we constructed an interface called the Perspectives Browser (PB). This interface provides non-technical users with an interactive visualization technique that makes differences between expected and observed frequencies clear for one family of statistical models. While highly interactive for small datasets, PB lacks the flexibility of most EDA systems. Instead, it represents a third approach to visual knowledge focused more specifically on learnability, responsiveness, robust-ness, and providing a satisfying user experience than on providing the most investigative power of EDA or VDM.

PB employs a novel kind of histogram that interactively displays deviations from a null-hypothesis distribution, allowing non-technical users to explore and discover underlying structures in the data that do not meet their expectations. When constructing a query, PB allows users to see (i) how many items match the current query, (ii) the distribution of items for each attribute, and (iii) whether the distribution is in accordance with the assumption that attributes are independent. To find dependencies, it is important to be able to compare the distribution over the entire dataset (the unconditional distribution) to that for a subset of interest (the conditional distribution). By observing the associations among attributes, users can find interesting questions to explore.

In order to better understand the knowledge discovery needs of non-technical users and to gain insights into how PB might meet these needs, we conducted an informal study using a database containing 37,000 works from the collection of the Fine Arts Museum of San Francisco. Participants appeared to use the interface for ordinary searching and browsing of datasets with hierarchically structured meta-data better than other commercial or research systems. In addition, participants were also able to make interesting discoveries while exploring the data. However, the sessions did suggest that our intended audience does not have an innate understanding of how or why they might want to find patterns within datasets. We are hopeful that over the next decade the idea of pattern finding will become more widely understood, just as information retrieval has been widely adopted, and we feel that tools that provide non-technical audiences the ability to find patterns will hasten this transition.

2 RELATED WORK

Information Retrieval Approaches

Information Retrieval (IR) approaches to knowledge discovery through data visualization use space two ways: by explicitly mapping attributes and by clustering. Clustering methods use vector space models with hundreds or thousands of dimensions to represent document content. Visualization of these models involves reducing the dimensionality to 2D or 3D, resulting in a visual space where absolute position is meaningless, but where similar documents tend cluster near one another [17]. Since these methods don't show meta-data attributes explicitly, they do not help users find associations.

IR visualizations that employ explicit spatial mapping usually focus on displaying only documents relevant to a specific query, as opposed to comparing the result set with the entire dataset. Therefore they don't support the statistical approach to pattern finding based on comparing observed and expected distributions. Examples of explicit mappings include showing relevance as a graphical property such as position or size. Even traditional ranked lists such as found in web search engines fall in this category. Metadata attributes like date and location may also be mapped to graphical properties, as in timelines or maps [3].

HierAxes

HierAxes have been used to plot documents in a space defined by hierarchical *x*- and *y*-axes [16], such as the ACM Classification system. Although only two dimensions are shown at a time, the dimensions on each axis can be changed interactively. 2D has the potential to show correlations between those dimensions better than the PBs use of multiple 1D visualizations. However, showing correlations for nominal attributes effectively is difficult. Indeed the examples in Shneiderman et al.'s work only mention patterns that can be seen easily in multiple 1D visualizations [16]. Further, subjects tended to become lost when they had filtered on dimensions that were no longer visible.

Multi-Dimensional Approaches

Of the known multi-dimensional information visualization techniques, parallel coordinates [8] is simpler than most and probably the most popular. This is the basis for PB's use of parallel histograms. More advanced techniques that use space hierarchically, such as Mosaic Displays [6], Worlds Within Worlds [5], Mihalisin et. al.'s visualizations of multivariate functions and distributions [14], and Trellis Displays [1], retain the advantages of orthogonal axes at a cost of being harder to learn.

Flamenco

Flamenco organizes image collections by multiple hierarchical attributes and supports incremental query construction that combines restrictions on attributes with text search [18]. The scenarios described in this paper are based on metadata created by the Flamenco project, and our interface supports the same style of incremental queries. In fact, any Flamenco usage scenario could be reproduced virtually gesture for gesture in our interface. However, in Flamenco feedback on conditional distributions is only shown textually. By default, attribute values are sorted alphabetically rather than by cardinality, and counts aren't even aligned, so understanding distributions is a cognitive task rather than a perceptual one. Further, the counts only show the number of works in a category that matches the query instead on comparing this number with the total number for a given attribute, so finding associations among attributes must be done externally by the user. For example, if the count for *Location=Europe* is higher than that for *Location=North America* given a query *Theme=Military*, it might be because a higher percentage of European works have a military theme, or it might simply be that there are more European works in the collection.

Flamenco filters are limited to one value per attribute. This is especially limiting when naturally quantitative attributes like *Date* are demoted to hierarchical ones by discretizing into centuries and decades. For instance, Flamenco can't filter for US works created between the Civil War and World War II, because that range does not coincide with either the 19th century or the 20th century, or to any decade within a century value. PB instead follows the Windows convention of using shift and control keys to allow users to select multiple attributes.

Relation Browser

Like Flamenco, the Relation Browser [13] is meant for IR rather than EDA; however, it contextualizes counts better that Flamenco. A successor to Query Previews [4], it shows both conditional and unconditional histograms for each metadata attribute, so it is possible to distinguish whether, for instance, a preponderance of works from Europe matching a query for military-themed works is due to an association between Europe and military, or just an abundance of European works overall. However perceiving the distinction requires visual comparison of the ratio between conditional and unconditional bar lengths for Europe compared to other locations. This is more difficult than the Perspective Browser's comparison between absolute bar lengths. Further, because the Relation Browser uses a single linear scale for bar heights, if only a small percentage of works match the query when comparing counts for two attribute values, the conditional bars may both be less than a pixel high, making precise comparisons impossible.

PaperLens

PaperLens is one of the few systems designed to allow broad audiences to find patterns in data [11]. Like the Perspectives Browser, it is highly visual and interactive. However, it is tailored for analyzing academic publications, while PB is domain independent.

3 Interface Design

Figure 1 shows the PB interface. We divided the interface into four columns labeled Query, Summary, Results List, and Selected Result; providing users with a natural left to right movement across the screen that roughly corresponds to Shneiderman's Information Seeking Mantra: overview first, zoom and filter, followed by details on demand [15]. To begin a query, users select one of the attributes labeled in the Query column. This expands the corresponding histogram in the Summary column. Once expanded, users can select one of the attribute values in the histogram. For instance, one experimental subject selected the attribute "Persons" and the value "Female Persons" in order to explore themes associated with works depicting women. Selecting a value both adds that value as a filter in the active query and, if the value has subcategories, expands a new histogram that displays them. For female persons, this would support further restricting the query to works depicting wives or mothers, for instance.

As users build queries by selecting different attribute values from different histograms, a natural language summary of the active query appears at the top of the Summary column. This helps users remember which attribute values they have selected. This is especially helpful for attribute values from histograms that are not expanded and therefore display no labels. At the same time, the Results List and Selected Result columns update to show images of the art works that match the active query. Users can select a specific item in the Results List column by either clicking with their mouse or navigating with the arrow keys.

Dual encoding of the histogram bars enables multivariate pattern discovery. Width shows the unconditional distribution over the whole collection, while height shows the conditional distribution given the current query. In Figure 1, the subject has added a second filter *Theme=military* and has chosen to view the attribute *Location*. While the relative bar widths show that there are more European works than North American ones in the collection, the heights show that the relative frequency of military-themed works depicting women is much higher for Europe, even after controlling for the difference in overall frequency (see accompanying video). We now describe the visual representation of the histograms in more detail.

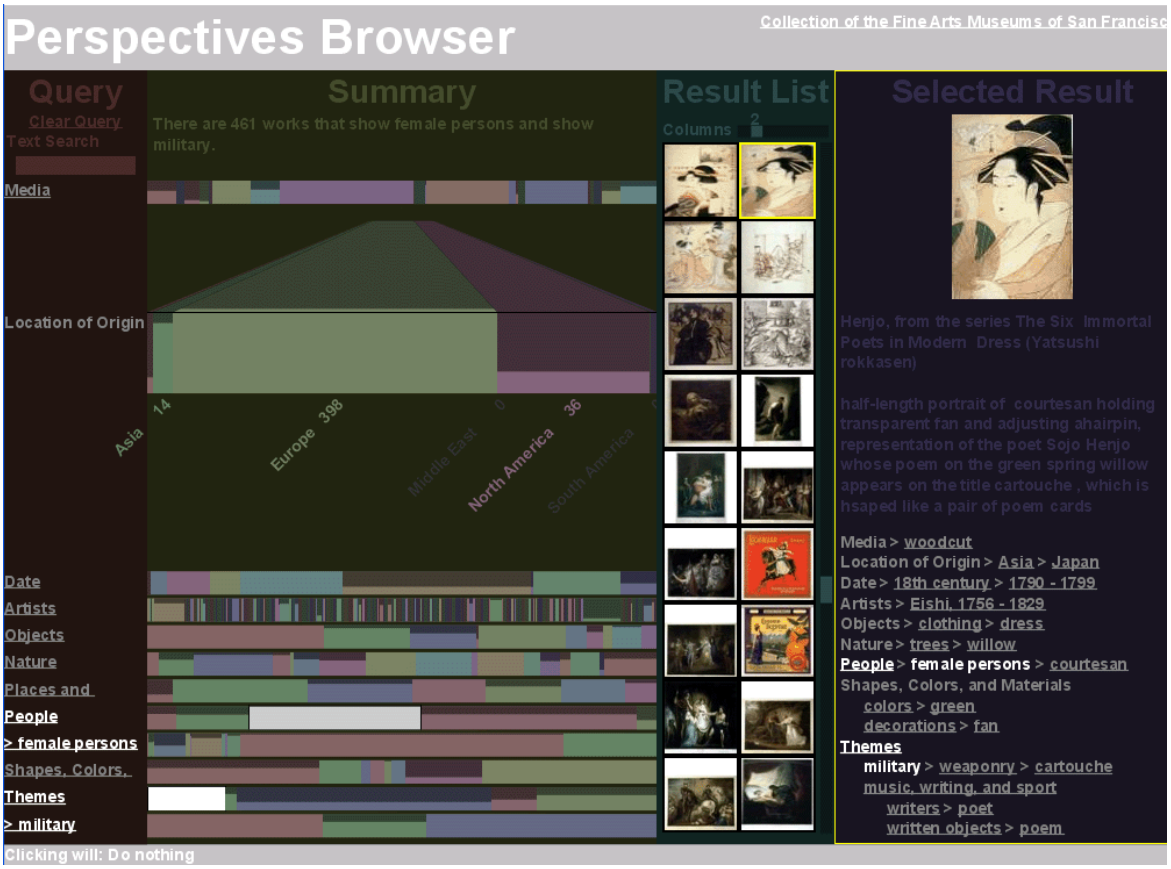

Figure 1 Perspectives Browser with a query for military-themed works of art depicting female persons.

Residual Histogram Visualization

A central task for EDA is finding underlying structure in data. In a statistics framework, this involves iteratively making models, investigating deviations from the models, refining the models, and repeating. For categorical data, one might start with the zero parameter model that all values are equally likely. If that model doesn't explain the data, one might take the empirical distribution of that variable as a model, but assume that other variables are independent of it. At each stage, the simplest model consistent with previous decisions is termed the null hypothesis. Only if that model is a poor match to the observed data are more complicated models considered.

Our audience does not naturally think in terms of models and residuals. Therefore we wanted PB to automatically factor the observed distribution into expected and residual components, and make the residual components salient. As users apply their common sense while iteratively refining the query, they can watch for patterns in the bar heights. Under this factorization, such patterns will automatically correspond to dependencies that aren't explained by simpler models.

Some current visualization designs show the difference between observed data and what is predicted under independence. For instance, Figure 2 shows a Mosaic Plot of the association between date and location in the art collection. We drew the figure to resemble Friendly's work on Mosaic Plots [6]. Mosaic Plots map cell width to the unconditional distribution of the *x*-variable, and heights in each column to the distribution of the *y*-variable conditioned on *x*. Thus areas represent the joint distribution. If the variables were independent, the cells would line up in rows, just like they line up in columns. Color shows the deviation from expectation in terms of standardized Pearson residuals: (*observed – expected*) / *expected*. Mosaic plots can show associations among any number of variables, though they become harder to discover as the number increases.

Even two-variable mosaic plots would most likely be too complicated for our audience. The number of rectangles (170 in the figure) is overwhelming, the correspondence between rectangles and labels is difficult to trace when the rows don't line up, and the many rectangles that are only one pixel in height or width can't be effectively compared. In order to simplify the visualization we chose to look at the expected vs. observed frequencies for one variable at a time, conditioned on a derived binary variable corresponding to a conjunctive query. We felt anyone who uses Google would be both familiar with and easily able to construct conjunctive queries.

Figure 3 shows our simplified mosaic plot where location is shown as before, but date is now binary, and specifies whether each art work matches the query *Date=20thCentury*. Since each 20th Century cell shares the same baseline, it is possible to draw a reference line (shown in black) where the cell boundaries would fall if location were independent of the query. The farther away the actual boundaries are, the greater the deviation from independence.

Figure 4 shows our redesign of the simplified mosaic plot, except that it leaves out space optimizations discussed below. Instead of using color we used brightness to distinguish the matched and unmatched set within an attribute. We chose instead to use color, in a repeating pattern, to distinguish different attribute val-

ues. This helped simplify the simplified mosaic plot even more, and made labeling of the *y*-variable (*20th Century* vs. *Not20th Century*) unnecessary. Participants in our evaluation had no trouble understanding that the height of the bright green bar labeled "Europe" in Figure 4 means that only about ¼ of European works match the query. Participants did not readily understand the concept of an expected value, but this is probably not crucial as long as they understand the meaning of differences in bar heights. Our modification of mosaic plots trades space complexity for time complexity. Users can read Figure 4 much more easily than Figure 2, but it shows less information at one time. When using our simplified plot, users can filter on each *Date* attribute one at a time and watch for changes to the Location histogram to reproduce the information show in Figure 2.

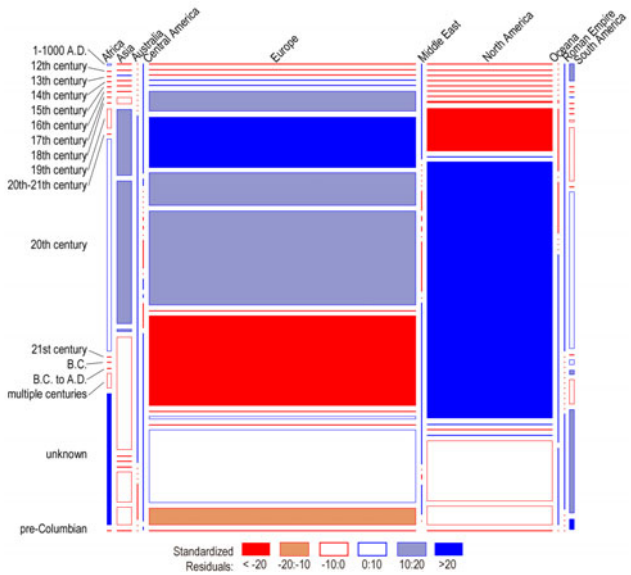

Figure 2 Mosaic plot of Date vs. Location of Origin. As in PB, attribute values are sorted alphabetically, which works poorly for ordered or quantitative attributes like Date.

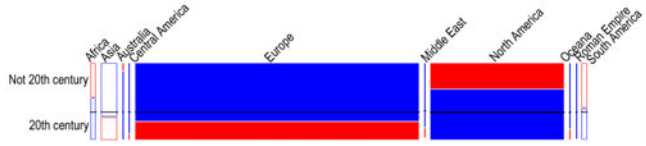

Figure 3 Mosaic plot aggregating most Date values into one, and rescaling *y* to save space. Black lines show expected boundary positions.

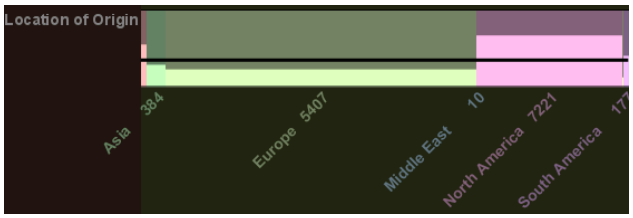

Figure 4 The same shapes as Figure 3, drawn in the style of the Perspectives Browser.

Space-saving Refinements

On Flamenco's opening screen, the table of attributes, values, and counts takes up more than 1024x768 pixels. The Perspectives Browser reduces the space required by attribute value labels and counts to about 600x150, again by trading space for time. Users must select an attribute in order to expand the histogram and see the labels, and only one histogram can be expanded at a time. This is similar to the fisheye technique for labeling papers from only a single selected year in PaperLens [11]. Hiding labels saves room for browser controls, attribute histograms, result thumbnails, and details about one result. These easily fit on a 1024x768 screen. To accommodate both distributions and result images, Flamenco requires users to scroll. It also requires users to view item details on a separate screen.

The attribute *Artists* has over 4000 values. Obviously they cannot all be labeled in PB given the layout of the screen. Instead a greedy algorithm first draws the label for the value under the mouse, if any. Remaining labels are drawn in order of their match to the query, starting with the largest. Labels that would occlude previously drawn ones are skipped. We use a similar algorithm to draw attribute bars, but instead of using the query match, we look at the total count for an attribute in the dataset. This causes Figure 4, the PB histogram, to have fewer bars and labels than Figure 3, the simplified mosaic plot.

In EDA, it is common to rapidly shift focus from overviews of an entire collection to filtered subsets orders of magnitude smaller. This creates a focus + context problem: how do you see the whole collection while allocating enough pixels to the focus so that patterns may be discerned? This problem shows up in mosaic plots as single-pixel rectangles. Fortunately, 1D visualizations are amenable to combining focus and context by distorting in the orthogonal dimension. For PB we vary the *y* scale with the degree of filtering; one pixel of height represents orders of magnitude fewer objects when zoomed compared to overviews of the whole. The *x* scale remains constant, so the full unconditional distribution is always visible; only the dependency effects are scaled. Under the null hypothesis, the expected bar heights are all equal, so a single scale can be chosen that focuses on the range of values near the expectation.

Since EDA is primarily qualitative, the main question is which values deviate significantly from expectation. Thus distortion for values far from the expectation is acceptable. Further, we minimize the perceived distortion by using a Perspective Wall [12]. Users' built-in 3D perspective correction helps soften the distortion of the *y* scale. We choose parameters so that the fold serves as a visual landmark, indicating where the null hypothesis distribution falls. Thus any bars that exceed the fold represent positive deviations from expectation, and bars that do not reach the fold represent negative deviations. For a more sophisticated audience, we could bevel or curve the fold to represent confidence intervals.

As a final space optimization, histograms for unselected attributes are scaled down and are cropped at the fold (see Figure 1). This still supports scanning for attributes strongly associated with the query, because bars shorter than expected are still noticeable. The bar areas must sum to the area below the fold, so the visible shortfalls equal the invisible excesses. Thus, the larger the dark area below the fold, the greater the association.

Exploration by Rollover

Hovering the mouse over an attribute value previews the distribution that would be seen by clicking, but without additional distortion (see Figure 5). This is the same technique as was used by the Relation Browser [13]. In addition, when hovering, the label and count (match/total) for the hovered attribute appears in the lower right corner of the screen. This allows users to easily see labels for attributes that are not expanded. By exploiting this hover feature, users can scan for values that are highly associated with the ex-

panded attribute, and then look at the value when an association is observed. Figure 5 shows the query *Persons=Female Persons* plus hovering over *Theme=religion*, rather than clicking on religion as in Figure 1. In both cases the brightest Europe bar represents 398 works while the North America bar represents 36. However the scaling in Figure 5 does not take into account the fraction of works with a religious theme. The drawback of this approach is that without distortion, hovering over a rare value may highlight distributions that are less than one pixel high, and therefore unobservable.

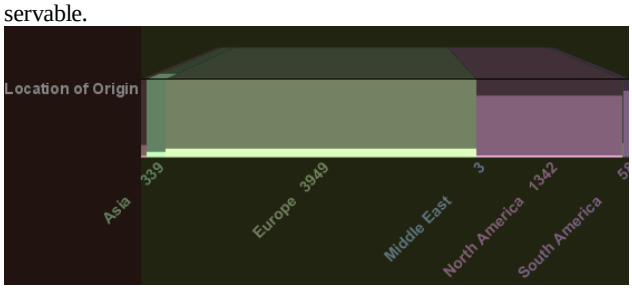

Figure 5 Using highlighting to see the women-religion-North America pattern, rather than selection.

Design Contributions

Compared to previous systems, PB displays more kinds of helpful information in less space. The design evolved over three iterations. The first mapped the bars for each attribute onto a literal cityscape, with one building for each attribute [citation deleted to preserve anonymity]. We wanted to explore dozens of dimensions, and didn't think they could all fit using a 2D layout. This design showed sample images on each “floor” of the buildings, and didn't attempt to show deviations from an expected distribution. However 3D navigation was difficult and the cityscape metaphor didn’t become as productive as we had hoped. We never find a use for streets, architectural style, or topography, for instance.

The second iteration had the look and feel of an ordinary web page, and was quite similar to Flamenco. There was an attempt to show distributions visually, and to show deviations from expectation. However these were overwhelmed by the sample images, and trying to explain the tick marks representing the null hypothesis distribution was unsuccessful. Subjects felt overwhelmed by the visual complexity, which required 1600x1200 pixels. They didn't find the integration of the thumbnails and attribute values to be particularly helpful.

In both designs, item details could only be seen by following a hyperlink. Seeing a full result list also required a separate web page in version two, and was not available at all in version one.

The current iteration was significantly influenced by ideas generated by students in a graduate interaction design studio. Each student was assigned to develop their own interactive visualization of the art collection. None of these projects showed all the attributes at once. Since the need to allow for dozens of text labels for each attribute monopolized much of the screen, this decision added much flexibility. However, these designs took the specific domain into account, for example showing location differently from date, making their designs less generalizable.

We wanted PB to be much more generalizable. As a quick test we also populated it with a database of academic papers from CiteSeer [10]. Since these papers are in pdf format, Adobe Acrobat can automatically extract figures to use as visual surrogates. We also used broadcast news stories collected by CMU's Informedia project [3], where keyframes serve as surrogates. There are qualitative differences in the attributes of these collections. Differences include the depth of the attribute hierarchy, attribute cardinality, whether attributes are nominal, ordinal, or quantitative, classification accuracy, and the number of values a single object may have for an attribute. Visualizing these additional databases helped keep the final design of the PB less tied to any particular type of information.

Other notable ideas from the students’ design projects included representing each object explicitly, and using animation to give more space to a focus attribute. In the InfoVis community, pixel-oriented visualizations have exploited the fact that computer monitors have millions of pixels, so even fairly large datasets can be shown without aggregation [9]. Although the idea is not used in the version shown in Figure 1, representing the result set as a grid of pixels effectively conveys the approximate size of the result set as filters are updated. While the proportional thumbs in the result list scrollbar contains the same information, it takes more conscious effort to understand. When given a task to find works meeting a certain description, one participant persisted for 15 minutes in a sequential scan of all the artworks until interrupted by the experimenter. She hadn't realized that she still had only looked at a tiny fraction of the images.

Like Flamenco, but to an even larger degree, supporting drill-through as well as drill-down aids exploration of the dataset. In Flamenco, clicking on an attribute value on an item detail page returns to the overview page and starts a new search with only that filter. In PB, there is only one page so the iteration can be much faster. In addition, clicking on an attribute value *adds* to the current query (or subtracts if you click on a value less specific than a previous filter term). Thus it is easy to explore within an area of interest without having to re-specify the area. The interface only changes the selected item when changes to the query remove it from the result set (or when the user clicks on another image in the result list). Thus, emulating the Flamenco behavior requires only one extra click. First use the Clear Query button (which will never change the selected item), and then click on the desired value.

The layout of the Selected Result column was designed for compactness. Descriptive text is guaranteed a fixed minimum height (20% of the window height). After that, space is allocated in priority order: images get first dibs, followed by metadata, with descriptive text getting any unused space. The metadata tree is laid out recursively with an $O(n^2)$ algorithm that does full look ahead for each sub-tree. Each sub-tree draws its root attribute value at the (x, y) location it is given. It then asks whether the rest of the subtree will fit to its right. If so, it lays out each of its children at $x' = x+$<*root label width*>, and at successive y' locations starting with y. Otherwise it lays out its children at $x' = x +$ <*indent width*>, and y' starting at $y +$<*line height*>. In case it is being called just to find out whether the parent should go to a new line, it tells its parent which of these choices it made.

Scrollbars are added to the descriptive text and metadata tree if needed, though most items in the databases we have tried so far don’t. Although PB supports scrolling through result sets, we believe that an expert user would rarely use this feature, beyond observing the thumb size to gauge the result size. Using the slider above the results to change the number of result thumbnail columns, the number of preview images visible at one time can be varied from 3 to 1000 (at 1024 × 768 screen resolution and normal amounts of browser decoration). Since the result order is unconstrained, scrolling is unlikely to reveal qualitatively different previews. (A possible improvement would be to explicitly randomize the order.) Thus thumbnails give useful feedback about the meaning of query terms and the biases of the collection at high levels of

overview. Once a user has zoomed in on an area of interest, he may want to examine all the results. However we expect these sets to be small and not require scrolling.

4 IMPLEMENTATION

Platform Support

The Perspectives Browser applet is built on top of Piccolo [2], which supports a 2D scene graph representation and provides hooks for picking and animation. Fortunately, PB doesn't draw anything on the angled panel of the Perspective Walls, so it is easy to achieve the 3D effect with 2D graphics. The same functionality was first prototyped in proce55ing [7], and the differences are instructive. proce55ing continually redraws the scene from scratch, rather than having updates be generated by events. (It does support handlers for mouse and keyboard events, but the application builder would have to keep track of dirty regions, which is a big part of the work done by a graphics package.) Thus the application builder avoids the complication of implementing efficient incremental updates. As a result, the code was much easier to write, and more compact by about a factor of two. It also provides a more uniform abstraction for drawing, even supporting a mixture of 2D and 3D with one set of methods

Applets vs. HTML look and feel

The implementation takes advantage of the asynchrony possible with applets, as opposed to html. Loading images and computing new conditional distributions when the query is updated are expensive. Fortunately image loading is less important to a subjective sense of responsiveness than feedback in the summary visualization. Besides updating counts for existing bars, query modifications that drill down to a new level of an attribute hierarchy require creating new bars. Fortunately, this process can be made fast. Creating the right number of bars (and their widths) is independent of the current query, so this information can be cached in the database. Further, only information about one hierarchy level for one attribute is needed. Once the new bars are created, the application begins animating them from a height of zero to the height for the selected attribute, while shrinking the previously selected attribute.

Based on perceptual psychology experiments, one second has become the rule of thumb for animating interface changes. This gives PB one second to compute the conditional distribution before the user really has a chance to notice the change in distribution. The perceptually important feedback is that animation starts quickly in response to mouse clicks. Often, query updates don't require downloading information about a previously unseen hierarchy level, and so initial feedback doesn't even require contacting the database. With a fast network connection, one second is plenty of time to get updated conditional counts (timings are based on a 1.6GHz, 512MB Pentium 4 desktop database server running Windows 2000, on the art database). Over dial-up, it can take up to 10 seconds for a query that generates non-zero counts for many attribute values.

Much effort has gone into optimizing the client code and the MySQL database. The database is the bottleneck, especially for larger databases like the video news collection; little CPU load is imposed by the client in the Piccolo version. On the database side, temporary tables are heavily used to store intermediate query results and share them among queries. There is a temporary table that holds the record numbers of the items matching the query, so the client knows the size of the result set before it has individual bar counts. This would enable beginning the animation of Perspective Wall parameters before animating the bars, but this hasn't been implemented. One could even start animating the bars to their expected height. Then the deviations would be animated separately, increasing their salience. MySQL supports memory-based tables with hashed indexes, which helps with these relatively small datasets. MySQL's JDBC implementation also supports a compressed communication protocol, which improves response time over slow connections.

Since PB initiates its animation within 100ms of mouse clicks, it feels more responsive than Flamenco, where every interaction requires loading a new web page. Flamenco's MySQL design doesn't seem to be optimized, either. The Relation Browser has to wait a similar amount of time for database queries on rollover, but feels slower in the general case because there is no immediate feedback. However the Relation Browser optimizes the special case of the empty query, by sending a table of counts for every possible rollover to the client. In this case, updates take less than 100ms. The number of counts in the table is quadratic in the number of values for all the attributes. Whether the extra time to download the table is justified is both dataset and task dependent.

5 EVALUATION

We conducted a small informal study as an initial evaluation of the PB interface. The study was meant to provide feedback that can influence the design through another round of refinement. We wanted to understand (i) if PB could help support a non-technical audience in knowledge discovery and (ii) how people without training in data mining or EDA conceive of using an interactive visualization to undertake knowledge discovery. We ran the study in two rounds. After the first round, because of how participants performed, we significantly modified both the study and the type of participants we were looking for.

For Round 1 participants consisted of two undergraduate students and one recent graduate who had just completed a Master's degree. The only instruction they were given was that they would be using a web site for an art museum collection. We asked them to answer two directed search questions and two knowledge discovery essay questions. Example of one directed search question: "*Find at least four 19th century Mexican art works that depict Jesus Christ. Do Mexican works depict Jesus Christ more often or less often than those of other countries in this period*?" Example of one essay question "*Compare depictions of the intellectual life of men vs. women. What objects and themes are associated with each sex, and how does this differ among cultures? Write 2 or 3 topic sentences, each suitable for expansion into an essay on the significance of the relationship you found.*"

For Round 2 participants consisted of two PhD students with non-technical backgrounds. One was working on a PhD in history and the other had a product design background and was beginning a PhD in HCI. Both had a little experience with statistics and one had limited programming experience. We decided to use PhD students because we felt they would have a more sophisticated view of the knowledge discovery task, given that they were beginning a career in research. We did not think the PB interface could teach knowledge discovery to an audience unfamiliar with this task, but we hoped it could provide a valuable resource to people who were taking a manual approach to looking for patterns. For Round 2 we quickly demoed the system and then asked each participant to *think aloud* while looking for interesting questions in the data. We wanted to see how many interesting questions they could find in 30 minutes.

Results

For Round 1, participants all completed the two information retrieval tasks. We expected them to refine the query, such as *Loca-*

tion = NorthAmerica => Mexico AND *Date = 19thCentury* AND *Theme = Religion => SpiritualBeing=> Jesus*, and then scan the results. However, they instead kept changing the query around, looking at one set of attributes at a time. When completing these tasks they used the search labels to expand the histograms in order to explore the attribute labels, and they used the Selected Result column to both refine their query using the listed links and to confirm that the artworks contained the elements they were looking for. During this process they almost completely ignored the histograms themselves and they never used the hover feature. None of the participants did much of anything with the essay question. This poor result with the essay motivated us to change the task to being knowledge discovery only, to explain how to use the interface for exploration, and to draw from a participant pool that had at least some familiarity with knowledge discovery, in this case PhD students.

For Round 2 both participants began by first trying to get an overview of the database followed by an exploration of an area in which they had some domain knowledge. This approach is not surprising given that some domain knowledge is require to know when expectations are not met. Both participants focused most of their efforts on comparing histograms and on using the hover feature to both read labels and counts for attributes in the collapsed histograms and to see the preview of the distribution. In addition, both participants used the thumbnails and detail view of the images to assist in better understanding the relationship between the abstract query and the concrete results.

Both participants were successful at discovering interesting questions for further exploration. The designer struggled a little more at this because his domain knowledge of the Middle Eastern art and of architecture was not a good match for the contents of the collection. The history student had more success applying his knowledge of the history of military technology when exploring the differences between works of art from North America and Europe that showed images of women and were classified with a military theme.

Location and date were the attributes explored most extensively. Once participants got involved in a single topic, then began to explore it looking for both changes over time, changes within different locations, or differences in the trends over time for different locations.

While participants found exploring with the histograms and hover feature to be quite easy, they did have trouble accurately interpreting their findings. Participants often constructed queries by selecting two attributes and then relating them to a third attribute. However, when they found something they did not expect, they never thought to evaluate the attributes one at a time. In several cases they encountered false positives that seems to indicate an association among three variables, but where one of the variables was actually irrelevant.

The following is an example of one of the participant's explorations to find an interesting question…

- Began by exploring military because he knows something about this. Knows about military from his initial exploration of the attributes
- Selects women, because he thinks there might be something interesting to discover in how women are portrayed in artworks with military themes
- Expands Location and notices that Europe has a greater percentage of artworks matching *military+women* than North America
- Expands Date and notices that there is a decrease over time for artworks that match *military+women*.
- Hovers over Europe and notices that the decreasing trend for military+women remains the same.
- Hovers over North America and notices that artworks that match military+women increase from the 19th to the 20th Century.
- Decides this is an interesting question for further exploration. Why do artworks matching military+women decrease in popularity in Europe from the 16th to the 20th century while increasing in North American from the 19th to the 20th century? Is this a real trend in the artworks that were produced in these regions, or is this instead an idiosyncrasy of this specific collection.

Discussion

In general the PB interface works very well for information retrieval tasks such as directed search and informal browsing. The study showed that participants could successfully use the PB interface to answer specific questions about the collection. In addition, the interface layout combining the Query, Summary, Results List, and Selected Result provided two clear benefits. First, it allowed users to more immediately understand what the attribute labels really meant. By examining the artworks that matched the active query, users could quickly see how the abstract query related to the concrete examples the system displayed. Second, the layout improved navigation over systems like Flamenco by allowing users to see a detailed result without navigating to a new screen and to even interact with the terms in the detail to refine their query.

The PB interface also allowed users with little training in statistics and programming to engage in interactive visual exploration in order to find interesting patterns in the data. However, in order to successfully engage in this task, participants needed to bring with them some experience in knowledge discovery; in this case their basic knowledge of a research process. In addition, the lack of a sophisticated knowledge of statistics may have influenced the number of false-positives that participants misinterpreted. It will be interesting to see if we can modify the interface to make the false positives more recognizable, or if the interface will require users with more training.

6 FUTURE WORK

Future work falls into two categories: refinement to the interface to guide users away from false correlations and more detailed and robust evaluation of the interface.

If PB is going to support a more general set of users, then something needs to be done to reduce the number of false positive discoveries. For example, one participant "discovered" that the use of the color pink has been decreasing in Europe over the past few centuries, while it has been increasing in North America. However this effect can be entirely explained by the fact that the percentage of all European works has been decreasing, in favor of those from North America. How can we discourage users from forming overly complicated hypotheses? In this case, controlling for Location would eliminate the dependency of Date on color use. Graphically, this can be accomplished by adjusting the widths of the Date bars when the user filters on Location. But how does the interface know to condition on Location and not Color?

One approach is to condition on all but the most recently imposed filter. This ensures that any observed pattern depends on that filter; however, the query might still involve other irrelevant filters. The Browser could also automatically gray out filters that do not contribute to the residual for the selected attribute. This

would prevent users from combining attributes with no relationship, but could diminish the exploratory feel of the interaction

Another approach, which demands more sophistication from users, is to explicitly visualize graphical models that make dependencies clear. These could be inferred automatically, as above, as well as serve as an interface for exploring different models.

The small studies have helped to both discover problems with the interface and to better understand how people approach knowledge discovery tasks. We plan additional iterations as we redesign to improve learnability and the quality of discoveries. In the future we plan to recruit subjects with varying backgrounds to find the best targets for more quantitative studies. We will also vary the datasets used in order to gauge the generality of the approach, including both image and non-image databases. Larger databases may require approximate count algorithms to maintain adequate responsiveness.

In parallel with evaluation on discovery tasks, we plan comparative studies with Flamenco and the Relation Browser on search and browsing tasks.

7 Conclusion

The Perspectives Browser provides an interactive visualization of a dataset with the goal of supporting knowledge discovery for general users who most likely do not have training in statistics or in programming. It uses a new visualization technique to make patterns in residuals salient, based on a model of independence among attributes. It is a special case of Mosaic Plots that is easier to understand, but requires more sequential search to find patterns. Perspective Walls enable precise comparison of bar heights while showing them in the context of the entire dataset.

PB uses a compact design that supports an overview plus detail organization into four columns that show the query, a summary of dozens of attributes, thumbnails of the results, and details for a single result. The design reduced the space devoted to labels by using animation to show only labels in the immediate focus area. In contrast, Flamenco requires scrolling and multiple pages, and the Relation Browser is limited to fewer than 10 attributes.

An iterative design process led to an appealing color scheme, layout, and animation that encouraged subjects to explore longer than with the previous html-based interface. The asynchronous updates supported by applets let us improve the perceived responsiveness despite the need to send slow queries to a database server.

The informal studies indicate that non-technical people with skill in knowledge discovery can use the interface to find interesting questions. Still, an important unanswered question is the size of the potential audience who would be interested in discovering patterns in databases. A mass audience will require intuitive and enjoyable interfaces.

8 Acknowledgements

This material is based on work supported by the Advanced Research and Development Activity (ARDA) under contract number H98230-04-C-0406 and NBCHC040037., and by DARPA STTR contract DAAH01-03-C-R171, subcontracted from MayaViz, Ltd. Bob Futernick of the Fine Arts Museums of San Francisco provided the images and text descriptions; Marti Hearst provided meta-data created by the Flamenco project, and gave helpful feedback on a previous iteration of the Perspectives Browser design.

References

1. Richard A. Becker, William S. Cleveland, and Ming-Jen Shyu, *The Visual Design and Control of Trellis Display*. Journal of Computational and Statistical Graphics, 1996. 5: p. 123-155. http://cm.bell-labs.com/stat/doc/trellis.jcgs.col.ps
2. B. B. Bederson, J. Grosjean, and J. Meyer, *Toolkit Design for Interactive Structured Graphics*. IEEE Transactions on Software Engineering, 2004. 30(8): p. 535-546. http://www.cs.umd.edu/hcil/piccolo/learn/Toolkit_Design_2004.pdf
3. Michael G. Christel. *Visual Digests for News Video Libraries*. in *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*. 1999. Orlando, Florida, United States: ACM Press. http://doi.acm.org/10.1145/319463.319633
4. K.; Doan, C. Plaisant, and B. Shneiderman. *Query Previews in Networked Information Systems*. in *Research and technology advances in digital libraries*. 1996. Washington; DC: IEEE Computer Society Press
5. Steven K. Feiner and Clifford Beshers. Worlds within Worlds: Metaphors for Exploring n-Dimensional Virtual Worlds. in User interface software and technology (UIST). 1990: ACM Press. citeseer.ist.psu.edu/feiner90worlds.html
6. Michael Friendly, *Visualizing Categorical Data*. 2000: BBU Press. 456
7. Ben Fry and Casey Reas, *Processing 1.0 _ALPHA_*. http://processing.org/
8. A. Inselberg and B. Dimsdale. Parallel Coordinates: A Tool for Visualizing Multi- Dimensional Geometry. in IEEE Conference on Visualization. 1990
9. D. A Keim and H.-P. Kriegel, *VisDB: Database Exploration using Multidimensional Visualization.* Computer Graphics & Applications, 1994. 14(5): p. 40-49. http://www.dbs.informatik.uni-muenchen.de/dbs/projekt/papers/visdb.ps
10. T. Kurc, C. Chang, R. Ferreira, A. Sussman, and J. Saltz. *Querying very large multi-dimensional datasets in ADR*. in *SC'99*. 1999. http://citeseer.nj.nec.com/kurc99querying.html
11. Bongshin Lee, Mary Czerwinski, George Robertson, and Benjamin B. Bederson. *Understanding research trends in conferences using paperLens*. in *CHI '05 extended abstracts on Human factors in computing systems*. 2005. Portland, OR, USA: ACM Press. http://doi.acm.org/10.1145/1056808.1057069
12. Jock D Mackinlay, George G Robertson, and Stuart K Card. The Perspective Wall: Detail and Context Smootly Integrated. in Human Factors in Computing Systems (SIGCHI). 1991. New Orleans, LA: ACM Press
13. Gary Marchionini, Carol Hert, Liz Liddy, and Ben Shneiderman. *Extending User Understanding of Federal Statistics in Tables*. in *Conference on Universal Usability*. 2000. http://www.ils.unc.edu/~march/CUU/tables.pdf
14. Ted Mihalisin, John Timlin, and J. Schwegler, *Visualizing Multivariate Functions, Data, and Distributions.* IEEE Computer Graphics and Applications, 1991. 11(13): p. 28-35
15. B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. in Proceedings of the IEEE Symposium on Visual Languages. 1996: IEEE Computer Society Press.
16. Ben Shneiderman, David Feldman, Anne Rose, and Xavier Ferre Grau. Visualizing digital library search results with categorical and hierarchical axes. in Proceedings of the fifth ACM conference on Digital libraries. 2000. San Antonio, Texas, United States: ACM Press. http://doi.acm.org/10.1145/336597.336637
17. James A. Wise, James Thomas, J., Kelly Pennock, David Lantrip, Marc Pottier, Anne Schur, and Vern Crow. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. in IEEE Information Visualization. 1995
18. prevPing Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. *Faceted Metadata for Image Search and Browsing*. in *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2003. Ft. Lauderdale, Florida, USA: ACM Press. http://bailando.sims.berkeley.edu/papers/flamenco-chi03.pdf