Data Object and Label Placement For Information Abundant Visualizations

Jia Li[#], Catherine Plaisant, Ben Shneiderman*[#]

Human-Computer Interaction Laboratory
[#]Department of Computer Science
*Institute for Systems Research
University of Maryland, College Park, MD 20742
{jiali, plaisant, ben}@cs.umd.edu

Aug 27, 1998

ABSTRACT

Placing numerous data objects and their corresponding labels in limited screen space is a challenging problem in information visualization systems. Extending map-oriented techniques, this paper describes static placement algorithms and develops metrics (such as compactness and labeling rate) as a basis for comparison among these algorithms. A control panel facilitates user customization by showing the metrics for alternative algorithms. Dynamic placement techniques that go beyond map-oriented techniques demonstrate additional possibilities. User actions can lead to selective display of data objects and their labels.

KEYWORDS

Timelines, data object placement, label placement, information visualization, control panel, metrics, visual feedback

1. INTRODUCTION

Mapmakers and now, visualization designers have realized that designing effective presentations for abundant information is a difficult task. Part of the problem is the large number of the data objects compared with the limited screen space. Maximizing the display of data content in a comprehensible way is a problem that has been addressed by many researchers. Mapmakers often turn to larger sheets of paper, but information visualization designers must work within a limited screen space. However, the dynamics of zooming, panning, and selective display can be powerful techniques.

Data objects are the essence of visualization systems, and therefore effective layouts are those that present large numbers of them and reveal semantic relationships among them. Since *labels* identify and explain the data objects, placing the labels directly on and around the data objects presents an integrated information overview. It frees the users' eyes from darting back and forth among the scattered elements on the screen, thus reducing users' time in the data comprehension process. We found that label placement is a challenging problem with few practical and satisfactory solutions, because of the following two issues:

1- Optimal labeling algorithms can be too computational expensive for interactive systems. While these algorithms work well for small sized problems, they are impractical due to their exponential nature. It is worth noting that labeling problems have been proven to be NP-hard. In map production systems, it is acceptable to have these algorithms run for days in order to generate a high quality of map. In interactive systems though, users place high demands and expectations on how long they can wait for the computer to respond. The frequently mentioned 2-second limit seems appropriate for many tasks [26].

2- Labels compete with data objects for the same limited screen space and data objects normally receive the greatest attention. Increasing data object density in a reasonable way makes the screen layout more compact, thus decreasing the need for scrolling. However, this leaves less space for placing legible and meaningful labels.

We have implemented a set of techniques in LifeLines, for medical patient records. LifeLines is a general visualization environment for personal histories [21]. LifeLines begins with a one-screen overview of the record in the metaphor of timelines, and users can then see more details using zooming tools or filters. One of the limitations of the early prototype is that too much space is left unused yielding a low information display. Our techniques address this limitation, but the efficacy of each technique varies with user communities, requirements and circumstances. We let users steer the decisions with a Control Panel that is equipped with feedback information. In addition to map-oriented static solutions, we propose dynamic solutions that take advantage of the interactive nature of computer displays.

2. RELATED WORK

2.1 Data Object Placement

Maximizing the display of data content in the limited screen space is one of the research goals in information visualization systems. Decisions need to be made on *what* data items to display and *how* they are laid out so that "users can see all of the possibilities and navigate among them" [25]. Two approaches have been widely adopted to address the data layout issue by focusing on "*what*" and "*how*" respectively [15].

1- More efficient data selection techniques are created to display data of interest in smaller chunks requiring less space. A good example of this is the work of Ahlberg [1], where a dynamic query interface provides continuous feedback to users as the graphical query is formulated (http://www.spotfire.com). Other examples are the Magic Lens, which encodes each operand of the query as filter [10] and Pad ++, which provides smooth zooming in a system that can work with large datasets [3] (http://www.cs.umd.edu/hcil/pad++).

   These systems do not have optimal layout strategies of the result set. Good global layouts may not apply well to localized data. The best layout algorithm depends on what information users are currently focused upon [15]. Therefore users should have control over the layout process so the resulting layout will reflect their current focus [18].

2- More efficient visual layouts are represented on-screen. Novel approaches to hierarchical information have been invented: Cone Tree layouts use three-dimensions [23], hyperbolic trees use the hyperbolic plane mapped onto a circular display region [17], and treemaps [24], use a space-filling two-dimensional rectangular layout.

The underlying structure imposes many constraints on where the data objects can be placed on the screen. However, this still leaves much room for varying data object placement. The developers of these systems, have just begun to explore alternative layouts.

2.2 Label Placement

Label placement has been a fundamental task in the field of cartography and GIS. Over 500 years, cartographers have collected a great deal of knowledge and rules of how to make a high-quality map. Imhof [14] illustrates these rules by giving examples of good and poor labeling. Automatic label placement has been proven mathematically as an NP-hard problem and it remains a research problem after twenty years of development. Research attention has thus shifted towards powerful heuristic methods that may not exhibit guaranteed performance bounds, but work acceptably in practice [7, 28].

ArcView, a commercial GIS mapping system, helps users analyze data in a spatial context. Its “Find Best Label Placement” combined with non-overlapping method works well in a non-dense scenario, where it places as many labels as possible (See Figure 1). However, it requires extended computing time for even moderate-sized datasets, and labels are not clearly associated with their data objects. Users are provided with several labeling options. They can auto-label either all the features or a selected set of features, change the font size, style, set the location of labels relative to their features, or allow and not allow overlapping labels. ArcView does not apply effective techniques for overlapped labels. It dramatically reduces visibility and overall quality even with small overlaps.

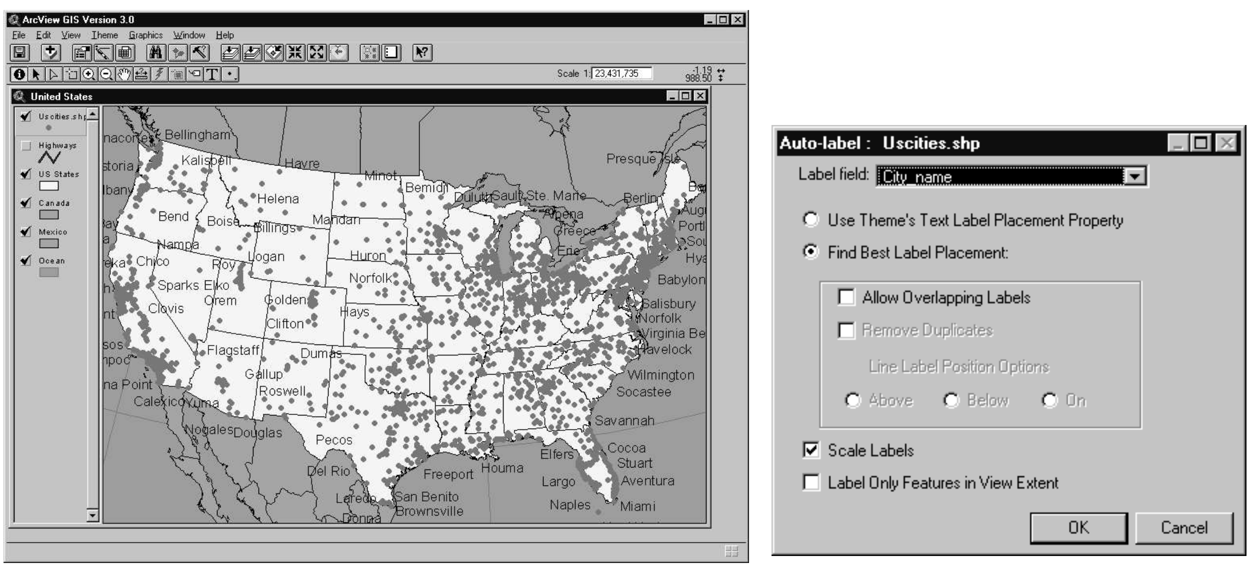


Figure 1: ArcView 3.0 map display and label control panel

The Hyperbolic browser [17], on the other hand, makes effective use of overlapped labels, as shown in Figure 2. It provides short and long labels and users can change font size easily. But still, the amount of text that the hyperbolic browser displays is a problem. The experimental task conducted to contrast the hyperbolic browser against a conventional 2-D scrolling browser with a horizontal tree layout, was particularly sensitive to this problem because of the length and overlap of URLs, and the ill-structured nature of the WWW hierarchy [17]. It reveals an important yet easily ignored factor of label placement - label content.

Interactive TimeLines [2] illustrates a poor design of labeling. It reduces the label legibility and at the same time, it leads to a low information graphics design. Generally, words should follow the ordinary writing direction from left to right, the so-called “clockwise direction” or “writing sense” [14].

Additional Controls
To decrease text size: **"Shift <"**
To increase text size: **"Shift >"**
To view or toggle long and short text for all nodes: **"o"**

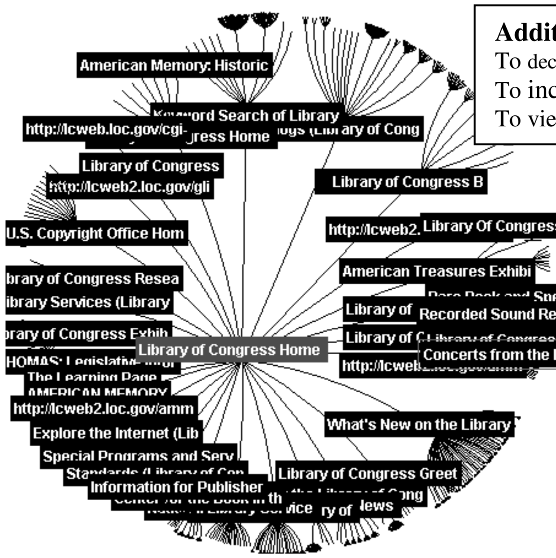


Figure 2: Hyperbolic Browser

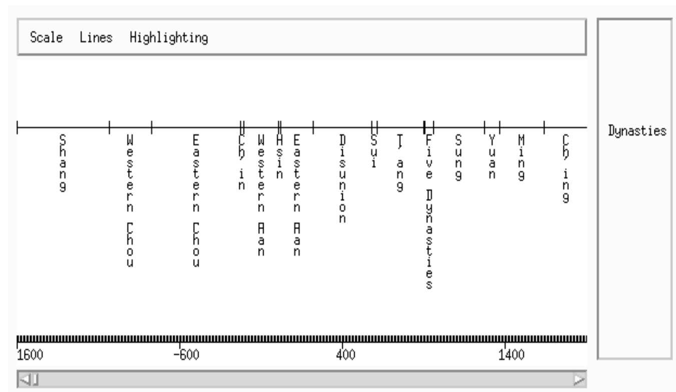


Figure 3: Interactive Timelines

Labeling by brushing [6], a direct manipulation technique developed by Cleveland, can selectively label data points that interest users. The labels can remain after the brush is moved away, if the mode is set to be *lasting*. This technique works nicely until more labels remain on the screen and they start to overlap with each other.

Another dynamic labeling technique, *text streaming* is proposed in the Bead exploration system [5], where a sample of labels is turned on and then a new sample follows. This successive sampling of labels is helpful in a way that it presents all the details by not cluttering the screen. However, it suffers stability problem since the changes are abrupt and users cannot foresee the next move.

3. EXPLORING THE LAYOUT DESIGN SPACE

In this section, we describe algorithms and techniques we have developed to address the placement problems. We implemented them in the LifeLines visualization system for medical patient records and we use this system to demonstrate these generally applicable techniques.

3.1 Data Object Placement

Establishing an underlying structure to organize data objects on the screen is a key step towards effective information visualization systems. LifeLines lays out temporal events horizontally across the time axis (x-axis) in the 2-D space. When it is applied to visualizing patient records, aspects like medical conditions, office visits, hospitalizations or medications are displayed as individual time lines. Line color and thickness illustrate relationships or significance [21]. An empirical study [18] showed that the LifeLines representation leads to faster response time than a textual design for tasks that involves interval comparisons and making inter-categorical connections.

While the starting and ending x-axis values of timelines are fixed by this structure, the freedom of placing timelines anywhere in the vertical space leads to a set of layout algorithms that can be designed to optimize space utilization or reveal more data relationships.

1- Compact Layouts
Figure 4a demonstrates the most compact version, i.e. “slow compact” of LifeLines. All the events are first sorted by their starting time. For each event, the algorithm searches all the lines from top to bottom for an available space to fit the event, i.e. the event will not overlap with other ones. If no space is found, a new line will be created to place the event. “Quick compact”, on the other hand, skips the sorting step. A default layout simply searches the bottom line for available space.

2- Attribute Based Layouts (Chronologically ordered or Event-name ordered)
Besides space utilization, attributes can be the criteria for data object placement. The “chronologically ordered” algorithm sorts the events by their starting time and places each event on a new line. An example is shown in Figure 4b. “Event-name ordered”, as in Figure 4c, lays out the events with the same name on one line if possible.

Figure 4a: Compact layout

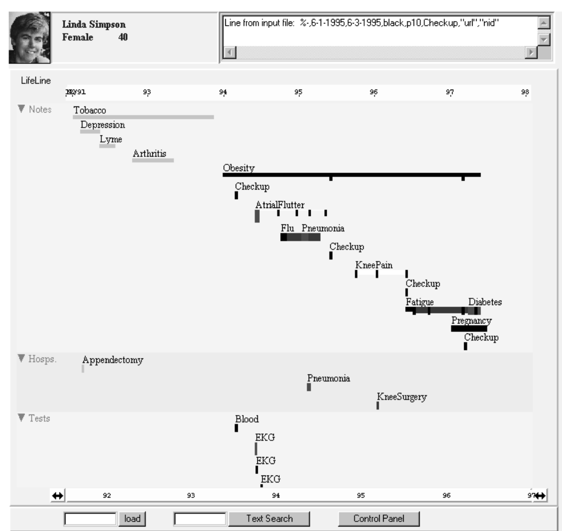


Figure 4b: Chronologically-ordered layout

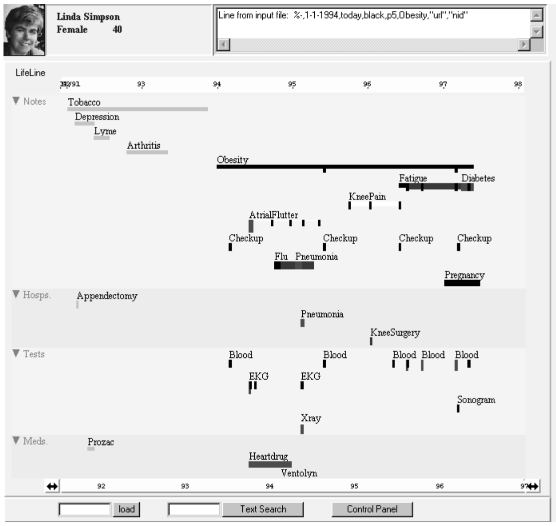


Figure 4c: Event-name ordered layout

Data Layout Algorithms	Compactness	Grouping	Occlusion
Default compact	6.08%	2	4%
Quick compact	8.17%	2	4%
Slow compact	9.59%	2	4%
Chronologically ordered	4.51%	3	0%
Event-name ordered	4.63%	3	4%

Table 1: Metric values for data layouts

Each one of these layouts provides certain benefits to users, but no single layout can always produce the best result. Compact layouts present a much richer screen when dealing with large records and minimize the need for scrolling. However, the grouping of events horizontally becomes less meaningful. A chronologically-ordered LifeLines helps users to review the events evolving across time. Unfortunately, a sparse data layout is likely to occur and inevitably, requires increased scrolling. An event-name ordered LifeLines groups similar events horizontally and users can gain insight into how many of those events occurred in the past and how frequently. In this case, screen space utilization depends heavily on the data itself.

We believe that research and practice will be advanced if useful criteria and metrics can be defined to compare layout algorithms. We have developed three metrics, *compactness*, *grouping* and *occlusion* to capture how well each layout strategy utilizes the space and reveals data relationships. We describe how to incorporate these metrics into the system in section 4.

1- Compactness is defined as:

(number of data object pixels / total number of pixels in the display area)

It ranges between 0 and 1 and the larger metric value indicates more compactness. A low compactness of data graphics is not desirable. It is suggested that the more data be shown within one display, the more effective and comparative user's eye can be [27]. However, very high compactness can make the data graphics more difficult for users to comprehend. Development of lower and upper bounds of this metrics will add tremendous value in evaluating the effectiveness of data placement algorithms. Table 1 shows the metric value of five layout algorithms against the same dataset.

2- Grouping is defined as:

(number of attributes used to group or order the dataset spatially)

A larger number indicates that more data dimensions are mapped spatially on the screen. For instance, all the 5 data layout algorithms have a minimal grouping value of 2, since the facet and aggregate name are the two attributes to gather the data together. The "event-named ordered" algorithm further groups similar events horizontally, thus increasing the metric value by 1.

3- Occlusion is defined as:
(number of data objects completely obscured / total number of data objects)

It ranges between 0 and 1 and the larger the value is, more data objects are completely overlapped. In LifeLines, data can be obscured because the graphing symbols are always rounded on each scale. As long as the objects are not completely overlapped, they can be visually detected without loss of much information.

3.2 Label Placement

Label placement is a crucial issue when dealing with large numbers of records. Our early prototype, as shown in Figure 5, illustrates the traditional labeling challenges:

1- Limited space to mark all labels.
Only 8 out of 14 events have labels in Figure 4 based on the early labeling rules, which are:
- ♦ The label is located at the right and above the data item
- ♦ The label will be dropped if it overlaps with previous ones.

2- Vague association with data objects:
For example, inside the square box of Figure 5, it's hard to tell which label if any is associated with the middle event. This limitation while pointed out in most literatures, has not been fully realized in the general GIS community. Visualization designers though, must ensure graphical integrity and accuracy. Any data ambiguity may mislead users to reach wrong conclusions or fail to spot critical information. As in this example, the same middle event color coded in red, is an unlabeled abnormal sonogram test. However, there is a chance that the doctor might read it as a blood test and makes to a wrong diagnosis. Labels should only be used to remove graphical ambiguity instead of introducing it.

Figure 5: LifeLines with poor labeling

Label Positions

Good name positions aids map reading considerably and enhance the esthetics of the map [14]. Based on Imhof's well-known guidelines, we have defined 4 candidate label positions for LifeLines data items (NE, NW, SE, SE) and their preference order is listed in Figure 6.

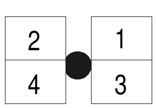


Figure 6: Candidate Label Positions and their preference orders

We chose to use the exhaustive search algorithm in which backtracking is performed, i.e. the algorithm returns to the most recently labeled item and considers the next available position. The algorithm continues until an acceptable labeling is found or until the whole search space is exhausted. Exhaustive search algorithms like these can become very expensive for even moderately sized problems. It turns out to be acceptable in the LifeLines case where each search space contains about 10-20 items. Figure 7 shows the result of applying this algorithm to the same dataset. In this case, all 14 items get labeled although some vague association problems still exist.

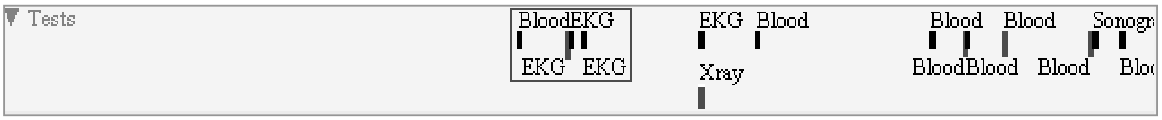


Figure 7: LifeLines with improved 4-candidate Labeling

Label connectors, shown in Figure 8a, are thus introduced to link the data objects with labels to clarify the association. However, it then leads to a more serious "crowing problem" [22] and lower data-to-ink ratio [27] since more ink in the graphics is now devoted to non-data items. In order to decrease the ink redundancy, we introduce a reduced label connection algorithm. Label connector links to the data only if the algorithm determines that the data object can be associated with more than one label, i.e., other labels reside in the labeling boundary of the current data object. The labeling boundary in LifeLines is defined as follows:

> If the x-axis range of data object is (x1, x2), then the x-axis labeling boundary is (x1 - deltax, x2 + deltax), while deltax defines how far away between the current data item and the labels of other data objects.

Figure 8b demonstrates the reduced label connector in LifeLines. While keeping the unique association between the data objects and their labels, the data graphics is less crowded than the previous one.

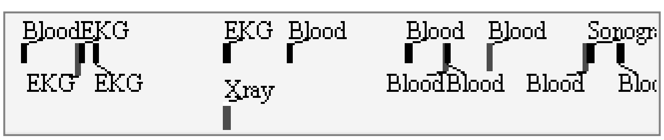
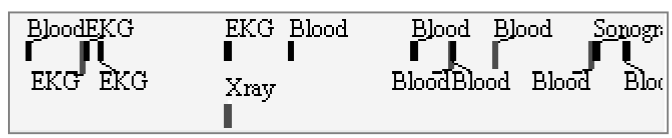


Figure 8a: LifeLines with label connectors



Figurer 8b: LifeLines with *reduced* label connectors

Semantic Labeling

Few system designers have explicitly looked at labeling techniques that take into account of semantic relationships or patterns among the data objects. Labels are used to explain the data and thus should reflect them. Three tactics will be presented here that captures different data characteristics: importance order, level of details and repetitive data.

A. Label Saliency

Saliency is a domain-specific measure of the relative importance or prominence of an event, and can refer either to particular events, characteristics of events, or classes of events [19]. For example, in LifeLines, abnormal events might be more significant and therefore, the labeling algorithm should allocate space resources to those data labels first. Appropriate tools should be provided to the users and domain experts to grant the importance order of those events.

B. Label Aggregation

Aggregation rules can be established when hierarchical data models are available. In LifeLines, events are grouped into aggregates and aggregates into facets. One of the rules can be defined as follows: label aggregates when the space does not permit labeling all the detailed event objects. For instance, as shown in Figure 9, a series of athenolol and propanolol are aggregated as beta-blockers. A high-level overview of the data set is presented rather than a partial set of individual data objects. Aggregation information, even though leaving out details, covers a complete data set and provides necessary cues for users to drill down to the details.

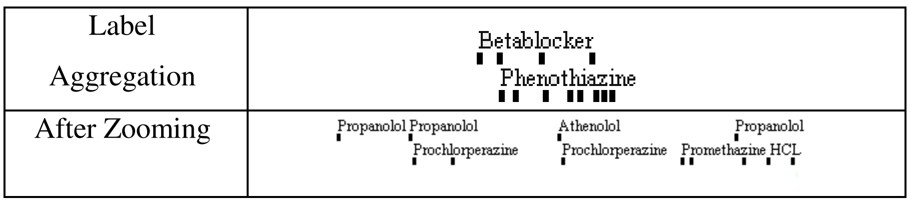


Figure 9: Label Aggregation for four drugs in two classes

C. Label Integration

A continuous series of events with the same name attribute can be tagged with a single label, eliminating duplicate texts and at the same releasing screen resources to other events. However, if that single label is too far away from some of the events, association will become vague again. Therefore, we will only discard the label for the event that already has a similar label residing in its labeling boundary. We applied this technique to the same medical test data sets and the result is illustrated in Figure 10. In the square box, notice that the three blood test events share the same label while the immediately followed event does not.

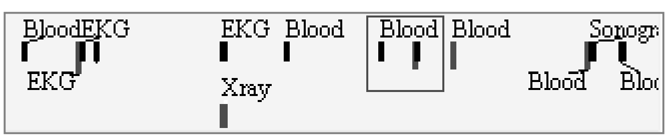


Figure 10: Label Integration

Metrics

We introduced three metrics to compare these labeling algorithms: “labeling rate”, “overlapping rate” and “association degree”.

1- Labeling rate is defined as,
(number of labeled objects) / (total number of objects).

It ranges between 0 and 1 and obviously, the higher the value is, the more objects are labeled.

2- Overlapping rate is defined as,

$\sum$(overlapped length / label length) / (total number of objects)

It ranges between 0 and 1 and the higher the value is, the more labels are overlapped.

3- Association degree is defined as

(number of objects that are clearly associated with labels) / (total number of objects)

but, when label connectors are used, the value will be 1. For an object to be clearly associated with its label, its label must not reside in the labeling boundaries of other objets.

In addition to these three, more metrics should be introduced to capture other important aspects of labeling. However, some of them are difficult to quantify. One good example is “readability”. Think of the scenario where very small fonts are chosen for labeling. Designers can attain high values of labeling rate and association degree, but the labels may be useless, if they are too small to be readable. Also, the readability metric plays a crucial role in evaluating the semantic labeling algorithm as well. The metric value, however, is heavily dependent on users’ perception and a standard way to quantify it is yet to be found.

3.3 Dynamic Placement Techniques

All the placement algorithms we have presented so far, are designed to produce a *static* data “map” that is highly comprehensible. Exploiting the dynamic and interactive nature of visualization systems opens the door to other useful techniques. For example, moving the mouse over the data object might cause the label to appear, thereby also clarifying the association. More extensive labeling can be “ballooned” out when the user is focused on a complex object. Another approach is to apply labels only to objects that are selected by dynamic queries.

A challenge of dynamic placement techniques is to balance display stability and best use of screen space. When users start to zoom in, many objects may fall out of the screen and if we do not allow re-layout, space will be underutilized (Figure 11a). At the same time, users have to scroll down to view other data objects. Figure 11b is the result after applying a re-layout operation, resulting in a more compactness of data graphics. However, the change of object locations may detract from users’ comprehension of the structure. The instability may be even more distracting when continuous re-layout is conducted during zooming.

Figure 11a: Zooming without re-layout leaves large blank areas with only 19 objects.

Figure 11b: Zooming with re-layout allows more data objects, 29 to be visible.

4. CONTROL PANEL COUPLED WITH FEEDBACK

Our control panel was designed to promote users' capability to tailor the systems based on their preferences, reasoning and goals. Appropriate feedback about the system can help foster user autonomy [11]. Combining these two together in the same user interface provides an integrated, informative and predictable environment to the users in their decision-making process.

As shown in Figure 12a, we incorporate the metrics described in section 3.1 into the LifeLines control panel with data layout options. The metric values are computed dynamically against the current dataset. Armed with these metrics, users may be able make more appropriate decisions for themselves. Similarly, we provide all the options of label placement algorithms described previously [Figure 12b]. Font size can be changed easily through a value slider. Label length can be truncated via a slider as well, to any number of characters within a pre-defined range. Feedback information on metrics is being added for these controls.

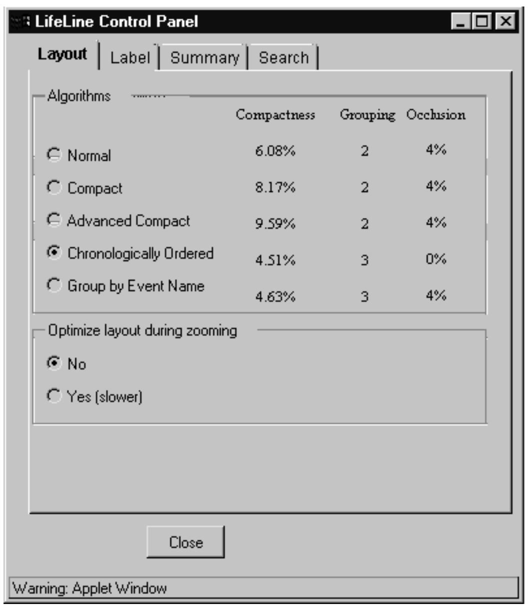


Figure 12a: Control panel with data object placement options

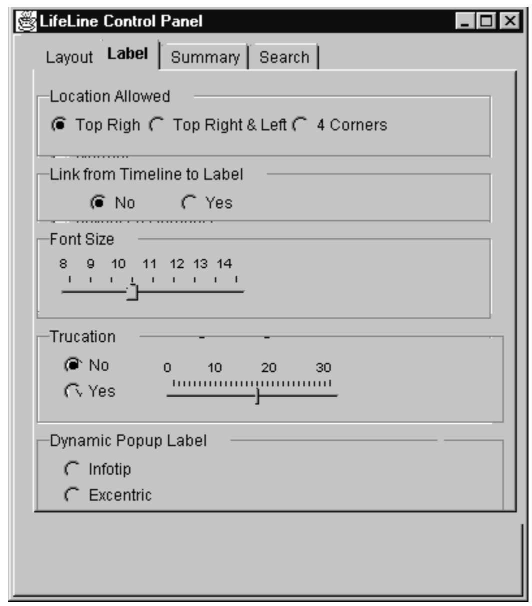


Figure 12b: Control panel with label placement options

5. CONCLUSION

Placement of data objects and their corresponding labels plays an important role in supporting information visualization. We have suggested algorithms and techniques to address these issues. Compact layouts have powerful advantages, but ultimately the screen will become too densely filled to be comprehensible. Therefore attribute-based approaches that allow users to selectively display data objects seem necessary.

We have developed metrics and actively used them in our control panel. Providing feedback about alternative placement algorithms or techniques can enable users to make appropriate choices to match their tasks. We believe that further study will lead to new metrics that will capture other important characteristics of the placement problem.

Static techniques for paper-based layouts should be explored, but the opportunities for dynamic techniques seem great. If task-related user actions can influence the placement of data objects and labels, then the right information can be made to appear more often. For example, if users move a cursor on to an X-ray object, then previous X-rays might be highlighted and labeled, thereby inviting physician exploration for comparison purposes. If users move a cursor on to a surgical procedure, the notes of the referring physician and the hospital records might be highlighted and labeled, thereby inviting physician exploration for background understanding. Additional tasks such as saving objects, navigating among a sequence of objects, and reviewing an entire history suggest other opportunities for dynamic techniques [20].

Control panel design to provide user control on the data object and label placement algorithms and techniques is a rich topic that deserves wider attention in the information visualization community.

REFERENCES

1. Ahlberg, C., Williamson C., and Shneiderman, B, Dynamic queries for information exploration: An implementation and evaluation. *Proc. of ACM CHI'92*, ACM, New York (1992), 619-626.

2. Allen, R. B., Interactive timelines as information systems interfaces. *Symposium on Digital Libraries*, Japan (August 1995).

3. Bederson, B. B., Stead, L., and Hollan, J. D., Pad++: Advances in multiscale interfaces. *Proc. of ACM CHI'94*, ACM, New York (1994) 315-316.

4. Brath R., Concept demonstration metrics for effective information visualization. *Proceedings of Information Visualization,* Phoenix, Arizona, (1997), 108-111.

5. Chalmers M., Ingram R. and Pfranger C., Adding imageability features to information displays. *UIST'96*, Seattle Washington USA. ACM

6. Cleveland W. S, *Visualizing Data*, (1993), Hobart Press, Summit, New Jersey

7. Christensen, J., Marks, J., and Shieber, S., An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics* 14, 3 (1995), 203-232.

8. Doerschler, J.S. and Freeman, H., A rule-based system for dense-map name placement. *Communications of the ACM* 35, 1 (1992), 68-79.

9. Edmondson, S., Christensen, J., Marks, J., and Shieber, S. M., A general cartographic labeling algorithm. *Cartographica* 33, 4 (1997) 13-23.

10. Fishkin, K. and Stone, M. C., Enhanced dynamic queries via movable filters. *Proc. of ACM CHI'95*, ACM, New York (1995), 415-420.

11. Friedman, B., User autonomy. *ACM SIGCHI Bulletin* 30, 1 (January 1998).

12. Harrison, B. L., Ishii, H., Vicente, K. J., and Buxton, W., Transparent layered user interfaces: An evaluation of a display design to enhance focused and divided attention. *Proc. of CHI'95*, ACM, New York, (1995).

13. Henry, T. R and Hudson, S. E., Interactive graph layout. *Proceedings of the ACM Symposium on User Interface Software and Technology*, (October 1991), 55-64.

14. Imhof E., Positioning names on maps. *The American Cartographer,* 2 (1975), 128-144.

15. Jones, C. V., *Visualization and Optimization.* Kluwer Academic Publishers, Norwell, MA (1996).

16. Kamba, T., Elson, S. A., Harpold, T., Stamper, T., and Sukaviriya, P. N., Using small screen space more efficiently. *Proc. of CHI'96*, ACM, New York, (1996), 383-390.

17. Lamping, J., Rao, R., and Pirolli, P., A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. *Proc. of CHI'95*, ACM, New York (1995) 401-408. URL: http://www.inxight.com/products/hw2/LibofCong.html.

18. Lindwarm, D., Rose, A., Plaisant, C., and Norman, K., Viewing personal history records: A comparison of tabular format and graphical presentation using LifeLines. *Behaviour & Information Technology* (1998, to appear).

19. Maybury, M. T., Generating summaries from event data. *International Journal of Information Processing and Management: Special Issue on Text Summarization*, (1995), 31(5): 735-751.

20. Plaisant, C., Carr, D., and Shneiderman, B., Image-browser taxonomy and guidelines for designers. *IEEE Software 12*, 2 (March 1995), 21-32.

21. Plaisant, C., Milash, B., Rose, A., Widoff, S., and Shneiderman, B., LifeLines: Visualizing personal histories. *Proc. of CHI '96,* ACM, New York, (1996), 221-227.

22. Plaisant, C. and Rose, A., Exploring LifeLines to visualize patient records. *Technical Reports CS-TR-3620,* Department of Computer Science, University of Maryland, College Park, MD (1997) URL: http://www.cs.umd.edu/hcil/Research/1997/patientrecord.html

23. Robertson, G. G., Mackinlay, J. D., and Card, S. K,. Cone trees: Animated 3d visualizations of hierarchical information. *Proc. of CHI'91,* ACM, New York (1991) 189-194.

24. Shneiderman, B., Tree visualization with tree-maps: A 2-d space-filling approach. *ACM Transactions on Graphics 11,* 1 (1992), 92-99.

25. Shneiderman, B. and Maes, P., Direct manipulation vs Interface agents. *Interactions* IV.6, ACM, New York (1997), 42-61.

26. Shneiderman, B., *Designing the User Interface,* Third Edition 1998, Addison Wesley Longman, Inc., Reading, MA

27. Tufte, E., *The Visual Display of Quantitative Information,* Graphics Press, Cheshire, CT, 1983.

28. Wagner, F. and Wolff, A., Map Labeling Heuristics: Provably good and practically useful. *Proc. of the 11th Annual ACM Symposium on Computational Geometry*, (1995), 109-118.