Dynamic Visualizations.

Name:	Paul Gokin
Course:	HF 730, Spring 2005
Instructor:	William Gribbons, Ph.D.
Date:	April 29, 2005

Several years ago a researchers at UCLA Berkeley estimated that about 1 exabyte of data is generated annually worldwide (Keim, 2001, p. 39). Naturally, the sheer volume and multidimensionality of this data makes the task of extracting insight from this sea of data daunting. This is probably the most significant human constraint here: we just don't have enough cognitive capacity to be able to juggle all this data by, for example, performing data transformations and what-if analyses in our minds, even if there was a way to activate all of the data at once in working memory.

With computerized analysis becoming commonplace, the emphasis is now on designing data mining systems that combine the power of computers with a human user's flexibility and creativity (Keim, 2001, p. 39), by allowing the human user to formulate complex queries and immediately see the results. Unfortunately, the results expressed in numeric terms (linear format) are difficult to interpret: patterns and relationships are obscured behind arbitrary symbols (which must be processed serially). Expressing query results visually, however, allows the human analyst to use the immense pattern-recognizing power of the massively parallel processes within the visual system. This ability of the human mind makes expressing massive, multivariate data sets visually a potentially attractive solution.

While early research focused on visualizing entities that related in concrete, even physical ways (i.e. spatially or temporally) (Spoerri, 1993, p. 11), much of the data being colleted for analysis is composed of entities whose relationships are abstract: logical rather than physical. This presents one of the main challenges faced by those who design information visualization systems: how to map logical relationships onto a physical display to ensure accurate, clear reading of data and emerging patterns. Of course, the challenge doesn't stop here, as it does with static graphs. The sheer volume of data makes the tools for filtering the data a necessity. In addition, enabling different levels of reading become the key to seeing relationships and patterns at different levels of data granularity. In general terms, then, the main task area addressed by dynamic visualizations is allowing the viewer to explore visually large collections of abstract multivariate data for patterns and connections at different levels of abstraction.

A dizzying array of visualization tools enabling this kind of data exploration have emerged over the last couple of decades. While these visualizations take different approaches to representing data visually and enabling the user to manipulate the representations, they share many common features. A few are discussed below.

Abstract relationships made concrete. Information visualization focuses on mapping abstract data onto the physical space (Keim, 2001, p. 40). Those tools that visualize abstract relationships—the degree of relatedness of items, containment of one item by another, etc.—illustrate the relationship in concrete, visual form. Most of the techniques for visualizing these relationships involve some type of Gestaltist grouping principles: proximity (semantic distance), similarity (i.e. color, size, shape), connectedness, etc. The benefit in amplifying cognition is clear: the human analyst can actually see both the existence and the nature of those relationships (i.e. strength of relationship through line thickness) in concrete form (rather conceptualize them) and do so preattentively, using the powers of the parallel processes in the visual cortex.

Overview first. A part of Ben Shneiderman's "Visual Information Seeking Mantra" (Shneiderman, 1996) overviews allow the analyst not only to see global patterns in the data, but also "size up" the data set to help determine the best strategy for exploring the data further. For example, if the analyst's task is to filter the data to a manageable set to be viewed, the number of data points in the overview may be used to determine how constraining the initial query should be.

Zooming in context. Many of the tools allow the viewer to enlarge an area of the visualized data set (without applying filters) for closer inspection, especially if some of the entities or relationships are obscured due to occluding one another. Unfortunately, straightforward zooming hides the data surrounding the zoomed-in area, obscuring the data's context. A viewer wishing to remain aware of the context has to rely on working memory to maintain context. This leaves less WM capacity to data analysis tasks. A solution chosen by many designs is to distort the data around the zoomed area to fit everything on a single screen using a lens metaphor. In fact, hyperbolic trees can only be viewed in this distorted, "focus+context," form (Lamping & Rao, 1994, p. 13).

In some visualizations, the focus+context principle has also been extended onto highlighting or brushing tools. For example, Inselberg's parallel coordinates visualization (Inselberg & Dimsdale in Ware, 2004, pp. 348-349) allows for highlighting of a specific set of data that satisfies certain parameters, while keeping the rest of the data visible in the background. The benefit of this technique is that the context is maintained—the viewer can see how the selected data is related to the entire data set. Node-link diagrams is another type of visualization where highlighting paths through a hierarchy or a network in this way can be effective.

Direct manipulation in viewing data and querying/filtering. Many visualizations allow the user to manipulate both the data display and the query formulation controls visually, though direct manual or mechanized manipulation (Luntzer's taxonomy from Tweedie, 1997, p. 375). For example, the hyperbolic tree browser allows the viewer to drag the tree presentation around (direct manual manipulation). Ahlberg & Shneiderman's Film Finder allows the user to manipulate the various sliders and buttons to construct queries (mechanized manual manipulation). One of the main benefits of direct manipulation in general is that it relieves the user from the cognitive load of generating linguistically-based commands (Shneiderman, 1997, p. 34). Therefore, it is only natural that direct manipulation be widely used in this highly visual task environment.

Visual queries. Data sets are not only thing that can be represented visually. Some specialized visualizations, like InfoCrystal and Filter Flow, allow the user to formulate queries visually as well. InfoCrystal (Spoerri, 1993) does it by allowing the user to manipulate a highly-stylized representation of a Venn diagram—which in itself if a graphical representation of overlapping data sets. Filter Flow (Young & Shneiderman, 1993) uses a pipe metaphor to let the viewer see how applying various constraints affects the number of results returned. The authors' studies indicate an increase in correct query compositions, comprehension of results, and higher subjective satisfaction using the Filter Flow method rather than text SQL queries (Young & Shneiderman, 1993). The subjective satisfaction findings are interesting, as they indicate that not only is a visual presentation more effective, it also has a positive affective impact.

Details on demand. Drilling down to see the details is an essential part of data exploration: "users, upon discovering patterns, need access to the actual data values" (Eick, 1995, p. 10). It only natural that detail on demand should be so common as to become an element in Shneiderman's "mantra." Drilling down can be done in a few different ways. Hoverpopups are one common technique. They're used effectively on SmartMoney.com's market map. In fact, multiple levels of drill-down are possible. On SmartMoney.com hovering over a stock square reveals just the basic information, while clicking it reveals more links, which in turn lead to even more information.

Maintaining context, while not as important as during zooming, is still an important design consideration here. Eick's SeeSoft application accomplishes this by using multiple linked views that are visually connected to each other, showing how the overview was "exploded" to show progressively finer levels of detail.

Instant feedback. A well-designed system can supplement the user's memory to the point that it may seem like an extension of the user's mind. This synergy is especially important is the visualization is used as a hypothesis-generating and testing tool—one of the primary purposes of dynamic information visualization identified by Keim (Keim, 2001, p. 40). Instant feedback allows fluid data exploration by filtering, brushing, zooming, switching perspectives, etc. There a clear working memory advantage here: quick response means that the memory trace of the hypothesis that prompted the interaction with the display has less time to fade and is more likely to be available in the WM when the display is updated with the results. The viewer does not lose the "train of thought."

Good fit with the user's task and type of data. Dynamic visualizations are usually created with a specific purpose: to answer a specific set of questions about data related in a specific way. For example, methods based on the node-link paradigm are good for letting the user explore hierarchically structured entities and find answers to questions like: "How is this entity related to that entity?," or "How many 'children' does this entity have?" In fact, even within this class of visualizations, there are variations that suit some situations better than others. Take a node-link diagram and a treemap. When arranged in a form of a tree, the node-link diagram makes the hierarchical relationships obvious (Ware, 2004, p. 217). On the other hand, a treemap can display many more items within the same amount of space (Ware, 2004, p. 217) and relate the items along an additional quantitative dimension by encoding it using the rectangle size representing each entity.

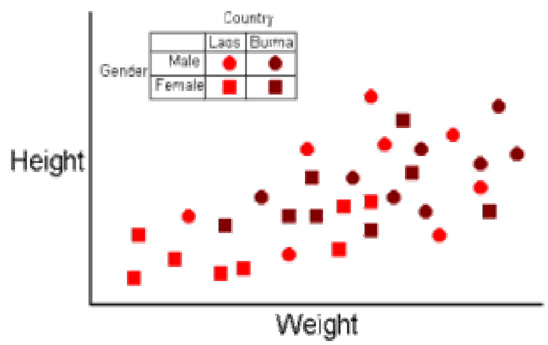
What makes matters more complicated here is that entities can often be related in several different ways, depending on which attribute is used as the basis for relationship, even within the same type of relationship (hierarchical, temporal, geographical, etc.) For example, animals in a zoo can be organized geographically by their location at the zoo vs. their location in the wild, or by how old their species is vs. how old the animal is, or by how large their species is in terms of numbers in the wild vs. how many of them live at the zoo, etc. A different type of visualization is required to most effectively show each of these types of relationships. In light of this kind of variety, it is no wonder that the match between the visualization and both the user's task and the data structure is a central theme in infoviz literature (Carr, 1999, p. 3; Keim, 2001, p. 43; Eick, 1995, p. 6).

Throughout this paper we've emphasized the ability of dynamic visualizations to reduce the viewer's working memory load through cleverly-designed viewing tools, direct manipulation, and instant feedback. There is one fundamental property that remains undiscussed: visually encoding data in a way that lets the analyst see important relationships and patterns at a glance. Jacques Bertin's answer to this problem is the central tenet of his image theory: "[T]he most efficient constructions are those in which any question, whatever its type or level, can be answered in a single instant of perception, in a single image." (Bertin, 1983, p. 11). Unfortunately, there are very few data dimensions that can be encoded using symbols placed together in a single graphic so that they are perceived preattentively and independently of one another. Bertin's limit here is three: two dimensions encoded spatially (naturally ordered planar variables) and one using the property of the glyph itself (visually ordered retinal variable) (Bertin, 1983, pp. 148-149).

Marc Green builds on Bertin, proposing that "the one retinal variable" rule can be extended to two or more by using animation (motion and flicker) and 3D (binocular disparity) to encode data

(Green, 1998, pp. 20-22). However, not all retinal variables can encode all data types equally well. For example, binocular disparity is highly selective—it is easy to preattentively group elements lying on the same plane even if they differ on another retinal dimension (like color or shape), but is not associative—it is difficult to group targets across different surfaces even if they share the same value of another retinal variable. In effect, disparity "gets in the way" of, or disrupts, the grouping established by another retinal variable.

Data encoding techniques have a significant effect on the effectiveness of user tasks at the perceptual level. Carefully selected encoding methods can enable certain comparisons, while preventing others. To use Green's example (see right), if the task requires that the viewer compare Laos to Burma disregarding gender, then the task will be successful, because similar shapes are associative: they are not different enough to disrupt the groupings created with brightness. If, on the other hand, the viewer was required to compare males and females (disregarding the country), then this coding would be ineffective: it is much more difficult to perceptually group glyphs by shape, ignoring their brightness. The point here is: data dimensions should be encoded using methods that make it easy for the user to make the required comparisons preattentively.



from Green, 1998, p. 9

The safest approach here seems to be to limit oneself to two planar and one retinal variable, as I have done in the visualization proposed below.

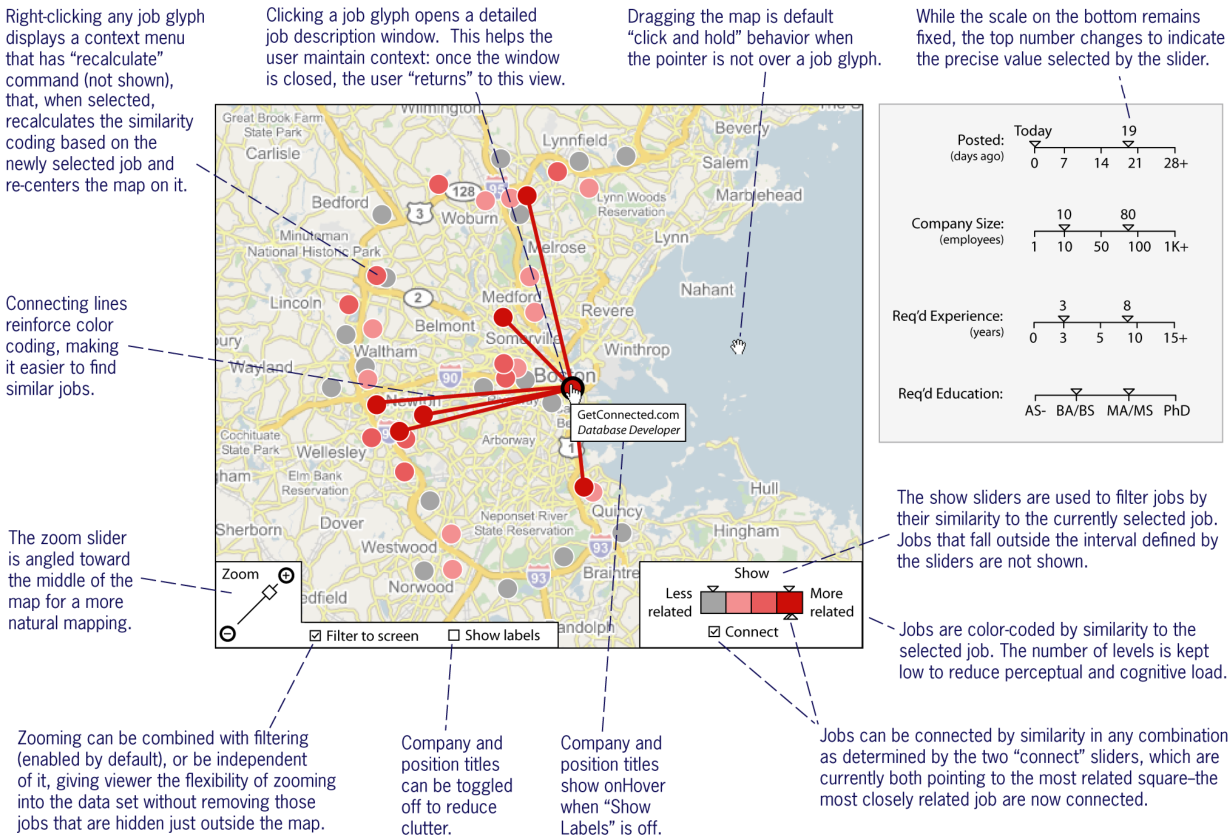
Case study: finding the perfect job by similarity.

Amazon.com went to the bank with the idea of showing its customers items similar to the one they are currently viewing. A similar principle can be applied to searching for jobs: once a job seeker finds one job that fits his or her requirements, the system can suggest jobs with similar descriptions. The strength of suggestions would be based on how similar job descriptions are. The more similarities, the stronger the recommendation. There are, of course, other factors beyond job description, like job location, required experience/education, and company size that are also factored into the job seeker's decision to apply. Unfortunately, finding the perfect job—the one that has the best combination of all of the pertinent variables—is difficult. Typical web sites present job seekers with long lists of jobs, most of which are unrelated to what the job seekers is looking for. The job titles are of not much help either, since different employers often call the same job using two different names, or have different duties listed under similarly-named jobs.

The purpose of the proposed dynamic visualization is to make it easier for the job seekers to see the "job landscape" and home in on the jobs that are right for them. It is designed to allow the viewer to answer questions like: "I really like this job. Are there a lot of jobs out there that have a similar description?" "Are there more like this in my area?" "Are there geographical areas which have more jobs like the one I want?" In addition, a relaxed initial search and instant feedback during filtering will yield fewer empty result sets. This approach may also encourage the viewer to consider more jobs that he might not have thought to search for: for example a job with an uninteresting title that is actually quite interesting based on its description. On top of this, a tired job seeker may overlook a potentially good job in a long list, but will notice it when it is plotted and visually connected to the job he knows is a good match. In this sense, the system can combat cognitive narrowing by explicitly pointing out options they may have overlooked.

Here's how a screen for this type of system might look:

This is a portion of the screen shown once the initial set of jobs has been returned by the site's search engine based on a job title, metro area, and, optionally, keyword search. The best match is automatically selected (black stoke around the red job circle) and the map is centered around it.



The advantage of the proposed implementation over traditional tabular presentations is that the two most important variables—the nature of the job[1] and its location—are both expressed spatially, which allows the viewer to make judgments about the relationship between the locations and nature of jobs. Jobs are "plotted" over a map. When a user selects a job of interest, the rest of the jobs with similar descriptions are rated by how similar they are to the selected one. Jobs are then color-coded based on how similar they are to the chosen one. In addition, the user can filter the job set on other variables. Finally, the user can click on the job and select to view more details about it.

The visualization distributes the decision making process between verbal and visual systems:

- **geography** is indicated visually, allowing the viewer to process these relationships visually. In effect, this transfers the cognitive load away from the verbal system, which

[1] Nature of the job is expressed through degree of similarity to the ideal.

would otherwise have to deal with translating verbal expressions of places into the representation of their physical relationship.

- **relatedness** is a highly abstract and complex relationship that would be prohibitively difficult for the cognitive system to estimate accurately across a large set of jobs. Does this mean that a graphical representation is the only choice here? Of course not. Relatedness could have been expressed as a number (1 through 4) and jobs could have been sorted on it in a list. However, this would have made it more difficult for the viewer to rapidly make decisions about location vs. job fit trade-offs. Visualizing both allows the viewer to process this relationship with much less cognitive effort.

Bibliography.

Bertin, J. (1983) *The semiology of graphics.* Madison, WI: University of Wisconsin Press.

Carr, D. (1999). Guidelines for Designing Information Visualization Applications. Proceedings of the 1999 Ericsson Conference on Usability Engineering. Retrieved on April 23, 2005 from: www.ida.liu.se/~davca/postscript/VizGuidelines.pdf.

Eick, S.G. (1995). Engineering Perceptually Effective Visualizations for Abstract Data. AT&T Bell Laboratories. Retrieved April 21, 2005 from http://citeseer.ist.psu.edu/rd/69894310%2C56924%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/2514/http:zSzzSzwww.bell-labs.comzSzuserzSzeickzSzbibliographyzSz1997zSzpercept.pdf/eick95engineering.pdf

Green, M. (1998). Toward a perceptual science of multidimensional data visualization: Bertin and beyond. ERGO/GERO. MS Word Version.

Keim, D.A. (August, 2001). Visual Exploration of Large Data Sets. *Communications of the ACM, 44(8)*, pp. 39-44.

Lamping, J., & Rao, R. (1994). Laying out and visualizing large trees using a hyperbolic space. *Proceedings of the 7th annual ACM symposium on User interface software and technology, 1994.* pp. 13-14.

Rao, R., & Card, S. (1994). The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence, 1994.* pp. 318-322.

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualization. *Proceedings of the 1996 IEEE Symposium on Visual Languages*, pp. 336-340.

Spoerri, A. (1993). InfoCrystal: A Visual Tool for Information Retrieval and Management. *Proceedings of the ACM Conference on Information and Knowledge Management, 1993*, pp. 11-20.

Shneiderman, B. (1997). Direct manipulation for comprehensible, predictable and controllable user interfaces. *Proceedings of the 2nd international conference on Intelligent user interfaces table of contents, Orlando, Florida, 1997.* pp. 33-39.

Tweedie, L. (1997). Characterizing interactive externalizations. *Proceedings of the SIGCHI conference on Human factors in computing systems.* pp. 375 - 382.

Young, D. & Shneiderman, B. (July 1993) A Graphical Filter/Flow Representation of Boolean Queries: A Prototype Implementation and Evaluation." *Journal of the American Society for Information Science, 44(6)*, p. 327.

Ware, C. (2000). *Information for Visualization: Perception for Design.* San Francisco, CA: Morgan Kaufmann.