# A Scalable Framework for Information Visualization

Matthias Kreuseler , Norma López , Heidrun Schumann
{mkreusel, lopez ,schumann}@informatik.uni-rostock.de
University of Rostock, Dept. of Computer Science, 18051 Rostock, Germany

## Abstract

*This paper describes major concepts of a scalable information visualization framework. We assume that the exploration of heterogenous information spaces at arbitrary levels of detail requires a suitable preprocessing of information quantities, the combination of different graphical interfaces and the illustration of the frame of reference of given information sets. The innovative features of our system include dynamic hierarchy computation and user controlled refinement of those hierarchies for preprocessing unstructured information spaces, a new Focus+Context technique for visualizing complex hierarchy graphs, a new paradigm for visualizing information structures within their frame of reference and a new graphical interface that utilizes textual similarities to arrange objects of high dimensional information space in 3-dimensional visualization space.*

## 1. Introduction

Visual exploration of complex information spaces has become one of the "hot topics" in computer graphics research. A variety of novel visualization paradigms and frameworks have been developed in recent years. Nevertheless achieving flexible visualizations i.e., preprocessing large quantities of unstructured heterogeneous information, displaying information context (e.g. frame of spatial or domain references) and supporting a variety of exploration tasks carry over entirely new qualities of problems. Some of the most important ones can be summarized as follows:

- *Reducing information and Obtaining structure:* The exploration of large unstructured information spaces requires information preprocessing in order to reduce the active data size to processible levels. In this regard "filtering out uninteresting items" and merging similar objects into groups are necessary. Suitable metrics have to be applied for obtaining similarities and structures in high-dimensional feature space. Furthermore the degree of abstraction has to be controlled interactively in order to browse information space at arbitrary levels of detail.
- *Visualizing information sets:* The success of visualization depends very much on its ability of supporting a variety of exploration tasks (e.g. overview, zoom in on items of interest or details on demand). Different visualization methods are required for revealing information structure and information contents (e.g. attribute values). Furthermore novel interaction techniques are needed for controlling the degree of abstraction within visual representations and for providing navigational aids in information space.
- *Visualizing the frame of reference:* Effective explorations of spatially referenced information (e.g. health data in certain areas) require the combination of an adequate display of the spatial frame of reference with the visualization of complex information structures. It is necessary to find an appropriate mapping between information and frame of reference. This problem hasn't been widely studied yet in the case of displaying complex graphs such as large hierarchies over geographical maps.

A variety of visualization methods have been developed in different domains. Among these are techniques for visualizing and interacting with hierarchies like Cone Trees [6] or Disc Trees [13] which use horizontal and vertical cones or discs to layout hierarchies. FSN [22] and Information Pyramids [1] exploit the metaphor of 3D information landscapes to depict large hierarchical information spaces. Other approaches such as Treemaps [14] and CHEOPS [5] are well known 2D techniques which use available screen space very effectively.

Several techniques have been developed for visualizing multidimensional information. These methods try to map correlations of objects in high dimensional information space to spatial correlations in a 2D or 3D presentation space. Among these are approaches like IVORY [8], VR-VIBE [4] and Narcissus [10] which exploit spring models to place objects according to their similarities, whereby similar objects are placed spatially close together. Other systems like Lyberworld[9] and SPIRE[24] use different visual metaphors like Relevance Spaces[9], Information Galaxies or Themescapes[24] in order to visualize document collections or results from data base retrieval. FOCUS[21] is an interactive table viewer which supports the exploration of complex object-attribute tables by a combination of a focus+context technique, a hierarchical outliner for large attribute sets and a general easy-to-use dynamic query mechanism.

Systems like Descartes [2] or Devise [7] provide solutions for visualizing geographically related information. Different types of icons, diagrams, colored faces and maps are used for depicting data within their spatial frame of reference. In contrast to that these systems do not support the visualization of rather complex information structures as for instance abstract node link graphs or hierarchies.

Most of the systems mentioned above solve, each in its own manner, some of the single problems introduced earlier in this section. But, up to now, there are still open questions of how to provide a flexible framework for solving those problems in a more general way.

We suggest a scalable visualization framework (cf. section 2) in order to address the introduced problems. Basically our framework integrates a *scalable preprocessing pipeline* for organizing large unstructured high-dimensional information spaces (see section 3) with several new *scalable visualization techniques* (cf. section 4) for visualizing information structure along with information contents. We propose a new paradigm for integrating the visualization of information structures and their spatial frame of reference in section 5. Future work and conclusions are covered in section 6.

## 2. Basic concept of a scalable framework

The design of a scalable visualization framework requires a formal and easily adaptable information model for describing information units and the general characteristics of the information space. Therefore we introduce *Information objects* $IO_i$ as the basic elements of our information model. The term information object denotes a necessary abstraction of the data for specifying information units. Each information object is characterized by a set of attributes which can have arbitrary continuous ranges of values in order to describe object properties. Information objects $IO_i$ can be text documents, files or real world objects like cars, houses or cities. A more formal transcription of our information model is given by Wünsche [25].

In order to solve the problems addressed in section 1 we propose a framework which integrates a scalable preprocessing pipeline and different visualization modules. Our preprocessing pipeline implements several algorithms (e.g. clustering, dynamic hierarchy computation and neural networks) for analyzing unstructured information spaces. Combining different techniques within a flexible framework helps to scale preprocessing with respect to the characteristics of the information space and users exploration tasks. In order to display preprocessing results and to explore information space graphically the framework offers several new visualization techniques as well.

### Scalable Preprocessing

Preprocessing information in order to gain structure, e.g. identifying groups of related information objects or forming meaningful subsets of the given data is a non-trivial task because there is no general mathematical framework or paradigm on how to build those groups or subsets. Basically our approach exploits similarities between information objects in high dimensional feature space. Therefore we have to provide adequate measures $S_{ij} = S(IO_i, IO_j)$ for calculating similarities between information objects $IO_i$ and $IO_j$.

As stated in [3] computing similarity measures is rather complicated because similarity can be defined in various ways, and often domain specific expertise is required for determining appropriate measures. Furthermore the decision if two objects are similar or not is specific to user goals. Lets consider an example. A number of firms are described by the volume of sales over a period of several years. As it is the objective to group those firms with similar sales rates within this time period Euclidean Distance or some Minkowski Distances [15] are sufficient measures. In contrast to that the Dot product or a Correlation coefficient [15] are appropriate if it is the intention grouping firms with similar sales growth within that period of time. Thus either of the different measures might be appropriate in certain cases.

Furthermore the applicability of a specific similarity measure depends on the basic data types of the information object's attribute values. Thus similarities might have to be computed from variables that are binary, nominal, ratio scaled or a combination of these (cf. [15] for further information about these data types).

Summarizing the discussion above we conclude that providing a single similarity measure is not sufficient for a flexible preprocessing of complex information spaces. Therefore our preprocessing pipeline offers a variety of different metrics and similarity measures : *Euclidean distance, $L_p$-metric, Mahalonobis distance, Dot product, Normalized dot product, Correlation coefficient, General M-coefficient and M-coefficient.* Moreover the pipeline can easily be extended by additional measures. In addition to flexibility regarding similarity measures our pipeline supports different algorithms for preprocessing information. Depending on exploration tasks the user can choose one of the following techniques:

- Self Organizing Maps [16] which are suitable for determining an overview of the entire collection and revealing the overall similarity structure between information objects in information space
- Dynamic hierarchy computation which can be controlled interactively in order to achieve sophisticated organizations of complex data sets and to reveal patterns and relationships among the data

### Scalable Visualization

The effective presentation of different aspects of a given information set (e.g. visualization of information structure or display of concrete attribute values) requires the combination of different visualization methods respectively scalable techniques which can be adopted to specific exploration goals. Our scalable visualization framework provides several visualization techniques. Beneath Highfields (cf. Figure 1), KOAN [18] and Parallel Coordinates [12] we introduce the new techniques *Magic-Eye-View* for displaying complex graphs and *ShapeVis* for depicting multidimensional information sets. Furthermore we propose a new approach which we named *Marching Sphere* for visualizing complex information structures with spatial dependencies.

## 3. Structuring and preprocessing information

Exploring information collections becomes increasingly difficult as the volume of information grows. Major problems arise due to visual clutter and the limited screen space as the number of objects exceeds some limits. Hence it is indispensable to apply suitable preprocessing for gaining structures, extracting relevant subsets of the information and for reducing the active data size to processible levels.

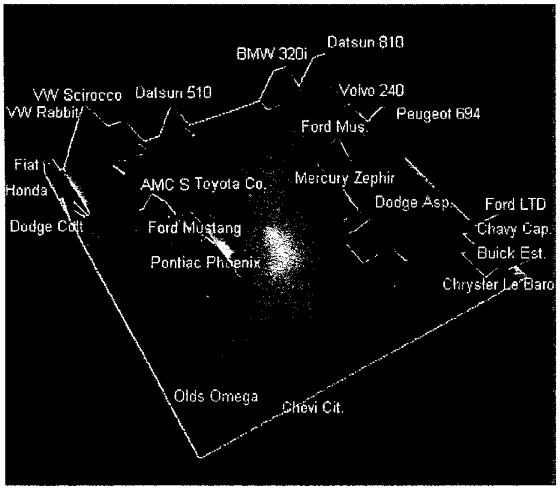


**Figure 1. Example of information organization based on self-organizing maps**

## 3.1. Self-organizing maps

Self-organizing maps (SOM) as introduced by Kohonen [16] provide an effective mechanism for preprocessing and organizing unstructured data. SOMs are able to extract groups of similar information objects and can be described as nonlinear projection from n-dimensional input space onto two-dimensional visualization space. A self organizing map consists of a two-dimensional network of neurons typically arranged on a regular lattice. Each cell is associated with a single randomly initialized n-dimensional reference vector. In the basic SOM algorithm the map is trained with a set of input vectors several times. For each input vector the map is searched for the most similar reference vector, called the winning vector. The winning vector is updated such that it more closely represents the input vector. Along with that the reference vectors in the neighborhood around the winning vector are also adjusted in response to the actual input vector. After the training phase reference vectors in adjacent cells represent input vectors which are close (i.e. similar) in information space. Thus SOMs provide a useful topological arrangement of information objects in order to display clusters of similar objects in information space.

Figure 1 illustrates the use of SOMs for structuring unorganized information spaces in our framework. The picture was generated from a car data set with 6 dimensions. Each peak in the map displays a cluster of similar objects. The number of objects within a single cluster is mapped onto the height of the peak. Color is used for displaying similarities between adjacent clusters where bright intensities denote a higher degree of dissimilarity. Thus SOMs are suitable for providing an overview of the entire information space.

## 3.2. Dynamic hierarchy computation

The dynamic hierarchy computation is another possible method to achieve predictable presentations of unstructured information spaces, even if the given data set is not a "natural" hierarchy. If an abstraction is used to organize data, it is important to remember that users may have different requirements when merging objects into groups. 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 similarity of the objects within those groups increase as the hierarchy is descended.

Dynamic hierarchy computation is carried out by adapted agglomerative clustering algorithms [15]. Based on one of the algorithms Single Linkage, Complete Linkage, Average Linkage, Ward, Median, Flexible Strategy and Zentroid [15], information objects $IO$ are merged into groups according to their similarities in information space. Therefore a symmetric $(n \; x \; n)$ similarity matrix $S$ is computed (with $n$ number of information objects $IO$ in information space) based on a single or on a combination of the similarity measures enumerated in section 2.

$$S = \begin{bmatrix} S_{1,1} & \cdots & S_{n,1} \\ \cdots & \cdots & \cdots \\ S_{n,1} & \cdots & S_{n,n} \end{bmatrix}$$

$$\begin{aligned} where \quad S_{i,j} &= S_{j,i} \quad \forall\, i,j = 1,\cdots,n \quad and \\ S_{i,i} &= 1 \quad \forall\, i = 1,\cdots,n \end{aligned}$$

The similarity matrix serves as a basis for a bottom up creation of a binary dendrogram (cf. Figure 2 left).

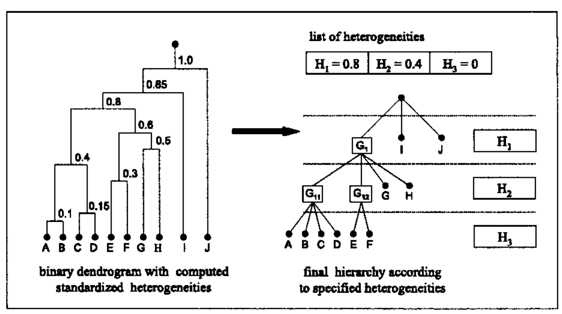


**Figure 2. Construction of the final Hierarchy tree with 3 levels based on the binary dendrogram**

In the first step we start merging the two most similar information objects $IO_i, IO_j$, i.e. where $S_{i,j} = max$ into the first group. Subsequently a new $(n-1 \; x \; n-1)$ similarity matrix is calculated and the next two closest objects (groups) are merged. This process continues until all information objects $IO_i$ are processed and the binary dendrogram is determined completely. A heterogeneity value, which denotes the average dissimilarity within a single group of objects, is calculated for each node in the binary dendrogram.

The hierarchy computation within our framework is scalable in terms of several similarity measures (c.f. section 2) and clustering algorithms. Furthermore it is 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 a user driven dynamic refinement of the hierarchy. The binary dendrogram (cf. Figure 2) which was computed previously provides such a basis. If the binary dendrogram has been determined the final hierarchy tree which represents the similarity structure of the information space is derived form it (cf. Figure 2). Therefore heterogeneity values have to be assigned with each level of the final hierarchy tree. These values denote the allowed average dissimilarities of the clusters at the levels in the final hierarchy tree (e.g., the maximum heterogeneity value $H_{max}$ is attached with the root node). These heterogeneity values can either be specified interactively by the user or determined automatically by our system in order to achieve optimized clustering of the data. Once the number of desired hierarchy levels and the heterogeneity values are specified the final hierarchy is derived from the dendrogram according to 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.
   (a) If not, proceed with the node's children at step 2.
   (b) 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.

Using the binary dendrogram is very efficient. Once the dendrogram is created we do not need time consuming recomputations of the similarity matrices for refining the hierarchy tree.

Thus complex information spaces can be browsed interactively in a top down like fashion by starting with an overview with only a few hierarchy levels (cf. Figure 3) and refining embodiments by increasing the number of hierarchy levels for determining more subtle patterns in the data (cf. Figure 4). The final hierarchy tree contains information objects *IO* at its leaves. The remaining nodes represent clusters which fulfil the heterogeneity conditions associated with each hierarchy level. The principle of hierarchy refinement is depicted in Figure 3 and Figure 4. As the number of levels is increased, bigger clusters are split up into smaller sub-clusters (see Fig. 3 and 4). Thus a stepwise exploration at arbitrary levels of detail is supported.

# 4. Visualization

Supporting a variety of different exploration tasks (e.g. displaying different aspects of given information sets) as well as processing different types of information (e.g. hierarchical information structures or unstructured multi-dimensional information spaces) requires several visualization methods or a combination of these methods. Therefore our framework provides a range of different techniques. Beneath known techniques (c.f. section 2)

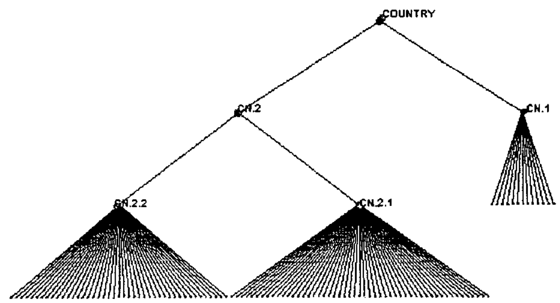


**Figure 3. Overview with 3 hierarchy levels**

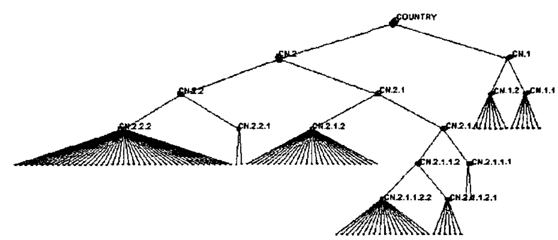

**Figure 4. Hierarchy refinement with 7 levels**

we propose the new Focus+Context technique *Magic-Eye-View* for displaying complex hierarchy graphs and an adapted version of our *ShapeVis* for visualizing multidimensional information sets.

## 4.1. Hierarchy Visualization

Visualizing the computed hierarchies becomes complicated as the number of levels and nodes increases. Standard 2D hierarchy browsers can typically display about 100 nodes [17]. 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 [20] or the Hyperbolic Browser [17] 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 propose 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.

**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 [19]. 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 $(\lambda, \phi)$. Thus the determined Cartesian coordinates can be mapped directly to spherical coordinates.

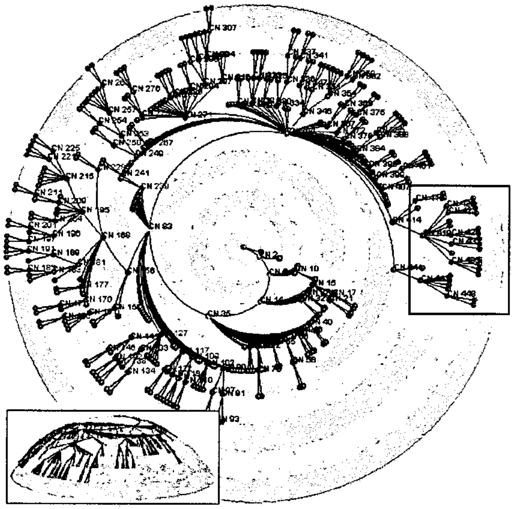

**Figure 5. Complex hierarchy graph with $\mathbf{p}_0$ at the origin**

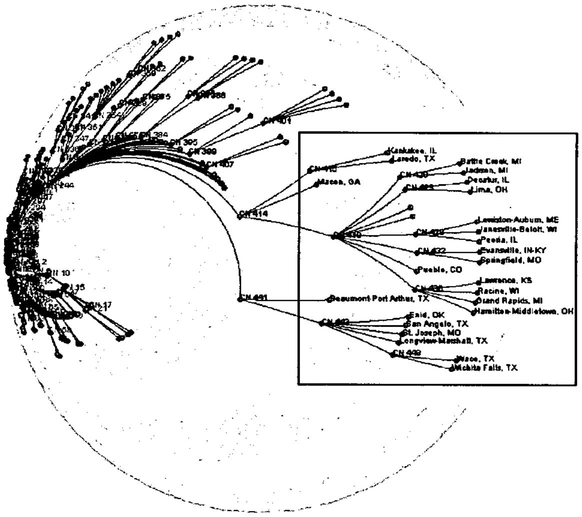

**Figure 7. Complex hierarchy graph with enlarged focus region**

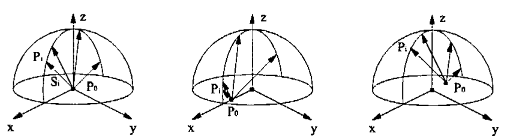

**Figure 6. Projection rays before and after moving $P_0$**

#### 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 introduce 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 $\mathbf{p}_0 = (0,0,0)$ through each of the $n$ nodal-points $\mathbf{p}_i$ (cf. Figure 6 left), 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 $\mathbf{p}_0$ can be moved arbitrarily, whereby the directions of the rays $S_i$ are retained (cf. Figure 6 middle and right). 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 $\mathbf{p}_0$. 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 $\mathbf{p}_0$ along the $X,Y,Z$-axis, the hemisphere can also be rotated, translated and zoomed. Compared to the Hyperbolic View [17] we introduce additional degrees of freedom for browsing hierarchies since we use change of focus along with conventional 3D navigation. Figures 5 and 7 demonstrate change of focus. Figure 5 shows a complex hierarchy graph mapped onto a hemisphere. The center of projection has been moved in Figure 7 in order to set the focus to the marked sub-graph. We introduce colored rings for minimizing the amount of confusion introduced by the distortion.

## 4.2. Visualization of Multi-dimensional information

We developed the new technique ShapeVis[1] for further exploration of multi-dimensional information sets (e.g. revealing attribute values of the data or determining object similarities within a cluster or at certain hierarchy levels). ShapeVis exploits an enhanced spring model for arranging n-dimensional information objects in 2(3)-dimensional visualization space according to their attribute values. For reasons of readability, we briefly sketch the basics of our model in this section.

#### Enhanced spring model

Information objects $IO$ are described by a set of $n$ attributes[2] which have continuous ranges of values. Thus each $IO$ in the n-dimensional information space is a $n$-tuple $(c_1,\ldots,c_n) \in \mathbb{R}^n$ with $(c_1,\ldots,c_n) > 0$. The $c_i$ with $i = 1,\ldots,n$ can be considered as the coordinates of the $IO$ in information space. (As an example, consider the $IO$ as text documents and the attributes as certain keywords. Then the coordinates $(c_1,\ldots,c_n)$ of an $IO$ are the frequencies of appearance of the key words in the document.)

Several approaches (e.g. [11]) use a classical spring model for mapping objects from n-dimensional information space onto 2(3)-dimensional visualization space. In the classical spring model every dimension of the information space is related to a point $\mathbf{d}_i \in \mathbb{R}^2(\mathbb{R}^3), (i = 1,\ldots,n)$ in the visualization space. An information object $IO = (c_1,\ldots,c_n)$ is mapped to a point $\mathbf{p}$ in visualization space using $n$ springs - from each dimension point $\mathbf{d}_i$ to $\mathbf{p}$. The stiffness of the springs are set to the values $c_1,\ldots,c_n$. Then the location of $\mathbf{p}$ is searched where the spring model is in balance.

[1]We use an adapted version of our technique introduced in [23] within the framework.

[2]The terms data *dimension* and *attribute* are used exchangeable in the following sections.

For fixed $\mathbf{d}_i$ this location can be computed explicitly:

$$p = \frac{\sum_{i=1}^{n} c_i \cdot \mathbf{d}_i}{\sum_{i=1}^{n} c_i} \tag{1}$$

The location of $\mathbf{p}$ gives spatially intuitive information about the information objects, i.e. the bigger the value of a certain attribute ($c_i$) the closer moves $\mathbf{p}$ towards $\mathbf{d}_i$. Furthermore objects with similar properties are spatially close in the visualization. Beneath these advantages the classical spring model introduces two major drawbacks.

1. *ambiguity:* Objects with different properties (coordinates $(c_1, \ldots, c_n)$ in information space) may collapse to the same point in visualization space (cf. [23]).
2. *insensitivity against coordinate scalings:* The information objects $(c_1, \ldots, c_n)$ and $(c_1 \cdot k, \ldots, c_n \cdot k)$ with $k > 0$ cannot be distinguished in the visualization because they are mapped to the same point.

In order to solve the problems mentioned above we assign an information object $IO$ not only with a point but with a small shape which is composed off basic geometric primitives. Size, location and orientation of these primitives are determined based on the following enhanced spring model.

As in the classical spring model, we place a fixed point $\mathbf{d}_i \in \mathbb{R}^2 (\mathbb{R}^3)$ for every dimension of the information space. We attach $n$ springs with the constant stiffness $c > 0$ to $\mathbf{p}$. The other ends of the springs are named $\mathbf{p}_1, \ldots, \mathbf{p}_n$. Now we consider $n$ more springs - from $\mathbf{p}_i$ to $\mathbf{d}_i$ with the stiffness $c_i$. The points $\mathbf{p}, \mathbf{p}_1, \ldots \mathbf{p}_n$ are free moveable, the points $\mathbf{d}_1, \ldots, \mathbf{d}_n$ are fixed. Then we search for the state of balance of this spring system. Figure 8 illustrates this principle. Applying this model, an information object $IO = (c_1, \ldots, c_n)$ is described by the $n+1$ points $\mathbf{p}, \mathbf{p}_1, \ldots, \mathbf{p}_n$ which can be computed explicitly by solving the linear system of equations (2)-(4):

$$\mathbf{p} = \frac{\sum_{i=1}^{n} w_i \cdot \mathbf{d}_i}{\sum_{i=1}^{n} w_i} \tag{2}$$

with

$$w_i = \frac{c_i}{c + c_i} \quad for\, i = 1, \ldots, n \tag{3}$$

Then $\mathbf{p}_1, \ldots, \mathbf{p}_n$ are obtained by:

$$\mathbf{p}_i = \frac{c \cdot \mathbf{p} + c_i \cdot \mathbf{d}_i}{c + c_i} \quad i = 1, \ldots, n \tag{4}$$

Obviously the locations of $\mathbf{p}, \mathbf{p}_1, \ldots \mathbf{p}_n$ depend on the attribute values $(c_1, \ldots, c_n)$ of the information object and on the value of constant $c$. Thus the points $\mathbf{p}, \mathbf{p}_1, \ldots, \mathbf{p}_n$ describe an information object $IO = (c_1, \ldots, c_n)$ uniquely. Thus we solve the problems introduced by the classical spring model.

**Obtaining geometric objects**

Even if the points $\mathbf{p}, \mathbf{p}_1, \ldots, \mathbf{p}_n$ describe an object uniquely - $n+1$ points are not suitable for visualizing information objects. We studied the use of small closed free-form-surfaces (c.f. [23]) for obtaining an intuitive imagination of the locations of $\mathbf{p}, \mathbf{p}_1, \ldots, \mathbf{p}_n$. But approximating point locations using free-form-surfaces becomes rather difficult when the number of information objects is

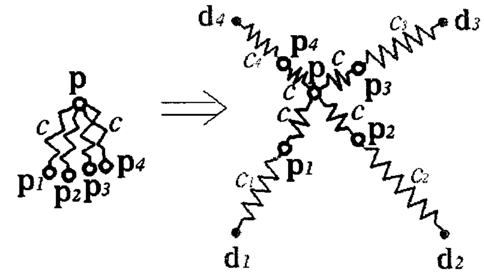

**Figure 8. Enhanced spring model for an information object $IO$ in 4-dimensional information space**

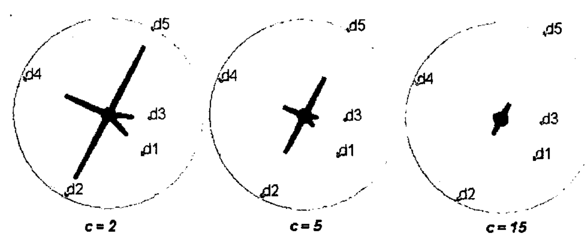


**Figure 9. Visualizing an object (1,2,1,1,2) with different parameters $c$.**

increased to several hundreds or thousands because of the large number of polygons required for generating smooth surfaces. (E.g, the geometric complexity of the visualization exceeded a million polygons when approximating 500 objects with a satisfying geometric resolution.)

Therefore we propose basic primitives ($n$ cylinders) for composing geometric objects out of the $\mathbf{p}, \mathbf{p}_1, \ldots, \mathbf{p}_n$. These $n$ cylinders tie up $\mathbf{p}$ and $\mathbf{p}_i$ in order to build the geometric shape which is assigned with the according information object. Location, Orientation and lengths of each cylinder depend on $\mathbf{p}, \mathbf{p}_1, \ldots, \mathbf{p}_n$ and the constant $c$. Thus the geomteric objects describe the information objects $IO = (c_1, \ldots, c_n)$ uniquely. This principle and the influence of parameter $c$ are illustrated in Figure 9. The strength of the deformation (length of cylinders) decreases, if parameter $c$ is increased. If the length of all cylinders of a geometric object is less than a certain threshold, we replace this object with a small sphere around $\mathbf{p}$. In this case we have the classical spring model. The parameter controlled deformation is very useful for visualizing a higher number of objects. First we obtain a global impression by visualizing all objects with a high parameter $c$. The objects are small points and we try to detect clusters. If we found a cluster, we zoom into it and decrease $c$ such that the deformation of the cylinders provides more information about the object properties (i.e. a long cylinder into the direction of a certain $\mathbf{d}_i$ denotes large data value of the according $c_i$ of this object). Figure 10 illustrates this principle. Our approach is applied to a data set which measures 6 demographic parameters of 106 countries. We placed one point $\mathbf{d}_i$ for each dimension of the data set in an

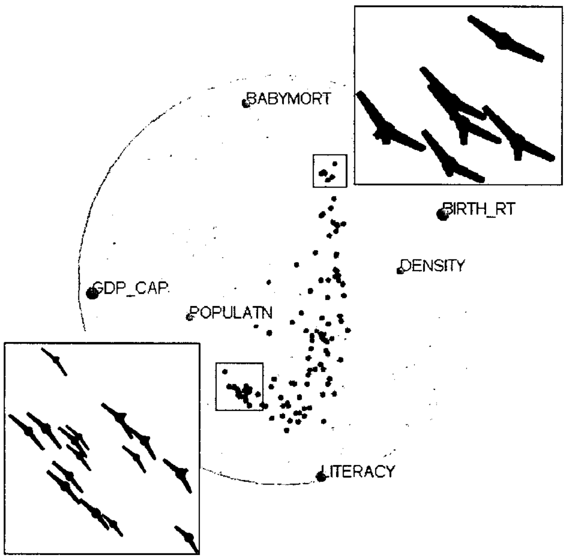


**Figure 10. Visualization of a demographic data set with 6 dimensions with deformable geometric objects.**

equidistant way on the surface of a sphere. The global clustering of the data can be obtained within the sphere. The objects in the upper right, which have big values in the dimensions *Baby mortality* and *Birthrate* move towards the according dimension points $\mathbf{d}_i$. Furthermore we can verify the assumption that these objects have big values in the dimensions *Baby mortality* and *Birthrate* by applying the deformation to the geometric objects. The cylinders which point towards the *Baby mortality* and *Birthrate* dimension points are much longer than the cylinders which point towards the remaining $\mathbf{d}_i$ (c.f. Figure 10 magnification of the upper cluster). In contrast to that the cluster lower left is characterized by countries with much bigger values with respect to the dimensions *Literacy* and *Gross Domestic Product* while the values of *Baby mortality* and *Birthrate* are rather small.

## 4.3. Combination of techniques

The techniques introduced above are combined in our framework in order to support flexible visualizations at arbitrary levels of detail. Therefore arbitrary subsets of the hierarchy can be selected for further exploration.

- *Selection of cluster nodes* - Each cluster node of the hierarchy tree can be selected. Color is used to distinguish between cluster nodes and object nodes, whereby the size of a cluster, i.e. the number of objects is mapped to the intensity of the 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 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.

Exploring clusters and levels with ShapeVis reveals basic information about attribute values and similarities between clusters and information objects. In order to identify concrete information contents, i.e. real attribute values, arbitrary shape objects can be selected and visualized with parallel coordinates [12]. Labeling the coordinate axis and displaying the data values provides more detailed information about each information object.

Figure 11 illustrates the combination of ShapeVis, Magic-Eye-View and Parallel Coordinates applied to a data set which describes 2440 houses with five attributes. The left picture shows the 2440 houses with ShapeVis and reveals three visual clusters. Exploration of single objects is rather complicated because of 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 is more meaningful to preprocess the data as introduced in section 3 in order to form manageable subsets. The picture in the middle of Figure 11 depicts the hierarchical representation of the data set. The three major clusters are represented by the hierarchy nodes at the first level. Furthermore the tree shows that these clusters are split up into smaller sub-clusters at the following levels. These sub-clusters can be selected for further exploration.

We selected the first level of the hierarchy tree and obtained the picture upper right which shows one representative for each of the three major clusters. Thus we can explore the relationships between the three clusters very easily by size, location and deformation of its graphical objects.

The picture lower right shows the use of parallel coordinates. In our example we selected a single information object (*Haus_516*) which belongs to the cluster *CN.1*. The diagram displays the concrete attribute values of (*Haus_516*) compared to the data values of the object which represents the whole cluster *CN.1*.

## 5. Marching Sphere

In many application domains (e.g. demographic research, health monitoring etc.) complex information structures are given within a spatial frame of reference. In general the usability of visual representations of given information can be enhanced significantly by displaying these frame of reference. Geographic Information Systems (GIS) provide various functions for displaying this spatial frame of reference but do not offer the functionality for depicting information structures like complex graphs or hierarchy trees.

We propose in our framework the *Marching Sphere* as a new approach for solving these drawback. The *Marching Sphere* combines the visualization of complex information structures and the display of the spatial frame of reference within the same visual representation. In order to achieve this we had to solve a range of problems:

- The visualization of spatially referenced information structures is rather complicated because of the high display complexity, since the information structures have to be visualized along with the spatial frame of reference.
- Techniques which generate compact embodiments of the given information set have to be applied such that the vi-

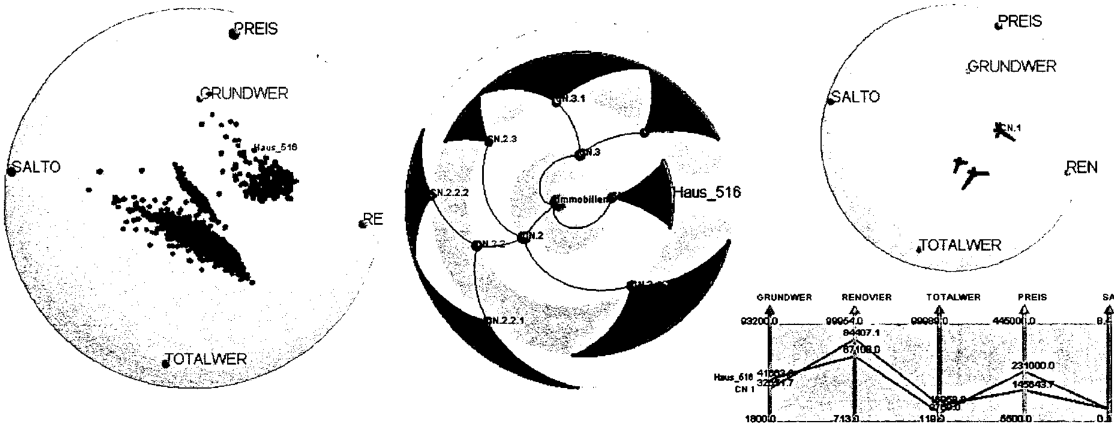


**Figure 11. Combination of Hierarchies, ShapeVis and Parallel coordinates**

sual representation of the information can easily be displayed within the geographical frame of reference.

- Suitable graphical representations have to be provided for the spatial (geographical) frame of reference. Furthermore an appropriate function has to be specified for mapping graphical representations of the information into the virtual frame of reference (e.g. onto the appropriate positions over the geographic maps).
- Interaction techniques are necessary for manipulating both embodiments of the geographical frame of reference and information structure in order to support a variety of exploration tasks.

### Displaying information structure

We use abstract 3-dimensional graphs for displaying information objects and revealing structural relations between information units. Basically we apply a technique called *KOAN* [18] (KOntext ANalysator), originated by SIEMENS. *KOAN* maps information objects from high-dimensional information space onto 3-dimensional visualization space according to the principle "contextual correlation $\approx$ spatial proximity", whereas *contextual correlation* denotes the similarity between information objects in information space. Thus similar objects are arranged spatially close in the graph. KOAN uses different types of nodes for depicting information objects (e.g. small cubes) and attributes (e.g. small spheres). Furthermore edges can be displayed between graph nodes in order to show whether objects or attributes are related to each other or not. This approach allows an easily understandable and compact visualization of complex information sets and shows structural relationships between units of information very intuitively.

### Displaying the spatial frame of reference

The visualization of the spatial frame of reference is based on ordinary 2-dimensional maps. Maps provide very intuitive visualizations of geographical areas and offer sufficient space for displaying further information.

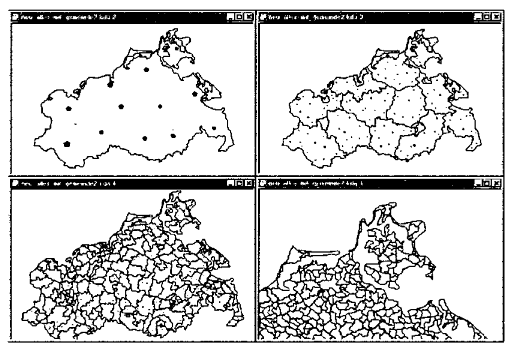

**Figure 12. Refinement of the geographical granularity**

We propose a hierarchical organization of these maps in order to display geographic areas at different levels of detail. This seems to be very useful because geographic areas usually contain subareas. Imagine for example the geographic structure of Germany. The country consists of several federal states, each of which contains a number of different counties. Counties are subdivided into zip code areas which are further split up into communities. Hence it is necessary to support different geographical resolutions in order to achieve a suitable visualization of the geographical frame of reference. Map refinement is illustrated in Figure 12. The picture shows a map of the German federal state Mecklenburg-Vorpommern in the 4 different resolutions - state, county, zip code area and community.

### Combination of information structure and geographical frame of reference

The *Marching sphere* implements the combination of both embodiments of information and spatial frame of reference. Therefore the 2-dimensional map is rendered in a virtual 3D scene. The 3-dimensional graph which represents the related information set is mapped into the virtual 3D scene as well, such that it is located above the area in which the information is given. The information graph is surrounded by a translucent sphere which is linked to the related area of the geographic map. Thus we provide unique mapping between information representation and frame of reference.

Placing complex graphs above each sub-area of a geographic map becomes increasingly difficult as the number of sub-areas grows or the complexity of the graph exceeds some limits. In order to avoid overlap of different graphs we propose the idea of the *Marching Sphere*. Basically we show only one complete graph at a time above an area of interest which can be specified interactively. In order to explore the information related to the remaining areas of the map, the sphere can be moved to arbitrary destination areas. The graph which was shown in the sphere previously is faded out and replaced with the graphical representation of the information related to the destination area. Thus the sphere can *'march'* throughout the whole geographical map driven by the user in order to display the information related to the sub-areas.

Furthermore the *Marching Sphere* provides a range of visual aids and interactions techniques for supporting a variety of exploration tasks:

- *Visualize context:* The information related to the areas around the sphere's current position can be shown along with the actual information graph. Therefore graphs with reduced complexity are determined and placed above the belonging areas.
- *Show history:* The areas which have been explored previously can be highlighted in the map along with the navigation path of the sphere.
- *Reveal details:* The information graph can be enlarged and rendered in a separate display area for revealing details.
- *Change geographical resolution:* The user can refine arbitrary areas of the map by selecting a more subtle geographical resolution. Along with that, the visualization of the related information is refined as well. (i.e, the information set related to the area which has been selected for refinement is split up into subsets whereby each of these subsets is assigned to the related sub-area of the selected area.)

Figure 13 illustrates the marching sphere applied to spatially related health information of the federal state Mecklenburg Vorpommern in Germany. Geographical overview is provided by the map which shows the different counties of the state. A county is selected and the related information is visualized as 3-dimensional graph which displays relations between certain diseases within that area. The smaller spheres around the selected area depict information objects related to the neighboring counties. The line on the map depicts the *exploration history*. In order to reveal further details such as node labels, the graph is magnified and rendered in a separate display area (c.f. Figure 13 upper right).

# 6. Conclusions and Future Work

This paper proposed a general framework for information visualization. The integration of preprocessing and visualization enables exploration of large information space at different levels of detail by providing an overview of the entire information space which can be arbitrarily refined by the user.

One of the major components of our framework is a flexible preprocessing pipeline. Several algorithms and similarity measures can be applied for structuring unorganized data and forming manageable subsets of complex information spaces. Especially the user controlled dynamic hierarchy computation is a suitable method to achieve predictable representations of given data and to support data analysis at arbitrary levels of detail.

We propose several new visualization techniques for displaying multi-dimensional and hierarchical information spaces. Furthermore our framework contains a new paradigm for exploring spatially referenced information structures.

However, there are still a number of challenges for future work. First of all evaluation of the introduced techniques needs to be performed to determine their effectiveness and to verify their applicability in different application domains.

Further work has to be done in order to enhance both the preprocessing and the introduced visualization techniques. In future research we would like to speed up hierarchy computation. 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 the dimension points in ShapeVis has to be investigated as well. Furthermore we would like to investigate animations for smoothly fading in and out information graphs in the *Marching Sphere.*

# 7. Acknowledgements

The authors wish to thank Prof. Schütt, Dr. Kolpatzik and Dr. Herzog from Siemens AG Munich for their constant support. Thanks also to Thomas Bürger, Thomas Nocke and Axel Oestereich for developing parts of the source code.

# References

[1] K. Andrews, J. Wolte, and M. Pichler. Information Pyramids (TM): A new Approach to Visualizing Large Hierarchies. In A. Varshney and D. S. Ebert, editors, *LBHT IEEE Visualization '97*. IEEE, Oct. 1997.

[2] G. Andrienko and N. Andrienko. Interactive Maps for Visual Data Exploration. In *ICA Commission on Visualization, Warsaw, 1998*. GMD - German National Research Center for Information Technology, 1998.

[3] M. Ankerst, S. Berchthold, and D. Keim. Similarity Clustering for an Enhanced Visualization of Multidimensional Data. In G. Wills and J. Dill, editors, *IEEE Information Visualization '98*. IEEE, Oct. 1998. ISBN 0-8186-9093-3.

[4] S. Benford, D. Snowdon, R. I. C. Greenhalgh, I. Knox, and C. Brown. VR-VIBE: A Virual Environment for Cooperative Information Retrieval. In H.-P. Seidel and P. J. Wills, editors, *Computer Graphics Forum, Vol. 14, Number 3, (Proceedings Eurographics '95)*. Eurographics Assocition, Aug. 1995. ISSN 0167-7055.

[5] L. Beudoin, M.-A. Parent, and L. Vroomen. Cheops: A compact explorer for complex hierarchies. In R. Yagel and G. M. Nielson, editors, *IEEE Visualization '96*. IEEE, Oct. 1996. ISBN 0-89791-864-9.

[6] S. Card, G. Robertson, and J. Mackinlay. The Information Visualizer - An Information Workspace. In *ACM SIGCHI*

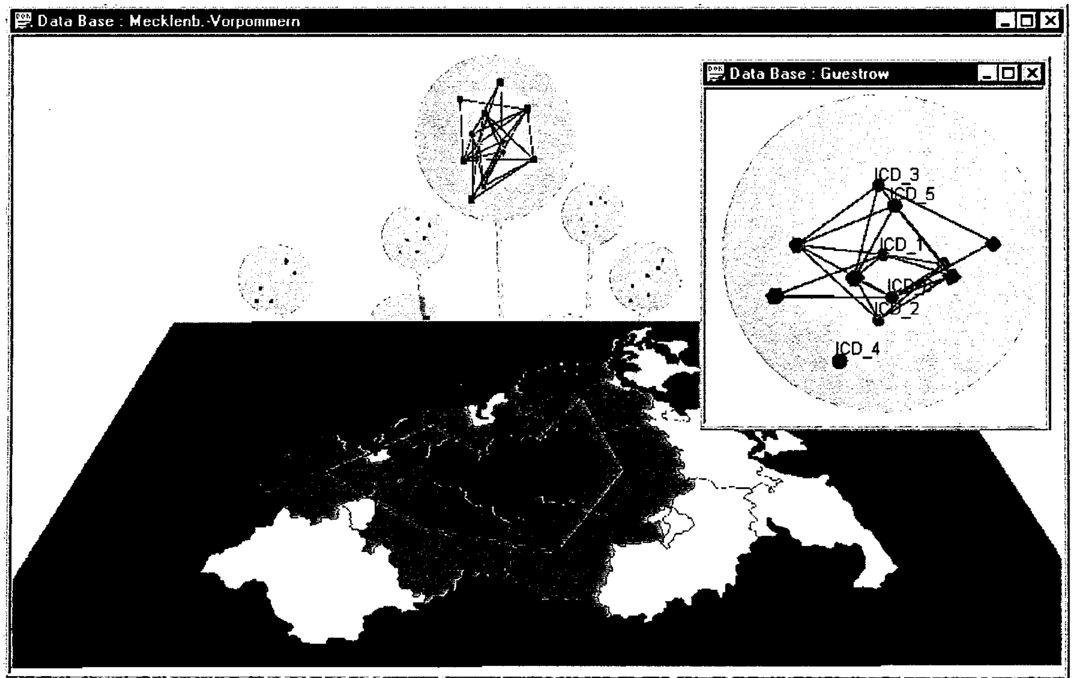


**Figure 13.** *Marching Sphere* **applied to spatially related health information of the federal state Mecklenburg Vorpommern**

'91 Conference on Human Factors in Computing Systems*. ACM, 1991.

[7] DEVise. *An Environment for Data Exploration and Visualization*. http://www.cs.wisc.edu/~devis/devise.html, 1998.

[8] M. Gross, T. Sprenger, and J. Finger. Visualizing Information on a Sphere. In J. Dill and N. Gershon, editors, *IEEE Information Visualization '97*. IEEE, Oct. 1997. ISBN 0-8186-8189-6.

[9] M. Hemmje. LyberWorld - A 3D Graphical User Interface for Fulltext Retrieval. In *ACM SIGCHI '95*. ACM, 1995.

[10] R. Hendley, N. Drew, A. Wood, and R.Beale. Narcissus: Visualizing Information. In N. Gershon and S. Eick, editors, *IEEE Information Visualization '95*. IEE, Oct. 1995. ISBN 0-8186-7201-3.

[11] P. Hoffmann, G. Grinstein, and E. I. Grosse. Dna Visual and Analytic Data Mining. In R. Yagel and H. Hagen, editors, *IEEE Visualization '97*. IEEE, Oct. 1997. ISBN 0-8186-8263-9.

[12] A. Inselberg and B. Dimsdale. Parallel Coordinates: A Tool for Visualizing Multidimensional Geometry. In *IEEE Visualization '90*. IEEE, Oct. 1990.

[13] C. Jeong and A. Pang. Reconfigurable Disc Trees for visualizing Large Hierarchical Information Space. In G. Wills and J. Dill, editors, *IEEE Information Visualization '98*. IEEE, Oct. 1998. ISBN 0-8186-9093-3.

[14] B. Johnson and B. Shneiderman. Tree-maps: A Space Filling Approach to the Visualization of Hierarchical Information Structures. In *IEEE Visualization '91*. IEEE, 1991.

[15] L. Kaufman and P. J. Roussew. *Finding Groups in Data - An Introduction to Cluster Analysis*. Wiley-Science Publication John Wiley & Sons Inc., Brussels, 1990.

[16] T. Kohonen. *Self Organizing Maps*. Springer Verlag, Berlin, 1995.

[17] J. Lamping, R. Rao, and P. Pirolli. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In *ACM CHI'95*. ACM, 1995.

[18] J. Panyr, U. Preiser, and T. Führing. Kontextuelle Visualisierung von Informationen. In *19. Oberhofer Kolloquium über Information und Dokumentation*, 1996.

[19] E. Reingold and J. Tilford. Tidier Drawing of Trees. *IEEE Transaction on Software Engineering*, 7(2):223–228, 1981.

[20] M. Sarkar and M. Brown. Graphical fisheye views. *Communications of the ACM*, 12(37):73–84, Dec. 1994.

[21] M. Spenke, C. Beilken, and T. Berlage. FOCUS: The Interactive Table for Product Comparison and Selection. In *ACM UIST '96*. ACM, 1996.

[22] J. Tesler and S. Strasnick. *FSN: The 3D File System Navigator*. Silicon Graphics Inc., ftp://sgi.sgi.com/sgi/fsn, 1992.

[23] H. Theisel and M. Kreuseler. An Enhanced Spring Model for Information Visualization. *Computer Graphics Forum*, 17(3), Sept. 1998.

[24] J. A. Wise, J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the Non-Visual: Spatial Analysis and Interaction with Information from Text Documents. In N. Gershon and S. Eick, editors, *IEEE Information Visualization '95*. IEEE, Oct. 1995. ISBN 0-8186-7201-3.

[25] V. Wünsche. Concepts for information visualization. *Computer Science Reports University of Rostock*, 21:113–132, 1998.