Information visualization using a new Focus+Context Technique in combination with dynamic clustering of information space

Matthias Kreuseler
University of Rostock
Dept. of Computer Science
D-18051 Rostock
++49 (381) 498-3417
mkreusel@informatik.uni-rostock.de

Heidrun Schumann
University of Rostock
Dept. of Computer Science
D-18051 Rostock
++49 (381) 498-3421
schumann@informatik.uni-rostock.de

ABSTRACT

This paper presents work in progress on our approach to visualizing multi-dimensional and hierarchical information. We propose two new graphical user interfaces, the Magic Eye View (MEV) and ShapeVis to explore information spaces. In order to cope with large information sets we combine MEV and ShapeVis with dynamic hierarchical clustering of information units.

The Magic Eye View, which implements a new Focus+Context technique, is used as the interface for visualizing those hierarchies. In order to support detailed exploration of the information space (e.g. analysis of certain cluster nodes or hierarchy levels) a new technique for visualizing multidimensional information is used. ShapeVis provides 2D or 3D representations of the information objects according to the selected subset of the information space. Objects are represented as small closed free-form-surfaces. The location, size and shape of these surfaces describe the original objects in the information space uniquely according to their properties.

Keywords

information visualization, multidimensional information space, hierarchy, clustering, spring model

1. INTRODUCTION

Visual exploration of complex information spaces has become one of the "hot topics" in scientific visualization. Over the last few years many techniques have been developed for visualizing different types of information. Among these are techniques for visualizing and interacting with hierarchies like Cone Trees [6] or Disc Trees [8] which use horizontal and vertical cones or discs to layout the hierarchies.

FSN [15] and Information Pyramids [1] exploit the metaphor of 3D information landscapes to depict large hierarchies. Other approaches such as Treemaps [9] and Cheops [4] are well known 2D techniques which use available screen space very effectively.

Several techniques have been developed for visualizing multi-dimensional information. These methods try to map correlations of objects in a higher dimensional information space to spatial correlations in a (2D or 3D) presentation space. Among these are approaches like KOAN [12], VR-VIBE [5] and Narcissus [7] which exploit spring models to place objects according to their similarities, whereby similar objects are placed spatially close together.

Despite many existing techniques, exploring information collections becomes increasingly difficult as the volume of information grows. Major problems arise due to the limited screen space and the applicability of existing techniques to only one particular class of information.

Thus the analysis of heterogeneous information spaces requires the combination of different visualization techniques with a suitable preprocessing of information quantities. One of the objectives of preprocessing is to extract relevant subsets in order to reduce the active data size to processible levels. With respect to this, hierarchies provide a valuable mechanism for organizing data, even if the given data set is not a "natural" hierarchy. The calculation and visualization of those structured abstractions helps to reveal patterns and structures in the data.

In this paper we will present our approach to visualizing multi-dimensional and hierarchical information. We propose the hierarchical clustering of a multidimensional information space so as to gain an overview of large volumes of quantitative continuous data. We describe a visualization system which uses a combination of hierarchy presentation in conjunction with a new Focus+Context technique and multi-dimensional visualization of particular subsets of those hierarchies for exploring multi-dimensional information at arbitrary levels of detail.

2. DYNAMIC HIERARCHY COMPUTATION

The dynamic hierarchy computation is one possible method to achieve predictable representations of given data. If an abstraction is used to organize unstructured data, it is important to remember that users may have different requirements when merging objects into groups. It might be sufficient, for instance, to merge all birds into a single group. However, a more detailed analysis may require further distinctions between singing birds, birds of prey or water birds. Thus we do not compute a fixed number of static groups. Instead, a nested sequence of groups is determined and organized into a hierarchy, whereby the requirements according to the homogeneity[1] of those groups increase as the hierarchy is descended.

In order to support the analysis of data at arbitrary levels of detail the computation of the hierarchy can be controlled interactively. An overview is provided by calculating hierarchies with only a few levels. These hierarchies can be refined for further investigations in order to reveal more subtle patterns and to identify smaller sub-clusters in the data.

The hierarchy computation is carried out by adapted agglomerative hierarchical clustering algorithms, whereby objects are merged into groups according to their similarities in the information space. We use the Euclidean distance for determining object similarities. This measure is easy to handle from a theoretical point of view, and – at least in some domains – also provides sufficient quality for practical purposes [2].

It's our objective to generate dynamic hierarchies under different aspects from the same information set. Therefore, we need a basis which can be used effectively for the dynamic refinement of the hierarchy. This basis is provided by a binary dendrogram (cf. Figure 1).

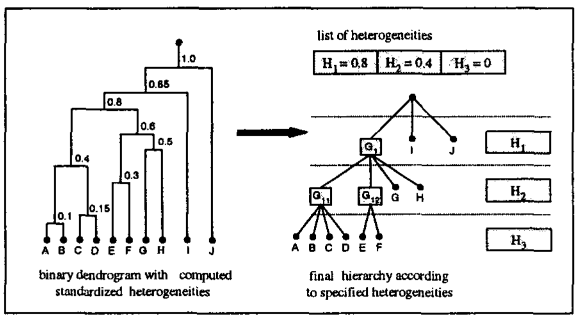

Figure 1: Construction of a Hierarchy with 3 levels based on the binary dendrogram

The binary dendrogram (cf. Figure 1) is build up based on the calculated object similarities by using one of the hierarchical clustering algorithms Single Linkage, Complete Linkage or Ward [3][10]. The values at the dendrogram nodes (cf. Figure 1) denote standardized heterogeneity values of the belonging groups.

In order to control the hierarchy computation the number of desired levels, and a heterogeneity threshold for each level, can be specified interactively. In a second pas the hierarchy is derived from the binary dendrogram according to these parameters by the following algorithm:

1) Create the root node of the final hierarchy tree (RHT) according to the dendrograms root node (RD).
2) Test if the heterogeneity of RD's children (max. 2) are less then the first (current) element in the heterogeneity list.
   2.1) If not, proceed with the node's children at step 2.
   2.2) If yes, i.e. the heterogeneity of a child node in the binary dendrogram is less than the current value in the list, insert this node into the final hierarchy. The belonging dendrogram's node's position of the inserted node is stored.
3) All new inserted nodes form new sub-trees within the final hierarchy. Execute step 1-2 for all those stored nodes with the next value in the heterogeneity list.
4) Iterate step 1-3 until the heterogeneity list is processed completely.

The final hierarchy tree contains information objects at its leaves. The remaining nodes represent clusters of multidimensional information objects which fulfil the specified heterogeneity conditions. Figure 2 and 3 illustrate the dynamic refinement of the computed hierarchy. We started from an overview in figure 2 with 3 levels. As the number of levels is increased to 6 the marked cluster in Figure 2 is split up into smaller sub-clusters in Figure 3. Thus a stepwise exploration at arbitrary levels of detail is supported.

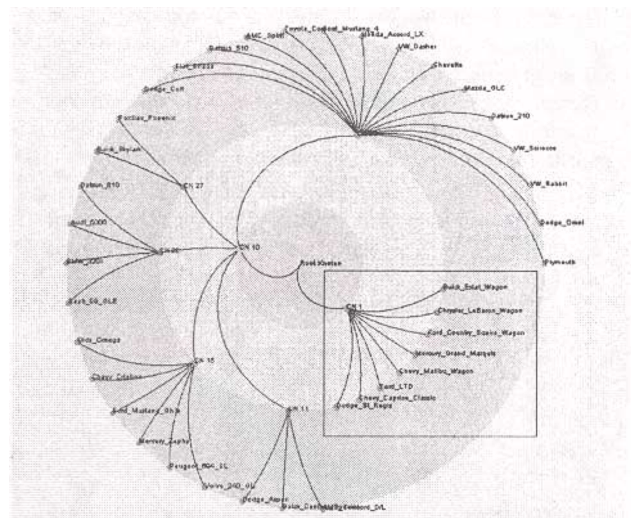

Figure 2: Overview with 3 hierarchy levels

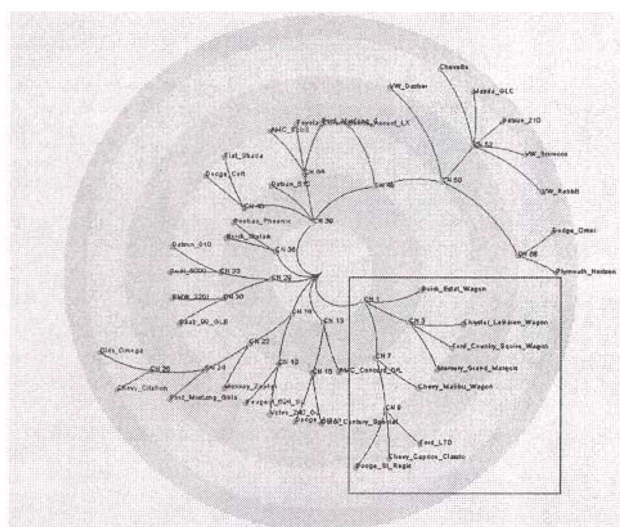

Figure 3: Hierarchy refinement with 6 levels

[1] Homogeneity denotes the average similarity between objects of an object set.

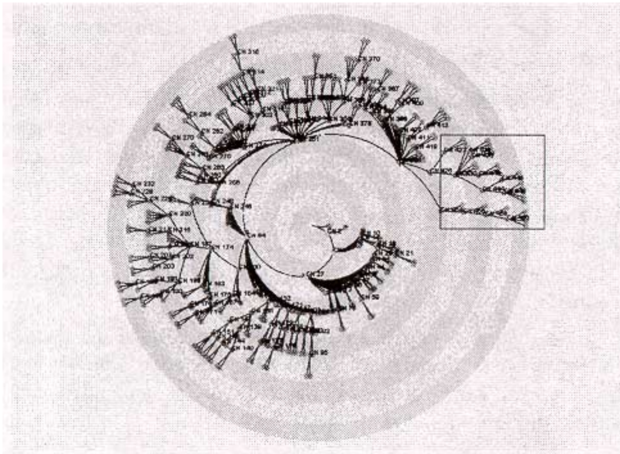

Figure 5: Complex hierarchy graph with P_0 at the origin

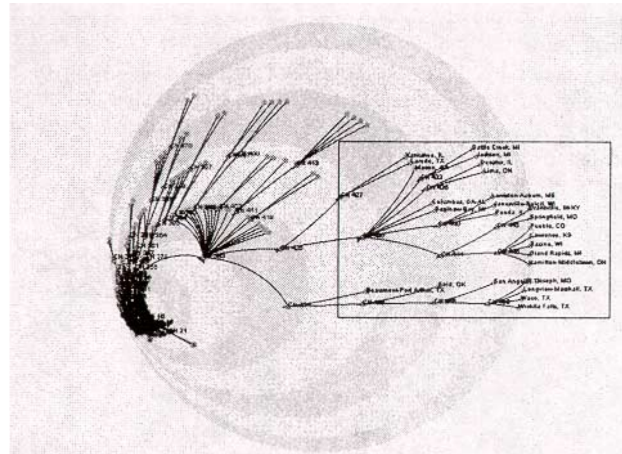

Figure 6: Hierarchy graph with enlarged focus region

3. HIERARCHY VISUALIZATION

Visualizing the computed hierarchies becomes complicated as the number of levels and nodes increases. Standard 2-d hierarchy browsers can typically display about 100 nodes [11]. Exceeding this number makes perceiving details difficult. Zooming and panning do not provide a satisfying solution to this drawback due to loss of context information. In order to solve these problems several Focus+Context techniques have been developed, e.g. Graphical Fisheye Views [14] or the Hyperbolic Browser [11] which exploit distortion to enlarge a focus area while preserving context information.

In order to achieve an additional degree of freedom for focussing arbitrary areas of the hierarchy graph, we implemented the new Focus+Context technique Magic Eye View. Our approach maps a hierarchy graph onto the surface of a hemisphere. We then apply a projection in order to change the focus area interactively by moving the center of projection.

3.1 Graph mapping onto the hemisphere

Laying out the hierarchy tree is done with a simple 2-d algorithm which is similar to the algorithm of Reingold and Tilford [13]. Thus we determine *(x,y)*-coordinates for each node of the hierarchy within a Cartesian coordinate system. The graph is then mapped onto the surface of a hemisphere. Each point on a sphere can be described uniquely by two angles *(• , •)*. Thus the determined Cartesian coordinates can be mapped directly to spherical coordinates.

3.2 Change of Focus

The objective of change of focus is to enlarge those parts of the graph which are in or near the focus region while the size of the remaining part is reduced. We implemented a projection in order to achieve this and to enable a smooth transition between the focus and context region.

Therefore we compute a ray S_i from the center of projection which is initially located at the origin $P_0=(0,0,0)$ through each of the n nodal-points P_i (cf. Figure 4a), i.e. the directions of these rays are determined by the nodes' initial positions which were ascertained by the layout algorithm. In order to change focus the center of projection P_0 can be moved arbitrarily, whereby the directions of the rays S_i are retained (cf. Figure 4 b,c).

New positions of the graph's nodes are obtained by computing the new intersection points of the rays S_i with the hemisphere. Thus the distances between nodes are increased or decreased depending on the position of P_0. By increasing the distance between nodes in the focus area we obtain more space to view the details while maintaining context information. As well as moving P_0 along the *X,Y,Z*-axis, the hemisphere can also be rotated.

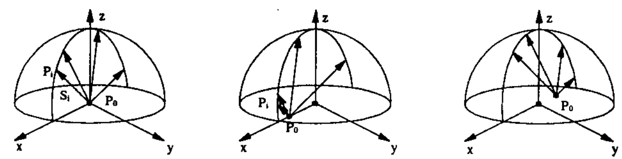

Figure 4a-c: Projection rays before and after moving P_0

Figures 5 and 6 demonstrate change of focus. Figure 5 shows a complex hierarchy graph mapped onto a hemisphere. The center of projection has been moved in Figure 6 in order to set the focus to the marked sub-graph

4. COMBINATION OF HIERARCHY AND MULTI-DIMENSIONAL VISUALIZATION

Computing hierarchies is a valid method for structuring multi-dimensional data and identifying subsets of similar objects. However, for further analysis of those subsets e.g. revealing attribute values of the data or determining object similarities within a cluster or at certain hierarchy levels we developed ShapeVis as a new technique for visualizing multi-dimensional information

4.1 ShapeVis

ShapeVis exploits an enhanced spring model [16] in order to arrange objects according to their similarities. Therefore we place n-points D_i, i.e. one point for each dimension of the data set, in an equidistant way on a sphere. Small closed free-form-surfaces (shapes) are used to depict information objects. Those shapes are attached with springs to each of the dimension points D_i. The locations of the shapes are determined by the spring model, i.e. the bigger the data value of a certain dimension the closer the shape moves towards that dimension point D_i. The shapes can be deformed in the direction of the dimension points D_i in order to depict attribute values and to solve ambiguities. The size of the deformation in a particular direction denotes the data value of that dimension. Thus multi-dimensional information objects are described uniquely by location, size and shape of their visual representations.

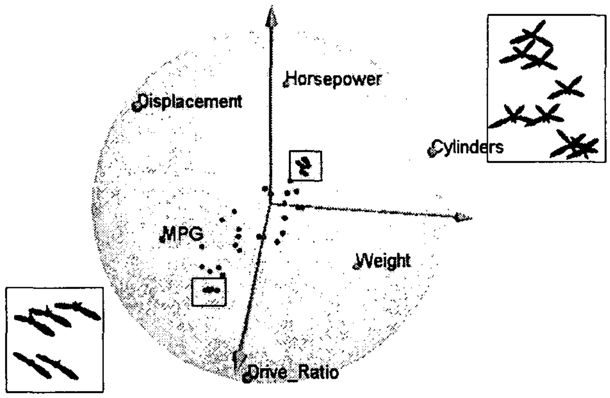

Figure 7: Visualization with ShapeVis

Figure 7 illustrates this principle. ShapeVis is applied to a car data set[2] with 6 dimensions. The objects in the cluster lower left have big values with respect to the dimensions MPG and Drive Ratio, whereas objects in the upper right cluster have large values with respect to the remaining dimensions. The small pictures left and right illustrate deformation.

4.2 Exploring clusters and levels

Arbitrary subsets of the hierarchy tree can be selected for further exploration.

- *Selection of cluster nodes* : We use color to distinguish between cluster nodes and object nodes within the hierarchy tree, whereby the size of a cluster, i.e. the number of objects is mapped to the cluster node's color. All objects of a selected cluster are visualized with ShapeVis in a separate display area.
- *Selection of hierarchy levels* : A representative is determined for each cluster node which resides at the selected level by calculating mean values of the data of all cluster members. ShapeVis is used to visualize those representatives and all remaining objects at the selected level.

Figure 8 a-c illustrate the combination of ShapeVis and Magic Eye View. The left picture shows a data set[3] of 329 American cities with ShapeVis. Exploration of single Shapes is complicated due to the dense object cloud. Reducing the size of the objects and zooming into the cluster is possible with ShapeVis but makes analysis difficult due to the vanishing dimension points. In this case it's more meaningful to preprocess the data as introduced in section 2 in order to form manageable subsets.

Figure 8b depicts the hierarchical representation of the data set. The graph has been enlarged around the root node. The 4 cities at the first level, which do not belong to a cluster, can be detected as outliers in Figure 8a as well.

Figure 8c shows a detail view of the first hierarchy level, i.e. level 1 has been selected for further exploration. The relations between the 4 cities and the 4 clusters, which represent the remaining American cities, are revealed by means of location, shape and size of their visual representations.

5. CONCLUSIONS AND FUTURE WORK

We provide an effective method for analyzing large amounts of multi-dimensional data at arbitrary levels of detail by combining dynamic hierarchy calculation with multi-dimensional visualization.

Evaluation of this techniques needs to be performed to determine their effectiveness and how to improve the introduced components. In order to apply our system in different application domains we will enhance the hierarchy calculation. In particular, the choice of appropriate similarity measures according to the domain needs to be investigated.

Further work will be done to enhance both of the graphical interfaces. Adaptive labeling of the hierarchy tree depending on the current focus area is desirable to avoid visual clutter through overlap of object labels. The 3D arrangement problem of dimension points in ShapeVis will also be studied.

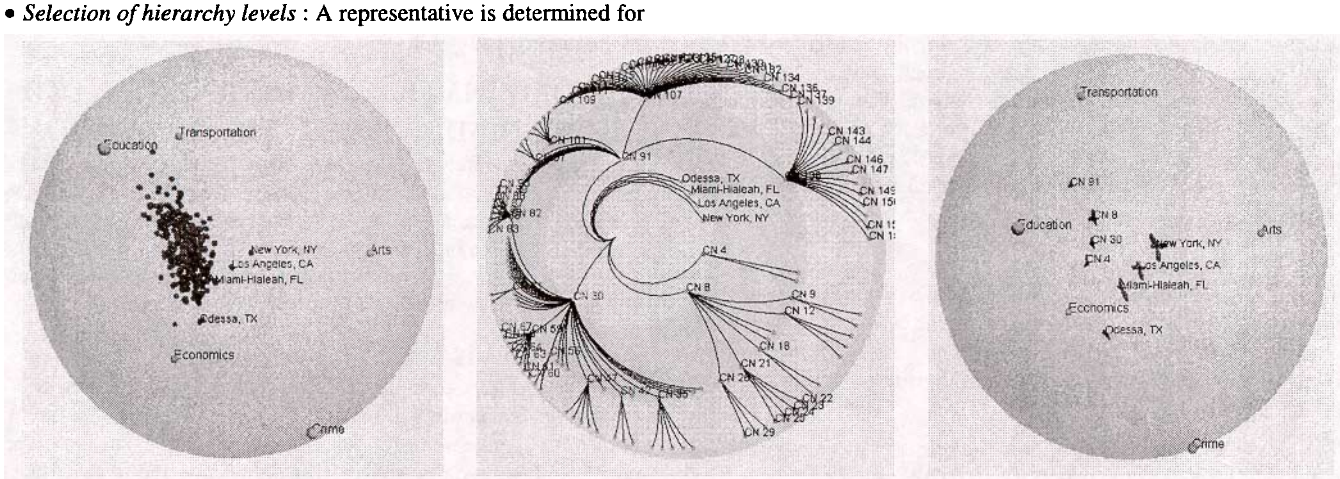

Figure 8a-c: Combination of hierarchy visualization and ShapeVis

[2] http://lib.stat.cmu.edu/DASL/Stories/ClusteringCars.html

[3] http://lib.stat.cmu.edu/datasets/places.data

6. ACKNOWLEDGMENTS

The author wish to thank Prof. Heidrun Schumann from Rostock University and Prof. Schütt from Siemens AG Munich for their constant support. Thanks also to Thomas Bürger and Thomas Nocke for developing parts of the source code. Special thanks to Jason Wood for proof reading our English.

7. REFERENCES

[1] K. Andrews, J.Wolte, M.Pichler. Information Pyramids: A new approach to Visualizing Large Hierarchies. In Proceedings LBHT IEEE Visualization 97, pp 49-52, Phoenix, 1997.

[2] M. Ankerst, S. Berchtold, D.A. Keim. Similarity Clustering for an Enhanced Visualization of Multidimensional Data. In Proceedings of Information visualization, 1998.

[3] K. Backhaus, B. Erichson, W. Plinke, R. Weiber. Multivariate Analysemethoden - Eine anwendungsorientierte Einführung. Springer Verlag , 1996, S. 261-321 (in German)

[4] L. Beaudoin, M-A Parent, L.C. Vroomen. Cheops: A compact explorer for complex hierarchies. In Proc. Visualization '96, pp. 62-63, San Francisco, October 1996

[5] S. Benford, D. Snowdon, C. Greenhalgh, R. Ingram, I. Knox and C. Brown. VR-VIBE: A Virtual Environment for Co-operative Information Retrieval. Proceedings Eurographics 95, Blakcwell Publishers, 1995

[6] S. K. Card, G. G. Robertson, J.D. Mackinlay. The Information Visualizer, An Information Workspace. In Proceedings of the ACM SIGCHI'91 Conference on Human Factors in Computing Systems, (New Orleans, LA, April 1991), pp. 181-188

[7] R. J. Hendley, N. S. Drew, A. M. Wood, R. Beale. Narcissus: Visualizing Information. Proceedings of Information Visualization 95 Symposium Atlanta, pp. 90-96, IEEE, 1995.

[8] C. Jeong, A. Pang. Reconfigurable disc trees for visualizing large hierarchical information space. In Proceedings of Information visualization, 1998, pp. 19-25, 149.

[9] B. Johnson, B. Shneiderman. Tree-maps: A Space Filling Approach to the Visualization of Hierachical Information Struc-tures. Proc. IEEE Visualization'91 (San Diego, CA 1991), pp. 284-291.

[10] L. Kaufman, P.J. Roussew. Finding Groups in Data - An Introduction to Cluster Analysis. A Wiley-Science Publication John Wiley & Sons, Inc., 1990, pp 47-48.

[11] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for viewing large hierarchies. Proc. CHI'95, pp 401-408, Denver, May 1995. ACM.

[12] J. Panyr, U. Preiser, T. Führing. Kontextuelle Visualisierung von Informationen. Proc. "19. Oberhofer Kolloquium über Information und Dokumentation", Oberhof, pp. 217-228, 1996 (in German)

[13] E.M. Reingold, J.S. Tilford. Tidier drawing of trees. In IEEE Transaction on Software Engineering, 7(2), pp.223-228, (1981)

[14] M. Sarkar and M. H. Brown. Graphical fisheye views. Communications of the ACM, 37 (12): 73—84, December 1994

[15] J.D. Tesler, S.L. Strasnick. FSN: The 3D file system navigator. Silicon Graphics, Inc., 1992. ftp://sgi.sgi.com/sgi/fsn.

[16] H. Theisel, M. Kreuseler. An Enhanced Spring Model for Information Visualization. Computer Graphics Forum, Vol 17, No 3, (Proceedings Eurographics 98), 1998.