A Flexible Approach for Visual Data Mining

Matthias Kreuseler and Heidrun Schumann

Abstract—The exploration of heterogenous information spaces requires suitable mining methods as well as effective visual interfaces. Most of the existing systems concentrate either on mining algorithms or on visualization techniques. This paper describes a flexible framework for Visual Data Mining which combines analytical and visual methods to achieve a better understanding of the information space. We provide several preprocessing methods for unstructured information spaces such as a flexible hierarchy generation with user controlled refinement. Moreover, we develop new visualization techniques including an intuitive Focus+Context technique to visualize complex hierarchical graphs. A special feature of our system is a new paradigm for visualizing information structures within their frame of reference.

Index Terms—Information visualization, multidimenisional information modeling, hierarchies, focus+context techniques, clustering, maps, information analysis.

1 Introduction

Exploration of complex information spaces has become one of the "hot topics" in many research fields, including computer graphics, data mining, pattern recognition, and learning, and other areas of statistics, as well as data bases and data warehousing. A variety of novel mining techniques, visualization paradigms, and frameworks have been developed in recent years. Nevertheless, extracting useful knowledge or models from observed data is still a complicated nontrivial process.

In this context, visualization offers a powerful means of analysis that can help to uncover patterns and trends hidden in unknown data. Additionally, visualization provides a natural method of integrating multiple data sets and has been proven to be reliable and effective across a number of application domains. Still, visual methods cannot entirely replace analytic nonvisual mining algorithms. Rather, it is useful to combine multiple methods during data exploration processes [31].

The new area of visual data mining focuses on this combination of visual and nonvisual techniques as well as on integrating the user in the exploration process. Integrating visual and nonvisual methods in order to support a variety of exploration tasks, such as identifying patterns in large unstructured heterogeneous information or displaying information context (e.g., frame of spatial or domain references), requires sophisticated mining, visualization and interaction techniques. This carries over entirely new qualities of problems. Some of the most important ones can be summarized as follows:

- *Extracting patterns and controlling the mining:* The exploration of large unstructured information spaces requires information preprocessing. In this regard "filtering out uninteresting items" and merging similar objects into groups are necessary in order to reveal hidden patterns. 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 supervise and steer the search for patterns during the mining process. This interaction is of outstanding importance to support explorations at arbitrary levels of detail.
- *Visualizing information sets:* The success of visual data analysis depends very much on its ability to support a variety of exploration tasks such as overview, zoom in on items of interest or details on demand. Different visualization methods are required for revealing information structure and information contents such as 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. In particular, we address the problem of displaying complex graphs over geographical maps, a problem that has not been widely studied.

Ankerst [4] classifies current visual data mining approaches into three categories. Methods of the first group apply visualization techniques independent of data mining algorithms. The second group uses visualization in order to represent patterns and results from mining algorithms graphically. The third category tightly integrates both mining and visualization algorithms in such a way that intermediate steps of the mining algorithms can be visualized. Furthermore, this tight integration allows users

- *The authors are with the Universität Rostock, FB Informatik, PostBox 999, 18051 Rostock, Germany.*
E-mail: {mkreusel, schumann}@informatik.uni-rostock.de.

Manuscript received 12 apr. 2001; accepted 10 July 2001.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number 114504.

to control and steer the mining process directly based on the given visual feedback.

A variety of visualization methods which have been developed in different domains can be classified into the first group referring to the classification given above. Among these are techniques 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 [10], VR-VIBE [6], and Narcissus [12], which exploit spring models to place objects according to their similarities, whereby similar objects are placed spatially close together. Other systems, like Lyberworld [11] and SPIRE [33], use different visual metaphors like Relevance Spaces [11], Information Galaxies, or Themescapes [33] in order to visualize document collections or results from data base retrieval. FOCUS [25] 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.

Other visual interfaces have been developed for visualizing and interacting with hierarchies, like Cone Trees [8] or Disc Trees [16], which use horizontal and vertical cones or discs to layout hierarchies. FSN [28] and Information Pyramids [2] exploit the metaphor of 3D information landscapes to depict large hierarchical information spaces. Other approaches, such as Treemaps [17] and CHEOPS [7], are well-known 2D techniques which use available screen space very effectively.

The visualization of mining models (category 2 of the classification of visual data mining approaches) can be found in [26], where hierarchical cluster structures are discovered and visualized based on implicit surfaces. Other examples are WebSOM [1], which applies color coded planes to visualize results of a Self-Organizing Map algorithm, or OPTICS [4], which displays hierarchical clusterings.

Systems like Descartes [3] or Devise [9] 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. These systems, however, 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. Up to now, there are still open questions of how to provide a flexible framework for solving those problems in a more general way.

The work reported in this paper was inspired by the research stated above. In Section 2, we briefly sketch our approach for modeling information space. We suggest a scalable visualization framework (cf. Section 3) 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 4) with several new *scalable visualization techniques* (cf. Section 5) for visualizing information structure along with information contents, as well as displaying and interacting with mining results. We propose a new paradigm for integrating the visualization of information structures and their spatial frame of reference in Section 6. Future work and conclusions are covered in Section 7.

2 Information Model

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. It's our goal to define a general model which is suitable for different domains and a variety of visualization applications.

References [30] and [31] use objects to represent information. In order to formalize this information representation, we introduces the concept of *information objects* IO_i to describe the *information space*. The term "information object" denotes a necessary abstraction of the data which represent the information. Information objects are concrete objects (e.g., documents, files, or real world objects like cars, houses, or cities) which may contain other information objects.

The information set $\mathbf{IM}$ is a discrete set of information objects.

$$\mathbf{IM} = \{IO_1, \ldots, IO_n\} \tag{1}$$

$$with \quad IO_i = IO_j \quad \Leftrightarrow \quad i = j \quad i, j, n \in \mathbb{N}.$$

Information objects are characterized by a set of attributes. Those attributes can have arbitrary continuous or categorical ranges of values in order to describe object properties and the characteristics of the information. The function $attr$ provides all attributes of a set of information objects.

$$attr(\{IO_1, IO_2, \ldots IO_n\}) = \{A_1, A_2, \ldots, A_k\} \tag{2}$$

$$with \quad A_i = A_j \quad \Leftrightarrow \quad i = j \quad i, j, k, n \in \mathbb{N}.$$

The attribute set $\mathbf{AM}$ is the set of all attributes A_i of the set of information objects.

$$\mathbf{AM} = attr(\{IO_1, \ldots, IO_n\}) \qquad n \in \mathbb{N}. \tag{3}$$

Those attributes define dimensions and span the *information space* $\mathbf{IR}$, whereas the ranges of attribute values define the scaling of the related axes of the information space.

The dimensionality of the information space $\mathbf{IR}$ is defined as the cardinality of the attribute set $\mathbf{AM}$.

$$dim(\mathbf{IR}) = |\mathbf{AM}|. \tag{4}$$

In other words, the attributes and their ranges of values represent the dimensions of the information space in our model. Thus, information objects IO_i can be understood as points within multidimensional information space.

In order to model arbitrary relations between IO_i which might either be given explicitly or obtained implicitly, we introduce the information structure $\mathbf{IS}$.

The information structure **IS** is defined as a relation on the information set **IS**:

$$\mathbf{IS} \subseteq \mathbf{IM} \times \mathbf{IM}. \quad (5)$$

The absolute value of **IS** may be 0, i.e., in some cases there may be no description of the relation between information objects.

Summarizing our model, the information space **IR** is defined by means of the information set **IM**, attributes which describe the information properties and represent the dimensions of **IR** and the information structure **IS**.

The information definition given above allows modeling of complex information spaces. Arbitrary visualization scenarios can be handled due to the use of attributes for characterizing information objects and the use of relations for describing connections between pieces of information. Spatially referenced information spaces can be described as well when treating the spatial frame of reference as a special attribute.

3 Basic Concept of a Scalable Framework

In order to solve the problems addressed in Section 1, we propose a framework which integrates a scalable preprocessing pipeline and different visualization modules. Basically, our preprocessing pipeline implements several algorithms, such as interactive filters, 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.

3.1 Scalable Preprocessing

Preprocessing large information spaces often requires reducing the active data size to processible levels without losing relevant information. Other preprocessing tasks, such as gaining structure, identifying groups of related information objects, or forming meaningful subsets of the given data, are nontrivial because there is no general mathematical framework or paradigm on how to build those groups or subsets.

In order to address these problems and to achieve flexibility in the exploration process, we propose two major methods for preprocessing information spaces within our framework. Interactive user-driven approaches are used for selecting dimensions or subsets of the information space manually. Algorithmic computational procedures are applied for obtaining structures and patterns in the data automatically.

3.1.1 Interactive Preprocessing

The objective of interactive preprocessing is user-driven information structuring and reduction in order to determine the information which is relevant for the visualization. This is achieved by user controlled filtering out of nonrelevant information. Our framework provides several interaction methods, such as sliders, mouse-based visual selections, etc., in conjunction with different visual previews onto the data set in order to support information selection processes. These interactive procedures are useful because they allow direct considerations of users' domain knowledge and exploration tasks during the preprocessing. Basically our framework offers the following three interactive preprocessing methods:

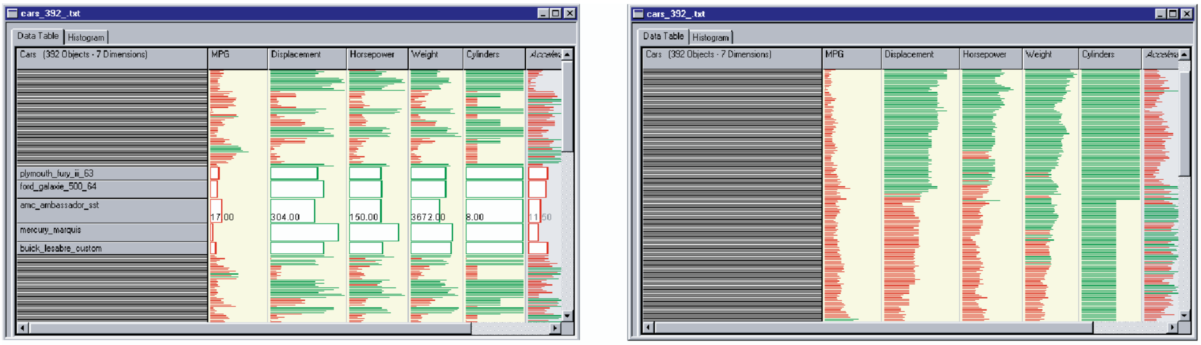
- **Interactive reduction of the number of dimensions** —Visual previews on multidimensional information sets are used for supporting the selection of those dimensions which might be most relevant for the visualization. These previews are created with a technique which we called Data-Table-View. The Data-Table-View reveals ranges of values, value distributions and correlations between dimensions in order to support a qualified selection of dimensions (see Section 4.1).
- **User-driven filtering of data ranges**—In conjunction with the preview, interactive sliders can be utilized for specifying value intervals such that only those information objects which fulfill predetermined value conditions are shown in the visualization.
- **Interactive hierarchy specification**—Based on the user-defined hierarchy approach introduced in [32], users can impose arbitrary hierarchical organizations on a given information set even if it is not a natural hierarchy. Due to this interactive strategy, users can bring domain-specific and task-specific knowledge to the hierarchy specification that can be utilized for obtaining structures and revealing patterns in the data.

3.1.2 Algorithmic Preprocessing

The algorithmic-based preprocessing approach exploits similarities between information objects in high-dimensional information 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 [5], 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. Let's 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 [18] are sufficient measures. In contrast to that, the Dot product or a Correlation coefficient [18] are appropriate if it is the intention to group firms with similar sales growth within that period of time. Thus, any 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. [18] for further information about these data types).



(a) (b)

Fig. 1. Table-based exploration of multidimensional data without (left) and with (right) similarity arrangement in order to reveal correlations.

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 [19], 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.

3.2 Scalable Visualization

An effective presentation of different aspects of a given information set including visualization of information structure or display of concrete attribute values requires the combination of different scalable visualization methods which can be adopted to specific exploration goals. Our scalable visualization framework provides several visualization techniques. Besides the Data-Table-View (cf. Fig. 1), Highfields and Icons (cf. Fig. 2), KOAN [21], and Parallel Coordinates [15], 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.

4 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 reducing both the dimensionality and the active data size to manageable levels. In this section, we will discuss some of the preprocessing techniques of our framework.

4.1 Interactive Reduction of the Information Space Supported by Visual Previews

A preview of the data in order to support qualified decisions as to whether a dimension of a multidimensional information set should be included into the visualization or not can be generated. We created a visual tool for this preview which we named the Data-Table-View. This method is very similar to the Table Lens introduced by Rao and Card [22]. The Table Lens integrates a common table, where information objects are arranged in lines, with graphical representations for depicting patterns and outliers in multidimensional information sets. It offers several graphical mapping schemes, along with a focus+context technique for exploring large tables effectively.

The Data-Table-View extends the Table Lens by introducing improved features for organizing the information objects within the table. This extended ordering mechanism,

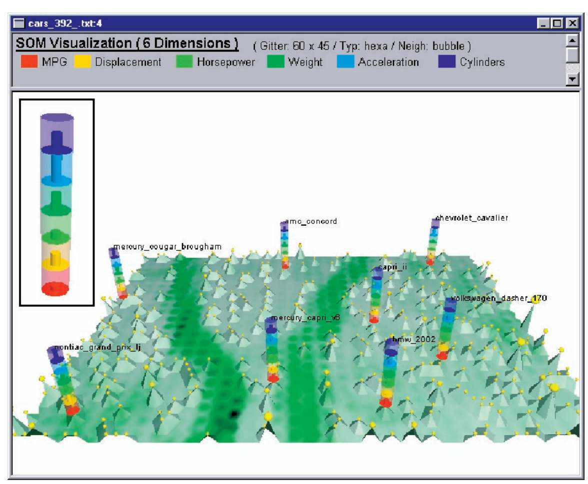


Fig. 2. Example of information organization based on self-organizing maps for a car information set.

which is based on Self-Organizing Maps [19] (cf. Section 4.2 for a brief discussion of the basic SOM algorithm), enables the analysis of correlations among table columns (dimensions of the information set). Our ordering approach does not sort the information objects within the table with respect to a particular dimension. Instead, all attributes are considered for the ordering process. Since our goal is to arrange information objects linearly for the data-table-view instead of organizing them on a two-dimensional grid, we are using the one-dimensional case of SOMs, which is proven (see [19]) to provide correct orderings as well. Thus, we obtain a sequence of information objects depending on their overall similarity in information space, i.e., similar objects are placed in successive table rows.

In order to reveal correlations among dimensions graphically, a bar representation is used where data values within table cells are mapped onto the length of a small bar. This principle is illustrated in Fig. 1. In our example, the table contains a car information set with 392 information objects and six dimensions. The left picture of Fig. 1 shows the data table without similarity arrangement of the information objects. The focus is set to a particular information object in order to reveal detailed data values. The similarity arrangement is applied in the right picture. Correlations among dimensions can be obtained very easily. Using this similarity arrangement reveals higher order correlations, i.e., correlations between more than two dimensions. This is shown in Fig. 1, where the first five dimensions are correlated.

This preview reveals valuable information for selecting relevant dimensions. Thus, a flexible interactive and qualified reduction of the dimensionality of arbitrary information sets is supported graphically.

4.2 Self-Organizing Maps

Self-organizing maps (SOM), as introduced by Kohonen [19], 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.

Fig. 2 illustrates the use of SOMs for structuring unorganized information spaces in our framework. The picture was generated from the information set of Fig. 2. 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 similarity.

Moreover, we introduce cylinder icons (cf. Fig. 2) for visualizing cluster properties, i.e., a small opaque cylinder is used for displaying the concrete value for each single variable of the map vectors. The height of the outer transparent cylinder corresponds to the maximum attribute value of the related dimension. Color is used to distinguish between the different dimensions. The different cylinders are composed into a single icon that is mapped on top of selected cluster peaks within the graphical representation.

Thus, SOMs are suitable for providing an overview of the entire information space by revealing clusters and cluster properties.

4.3 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 [18]. Based on one of the algorithms, Single Linkage, Complete Linkage, Average Linkage, Ward, Median, Flexible Strategy, and Zentroid [18], information objects IO are merged into groups according to their similarities in information space. Therefore, a symmetric $(n\ by\ n)$ 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 3.1.

$$S = \begin{bmatrix} s_{1,1} & \cdots & s_{n,1} \\ \cdots & \cdots & \cdots \\ s_{n,1} & \cdots & s_{n,n}, \end{bmatrix} \quad where$$
$$\begin{aligned} s_{i,j} &= s_{j,i} \quad \forall\, i,j \,=\, 1,\cdots,n \quad and \\ s_{i,i} &= 1 \qquad \forall\, i \,=\, 1,\cdots,n. \end{aligned}$$

The similarity matrix serves as a basis for a bottom up creation of a binary dendrogram (cf. Fig. 3 left).

We create the first group by merging the two most similar information objects. That is, we combine IO_i, IO_j for i, j, where $s_{i,j} = max$. Subsequently, a new $(n-1\ by\ n-1)$ 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 (cf.

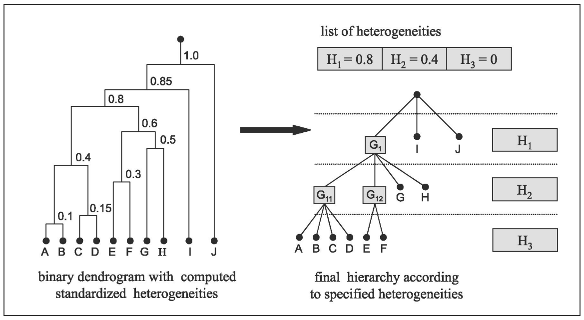


Fig. 3. Construction of the final Hierarchy tree with three levels based on the binary dendrogram.

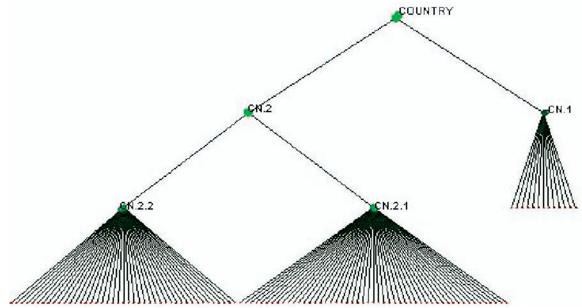


Fig. 4. Overview with three hierarchy levels.

Section 3.1) 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. Fig. 3) 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 from it (cf. Fig. 3). 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 effective 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 position of the inserted node is stored.
3. All new inserted nodes form new subtrees within the final hierarchy. Execute Steps 1-2 for all those stored nodes with the next value in the heterogeneity list.
4. Iterate Steps 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 and refining embodiments by increasing the number of hierarchy levels for determining more subtle patterns in the data. The final hierarchy tree contains information objects IO at its leaves. The remaining nodes represent clusters which fulfill the heterogeneity conditions associated with each hierarchy level. The principle of hierarchy refinement is depicted in Fig. 4 and Fig. 5. As the number of levels is increased, bigger clusters are split up into smaller subclusters. Thus, a stepwise exploration at arbitrary levels of detail is supported.

5 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, such as hierarchical information structures or unstructured multidimensional information spaces, requires several visualization methods or a combination of these methods. Therefore, our framework provides a range of different techniques. Besides known techniques (cf. Section 3.2), 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.

5.1 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 [20]. 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, including Graphical Fisheye Views [24] or the Hyperbolic Browser [20], which

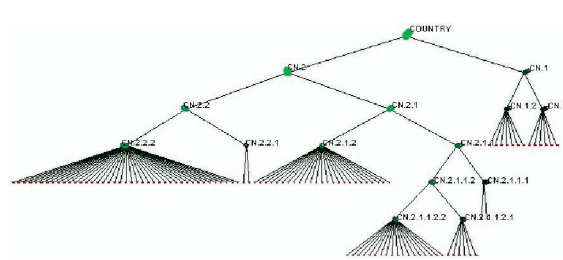
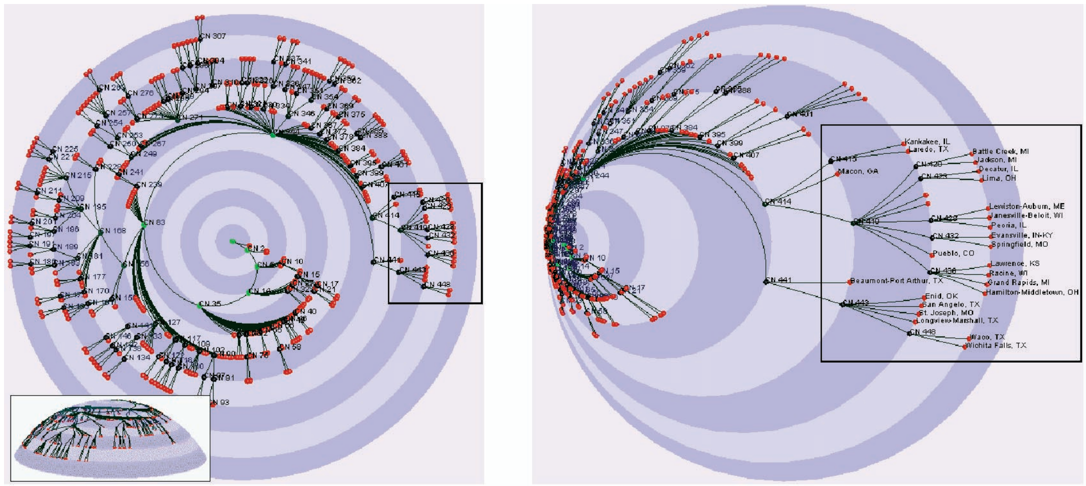


Fig. 5. Hierarchy refinement with seven levels.



(a) (b)

Fig. 6. Complex hierarchy graph with $\mathbf{p}_0$ at the origin (left) and with enlarged focus region (right).

exploit distortion to enlarge a focus area while preserving context information. In order to achieve an additional degree of freedom for focusing arbitrary areas of the hierarchy graph in conjunction with providing orientation support, 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.

5.1.1 Graph Mapping onto the Hemisphere

Laying out the hierarchy tree is done with a simple 2D algorithm which is similar to the algorithm of Reingold and Tilford [23]. 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 (λ, ϕ). Thus, the determined Cartesian coordinates can be mapped directly to spherical coordinates.

5.1.2 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 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. Fig. 7 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. Fig. 7 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 [20], we introduce additional degrees of freedom for browsing hierarchies since we use change of focus along with conventional 3D navigation. Fig. 6 demonstrates change of focus. Fig. 6 left shows a complex hierarchy graph mapped onto a hemisphere. The center of projection has been moved right in Fig. 6 in order to set the focus to the marked subgraph. We introduce colored rings for minimizing the amount of confusion introduced by the distortion.

5.1.3 Enhancements of the Magic-Eye-View

Despite depicting computed hierarchical abstractions, many visualization applications require the display of natural hierarchies as well (e.g., file systems, company structures, etc.). Therefore, one goal of our research is focused on enhancements of the Magic-Eye-View, especially with respect to scalability and a general usability of the approach. Basically, we have explored two major directions:

- Integration of adaptive features into our Focus+ Context technique in order to achieve effective Level of Detail strategies and navigational aids for the exploration of large graphs. We apply the research reported in [13] for enhancing the Magic-Eye-View. The basic idea here is to provide visual clues for navigating graphs by either interactive or automatic folding and unfolding subtrees in combination with mapping of so-called Strahler numbers. Strahler numbers, which denote the complexity of a node's subtree, are computed for all nonleaf nodes and mapped onto color and width of the incoming edges

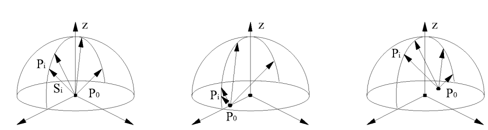


Fig. 7. Projection rays before and after moving P_0.

of these nodes. Thus, the visual mapping of the Strahler numbers indicates in which direction a tree really grows, i.e., where the user should go to explore interesting subtrees [13]. We found that the introduction of the features described above increases the number of manageable nodes by about three to four times.

- Adaptation of the enhanced Magic-Eye-View for visualizing hierarchies on pocket-sized devices with small displays such as PDAs (Personal digital assistant) or wireless web browsers. Currently, we are investigating methods for adapting our technique to PDAs. Since real interactive 3D visualizations of large graphs exceeds the capability of PDAs, we use a 2D representation of the Magic-Eye-View by projecting it onto a circular display region. At this early stage of our research, we believe that the extensions of the Magic-Eye-View stated above are valuable mechanisms for adapting our technique to very small screens. Addressing the problems of limited input facilities, we will provide additional support for navigating hierarchies by utilizing an interaction technique called the *event horizon* suggested in [27]. The key idea of this model is that the display can be compressed and expanded by moving objects radially farther away or closer to an event horizon in the middle of the screen. Using this principle in conjunction with a hierarchy tree, we obtain an intuitive method for showing only certain hierarchy levels, such as upper levels or lower levels.

Summarizing the discussion above, we suggest the combination of three different categories for interacting and navigating hierarchies on PDAs, which are: focusing arbitrary areas of the graph, as introduced with the Magic-Eye-View, navigational aids based on visual mappings, along with folding/unfolding of complex subtrees, and the utilization of the *event horizon* paradigm.

Due to these extended features, it is possible to support explorations at different levels of detail in order to facilitate *top-down*-like visualization scenarios such as:

- Start with an initial display of the graph with folded (hidden) subtrees for providing a first overview of the whole hierarchy. At this point, visual mappings of the Strahler numbers indicate the directions the graph really grows.
- Refine the embodiments interactively according to users' exploration tasks by unfolding "subtrees of interest" for revealing more subtle structures of the hierarchy.
- Reveal precise information details while maintaining context information by applying change of focus operations or *event horizon* interactions.

The principles of applying the adapted Magic-Eye-View on an HP Journada PDA are illustrated in Fig. 8.

5.2 Visualization of Multidimensional Information

We developed the new technique ShapeVis[1] for further exploration of multidimensional information sets (e.g.,

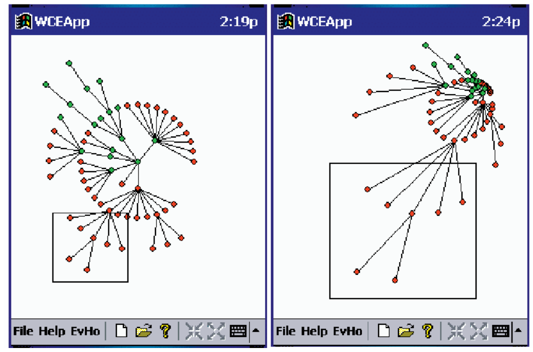


Fig. 8. Adapted Magic-Eye-View for hierarchy visualization on PDAs without and with focused subgraph.

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 two(three)-dimensional visualization space according to their attribute values. For reasons of readability, we briefly sketch the basics of our model in this section.

5.2.1 Enhanced Spring Model

Information objects IO are described by a set of n attributes which have continuous ranges of values. Thus, each IO in the n-dimensional information space is an 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., [14]) use a classical spring model for mapping objects from n-dimensional information space onto two(three)-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. For fixed $\mathbf{d}_i$, this location can be computed explicitly:

$$p = \frac{\sum_{i=1}^{n} c_i \cdot \mathbf{d}_i}{\sum_{i=1}^{n} c_i}. \tag{6}$$

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 $\mathbf{p}$ moves toward $\mathbf{d}_i$. Furthermore, objects with similar properties are spatially close in the visualization. Despite 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. [29]).
2. *Insensitivity against coordinate scalings:* The information objects $(c_1, \ldots, c_n)$ and $(c_1 \cdot k, \ldots, c_n \cdot k)$ with

1. We use an adapted version of our technique introduced in [29] within the framework.

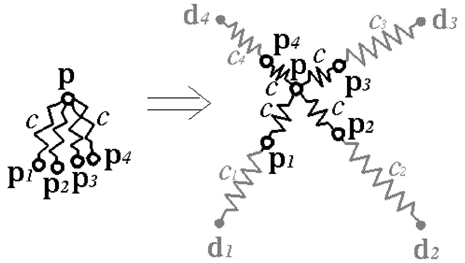


Fig. 9. Enhanced spring model for an information object IO in four-dimensional information space.

$k > 0$ 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 of 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. Fig. 9 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 (7)-(9):

$$\mathbf{p} = \frac{\sum_{i=1}^{n} w_i \cdot \mathbf{d}_i}{\sum_{i=1}^{n} w_i} \tag{7}$$

with

$$w_i = \frac{c_i}{c + c_i} \quad for\ i = 1, \ldots, n. \tag{8}$$

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

$$\mathbf{p}_i = \frac{c \cdot \mathbf{p} + c_i \cdot \mathbf{d}_i}{c + c_i} \quad i = 1, \ldots, n. \tag{9}$$

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.

5.2.2 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 (cf. [29]) for obtaining an intuitive imagination of the locations of $\mathbf{p}, \mathbf{p}_1, \ldots, \mathbf{p}_n$. Approximating point locations using free-form-surfaces becomes rather difficult when the number of information objects is 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).

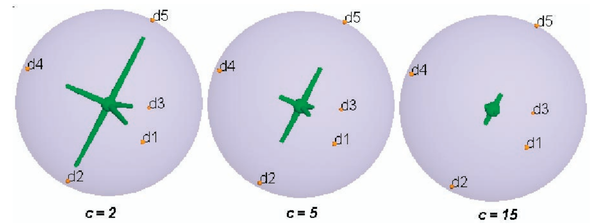


Fig. 10. Visualizing an object (1, 2, 1, 1, 2) with different parameters c.

Therefore, we propose basic primitives (n cylinders) for composing geometric objects out of $\mathbf{p}, \mathbf{p}_1, \ldots, \mathbf{p}_n$. 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 length of each cylinder depend on $\mathbf{p}, \mathbf{p}_1, \ldots, \mathbf{p}_n$ and the constant c. Thus, the geometric objects describe the information objects $IO = (c_1, \ldots, c_n)$ uniquely. This principle and the influence of parameter c are illustrated in Fig. 10. 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 find 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 in the direction of a certain $\mathbf{d}_i$ denotes a large data value of the corresponding c_i of this object). Fig. 11 illustrates this principle. Our approach is applied to

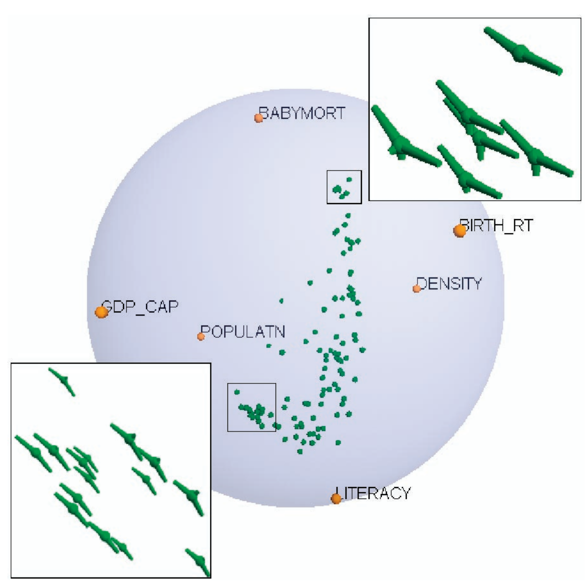


Fig. 11. Visualization of a demographic data set with six dimensions and deformable geometric objects.

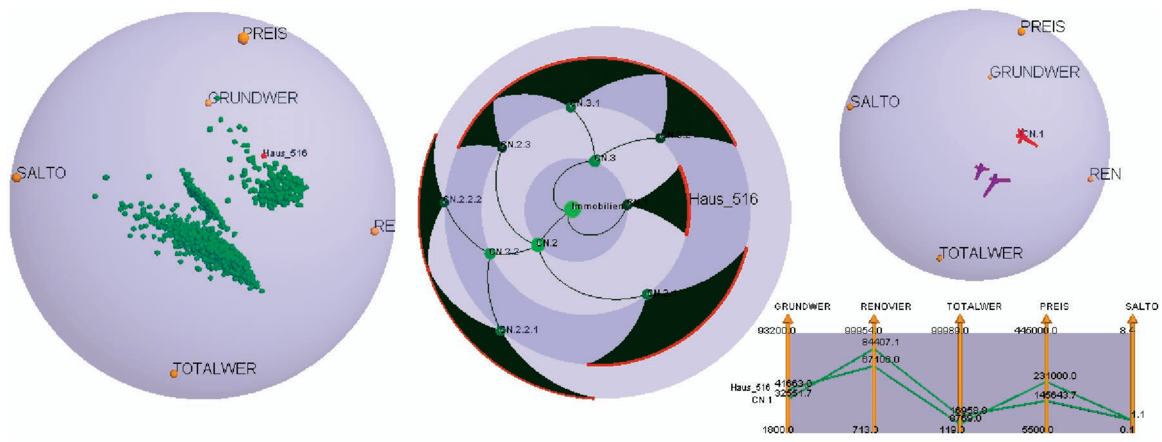


Fig. 12. Combination of Hierarchies, ShapeVis, and Parallel coordinates.

an information set which measures six demographic parameters of 106 countries. We placed one point $\mathbf{d}_i$ for each dimension of the information space in an 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 toward the corresponding 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 toward the *Baby mortality* and *Birthrate* dimension points are much longer than the cylinders which point toward the remaining $\mathbf{d}_i$ (cf. Fig. 11 magnification of the upper cluster). In contrast to that, the cluster in the 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.

5.3 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 such as real attribute values, arbitrary objects can be selected and visualized with parallel coordinates [15]. Labeling the coordinate axis and displaying the data values provides more detailed information about each information object.

Fig. 12 illustrates the combination of ShapeVis, Magic-Eye-View, and Parallel Coordinates applied to an information set which describes 2,440 houses with five attributes. The left picture shows the 2,440 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 points at the surface of the sphere. In this case, it is more meaningful to preprocess the data, as introduced in Section 4, in order to form manageable subsets. The picture in the middle of Fig. 12 depicts the hierarchical representation of the information 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 subclusters at the following levels. These subclusters can be selected for further exploration.

We selected the first level of the hierarchy tree and obtained the picture on the 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 on the 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*.

6 Marching Sphere

In many application domains (such as 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 frames 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 drawbacks. 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 visual representation of the information can be easily 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 objects 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.

6.1 Displaying Information Structure

We use abstract three-dimensional graphs for displaying information objects and revealing structural relations between information units. Basically, we apply a technique called *KOAN* [21] (KOntext ANalysator), originated by SIEMENS. *KOAN* maps information objects from high-dimensional information space onto three-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.

6.2 Displaying the Spatial Frame of Reference

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

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 Fig. 13. The picture shows a map of the German federal state Mecklenburg-Vorpommern in the four different resolutions —state, county, zip code area, and community.

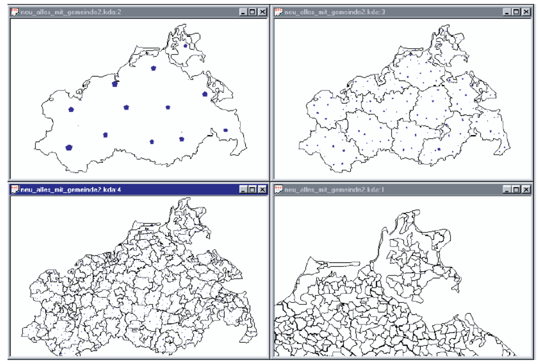


Fig. 13. Refinement of the geographical granularity.

6.3 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 two-dimensional map is rendered in a virtual 3D scene. The three-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 subarea of a geographic map becomes increasingly difficult as the number of subareas 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 subareas.

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

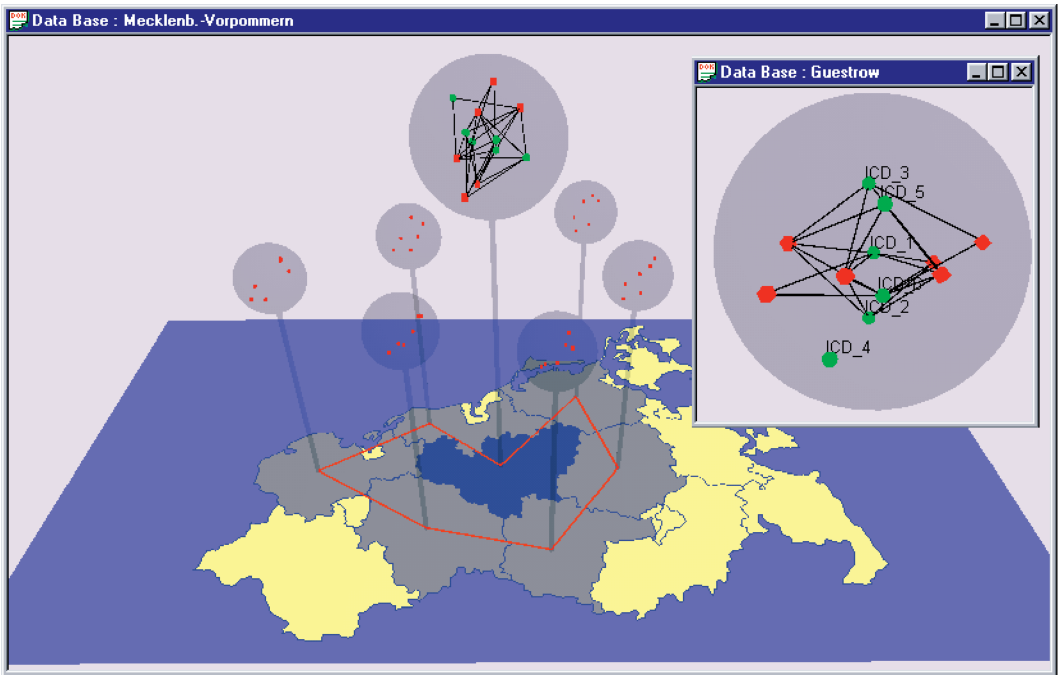


Fig. 14. *Marching Sphere* applied to spatially related health information of the federal state Mecklenburg-Vorpommern.

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 subarea of the selected area.)

Fig. 14 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 a three-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 (cf. Fig. 14 upper right).

7 Conclusions and Future Work

This paper proposed a general framework for information visualization. The integration of preprocessing and visualization enables exploration of large information spaces 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 finding patterns in unstructured data. 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 multidimensional 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, further evaluation of the framework needs to be performed to determine its effectiveness and to verify its general 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 by improving the effectiveness of the dendrogram calculation. 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. One idea which we would like to study is the utilization of SOMs for solving 2D and 3D arrangement problems. Furthermore, we would like to investigate animations for smoothly fading in and out information graphs in the *Marching Sphere*.

ACKNOWLEDGMENTS

The authors wish to thank Prof. Schütt and Dr. Kolpatzik from Siemens AG Munich for their constant support. They want to thank Holger Theisel and Norma López for many ideas and discussions. Thanks also to Thomas Bürger, Thomas Nocke, and Axel Oestereich for developing parts of the source code.

REFERENCES

[1] "WEBSOM—Self-Organizing Maps for Internet Exploration," http://websom.hut.fi/websom/, 1996.

[2] K. Andrews, J. Wolte, and M. Pichler, "Information Pyramids (TM): A New Approach to Visualizing Large Hierarchies," *Proc. IEEE Visualization '97,* A. Varshney and D.S. Ebert, eds., Oct. 1997.

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

[4] M. Ankerst, "Visual Data Mining with Pixel-Oriented Visualization Techniques," *Proc. Workshop Visual Data Mining,* 2001.

[5] M. Ankerst, S. Berchthold, and D. Keim, "Similarity Clustering for an Enhanced Visualization of Multidimensional Data," *Proc. IEEE Information Visualization '98,* G. Wills and J. Dill, eds., Oct. 1998.

[6] S. Benford, D. Snowdon, R.I.C. Greenhalgh, I. Knox, and C. Brown, "VR-VIBE: A Virtual Environment for Co-Operative Information Retrieval," *Computer Graphics Forum,* H.-P. Seidel and P.J. Wills, eds., vol. 14, no. 3, Aug. 1995.

[7] L. Beudoin, M.-A. Parent, and L. Vroomen, "Cheops: A Compact Explorer for Complex Hierarchies," *Proc. IEEE Visualization '96,* R. Yagel and G.M. Nielson, eds., Oct. 1996.

[8] S. Card, G. Robertson, and J. Mackinlay, "The Information Visualizer—An Information Workspace," *Proc. ACM SIGCHI '91 Conf. Human Factors in Computing Systems,* 1991.

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

[10] M. Gross, T. Sprenger, and J. Finger, "Visualizing Information on a Sphere," *Proc. IEEE Information Visualization '97,* J. Dill and N. Gershon, eds., Oct. 1997.

[11] M. Hemmje, "LyberWorld—A 3D Graphical User Interface for Fulltext Retrieval," *Proc. ACM SIGCHI '95,* 1995.

[12] R. Hendley, N. Drew, A. Wood, and R. Beale, "Narcissus: Visualizing Information," *Proc. IEEE Information Visualization '95,* N. Gershon and S. Eick, eds., Oct. 1995.

[13] I. Herman, M. Delest, and G. Melançon, "Tree Visualisation and Navigation Clues for Information Visualisation," *Computer Graphics Forum,* vol. 17, no. 2, pp. 153-165, June 1998.

[14] P. Hoffmann, G. Grinstein, and E.I. Grosse, "Dna Visual and Analytic Data Mining," *Proc. IEEE Visualization '97,* R. Yagel and H. Hagen, eds., Oct. 1997.

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

[16] C. Jeong and A. Pang, "Reconfigurable Disc Trees for Visualizing Large Hierarchical Information Space," *Proc. IEEE Information Visualization '98,* G. Wills and J. Dill, eds., Oct. 1998.

[17] B. Johnson and B. Shneiderman, "Tree-Maps: A Space Filling Approach to the Visualization of Hierarchical Information Structures," *Proc. IEEE Visualization '91,* 1991.

[18] L. Kaufman and P.J. Roussew, *Finding Groups in Data—An Introduction to ClusterAnalysis.* Brussels: John Wiley & Sons, 1990.

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

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

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

[22] R. Rao and S. Card, "The Table Lens: Merging Graphical and Symbolical Representations in an Interactive Focus+Context Visualization for Tabular Information," *Proc. Conf. Human Factors in Computing Systems,* Apr. 1994.

[23] E. Reingold and J. Tilford, "Tidier Drawing of Trees," *IEEE Trans. Software Eng.,* vol. 7, no. 2, pp. 223-228, Feb. 1981.

[24] M. Sarkar and M. Brown, "Graphical Fisheye Views," *Comm. ACM,* vol. 37, no. 12, pp. 73-84, Dec. 1994.

[25] M. Spenke, C. Beilken, and T. Berlage, "FOCUS: The Interactive Table for Product Comparison and Selection," *Proc. Ninth Ann. ACM Symp. User Interface Software and Technology (UIST '96),* 1996.

[26] T. Sprenger, R. Brunella, and M. Gross, "H-BLOB: A Hierarchical Visual Clustering Method Using Implicit Surfaces," *Proc. IEEE Visualization,* T. Ertl, B. Hamann, and A. Varshney, eds., 2000.

[27] A. Taivalsaari, "The Event Horizon User Interface Model for Small Devices," Technical Report TR-99-74, Sun Microsystems Laboratories, http://www.sunlabs.com/technical-reports/1999/, 1999.

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

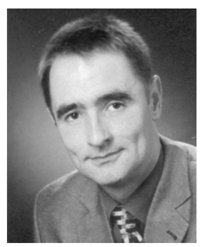
[29] H. Theisel and M. Kreuseler, "An Enhanced Spring Model for Information Visualization," *Computer Graphics Forum,* vol. 17, no. 3, Sept. 1998.

[30] S. Wendt, *N ichtphysikalische Grundlagen der Informationstechnik.* Berlin: Springer Verlag, 1991. (In German).

[31] C. Westphal and T. Blaxton, *Data Mining Solutions—Methods and Tools for Solving Real-World Problems.* New York: John Wiley & Sons, 1998.

[32] R.M. Wilson and R.D. Bergeron, "Dynamic Hierarchy Specification and Visualization," *Proc. IEEE Information Visualization '99,* G. Wills and D. Keim, eds., Oct. 1999.

[33] 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," *Proc. IEEE Information Visualization '95,* N. Gershon and S. Eick, eds., Oct. 1995.



Matthias Kreuseler is a PhD candidate and has a research position at the University of Rostock. He received the MA degree in computer science from the University of Rostock (1996). His work and research is focused on information visualization, visual data mining, and computer graphics. He has been working on the development of new techniques for visualizing multidimensional and hierarchical information as well as on a scalable framework for information visualization. Recently, he worked as a research intern at the AT&T Shannon Labs, New Jersey, on the navigation and visualization of very large graphs.

Heidrun Schumann graduated from the University of Rostock (1977 Master's degree, 1981 PhD, 1989 Habilitation). She is the head of the Computer Graphics Research Group at the Department of Computer Science in Rostock. Her research profile covers scientific and information visualization, mobile and context-driven visualization, photorealistic rendering, multimedia presentation generation, and image transmission and display. She headed the cross-institutional research group MoVi—Visualization of Multimedia Information on Mobile Computer Systems, supported by the German Research Society DFG. Her current research projects supported by research institutions or industry focus on the development of a scalable framework for information visualization, on the visualization of climate data and human health data, as well as on problems of user defined image transmission.