Screen Layout Methods for Multidimensional Visualization

Matthew O. Ward
Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609 USA

Daniel Keim
Institute for Computer Science
University of Munich
Munich, Germany D-80802

Abstract

The location of graphical entities such as data points, glyphs, or icons in a display is important to convey significant information about the relations between entities. The screen layout problem manifests itself in many areas, including data and information visualization, and many strategies have been developed in each domain. The goal of this paper is to identify and unite a variety of screen layout techniques from different disciplines, correlate techniques with implicit and explicit relations within the entities being visualized, and suggest new methods for placement and interaction with the resulting entities. We will focus on the placement of glyphs, which are graphical entities whose visual attributes are controlled by different fields of multivariate data points. However, many of the results are pertinent to the screen layout of icons and data points as well.

Entity Relationships and Screen Layout Criteria

Entities to be placed on a screen may or may not have quantifiable relations with each other. Linear orders may be derived by sorting the entities based on some attribute of the entity. Hierarchical ordering relations can be explicit, as in organizational charts or directory structures, or derived, such as by using clustering or partitioning algorithms. A relation which is perhaps more powerful than ordering is that of a distance or similarity metric, where a value exists for each pair of entities indicating the degree of similarity or difference.

A major criterion for effectively conveying information graphically is that distortion of the actual information is minimized. Thus not only should the viewer be able to infer correct relational information (semantically similar entities should be placed near each other on the screen), but false relations (entities which are near on the screen, but not semantically close) should not be presented. Another issue is the effect of occluding entities; in most situations we would like to perceive each entity individually, which dictates against occlusion. Finally, we want to make effective use of the empty space on the screen; too much empty space may be an indication that not enough space is being allocated for glyph display, while not enough empty space may make relations, especially those involving clustering, difficult to perceive.

Common Techniques

The most common strategy for glyph placement is either to divide the screen into M roughly equal sized regions and assign each glyph to a successive slot based on some ordering [5] or use two numeric attributes of the glyph to determine a position. The former technique, while making effective use of screen space, may fail to convey relationships other than linear ordering. The latter technique can display more extensive relationships, but may suffer from occlusions and poor space utilization.

Some Improved Placement Techniques

For linearly ordered entities, there are several methods to convey more information using placement. VisDB [3] uses a circular arrangement of sorted colored entities, with successive similar entities forming rings in the display (Fig. 1). Similarly, the Recursive Pattern technique [4] uses recursively defined raster sweeps of smaller and smaller regions, so that entities near each other in order are close to each other on the screen (this is related to space-filling curves). This technique may be extensible to handle hierarchically ordered entities as well. Hierarchically ordered entities can also be displayed using tree-maps [2], which partition space either vertically or horizontally in a recursive, alternating fashion (Fig. 2).

Entities with distance/similarity relationships can be placed using dimensionality reduction techniques such as multidimensional scaling [1] or principal components analysis (Fig. 3). Another strategy is to perform clustering on the entities and use either a space partitioning algorithm such as tree-maps or a spring model to force separation between clusters while encouraging bonding within clusters.

Given a placement strategy, there are many strategies one could use to improve the interpretability of the resulting entity placement. To reduce occlusion we can apply a jitter algorithm or use a spring model to force separation. To eliminate clutter, entities can be manually deleted or moved to a designated "uninteresting" area. Finally, groups of related entities can be merged into "mega-entities", which can be automatically unmerged when the user zooms into a particular region of the data space.

Summary

In this paper we've attempted to describe a broad spectrum of techniques for placing glyphs on a display, using the relationships explicit or implicit in the data to determine appropriate placement strategies. Future work will include implementing a single environment for experimenting with glyph placement and performing user studies to gauge the effectiveness of the various placement strategies and enhancement techniques.

References

[1] Bentley, C.L., and Ward, M.O., "Animating multidimensional scaling to visualize n-dimensional data sets," IEEE Information Visualization '96, pp. 72-73, October, 1996.

[2] Johnson, B. and Shneiderman, B., "Tree-maps: a space-filling approach to the visualization of hierarchical information structures, IEEE Visualization '91, pp. 189-194, October, 1991.

[3] Keim, D.A., and Kriegel, H.-P., "VisDB: database exploration using multidimensional visualization", Computer Graphics and Applications, pp.40-49, September, 1994.

[4] Keim, D.A., Kriegel, H.-P., and Ankerst, M., "Recursive pattern: a technique for visualizing very large amounts of data," IEEE Visualization '95, pp. 279-286, October, 1995.

[5] Ward, M.O., "XmdvTool: integrating multiple methods for visualizing multivariate data," IEEE Visualization '94, pp. 326-333, October, 1994.

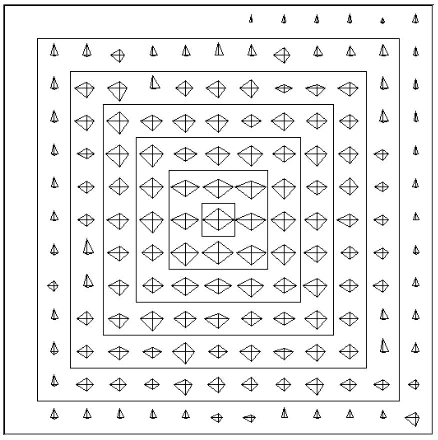

Figure 1: Circular linear ordering (Iris data set).

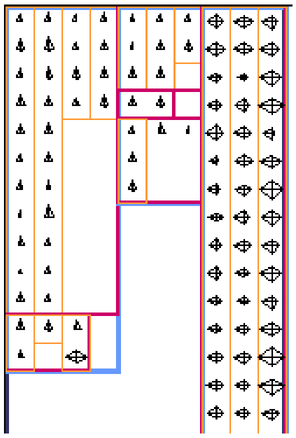

Figure 2: Recursive hierarchical placement.

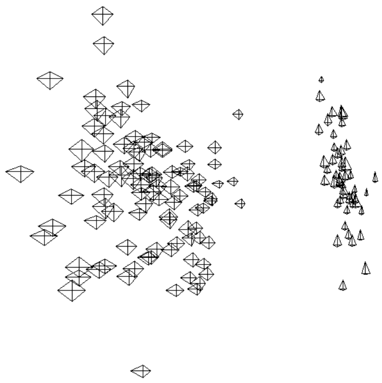

Figure 3: Position based on principal components analysis.