Navigating Hierarchies with Structure-Based Brushes

Ying-Huey Fua, Matthew O. Ward and Elke A. Rundensteiner
Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609
{yingfua,matt,rundenst}@cs.wpi.edu *

Abstract

Interactive selection is a critical component in exploratory visualization, allowing users to isolate subsets of the displayed information for highlighting, deleting, analysis, or focussed investigation. Brushing, a popular method for implementing the selection process, has traditionally been performed in either *screen space* or *data space*. In this paper, we introduce the concept of a *structure-based brush*, which can be used to perform selection in hierarchically structured data sets. Our structure-based brush allows users to navigate hierarchies by specifying focal extents and level-of-detail on a visual representation of the structure. Proximity-based coloring, which maps similar colors to data that are closely related within the structure, helps convey both structural relationships and anomalies. We describe the design and implementation of our structure-based brushing tool. We also validate its usefulness using two distinct hierarchical visualization techniques, namely hierarchical parallel coordinates and tree-maps.

Keywords: Brushing, hierarchical representation, interactive selection, exploratory data analysis.

1 Introduction

A hierarchical organization on a dataset provides a framework for viewing information at varying levels-of-detail. There have been numerous research efforts in finding effective methods to present and explore hierarchical information, such as Tree-Maps [7, 12], Cone-Trees [10] and Reconfigurable Disc Trees [6]. However, most of these methods provide only modest modes of interactions for navigating the hierarchy. In this paper, the particular way of representing the tree structure is not of concern; rather we are interested in finding ways to effectively navigate through such hierarchies. Navigation plays an important role in aiding users to find their way through the complex structure: to see where they are, what information is available and how to access information of interest, while avoiding getting lost wandering in some isolated subspace.

Recently, there has been increased interest in visual exploration of hierarchies [3, 8, 11]. Hierarchy visualizations are evident in many commercial applications, such as Microsoft Windows Explorer, Norton Commander, and so on. The major disadvantage of such interfaces is that there is a limited display space for the tree. Hence, it is easy to lose context, for instance, if some subdirectories get very large.

Often we are interested in exploring a particular subspace of interest besides having an overview of the hierarchical structure. One way of achieving this is through *brushing*. Brushing is a *direct* and *data-driven* metaphor. It is an interactive process for selecting subsets of data or localizing a subspace within an N-dimensional space [9, 13, 16]. Many useful operations, such as highlighting, deleting, masking, or aggregation, may then be performed on elements that lie within the brushed subspace. This paper presents a new technique to navigate hierarchical space by extending the brush metaphor to structure. Our *structure-based brush* serves as a mechanism to select subsets of the hierarchical structure for further analysis or drill-down/roll-up operations.

Section 2 of this paper gives an overview of the basics of brushing. Section 3 gives a description of structure-based brushes and the containment criteria. Section 4 describes the creation and manipulation of our brush. Following that in Section 5, we illustrate the usefulness and generality of our tool by applying it to two hierarchical visualization techniques, namely, hierarchical parallel coordinates [2] and tree-maps [7, 12]. We conclude by summarizing our contributions and outlining plans for future work.

2 Brush Basics

Selection is a process whereby a subset of entities on a display are isolated for further operations, such as highlighting, deleting, or analysis. Wills [15] defined a taxonomy of selection operations, classifying techniques based on whether memory of previous selections is maintained or not, whether the selection is controlled by the underlying data (we term this *data-driven*) or not, and what specific interactive tool (e.g., brushing, lassoing) is used to differentiate an area of the display. He also created a selection calculus which enumerates all possible combinations of actions between a previous selection and a new selection (replace, add, subtract, intersect, and toggle) and attempted to identify configurations of these actions which would be most useful.

Brushing is the process of interactively painting over a subregion of the data display using a mouse, stylus, or other input device that enables the specification of location attributes. It has been used as a method for performing selection in graphics for many years. The principles of brushing were first explored by Becker and Cleveland [1] and applied to high dimensional scatterplots. In this system, the user specified a rectangular region in one of the 2D scatterplot projections and based on the mode of operation, points in other views corresponding to those falling within the brush were highlighted, deleted or labelled. Ward and Martin [13, 9] extended brushing to permit brushes to have the same dimensionality as the data (N-D instead of 2-D). They also explored the concepts of mul-

*This work is supported under NSF grant IIS-9732897.

tiple brushes, composite brushes (formed by logical combinations of brushes), and *fuzzy* brushes, which allow points to be partially contained within a brush. Haslett et al. [4] introduced the ability to show the average value of the points that are currently selected by the brush.

One common method of classifying brushing techniques is by identifying in which space the selection is being performed, namely screen or data space. This can then be used to specify a *containment criteria* (whether a particular point is inside or outside the brush). In *screen space* techniques, a brush is completely specified by a 2-D contiguous subspace on the screen. In *data space* techniques, a complete specification consists of either an enumeration of the data elements contained within the brush or the N-D boundaries of a hyperbox which encapsulates the selection. In this paper, we introduce a third category, namely *structure space* techniques, which allows selection based on structural relationships between data points.

In addition, brush manipulation may be *direct* or *indirect*. Direct manipulation refers to the ability to interactively control brush creation and manipulation by mouse (or other locator) actions on the data display itself. On the other hand, indirect manipulation refers to the use of separate widgets such as sliders to specify or manipulate the brush coverage. Direct manipulation is generally preferred for data-driven operations (such as isolating an interesting subset of the display), while user-driven operations (such as a range query) are often easier to specify with indirect methods.

3 Structure-Based Brushes

The *structure* of a data set specifies relationships between data points. This structure may be explicit (e.g., categorical groupings or time-based orderings) or implicit (e.g., resulting from analytic clustering or partitioning algorithms). Examples of structures include linear orderings, hierarchies, and directed acyclic graphs. In this work, we focus on hierarchical relationships of data as it is one of the techniques that is receiving substantial attention as a means for effectively coping with large data sets.

A *hierarchy* or *tree* is a convenient mechanism for organizing large data sets. By recursively clustering or partitioning data into related groups and identifying suitable representative information (summarizations) for each cluster, we can examine the data set methodically at different levels of abstraction, moving down the hierarchy (*drill-down*) when interesting features appear in the summarizations and up the hierarchy (*roll-up*) after sufficient information has been gleaned from a particular subtree.

There has been much research into the graphical presentation of hierarchical or tree-structured information [7, 12, 10, 6]. However, most techniques for interacting with the resulting displays have focused on modifying the viewing parameters or distorting the structure to help identify features of interest. We have created a suite of techniques, which we term *structure-based brushing*, aimed at supporting the interactive exploration of large data sets that are either implicitly or explicitly clustered into a hierarchical structure.

As mentioned in Section 2, brushing requires a containment criteria. For our first containment criteria, we augment each node in the hierarchy, that is each cluster, with a monotonic value relative to its parent. This value can be for example the level number, the cluster size/population, or the volume of the cluster's extents (defined by the minimum and maximum values of the nodes in the cluster). This assigned value determines the control for the level-of-detail. By choosing a continuous control variable, such as cluster size, the traversal of the tree through different levels-of-detail can be smooth transitions instead of abrupt screen changes. In addition, this concept can be extended to allow the specification of a subset of levels-of-detail. Hence, instead of selecting a single level of detail, we might want to select multiple levels-of-detail (which we term the *level-of-detail range*). Formally, we define a level-of-detail variable L, with range l_{min} to l_{max} and a value range $[l_a, l_b]$ such that $l_{min} \leq l_a \leq l_b \leq l_{max}$.

Our second containment criteria for structure-based brushing is based on the fact that each node in a tree has extents, denoted by the left- and right-most leaf nodes originating from the node. In addition, it is always possible to draw a vertically oriented tree in such a way that the horizontal position of each node (and, in fact, all of its children) falls between its extents. These extents ensure that a selected subspace is contiguous in structure space. Formally, we define an extents variable E, with range e_{min} to e_{max} and an extent range $[e_c, e_d]$ such that $e_{min} \leq e_c \leq e_d \leq e_{max}$.

Thus a structure-based brush is defined by a subrange of the structure extents and level-of-detail variables, namely $[l_a, l_b, e_c, e_d]$.

4 Creation and Manipulation of Structure-Based Brush

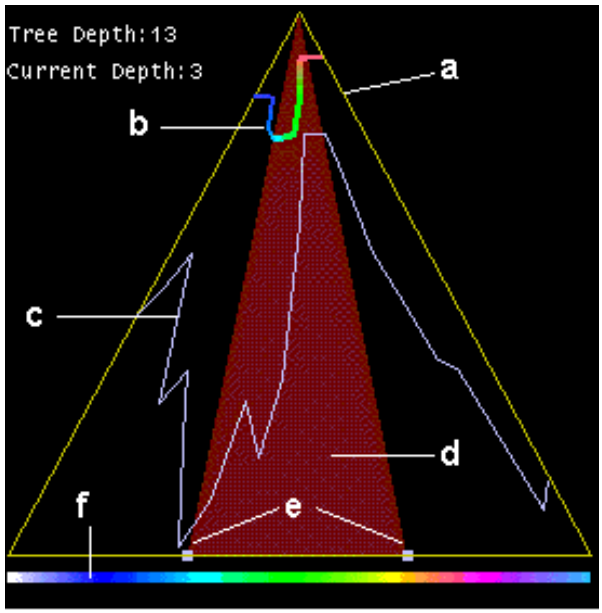

Figure 1: Structure-based brushing tool. (a) Hierarchical tree frame; (b) Contour corresponding to current level-of-detail; (c) Leaf contour approximates shape of hierarchical tree; (d) Structure-based brush; (e) Interactive brush handles; (f) Colormap legend for level-of-detail contour.

Figure 1 shows our *structure-based brushing interface*. The triangular frame depicts the hierarchical tree (see (a)). The leaf contour (see (c)) depicts the silhouette of the hierarchical tree. It delineates the approximate shape formed by chaining the leaf nodes. The colored bold contour (see (b)) across the tree delineates the tree cut that represents the cluster partition corresponding to the specified level-of-detail. We use a *proximity-based coloring* scheme in assigning colors to the partition nodes [2]. In this scheme, a linear order is imposed on the data clusters gathered for display at a given level-of-detail. This linear order is directly derived from the order in which the tree is traversed when gathering the relevant nodes for a given level-of-detail. In our implementation, we adopt the in-order tree traversal. Colors are then assigned to each cluster by looking up a linear colormap table (see (f)). The same colors are used for the display of the nodes in the corresponding data display. The two movable handles (see (e)) on the base of the triangle,

together with the apex of the triangle, form a wedge in the hierarchical space (see (d)).

Our structure-based brushing tool supports both direct and indirect manipulation. Sets of elements may be directly selected by positioning the wedge handles so as to bound the range of colors spanned by the elements. This is made possible due to the direct color correspondence between the data display and the structure display. Moreover, similar elements are selected as a group, since by our coloring criteria, similar elements are drawn in similar colors. The wedge handles can be adjusted at either end or simply translated to bound the desired set of elements. Indirect manipulation is provided through the use of sliders for the range of extents and values, in case the user prefers such mode of interaction.

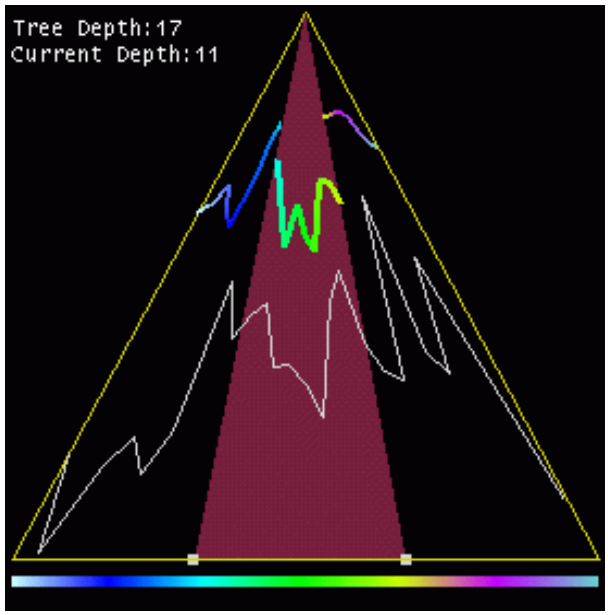

Figure 2: Structure-based brushing at two different levels-of-detail.

In a hierarchical organization, drill-down and roll-up operations are commonly used to explore the hierarchy. Our tool supports a global drill-down and roll-up operation, that is, the current level-of-detail can be adjusted by dragging the colored contour vertically. The data display changes to reflect more details when the contour is adjusted vertically downwards while showing more and more abstract views of the data when the contour is adjusted vertically upwards.

Besides a global drill-down/roll-up operation, our tool also allows the user independent control of the level-of-detail of the brushed and unbrushed region. That is, the colored contour in the brushed region can be adjusted independently of the contour segments outside the brushed region, and vice versa (See example in Figure 2). We term this *selective drill-down/roll-up*. This separate mode of control gives the user the flexibility to view the hierarchical structure at two different levels-of-detail at the same time.

It is important to contrast structure-based brushing with traditional data-based brushing. In a traditional user-driven brushing operation, to specify a region of interest in a multivariate data display, the user sets upper and lower bounds for each dimension. In data-driven brushing, the user paints over groupings of interesting data. Neither of these approaches is suitable for isolating data elements which are *structurally* related. Rather, their focus is on the *values* of the data. Clearly, structure-based brushing provides new, and potentially invaluable, functionality beyond data-based brushing.

5 Case Studies

We illustrate the usefulness and general applicability of our tool by applying it to two hierarchical visualization techniques: hierarchical parallel coordinates [2] and tree-maps [7]. These case studies demonstrate the functionalities of our new brush, its usefulness, and difference from alternative techniques.

We use a 5-dimensional 16,000 element dataset formed by combining SPOT, magnetics, and radiometrics (three channels) remote sensing datasets from the Grant's Patch region of Western Australia. The hierarchical clustering was achieved by processing the data with the BIRCH algorithm [17], which can handle large scale data sets efficiently.

5.1 Interacting with Hierarchical Parallel Coordinates using Structure-Based Brushes

Parallel coordinates is a multivariate visualization technique pioneered in the 1980's which has been applied to a diverse set of problems [5, 14]. In this technique, each data dimension is represented as a (horizontal or) vertical axis, and the N axes are organized as uniformly spaced lines. A data element in an N-dimensional space is mapped to a polyline that traverses across all of the axes, intersecting each axis at a position proportional to its value for that dimension.

Hierarchical parallel coordinates [2] is a new extension that we have developed for visualizing large data sets on parallel coordinates. In hierarchical parallel coordinates, the data is structured as a hierarchy of clusters, and the display shows summarizations of the clusters at a certain level of detail. Many display options exist, including showing cluster centers (which look identical to traditional parallel coordinate displays), extents (which manifest themselves as variable width bands encasing the centers), population (mapping to opacity), and other cluster statistics. Distortion techniques, structure-based coloring, and selective fade-in/fade-out are available to help reduce clutter and expose structure.

Figure 3 shows the display of hierarchical parallel coordinates at a given level-of-detail. Each polyline across the axes displays the mean value of its cluster. The number of polylines spanning across the screen corresponds to the number of clusters at the given level-of-detail. The lines in the data display are painted with the corresponding color of the structure-based display with the color red reserved as a highlighting color. With our brushing tool, the user simply adjusts the handles at the base of the triangle wedge to bound the extents of interest. The selected clusters are drawn in bold red, indicating they are being brushed. Next, we demonstrate the usefulness of the selective drill-down/roll-up operations.

Figures 4 and 5 show two images of hierarchical parallel coordinates at different levels-of-detail. Figure 4 displays the initial state, with all clusters at the same level-of-detail. The user can then brush the cluster(s) of interest by adjusting the handles at the bottom of the wedge on the structure-based brush interface. Next, by "pulling" the brushed contour vertically downwards, we can view the selected clusters at a higher level-of-detail while maintaining the same level-of-detail for the unselected clusters. This results in the display shown in Figure 5. We have turned off the red encoding of the brushed clusters to convey the actual colors of the clusters that correspond to the colored contour. The usefulness of the selective drill-down/roll-up feature is evident here – users have the flexibility to see an isolated view or to manipulate the region of interest while minimizing the distraction from data lines not falling in that region.

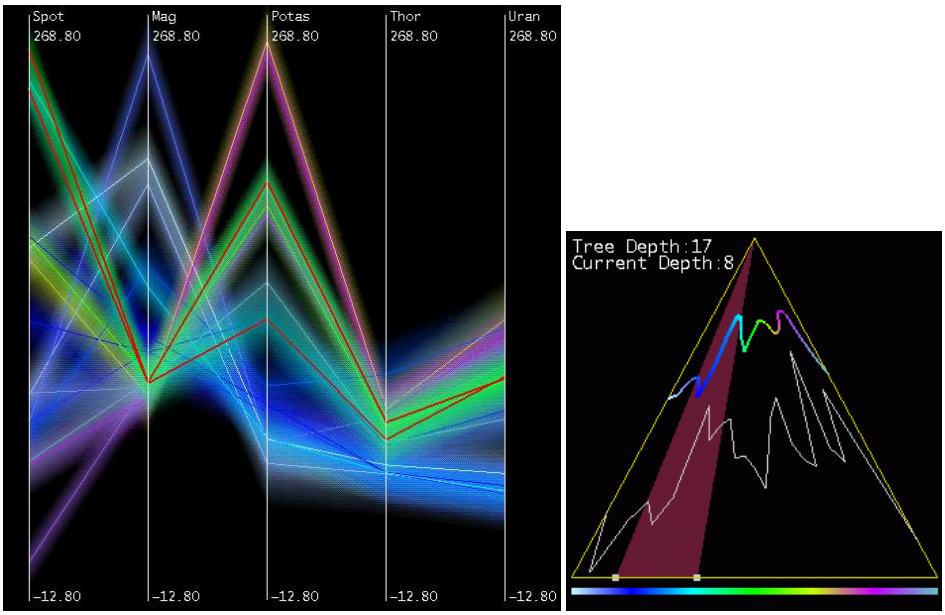

Figure 3: A hierarchical parallel coordinates display of a remote sensing dataset with the selected cluster painted in bold red to reflect that it is currently being brushed in the structure-based tool. The image on the right shows the corresponding level-of-detail indicated by the colored contour in the structure-based brush with the brushed region indicated by the wedge. In this case, we observe that the selected clusters share the same mean value for magnetics and uranium contents, and have high SPOT contents. (See Color Plates).

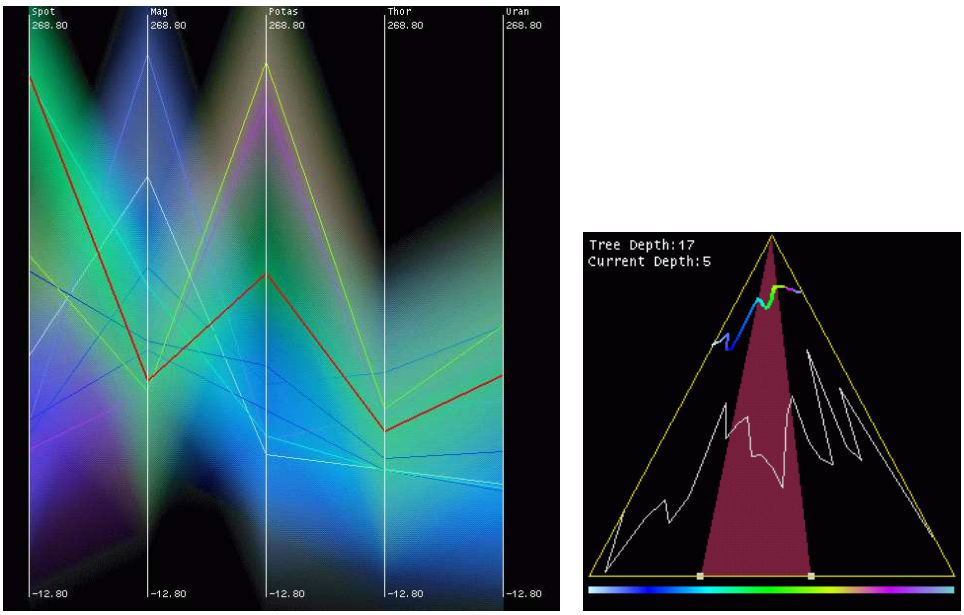

Figure 4: A hierarchical parallel coordinates display of a remote sensing data with the selected and unselected clusters at the same level-of-detail. Notice that the selected cluster which is drawn in bold red has relatively low mean levels of magnetics and thorium contents. The colored contour in the structure-based brush indicates the current level-of-detail. (See Color Plates).

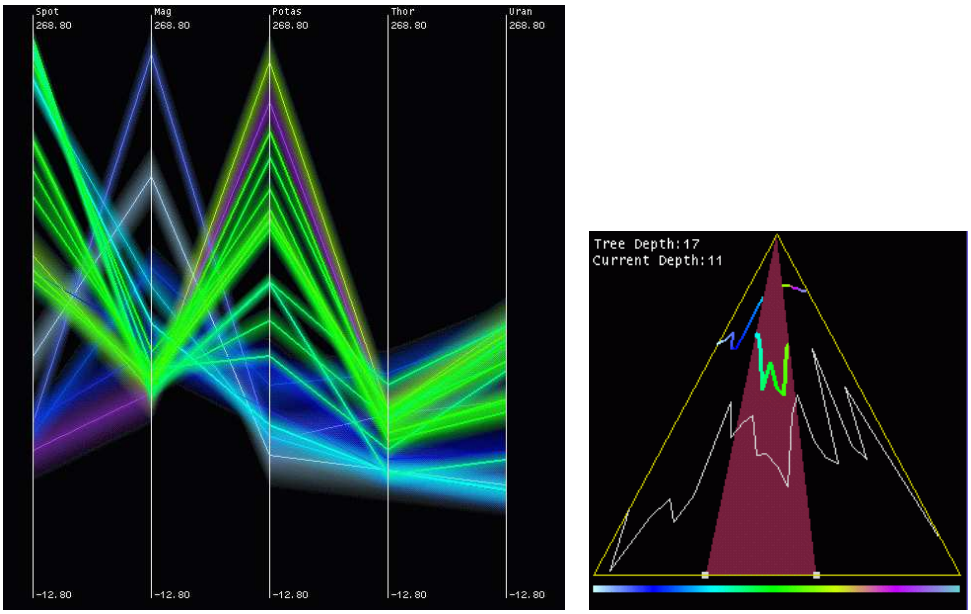

Figure 5: A hierarchical parallel coordinates display of a remote sensing dataset with the selected cluster drawn at a higher level-of-detail compared to the unselected clusters. The left image shows the effect on the selected cluster (indicated by the bold red lines in Figure 4) when it expands to show more detail. In this case, we display the original colors of the selected lines rather than painting them bold red in order to reveal the actual colors encoded for the clusters. Moreover, in order to display the lines clearer, we reduce the bands around the lines via extent scaling [2]. These clusters exhibit trends similar to their parent cluster, that is having relatively low mean levels of magnetics and thorium contents. The corresponding levels-of-detail are indicated by the structure-based brush on the right. (See Color Plates).

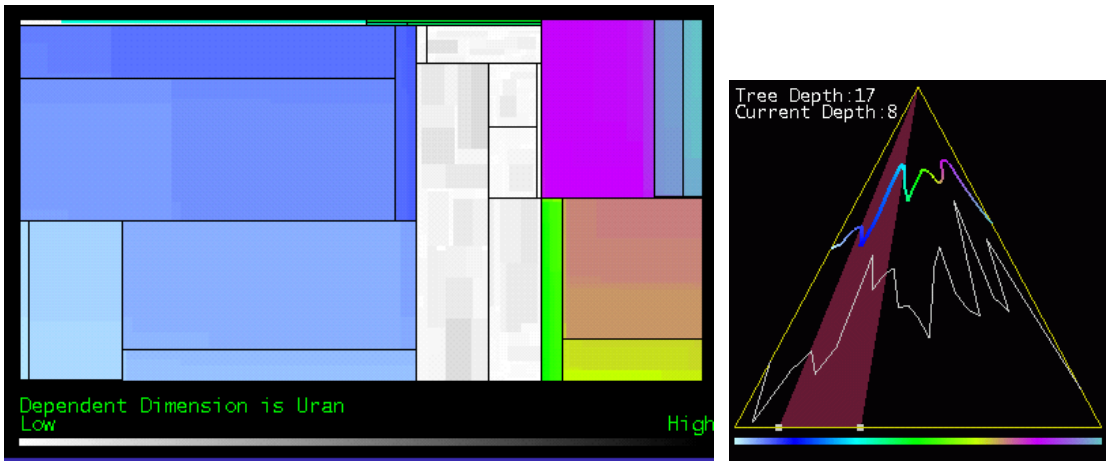

Figure 6: A tree-map display of the remote sensing dataset with the selected clusters painted with the color of the dependent variable, uranium. By mapping the value of the dependent variable to a greyscale colormap where high values are mapped to darker colors, we observe that the selected clusters have relatively low mean levels of uranium content.

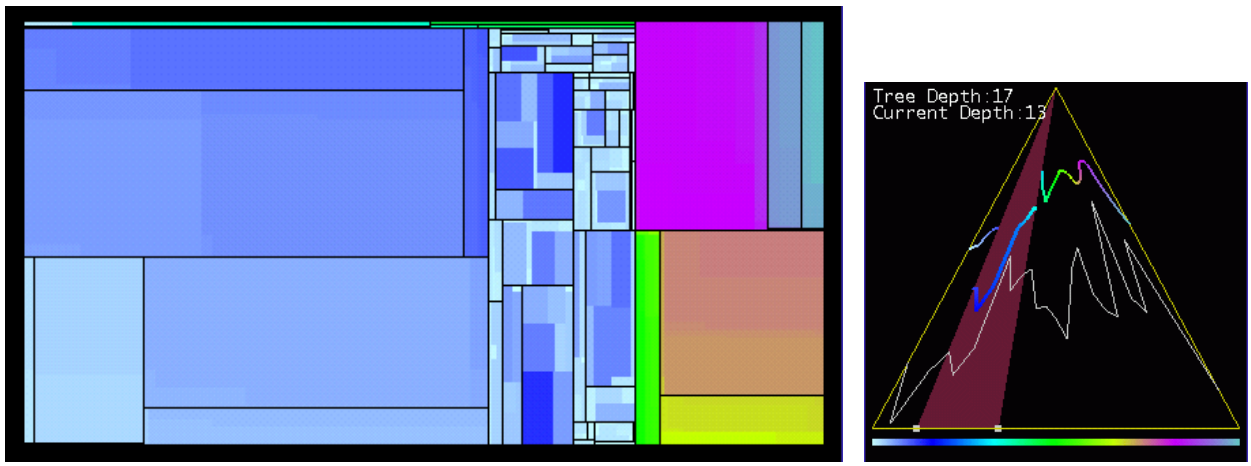

Figure 7: A tree-map display of a remote sensing dataset with the selected clusters at a higher level-of-detail compared to the unselected clusters. In this case, we color the selected rectangular regions with the corresponding color on the structure display. The structure-based brush is shown on the right. (See Color Plates).

5.2 Interacting with Tree-Maps using Structure-Based Brushes

A tree-map [7, 12] is a space-filling method for presenting hierarchical, univariate data. It is formed by taking a rectangular display area and recursively subdividing it based on the tree structure, alternating between horizontal and vertical subdivision, and filling the terminal rectangular regions with a color based on the dependent variable. In our modified version of the tree-map, we can choose to fill the color of the rectangular region based on a dependent variable or with its corresponding color on the structure display (structure-based coloring).

Figure 6 shows the display of a tree-map at a given level-of-detail as indicated by the colored contour in the structure-based brush. As in the hierarchical parallel coordinates display (Section 5.1), the clusters of interest can be selected by bounding the corresponding color on the structure-based brush interface. The color of the selected clusters on the tree-map then change to reflect the value of the dependent variable of the tree-map.

Next, the user can view the selected cluster on the tree-map at a higher level of resolution by "pulling down" the brushed segment of the colored contour in the structure-based brush. The resulting state of the structure-based brush and the cluster display are shown in Figure 7. In this case, we choose to paint the nodes using structure-based coloring. We can observe the relative sizes of each subcluster from further subdivisions of the rectangle. Since all these observations are isolated from the unselected clusters, it gives the user an uncluttered view of the regions of interest. To make similar observations for other clusters, the users can simply translate the wedge if they desire the same brush size, or adjust the handles at the corners of the wedge to define a totally new brush.

6 Conclusion and Future Work

This paper presents a new technique for navigating hierarchies, called structure-based brushing, that is an extension of the data-based brush metaphor. With structure-based brushing, it is possible to select a subset of a hierarchy and explore the selected space in varying levels-of-detail using the drill-down/roll-up operations pair. In addition, we demonstrate the usefulness of our tool with case studies on two hierarchical visualization techniques, namely hierarchical parallel coordinates and tree-maps. Examples of its use can be found at http://davis.wpi.edu/∼xmdv.

Our current structure-based brushing tool has a few limitations. First, the extent-based subranging assumes that the order of the branches is fixed. With a different order of the clusters, the color assignment will be different, and hence the selection. Second, for very wide hierarchies it might be difficult to select narrow substructures. Third, with our coloring strategy, adjacent clusters may be assigned indistinguishable colors if the number of clusters is very large.

Our future work will be aimed at addressing these limitations, as well as to perform extensive testing of the tool over different domains and with other hierarchical multiresolution visualization techniques. In particular, we are interested in using zooming/distortion techniques in both structure space and color space to facilitate precise operations in dense structures. We are also planning to investigate methods for dynamically reordering cluster branches (when ordering isn't data-driven) to more easily enable the comparison of multiple, isolated branches. While this could be accomplished using multiple composite brushes [9], dynamic reorganization may lead to simpler exploratory interactions.

References

[1] A. Becker and S. Cleveland. Brushing scatterplots. *Technometrics, Vol 29(2), p. 127-142, 1987*, 1987.

[2] Y. Fua, M. Ward, and E. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. *Proc. of Visualization '99*, Oct. 1999.

[3] G. Furnas. Generalized fisheye views. *Proc. of Computer-Human Interaction '86, p. 16-23*, 1986.

[4] J. Haslett, R. Bradley, P. Craig, A. Unwin, and G. Wills. Dynamic graphics for exploring spatial data with application to locating global and local anomalies. *Statistical Computing 45(3), p. 234-42, 1991*, 1991.

[5] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multidimensional geometry. *Proc. of Visualization '90, p. 361-78*, 1990.

[6] C. Jeong and A. Pang. Reconfigurable disc trees for visualizing large hierarchical information space. *Proc. of Information Visualization '98, p. 19-25*, 1998.

[7] B. Johnson and B. Shneiderman. Tree maps: A space-filling approach to the visualization of hierarchical information structures. *Proc. of Visualization '91, p.284-91*, 1991.

[8] Y. Leung and M. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction Vol. 1(2), June 1994, p. 126-160*, 1994.

[9] A. Martin and M. Ward. High dimensional brushing for interactive exploration of multivariate data. *Proc. of Visualization '95, p. 271-8*, 1995.

[10] G. Robertson, J. Mackinlay, and S. Card. Cone trees: Animated 3d visualization of hierarchical information. *Proc. of Computer-Human Interaction '91, p. 189-194*, 1991.

[11] D. Schaffer, Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, and M. Roseman. Navigating hierarchically clustered networks through fisheye and full-zoom methods. *ACM Transactions on Computer-Human Interaction, Vol. 3(2), p. 162-88*, 1996.

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

[13] M. Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. *Proc. of Visualization '94, p. 326-33*, 1994.

[14] E. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association, Vol. 411(85), p. 664*, 1990.

[15] G. Wills. Selection:524,288 ways to say this is interesting. *Proc. of Information Visualization '96, p. 54-9*, 1996.

[16] P. Wong and R. Bergeron. Multiresolution multidimensional wavelet brushing. *Proc. of Visualization '96, p. 141-8*, 1996.

[17] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: an efficient data clustering method for very large databases. *SIGMOD Record, vol.25(2), p. 103-14*, June 1996.