

DRAFT



RESEARCH ARTICLE

Visualization of Attributed Hierarchical Structures in a Spatio-Temporal Context

S. Hadlak*, C. Tominski, H.-J. Schulz, and H. Schumann

University of Rostock, A.-Einstein-Str. 21, D-18059 Rostock, Germany

(Month Day, 2010)

When visualizing data, spatial and temporal references of these data often have to be considered in addition to the actual data attributes. Nowadays, structural information is becoming more and more important. Hierarchies, for instance, are frequently applied to make large and complex data manageable. Hence, a visual depiction of hierarchical structures in space and time is required.

While there are several techniques addressing specific aspects of spatio-temporal visualization, approaches that cope with space, time, data, *and* structure are rare. With this paper we take a step to fill this gap. By combining various well-established concepts we achieve a reasonably complete visualization of all of the aforementioned aspects, where our focus is on hierarchical structures. We embed hierarchies directly into regions of a map display using variants of the point-based layout. Layering and animation are applied to visualize temporal aspects. Depending on analysis goals, users can switch between representations that emphasize data attributes or hierarchical structures. Interaction techniques support users in navigating the data and their visualization. We demonstrate the usefulness of our approach by adapting it to implement a visualization for spatio-temporal human health data.

Keywords: Hierarchy visualization; Geo-visualization; Time visualization

1. Introduction

Data visualization in GIS commonly considers qualitative and quantitative data attributes that are given in a spatial frame of reference. In many cases these data attributes are subject to change, that is, they depend on time. Traditional approaches are capable of effectively representing such data on the regions of a map (e.g., animated choropleth maps).

*Corresponding author. Email: hadlak(at)informatik.uni-rostock.de

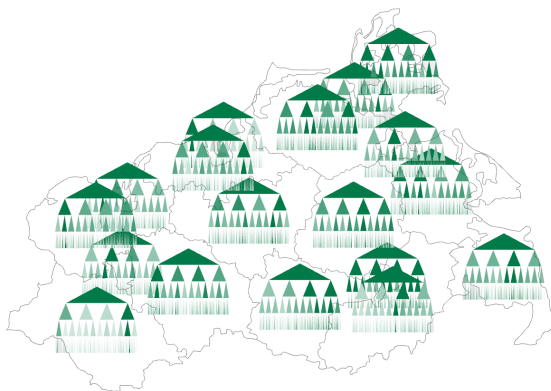


Figure 1. A map showing counties, each of which superimposed with a data hierarchy.

In our research, we face the challenge of visualizing not only plain data values per region, but a whole hierarchy¹ of data values per region, where the structure of the hierarchy *and* its associated data values are subject to change. The literature provides a wealth of approaches to create expressive visual mappings of hierarchies. However, as Figure 1 demonstrates, simply superimposing layouts of the hierarchies on the regions of a map is not a good solution. The problem is that a layout could obscure region borders and could overlap with other regions or with other hierarchies, leading to occlusion of possibly important information. To mitigate these problems, we developed novel methods for visualizing attributed hierarchical structures in space and time.

In Section 2, we explain basic definitions that will be used throughout this article and review related work in visualizing spatio-temporal information. In Section 3, we introduce a general approach for visualizing attributed hierarchies that have an anchor in geographic space. To overcome the problem of overlapping and cluttering we went for a direct embedding of the hierarchy into the map display by using a point-based layout, which efficiently uses the available screen real estate. Animation, multiple views, and a three dimensional layering approach, similar to the classic space-time-cube, are applied to encode temporal aspects. Data attributes that are attached to the hierarchy are visualized by color-coding and dedicated 3D glyphs. Interaction facilities allow users to navigate in space and time. We demonstrate our approach in Section 4 by focusing on the visual representation of the hierarchical structure of time and associated data values from spatio-temporal human health data. Conclusion and future work are given in Section 5.

2. Basic Notations & Related Work

Next, we briefly explain basic notations that we will use through this article. Then, a review of related work will be presented including visualization of spatio-temporal data and visualization of hierarchies in space and time.

2.1. Basic Notations

Graph structures in general describe entities and relations between these entities. Our approach focuses on hierarchies (i.e., directed acyclic graphs) and associated data given

¹The data hierarchies we consider here are not related to a hierarchical subdivision of space as commonly seen in GIS, but may be arbitrary hierarchical structures (e.g., structure of time, clustering hierarchies, or organizations).

in a spatial and temporal frame of reference. We assume that geographic *space* is modeled as a set of disjoint two-dimensional *regions*

$$S = \{R \mid R \subset \mathbb{R}^2\}$$

where $R \neq Q \rightarrow R \cap Q = \emptyset$ holds for any two regions $R, Q \in S$. The time axis is defined as an ordered set of time steps

$$T = (t_1, \dots, t_n)$$

with $n \in \mathbb{N}$. We define

$$H_{R,t_i} = (V_{R,t_i}, E_{R,t_i})$$

as the attributed hierarchy for region $R \in S$ at time step $t_i \in T$. Such a hierarchy consists of a set of *nodes*

$$V_{R,t_i} \subset V, \quad V = \bigcup_{R \in S, t_i \in T} V_{R,t_i}$$

of which one is a designated *root* node $r_{R,t_i} \in V_{R,t_i}$ and a set of edges

$$E_{R,t_i} \subseteq (V_{R,t_i} \times V_{R,t_i}) \subset E, \quad E = \bigcup_{R \in S, t_i \in T} E_{R,t_i}.$$

Data attributes of nodes and edges are defined by means of mappings

$$d_V : S \times T \times V \rightarrow \{D_1^V \times \dots \times D_p^V\}$$

$$d_E : S \times T \times E \rightarrow \{D_1^E \times \dots \times D_q^E\}$$

where D_*^V and D_*^E are the domains of *node attributes* and *edge attributes*, respectively.

There are various examples for data that follow this definition. One example regards spatio-temporal data that are too large to handle in a simple manner. Hierarchical aggregation often helps in dealing with huge data sets. Depending on analysis tasks and goals, it is not uncommon that an aggregation hierarchy is computed for each region and each time step, that is, the hierarchy may be different from region to region and from time step to time step. Another example are data that are given with respect to an existing hierarchy, as for instance the hierarchical structure of time itself (i.e., year, quarter, month, etc.). In this case, the hierarchical structure remains fixed, but the attributes of the nodes of the hierarchy vary over space and time. For a last example, we may think of organizational structures, e.g., prime minister, ministers, chief officers, etc. of administrative geographical units. Such structures obviously differ among countries and change over time.

From the previous definition and examples one can see that several aspects need to be taken care of for visualization. First, multiple hierarchies must be embedded in the spatial frame of reference. In fact, each geographic region has its own sequence of hierarchies. How to visualize the sequences in order to represent changes over time is the second question to be investigated. Finally, the data attributes associated with nodes and edges should be communicated. The attributes may be external data attached to the graph

structure or may be internal system-computed properties derived from the hierarchy (e.g., depth).

Spatial and temporal references also demand for appropriate interaction techniques: Browsing back and forth in time to view different parts of the time axis, as well as navigating in space to visit different geographic regions are crucial for data exploration.

2.2. Related Work

To our knowledge, the aforementioned needs have been satisfied only partially in previous work. Nonetheless, the literature provides a variety of interesting solutions for specific aspects.

2.2.1. Spatio-Temporal Data Visualization

The visualization of geo-spatial data is mainly based on map displays (see MacEachren (1995)). Cartography and geovisualization provide approaches and guidelines that help in generating efficient map displays in concert with expressive techniques for presenting data on (interactive) maps (see Slocum *et al.* (2008)). When visualizing time-dependent data, there are basically two options to choose (see Aigner *et al.* (2007)). First, one can map time (in the data) to time (in the real world), commonly implemented as animations or slide shows, or as interactive navigation in time. Second, it is possible to map data time to presentation space. All techniques that align data items with a time axis belong to this class.

Approaches that have to visualize both spatial and temporal dependency commonly pursue a combination of map displays with additional temporal encoding (see Kraak and Ormeling (1996), Andrienko and Andrienko (2006)). Animated maps are used in many cases, since they provide a good overview and convey major trends in the data. However, Tversky *et al.* (2002) pointed out that detailed visual comparison is difficult with animations. In this case, techniques that show several time steps in a single visual representation are better suited; small multiple displays (see Tufte (1986)) or the classic space-time-cube (see Kraak (2003)) are prominent examples.

Most techniques for spatio-temporal visualization focus on data, not on hierarchical structures. Such structures are often seen as isolated objects for which separate visualization techniques can be applied.

2.2.2. Approaches to Represent Hierarchies

As Battista *et al.* (1999) describe, classic node-link displays require a spatial layout of the hierarchy. Various algorithms are known to generate linear or radial layouts in two-dimensional, three-dimensional, and also hyperbolic space (e.g., Reingold and Tilford (1981), Robertson *et al.* (1991), Munzner (1997)). In contrast to node-link displays, which explicitly visualize node relations (i.e., edges), implicit hierarchy visualizations generate special node arrangements that allow users to derive relationships mentally. The most prominent example are classic treemaps by Shneiderman (1992) and their manifold derivatives (see Schulz *et al.* (2010a) for a survey).

Just like spatio-temporal data visualization often neglects structural aspects, so does classic hierarchy visualization disregard spatio-temporal aspects. Nonetheless, first steps have been taken to combine hierarchies and space as well as hierarchies and time.

2.2.3. Representing Hierarchies in Space

Multiple views are one possibility to represent hierarchical structure with a spatial reference. The idea is to provide interactive linking facilities to allow users to mentally associate map display and hierarchy visualization, which are shown in separate views. An expressive example is Jern *et al.* (2009)'s combination of treemaps and choropleth maps. On the other hand, adapted maps like cartograms or RecMaps by Heilmann *et al.* (2004) are used. Such maps are better suited to integrate hierarchies into the map display. A similar approach is pursued by Wood and Dykes (2008). They modified treemaps in a way such that positions of nodes in the treemap layout reflect node locations in geographical space. Despite these examples, however, not much research has investigated a direct integration of hierarchy representations into regions of maps.

2.2.4. Representing Hierarchies in Time

Two concepts assist in visualizing time-varying hierarchies. One is the cumulative approach, which creates a super hierarchy that unites nodes and edges of all time steps. This super structure can be visualized to gain a static overall view of the hierarchy. An example are Tu and Shen (2007)'s Contrast Treemaps, which support visual comparison. The other class of approaches is based on generating sequential views on time-varying hierarchies. The idea is to visualize the hierarchy at each time step, for instance by means of animations showing one time step after the other (e.g., TimeTrees by Card *et al.* (2006)), or by drawing the hierarchy on different temporal layers (e.g., Code Flows by Telea and Auber (2008)). Conceptually both examples corresponds to mapping time to time (in an animation) and time to space (for layering), respectively.

In summary, we see that several solutions do exist to visualize data in space and time and to visualize hierarchies (isolated, in space, *or* in time). However, techniques to combine data and hierarchy visualization and representations of spatial and temporal contexts are still scarce. In the next section, we present an approach that offers such a combined strategy as a piece to fill this gap.

3. Visualizing Hierarchies in Space and Time

Our general approach addresses four issues: (1) Embedding of a hierarchy representation into the map display, (2) visualization of temporal aspects, (3) encoding of data associated with the hierarchy, and (4) appropriate interaction for navigation in space and time.

3.1. Embedding Hierarchies in Irregular Shaped Regions

Due to the number of aspects that need to be visualized, we have to use screen real estate efficiently. Therefore, we went for a direct embedding of the hierarchy into the map display. A problem concerning this embedding are the usually irregular shapes of geographic regions. Most hierarchy layouts, however, assume specific shapes, for example rectangles or circles, and cannot be adapted easily to handle irregular shapes. On the other hand, an algorithm that adapts the layout of a given hierarchy to irregular shapes can lead to very different visual results. In other words, layouts could vary a lot for the regions of a map, impeding comparison of data associated with the hierarchy.

To tackle these problems, we propose two different approaches for embedding a hierarchy into an irregular region. The first approach is more data driven and aims to use the space efficiently showing a maximum number of nodes and therefore most of the

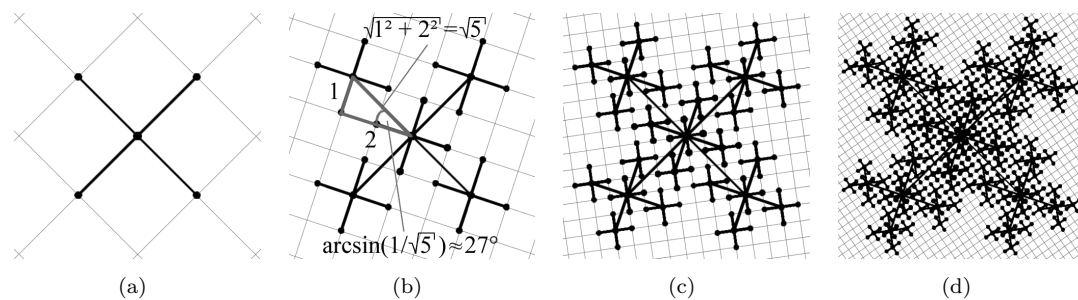


Figure 2. Computing a layout in a point-based manner.

associated data. The second approach sacrifices adaptiveness to irregular shapes in favor of better readability of the visual representation. It is less space-efficient, but facilitates comparison tasks. Both approaches are extensions of the point-based layout that will be briefly introduced at first.

3.1.1. Point-Based Layout

The point-based approach described by Schulz *et al.* (2010b) uses a space-efficient layout that allocates a unique pixel for each hierarchy node. The mechanism behind this approach is inspired by point-based computer graphics. It can be summarized as follows:

- (a) The root node is placed in the center and its first 4 children are arranged around it forming a regular grid (see Figure 2(a)).
- (b) The positions of the next 4 children of the root are obtained by rotating the grid by $\approx 27^\circ$ and scaling it by a factor¹ of $1/\sqrt{5}$. The first 4 children of each of the previously laid out nodes are positioned with the very same rotation and scaling scheme (see Figure 2(b)).
- (c) This procedure is repeated to layout the next 4 children of the root node, the next 4 children of the nodes laid out in step (a), and the first 4 children of the nodes that have been added in step (b) (see Figure 2(c)).
- (d) The procedure continues until all children have been associated with a unique layout position (see Figure 2(d)).

The rotation and scaling scheme generates a fixed layout that can be pre-computed independently. To visualize a concrete hierarchy, the only necessary step is to assign nodes to the pre-computed node positions, which is done in a well-defined way. In case the number of children is not divisible by 4, positions are left unoccupied. Due to pre-computation and use of unique pixels, larger hierarchies can be visualized effectively.

3.1.2. Layout to Use Space Efficiently

Thanks to the fixed nature of the point-based layout we can quite easily adapt the final shape of the hierarchy representation by neglecting certain locations during the assignment phase. To achieve a better adaptation of the hierarchy layout to the underlying geographical region, we operate on *areas* subdivided into *subareas*. We use the term *area* to denote the display space covered by a region. Note that in this sense, subareas are just a geometrical means for computing the layout of subtrees; they are not related to any organizational structure of geographical space as we are not visualizing the spatial hierarchy. Subtrees will be assigned to subareas depending on their sizes. We suggest the following recursive procedure for finding a good placement of subtrees, eventually resulting in a layout of the hierarchy:

¹These two factors describe the relation between the different regular grids shown in Figure 2 and guarantee a space filling placement. More information on this and other pairs of factors can be found in Schulz *et al.* (2010b).

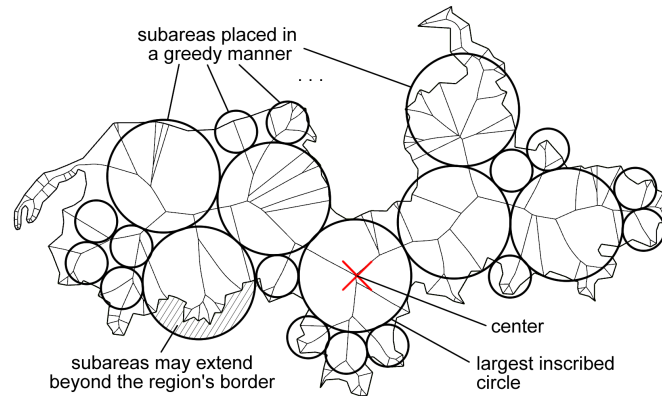


Figure 3. Subdivision of an irregular area along its skeleton.

- (1) Compute center of area
- (2) Subdivide area into subareas
- (3) Map subtrees onto subareas
- (4) Apply procedure recursively

To embed a hierarchy H_{R,t_i} into the area A corresponding to the region $R \in S$ one starts with the computation of the center $c \in A$. Secondly, A is subdivided into k subareas A^1, \dots, A^k , where k is the number of subtrees H_{R,t_i}^j ($1 \leq j \leq k$) below the root r_{R,t_i} . Then, subareas and subtrees are sorted by size and are assigned to each other accordingly. This procedure continues recursively for each subarea and its associated subtree. While steps 3. and 4. are straight-forward, steps 1. and 2. should be explained in more detail.

Step 1: Compared to regular and convex shapes, it is not a simple task to determine a central position c of an irregular area A , which is necessary to place the root node of a subtree. Obviously, a center has to be inside the area and it should have a maximum distance to the area boundary. Several possibilities exist to choose a center point, but none is without problems. For example, the barycenter of a concave area may be outside of that area, while the center of the largest inscribed circle may turn out to be far from being central. We utilize the barycenter, the center of the largest inscribed circle, and the skeleton point with the highest importance as described by Telea and van Wijk (2002) and evaluate the results of the different methods. For each center candidate, we compute the visibility polygon, namely that part of the area that can be seen from the candidate. Since one can interpret this as a measurement of centrality, we choose the candidate with the largest visibility polygon.

Step 2: The chosen center is the starting point for area subdivision. The first subarea A^1 is defined as the largest inscribed circle around the area's center $c \in A$ (see Figure 3). In a greedy manner, we then place further subareas of the same size as A^1 along the area's skeleton. If no further area of that size can be placed the greedy procedure continues with a search for smaller subareas¹. We stop this procedure when the size of subareas is getting too small (ca. 1% of the size of A^1) or a user chosen number m of subareas has been created. Both termination conditions are independent of the hierarchy, which allows us to precompute the subdivision and reuse it for different hierarchies. Depending on application needs, other shapes than circles may be used as long as they fit entirely inside the inscribed circle of A^1 . Furthermore, subareas do not necessarily have to lie completely inside the irregular area's shape, but may slightly extend beyond it.

After the subdivision has been computed, we assign the k subtrees of a hierarchy to the m subareas. If $m < k$, we start splitting the largest A^i into further subareas to generate

¹Searching for subareas of size $\frac{1}{5}A^1, \dots, \frac{1}{54}A^1$ has proven practical.

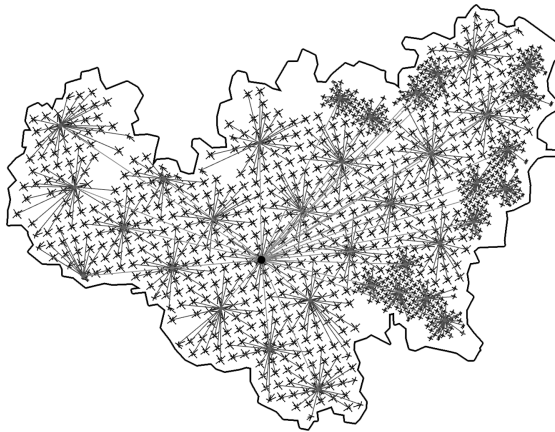


Figure 4. A hierarchy layout with 5517 nodes embedded into an irregular region.

enough subareas. If $m > k$ we can even reunite some A^i to get closer to k , and thus, to use space more efficiently. Then, we continue with the procedure recursively. In summary, this approach is capable of embedding a hierarchy layout directly into a 2-dimensional geographic region of irregular shape, while being computationally and space efficient. A result of this layout is illustrated in Figure 4.

3.1.3. Layout to Facilitate Visual Comparison

The previously described algorithm adapts the layout of a hierarchy quite well to the shape of a region. However, this comes at the price of losing comparability between regions. The most simple solution to ensure comparability is to restrict the layout to the largest inscribed circle of the corresponding area only. Then all layouts would be easily comparable varying only in size. However, space would not be used efficiently as space outside the inscribed circle would be left unused. We need to find a compromise between being adaptive for efficient use of space and not being adaptive to cater for comparability of layouts.

To this end, we use the space-efficient adaptation only down to a certain level of the hierarchy, and a fixed layout is used for the lower levels. This means that at higher levels of the hierarchy a coarse adaptation to the underlying shape is achieved, while lower levels are kept comparable between regions by resorting to a non-adaptive layout. The level at which to switch the layout strategies is dependent on the data and may also be set interactively by the user.

We implement this concept by combining a modified version of our space-efficient layout with a regular non-adaptive layout (e.g., the point-based layout or any of the layouts mentioned in Section 2.2.2). In the first step, a low number of hierarchy levels¹ is laid out as described in the previous section, but with the following modifications:

- There will be no merging of subareas if $m > k$.
- The subareas placed on the chosen depth level must not extend beyond the irregular area's shape.

These restrictions are necessary to ensure that the same initial shape, varying only in size, is available for the fixed layout of lower subtrees. Additionally, these subareas are ordered (e.g., by their x-coordinate) to generate a replicable visual mapping for each region and thus to simplify the search for similar subtrees across different regions. Then, in the second step, the layout for the remaining subtrees is computed using standard al-

¹We suggest a maximum depth of three to avoid too small subtrees at lower levels.

gorithms that produce comparable visual representations. An example for this combined layout strategy will be showcased later in Section 4. In the case that the sizes of the regions differ too much or have to correctly reflect the statistical data associated with the hierarchy (e.g., the population of the region) a combination of our approach with well established transformations of the map display (e.g., cartograms) is possible.

3.2. Representing Temporal Aspects

For hierarchies, several changes may occur over time. First, the sets of nodes and edges may change from one time step to another, that is $V_{R,t_i} \neq V_{R,t_{i+1}}$ and $E_{R,t_i} \neq E_{R,t_{i+1}}$ for $R \in S$ and $t_i, t_{i+1} \in T$. Nodes can be added or deleted. The same holds true for edges, which also implies that a node can move in the hierarchy – exactly then when it is connected via different edges. As node and edge sets change, so do global properties of the hierarchy (e.g., maximum depth) and node attributes (e.g., number of children). We also indicated that external data attributes d_V and d_E , which are associated with nodes and edges, respectively, are usually subject to change. Often the analysis of the evolution of data attributes over time is the primary goal of visualization.

We combine several concepts to communicate temporal aspects of the data. For an initial overview, simple animation of the map display is useful. When it comes to discerning details in the course of time, visualizations that show a sequence of time steps are better suited. Therefore, we follow two different approaches. These are three dimensional layering and multiple views. In the layering approach, the third dimension of the presentation space represents the time axis (analog to the space-time-cube approach, see Kraak (2003)). To this end, we consider a subset of time steps, by default a successive series (t_i, \dots, t_{i+s}) with $1 \leq i < i + s \leq n$, but the user can override this default with any ordered set of time points. For each time step we render a separate layer L_i representing the map and the embedded hierarchy layouts. These semitransparent layers are aligned with the time axis that emanates perpendicularly from the base map layer L_i . To facilitate identification of the aforementioned changes in between two layers, visual cues are added (see Figure 5):

- Differently colored *links* connect subsequent layers to indicate nodes that have moved or whose attribute values have changed significantly. Significance is determined by a user-selectable threshold. According to a thermometer scale, we visualize positive attribute changes as red links, negative changes are shown in blue. Links representing node movements are colored with a shade of gray.
- Addition or deletion of nodes and edges is visualized by *spikes*. Blue spikes point (not connect) from a layer L_i to the subsequent one L_{i+1} to indicate objects that exist at t_i but not at t_{i+1} (deletion), whereas red spikes point from L_i to L_{i-1} to indicate addition of objects at t_i .

The layering approach in combination with the described visual cues allows users to compare successive time steps more closely. Especially as we are using just small points for representing the hierarchy's nodes, this explicit visualization of changes helps to find even small changes in the data that could barely be seen otherwise. However, to avoid problems caused by overplotting and visual cluttering, only a smaller subsequence of time steps (e.g., $s < 10$) can be represented in a single visualization. To alleviate this restriction and to allow for comparison of time steps that may be located anywhere on the time axis, multiple views are provided. Each view may be configured independently to show separate portions of the temporal domain, either a single time step using a base

map or a subsequence of time steps using the layering method. We employ a docking framework (www.infonode.net) that allows users to arrange views according to their needs. It is possible to align views in a row to resemble a temporal sequence, or to set up larger focus views and smaller views for those parts of the time axis the user deems to be contextual.

3.3. *Encoding Associated Data Values*

The question that remains is how to encode the data attributes associated with a hierarchy. Since the display is already packed with information, only little space is available for data visualization. Therefore, it makes sense to follow a two-fold strategy: The user can decide to focus on structural aspects or on data aspects. If structure is selected, only one data attribute is encoded to the nodes and edges of the hierarchy layout. We use color, size, and shape variations sparingly to keep structural aspects in the focus. If data aspects should be emphasized, the hierarchy layout is dimmed and additional visual representations of data attributes are faded in. Pencil and helix glyphs as described by Tominski *et al.* (2005) fit well into our space-time-cube visualization. They are applied to show the development of multiple data attributes over time. Figure 6 shows pencil and helix glyphs side by side. Pencil glyphs are suited to visualize linear development, whereas helix glyphs are useful for exploring cyclic patterns. The glyphs are positioned with respect to nodes that users may select in the base map, and each glyph represents the data attributes of its associated node via color-coding. Optionally, the regions of the base map may be color-coded to visualize a single aggregated data value (e.g., to indicate high variance for a region over time).

3.4. *Interaction for Data Exploration*

The previous paragraphs describe visualizations that can be quite complex. As we visualize complex data, the visualization is endangered by visual cluttering. Therefore, it is crucial to provide effective methods for user interaction.

To keep the visual load at a digestible level, our approach operates on user-chosen subsets of regions, time steps, nodes, and data attributes. Where possible, we allow for direct selection from the visualization (e.g., selection of regions or nodes) and provide dedicated user interface elements (e.g., calendar views to select time spans) otherwise. Visualization parameters can be set analogously.

A particular concern is the navigation of the map display because we are facing the challenge of 3D navigation. Our approach combines visualization techniques that are best viewed as a 2D projection (e.g., hierarchy embedded in map regions, color coding) with techniques that intrinsically require 3D interaction (e.g., layering of time steps, 3D glyphs). Therefore, a single interaction metaphor alone cannot satisfy interaction needs. As a consequence, we combine object-oriented navigation (i.e., the user manipulates objects in the 3D scene) with user-oriented navigation (i.e., the user moves through the 3D scene). To alleviate 3D occlusion problems, we allow for easy orbiting of points of interest (e.g., a selected glyph or node in a hierarchy layout). A special mode for moving along the z-axis facilitates navigation in time. We also provide a “panic button” that resets the visualization to a well-defined default view.

4. Adaptation to a Specific Visualization Scenario

Our approach provides the general means to visualize hierarchies in a spatio-temporal setting. We will now show how it can be applied and adapted to a concrete visualization problem, namely the visual exploration of human health data. These data provide information about how many people suffered from a particular disease, in a specific geographic region, during a certain temporal period. In previous work, we focused on the visualization of qualitative and quantitative data attributes (see Tominski *et al.* (2008)), but did not consider any hierarchical structure in the data. As a matter of fact diseases, geographic regions, and time, all are organized in a hierarchical structure: ICD10 classification of diseases, administrative districts (state, district, postal code area), and calendar system (year, quarter, month, week, day), respectively. As the focus of this special issue is on temporal aspects, we chose to use the hierarchical structure of time for demonstration purposes. This also allows us to complement the linear representation of temporal aspects as described in the previous section by a hierarchical representation of the dimension of time by adapting our general layout approach.

4.1. Visualization According to the Hierarchical Structure of Time

Health data may contain various interesting temporal patterns. The visualization that we will present addresses the following three questions:

- (1) How are diseases distributed during the week? (weekly pattern)
- (2) Is the number of cases dependent on the season? (annual pattern)
- (3) Are cases uniformly or non-uniformly distributed? (overall pattern)

The weekly pattern describes the behavior of sick people seeking medical care. Most people hope to recover naturally over the weekend, but are more likely to consult a doctor early in a week to get at least a sickness certificate. A deviation from this pattern might indicate a critical emergency. The annual pattern describes if and how a disease depends on the season. For instance, most cases of influenza generally occur during fall and winter. A larger number of influenza cases in summer might hint at an import of the virus from somewhere else. The last pattern considers the overall temporal distribution of cases: is there a uniform distribution of diagnoses over a larger period or are they highly accumulated at certain points in time (e.g., many injuries after a sport festival or a car accident)?

There already exist a number of powerful techniques such as spiral displays or time line plots for visualizing one of the mentioned patterns or finding peaks in the time line. But in order to assist users in answering all three questions, we construct a temporal hierarchy and use an adapted layout to present it on the map. As the data is given for years, quarters, months, weeks and days we use this as the basis of our temporal hierarchy, but other hierarchical decompositions of the time (e.g., by seasons) are also possible. In the layout step the year is placed as the root node in the center, the four quarters are positioned clockwise around the year, and the months are placed clockwise on another *ring* around the corresponding quarter. Around each month, its weeks are placed. Finally, days will be positioned on smaller rings around the week nodes. A small gap between Mondays and Sundays is inserted to indicate beginning and orientation of a week. Days that do not fit into the four week pattern, usually the first and last days of a month, are located directly around the month nodes: the month's first days to the left and its last days to the right.

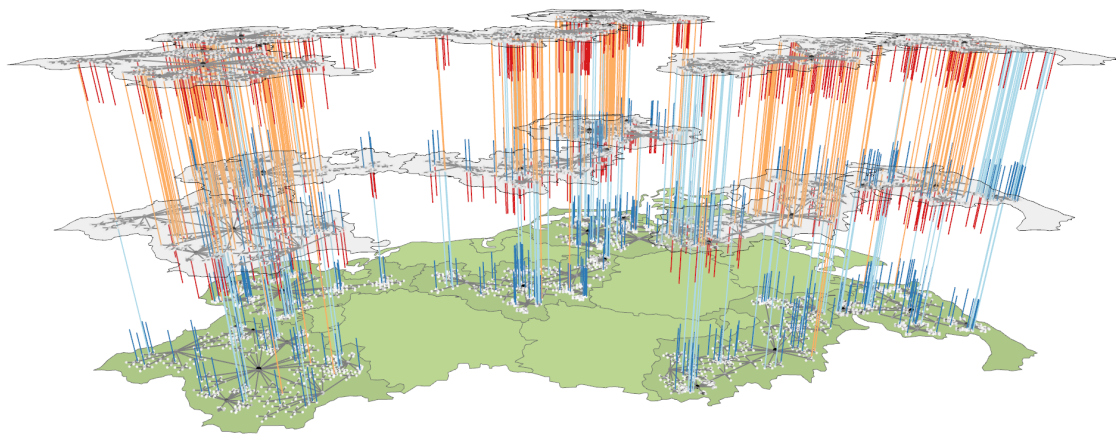


Figure 5. Visualization of hierarchies for selected regions and three time steps. – Colored links between layers indicate significant increase (red) or decrease (blue) of node attributes. Addition and deletion of nodes is shown via red and blue spikes, respectively.

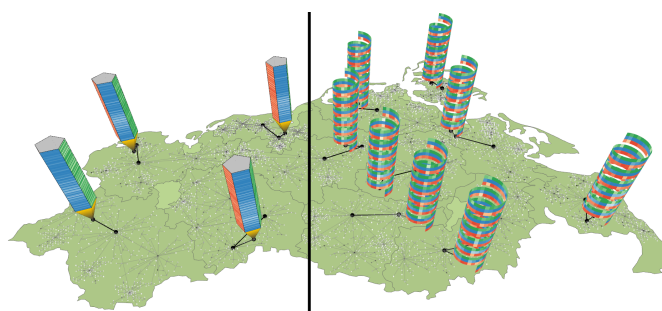


Figure 6. Pencil and helix glyphs visualize data attributes associated with hierarchy nodes.

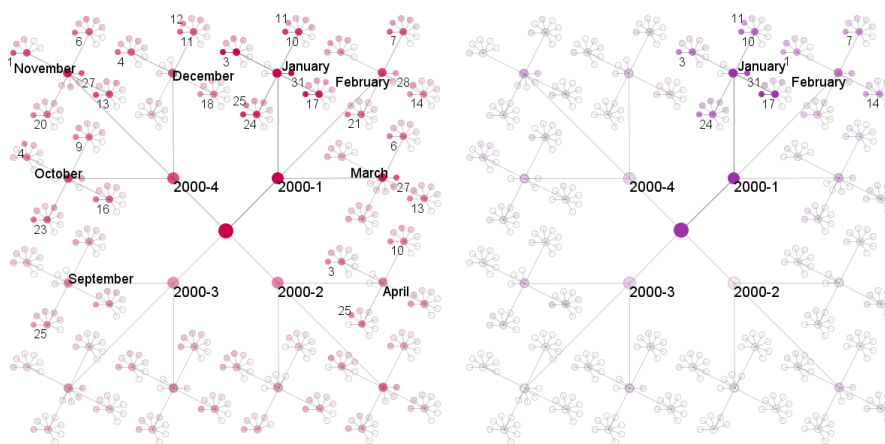


Figure 7. Visualization according to the hierarchical structure of time. – Left: number of people with problems with the upper respiratory tract; right: number of people suffering from influenza.

In this hierarchy, the leaves contain data measured on a daily basis and the internal nodes (week, month, quarter, year) contain data aggregated by summing up data values along the hierarchy. A simple color coding can be used to visually encode data values. Using white for no cases and a fully saturated color for the maximum number of cases accentuates the nodes with higher number of cases and improves the interpretation of the data value distribution. Additionally, labels are shown for nodes whose values exceed a user-chosen threshold, which further accentuates important nodes. Figure 7 shows layout and color coding for two diagnoses, influenza and diseases related to the upper

respiratory tract. The adapted layout allows for an easy interpretation of the underlying data especially emphasizing the interesting patterns and for comparison with different diseases or years. With regard to the assumption that people get sick certificates more often in the beginning of a week, one can clearly see that the most saturated colors (high number of cases) appear for the first days of a week. The clockwise positioning of the quarter and month nodes provides self-contained overview of the annual pattern. For instance, influenza is seasonal and occurs mainly in the cold winter months, as it can be seen from the dark colored nodes in the upper right corner of the visualization, pointing to the first three months of the year. On the other hand, diagnoses generally related to the upper respiratory tract are more evenly distributed over the year. Furthermore, the visualization reveals a relation between both diagnoses: at days with larger numbers of people suffering from influenza, there are also more people that have problems with the upper respiratory tract. This is not surprising as influenza has a direct impact on the upper respiratory tract, but it demonstrates quite well the effectiveness and comparability of the visualization. The fixed layout also increases the users' awareness of missing nodes or nodes with only few cases, as such nodes result in unused or white space in the visualization.

4.2. *Embedding Into the Map Display*

Our goal is to compare the temporal trends of different diseases for the regions of a map. For this purpose, we compute layouts (for the time hierarchy as described before) for all regions of a map, for a selected diagnosis. Each layout is placed in the largest inscribed circle of its corresponding region. Depending on the available display space different visual abstractions of the layout are rendered. The higher resolution node-link representation is used if plenty of display space is available (e.g., when the user has zoomed into the map). If display space is limited (e.g., when dealing with small regions), we resort to an iconic representation of the layout (similar to Slingsby *et al.* (2008)). With decreasing display space the number of shown hierarchy levels and therefore the number of nodes inside the iconic representation is also reduced showing then just a clear overview of the higher levels. The iconic rendering in combination with a high resolution detail view is illustrated in Figure 8. One can see that many regions share the same temporal trend as the highlighted Rostock region, showing high occurrences of influenza in the first two months of the year. Some regions show high numbers of affected people also at the end of the year, which might be an indication that the inhabitants of these regions are less vaccinated against influenza.

In order to visualize more than one disease at a time we have to place multiple layouts per region. We subdivide regions (see Section 3) and embed the layouts into the inscribed circles of the subregions. To enable users to identify the same disease in different regions we encode each disease with a unique hue, where we allow similar diseases to share similar hues (e.g., “influenza” and “upper respiratory tract” use hues of red). The iconic representation on the map allows for an initial comparison of diseases between different regions. Different scopes of comparison are supported by offering different ways of mapping data to color (see Figure 9):

- (a) global comparison – data values are normalized between 0 and the overall maximum number of cases per day
- (b) region-local comparison – data values are normalized between 0 and the maximum number of cases per day for all diseases *on a per region basis*

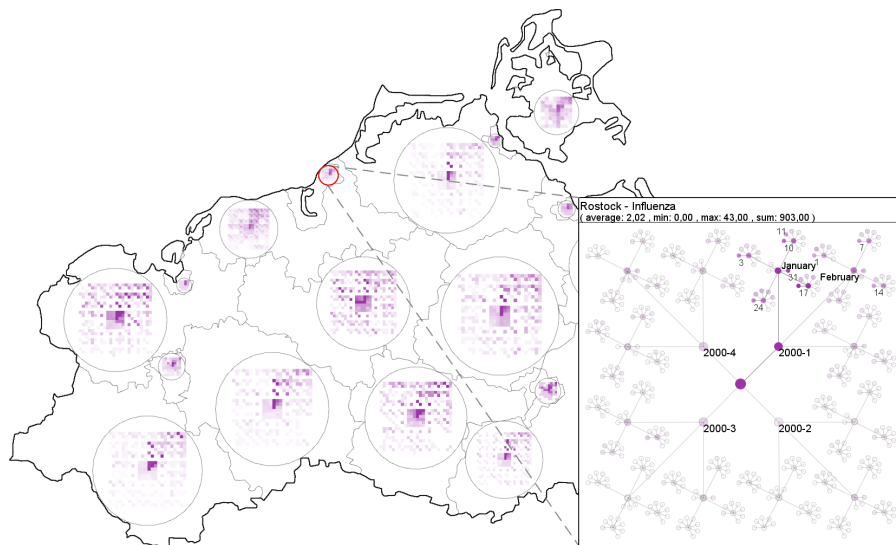


Figure 8. Visualization of influenza in Mecklenburg-Vorpommern for the year 2000.

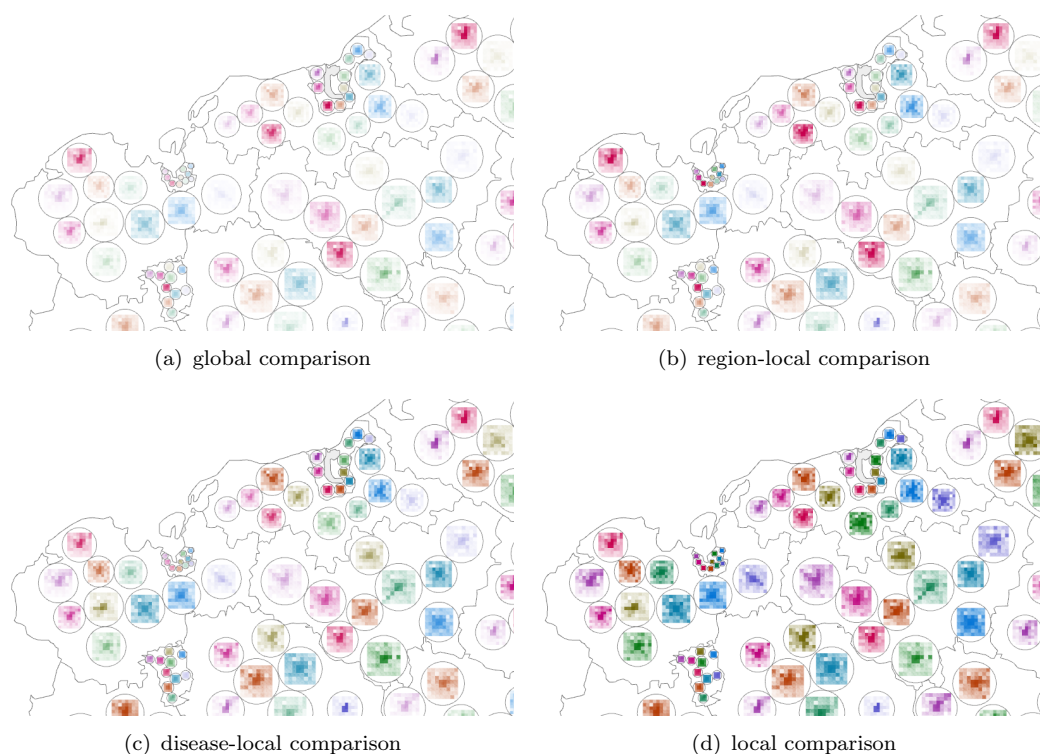


Figure 9. Different scopes of comparison of multiple diseases (denoted by different hues) in Mecklenburg-Vorpommern for the year 2000.

- (c) disease-local comparison – data values are normalized between 0 and the maximum number of cases per day for all regions *on a per disease basis*
- (d) local comparison – data values are normalized between 0 the maximum number of cases per day *on a per disease and per region basis*

Global comparison can be used, for example, to find the disease and/or region with the highest number of cases at all. Region-local comparison can be used for comparing just the different diseases of each area to find similar trends between them. Disease-local comparison supports finding deviations in the temporal patterns, which might indicate local dependencies of a disease. Local comparison restricts the scope of interest to the

analysis of the temporal evolution of each disease for each region.

The visualization described so far shows multiple diseases and multiple regions, but the hierarchy layout shows a single year only. To display multiple years, the layered approach described in Section 3.2 can be used. In the special case that just one disease needs to be shown, we can alternatively embed multiple layouts, each of which representing a different year, into subregions.

Eventually, we have arrived at a visualization that allows users to analyze temporal patterns of multivariate human health data in their spatial distribution on a map display.

5. Conclusion

This work described the visualization of hierarchical structures and associated data in time and space. We combined several techniques and concepts to create a solution that integrates spatial, temporal, structural, and data aspects. Key points in our combined strategy are the point-based layout to embed hierarchies into regions of a map, the layering of time steps (and corresponding visual cues), the integration of dedicated data representations, and interactivity. We adapted our general approach into a visualization for human health data that allows users to identify spatial and temporal patterns.

We understand our work as a step towards visualizing hierarchies in time and space, not as a final solution. Many research questions need to be answered in the future. How can we handle the spatial hierarchy, can our approach be adapted for showing it explicitly inside a region? How can we deal with large numbers of unevenly distributed time steps? Is it possible to find better ways to tackle overplotting of layers, for example by grouping the visual cues between the layers? Does our approach scale for globe-like visualization and 3D terrains? How can data attributes and structural aspects be combined more efficiently? How can we further facilitate comparison tasks and detection of patterns?

The focus+context concept seems to be a good starting point in the search for valid solutions to these problems. Depending on data characteristics, analytical needs, and user interests, relevant parts should be automatically highlighted in the visualization, others should be dimmed. The natural structure of time (years, quarters, months, weeks, days, etc.) as well as different scales of space (continents, countries, counties, etc.) will surely prove useful in this endeavor. Moreover, knowledge about the effectiveness of visual encodings is required, which in turn means that evaluations need to be conducted. We believe that the described visualization and interaction facilities help users in exploring attributed hierarchies in a spatio-temporal context. However, we also think that much more research is required to understand visualization and interaction needs and to design solutions that adapt themselves in a user-centered and task-oriented manner.

Acknowledgement(s)

We gratefully acknowledge support by “VisMaster” and “dIEM oSiRiS”.

References

- Aigner, W., *et al.*, 2007. Visualizing time-oriented data - A systematic view. *Computers & Graphics*, 31 (3), 401–409.
- Andrienko, N. and Andrienko, G., 2006. *Exploratory Analysis of Spatial and Temporal Data*. Berlin, Germany: Springer.
- Battista, G.D., *et al.*, 1999. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall.

- Card, S.K., *et al.*, 2006. TimeTree: Exploring Time Changing Hierarchies. *In: Proc. of IEEE Symposium on Visual Analytics Science And Technology*, 3–10.
- Heilmann, R., *et al.*, 2004. RecMap: Rectangular Map Approximations. *In: Proc. of IEEE Symposium on Information Visualization*, 33–40.
- Jern, M., Rogstadius, J., and Aström, T., 2009. Treemaps and Choropleth Maps Applied to Regional Hierarchical Statistical Data. *In: Proc. of International Conference Information Visualisation*, 403–410.
- Kraak, M.J., 2003. The Space-Time Cube Revisited from a Geovisualization Perspective. *In: Proc. of International Cartographic Conference*, 1988–1995.
- Kraak, M.J. and Ormeling, F., 1996. *Cartography: Visualization of Geospatial Data*. Prentice Hall.
- MacEachren, A.M., 1995. *How maps work: Representation, Visualization and Design*. New York, USA: The Guilford Press.
- Munzner, T., 1997. H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space. *In: Proc. of IEEE Symposium on Information Visualization*, 2–10.
- Reingold, E.M. and Tilford, J.S., 1981. Tidier Drawings of Trees. *IEEE Transactions on Software Engineering*, 7 (2).
- Robertson, G.G., Mackinlay, J.D., and Card, S.K., 1991. Cone Trees: animated 3D visualizations of hierarchical information. *In: Proceedings of the SIGCHI Conference on Human factors in computing systems*, 189–194.
- Schulz, H.J., Hadlak, S., and Schumann, H., 2010a. The Design Space of Implicit Hierarchy Visualization : A Survey. *IEEE Transactions on Visualization and Computer Graphics*, To appear.
- Schulz, H.J., Hadlak, S., and Schumann, H., 2010b. Point-Based Visualization for Large Hierarchies. *IEEE Transactions on Visualization and Computer Graphics*, To appear.
- Shneiderman, B., 1992. Tree Visualization with Tree-maps: 2-d Space-Filling Approach. *ACM Transactions on Graphics*, 11 (1), 92–99.
- Slingsby, A., Dykes, J., and Wood, J., 2008. Using Treemaps for Variable Selection in Spatio-Temporal Visualization. *Information Visualization*, 7 (3), 210–224.
- Slocum, T.A., *et al.*, 2008. *Thematic Cartography and Geovisualization*. Prentice Hall.
- Telea, A. and Auber, D., 2008. Code Flows: Visualizing Structural Evolution of Source Code. *Computer Graphics Forum*, 27 (3), 831–838.
- Telea, A. and van Wijk, J.J., 2002. An augmented Fast Marching Method for computing skeletons and centerlines. *In: Proc. of Symposium on Data Visualisation*, 251–260.
- Tominski, C., Schulze-Wollgast, P., and Schumann, H., 2005. 3D Information Visualization for Time Dependent Data on Maps. *In: Proc. of International Conference on Information Visualisation*, 175–181.
- Tominski, C., Schulze-Wollgast, P., and Schumann, H., 2008. Visual Methods for Analyzing Human Health Data. *Nilmini Wickramasinghe and Eliezer Geisler (editors), Encyclopedia of Healthcare Information Systems*, 1357–1364.
- Tu, Y. and Shen, H.W., 2007. Visualizing Changes of Hierarchical Data using Treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 13 (6), 1286–1293.
- Tufte, E.R., 1986. *The visual display of quantitative information*. Cheshire, CT, USA: Graphics Press.
- Tversky, B., Morrison, J.B., and Betrancourt, M., 2002. Animation: Can It Facilitate?. *International Journal of Human-Computer Studies*, 57 (4), 247–262.
- Wood, J. and Dykes, J., 2008. Spatially Ordered Treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 14 (6), 1348–1355.