Hierarchical Volume Visualization of Brain Anatomy

Jean-Paul Balabanian[1], Martin Ystad[2], Ivan Viola[1], Arvid Lundervold[2], Helwig Hauser[1], and Eduard Gröller[3,1]

Department of {Informatics[1] | Biomedicine[2]}, University of Bergen, Norway
Vienna University of Technology, Austria[3]
Email:{firstname.lastname}@{biomed|ii}.uib.no, groeller@cg.tuwien.ac.at

Abstract

Scientific data-sets often come with an inherent hierarchical structure such as functional substructures within organs. In this work we propose a new visualization approach for volume data which is augmented by the explicit representation of hierarchically structured data. The volumetric structures are organized in an interactive hierarchy view. Seamless zooming between data visualization, with volume rendering, and map viewing, for orientation and navigation within the hierarchy, facilitates deeper insight on multiple levels. The map shows all structures, organized in multiple hierarchy levels. Focusing on a selected node allows a visual analysis of a substructure as well as identifying its location in the hierarchy. The visual style of the node in focus, its parent and child nodes are automatically adapted during interaction to emphasize the embedding in the hierarchy. The hierarchy view is linked to a traditional tree view. The value of this new visualization approach is demonstrated on segmented MRI brain data consisting of hundreds of cortical and sub-cortical structures.

1 Introduction

Research in information visualization has many examples of visualizing hierarchical data such as trees and graphs. Scientific data often has an inherent hierarchy that is in many cases not fully exploited during visualization. In the medical domain it is often easy to describe the inherent hierarchical nature of the data. The human body can be semantically divided into several structures that have a hierarchical relationship with each other. For example the arm can be substructured into upper arm, forearm, and hand. The hand can be further divided into fingers and palm. Another example of a hierarchical structure, and also the one we are here focusing on, is the brain. The anatomical hierarchical subdivision of the brain starts with the separation of the left and right hemispheres, then the cortical and sub-cortical areas, followed by subdivision into different lobes, consisting of several gyri and other structures.

In medical education it is difficult to convey this 3D spatial relationship by the use of textbooks. Thus, medical students have to perform training on cadavers in order to acquire this kind of knowledge. The amount of information that is possible to extract from a textbook is to a significant amount related to the contained illustrations. The amount of knowledge gained from cutting into a real brain is also limited. Cutting open one structure to study its sub-structures will make the higher level structure unusable for further studies due to its irreversible modification. It is also possible to study brain data by looking at MRI slices, but analyzing such slices requires reasonable expertise. 3D volume visualization can help in visualizing the structures. However it is difficult to infer hierarchical and semantic information from these visualizations, especially when many structures are to be investigated.

Our approach is based on two different types of data. One is 3D anatomical data from MRI, with binary segmentation masks, and the other is abstract hierarchical information inferred from the 3D data. The proposed approach in this paper tries to not only show the anatomical structure but to integrate hierarchical semantics and volume information in the same visualization. The visualization combines the field of scientific visualization with information visualization by rendering a hierarchical layout in the same view as the volume rendering. Figure 1 shows a closeup example of this combined view.

The major contribution of this work is the combined visualization of scientific volume data with inherent hierarchies. We provide a seamless interface that enables an integrated interaction between abstract hierarchies and scientific data. We do this by creating an overview map where the hierarchy of the data is represented and where it is possible to zoom in to reveal knowledge about the volume data. At the volume data level we change the visual representation of structures with auto-styling so that the hierarchical relationship between the structures becomes evident in the spatial domain. Using the novel concept of raycasting portals we are able to render more than 150 structures with volume rendering at the same time.

This paper is organized as follows: In the following section we present an overview of existing visualization techniques that relate to our work. In Section 3 we describe our approach to visualize hierarchical data and present results in Section 4. In Section 5 we discuss the results and mention future work. Finally we conclude in Section 6.

2 Related Work

We aim at volume rendering with the look-and-feel of medical textbooks such as the anatomical atlas by Sobotta [16]. Volume rendering has become a large field of research. The GPU-based rendering approaches that we build on are described by Engel et al. [3] and Krüger and Westermann [8]. The illustrative results are produced with style transfer functions as proposed by Bruckner and Gröller [2].

Hierarchical data are easier to navigate and to gain knowledge from if an appropriate interaction metaphor and visualization is used. The evaluation done by Wang et al. [19] confirms this. Hierarchical information is often visualized as a tree. The information visualization community has done extensive research in the field of visualizing and navigating hierarchical data. One type of approach maximizes the utilization of the available screen-space, called space-filling techniques, such as tree-maps [14, 15], information slices [1], and the InterRing [21]. They also indicate size measures associated with the data. In visualizing a file-system, for example, the tree-map technique uses the size of a file or directory as a measure of the size of a structure. Since the data is hierarchical, the size coding is applied recursively and the space occupied by a parent node is subdivided by its children. The visualization used to show the parent-child relationship is depicted with rectangles inside rectangles. Similarly, information

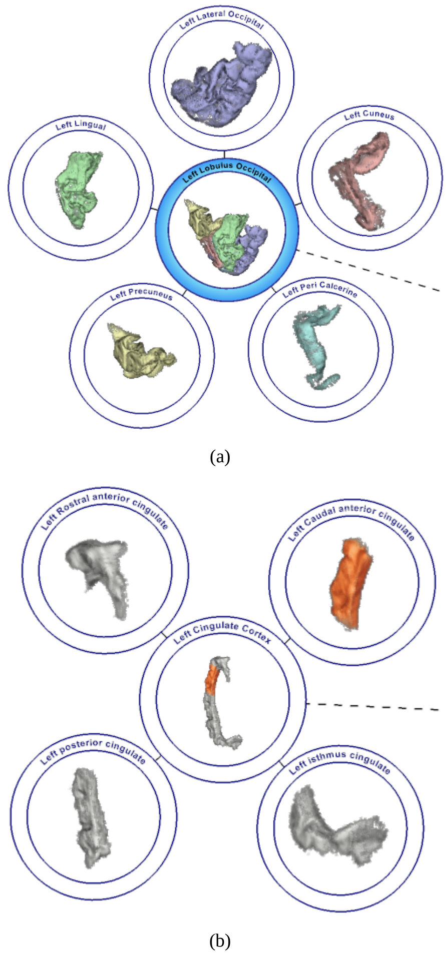


Figure 1: (a) The occipital lobe is colored to indicate the hierarchical relationship using *auto-styling*. (b) Interactive change of the visual representation of one structure in the cingulate cortex.

slices and InterRing use cascading circles and visualize the size measures as sector pieces.

Other techniques visualize trees without giving an indication of the relative sizes of the hierarchies. Cone Trees [13] and hyperbolic trees [10], for example, create a navigatable space with nodes in 3D. Other techniques such as RINGS [18] and Balloon trees [9] position nodes radially in 2D. The latter approaches have some similarities with our technique to lay out hierarchical data.

Interaction with and navigation of hierarchical

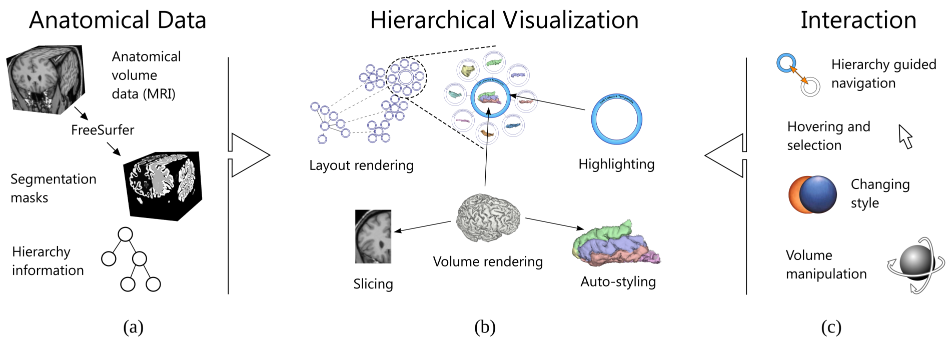


Figure 2: Hierarchically boosted volume visualization. (a) The data basis. (b) Components of the hierarchical visualization. (c) Interaction metaphors.

data is also a topic of research. An example is automatic panning and zooming [20] which efficiently moves from one node to another while preserving the overview by immediately zooming out to show the context. Other approaches let the user focus on some region of interest. InterRing lets the user expand a hierarchical level of interest. The other levels are reduced automatically to accommodate the region of interest. Another approach to visualizing the region of interest especially, was suggested by Stasko and Zhang [17]. The outer or inner part of a radial visualization is used as a special area to render the region of interest. Another interesting way of performing focus+context visualization is based on non-linear magnification lenses [7].

Other visualization techniques try to make the tree and graph visualizations more sparse by reducing the number of connection lines between nodes. Examples are edge bundling [6], or changing the thickness of connection lines such as in arctrees [11]. Herman et al. [5] provide an exhaustive survey on trees and hierarchy interaction.

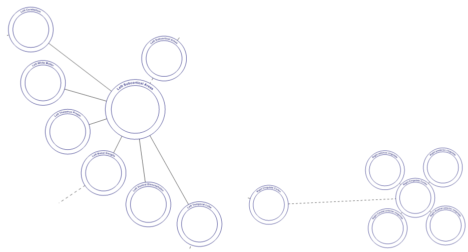
FreeSurfer [4] is a set of tools for the study of cortical and sub-cortical anatomy. It provides automated parcellation of the cerebral cortex and labeling of sub-cortical tissue classes in MRI volumes.

Previous work that proposes techniques to visualize the hierarchical nature of the brain has been proposed by Pommert et al. [12]. Their technique considers several different types of hierarchies. The user has to actively select a structure, then select what type of hierarchical information is interesting from a popup menu. Our technique differs significantly from their approach. In our visualization, for example, the hierarchy is the context that the user is navigating in. When focusing on a feature more hierarchical information is automatically provided.

Sources that describe techniques to combine hierarchy visualization and scientific visualization in the same context are scarce. The closest solution to resemble our technique is volume rendering of segmented structures with one structure highlighted and the other structures as context.

3 Spatial Data with Hierarchical Semantics

We integrate two spaces, an abstract space with a hierarchy and a data space where the volume data is defined. We enable seamless zooming between the hierarchical model and the anatomical data in the spirit of the focus+context metaphor. We propose a tree layout of the hierarchical data where each node shows a volume rendering of the semantically associated structure and a descriptive label. We call this the *context view*. It is crucial that rendering and navigation of this view is interactive. The navigation includes zooming from the context view to the volume data and the hierarchically guided exploration of this view. Figure 2 illustrates the different aspects of our approach. Figure 2(a) shows the available data basis, in Figure 2(c) the interaction possibilities are shown, and in Figure 2(b) the visualization techniques that create the final results are shown. The visualization changes according to the user interaction. Some of the visualization techniques are only active during specific interactions, others are



(a) (b)

Figure 3: Layout patterns for the context view. (a) A fan pattern, filling the area of a sector. (b) A cluster layout around a replicated group node in the center with a dashed line connecting it with its original.

active during the entire interaction process.

3.1 Hierarchy in the Data

In Figure 2(a) three types of data are listed as input for our approach. The data which we visualized here is the `Bert` data-set as provided by FreeSurfer. The data is T1 weighted MRI and FreeSurfer automatically generates binary segmentation masks for many structures in the brain. This process is based on an brain atlas technique and it takes approximately 20 hours. The segmentation masks that we use are the ones generated for the cortex and the sub-cortical areas of the brain (APARC+ASEG). The segmentation masks represent small regions and structures that by themselves do not form a hierarchy. We have created a hierarchical tree that associates segmentation masks with labels and labels with groups that are semantically meaningful. For example cortical ridges, denoted as gyri, are grouped together to form larger structures called lobes. These groups are part of other groups, such as lobes that are part of the cerebral cortex. The hierarchical groupings stop with the hemispheres of the brain. The resulting data-set contains a hierarchical overview that anatomically and hierarchically describes the brain and its structures.

3.2 Hierarchy Visualization

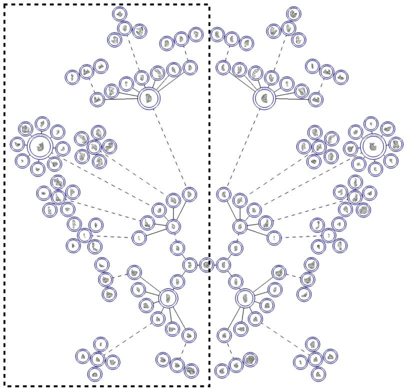
The hierarchical information of the data is visualized in two ways. First, a context view is generated that illustrates the hierarchical structure. A node-link diagram is used. Every node has links to its children and to its parent and all nodes are labeled. The automatic layout scheme attempts to create a context view that takes advantage of symmetries to support the orientation of the user. Second, child and parent nodes are displayed close to each other. A unique coloring of sub-structures indicates the hierarchical structure. An example of this can be seen in Figure 1(a).

We have defined semantics for the hierarchical information associated with the data. A leaf node is a segment node. These nodes have a direct correspondence with a segmentation mask. Nodes with children are group nodes, and group nodes that only contain leaf nodes are called leaf-clusters. This semantic is used for context view creation, auto-styling, and volume rendering. The leaf nodes of a leaf-cluster are shown in Figure 3(b). In the same figure the two nodes connected with a dashed line are group nodes.

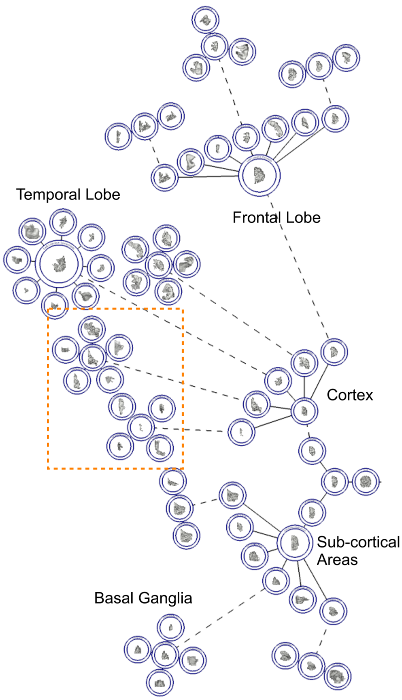
To optimize screen-space utilization we place structures as close to each other as possible, when we create the context view. In addition, structures should be positioned in such a way that the hierarchical relations are self evident. The user should fast and easily recognize the different structural features as generated in the context view.

The design choices of the context view are derived from the hierarchical nature of the data (Figure 4(a)). We place structures in a radial pattern and assign sub-structures to fractions of the sectors that are occupied by higher level structures. The left half and the right half of the layout correspond to the left and right hemispheres of the brain. Each quadrant represents a high level feature. These features are the cortex in the upper quadrants and the sub-cortical areas in the lower quadrants. Sub-structures of these features are given as fractions of these four sectors. Groups get a fraction of their parent's sector based on the number of children.

Some of the structures at the leaf level in the hierarchy have many siblings. In the case of leaf-clusters we want to minimize the occupied screen-space. Nodes in a leaf-cluster are positioned around a central point without any overlap. For helping the user to keep the context in mind, the group node of the leaf-cluster is replicated in the center of the cluster (Figure 3(b)). The clustering reduces the area occupied by sibling nodes relative to the sector size. If we do not cluster the sibling nodes in this manner the space needed to draw all the sibling



(a)



(b)

Figure 4: A context view (a) showing the entire hierarchical layout with a close up below (b). The right hemisphere is indicated in (a) as the dashed rectangle. Some features have been labeled in the figure to indicate their position. The close up of the region enclosed by the orange dashed rectangle is shown in Figure 1.

nodes increases, causing the nodes to move further away from the original parent node. Groups that only have group nodes or a combination of group nodes and leaf nodes as children are placed in a fan pattern, positioned at a distance where nodes do not overlap. This means that the nodes must be moved to a distance where all nodes can be positioned on an arc within the sector bounds. The group node is in this case replicated as well but it is positioned between the fan and the original group node (Figure 3(a)). At the two highest levels. i.e., the brain and the two hemispheres, we do not replicate the group nodes. The size of the rendered replicated group node is adjusted depending on the number of siblings. In a leaf-cluster, for example, the replicated group node increases in size when the number of siblings creates a circle that is much larger than the minimum node size (Figure 8).

The position where we place a replicated group node is also the position we use to bundle the connection lines between parent and children. In Figure 3 this can be seen as the connection lines from leaf nodes and group-nodes to the replicated group node of their parent. This makes the overview less cluttered. Drawing one line between a group node and a replicated group node and then one line from each child to the replicated group node is more space efficient than one line per child node to the original group node.

The Figure 4(a) shows the complete hierarchy of the brain with every node rendered in a ring. The ring is rendered as two concentric circles with different radii. The orange dashed square in Figure 4(b) represents the zoomed area given in Figure 1. In this closeup we can see most of the different types of visualizations from our approach. In Figure 1 there are two structures organized in two leaf-clusters, i.e., the occipital lobe and the cingulate cortex. When a node is highlighted, its ring is rendered in blue gradients. In case of replicated group nodes the highlighting is done for both nodes. Each node has an associated label which is placed on a curve on the top half of the ring.

When the user is looking at the context view, lines provide hierarchical information, such as child-parent relationships. The algorithm draws lines recursively between parent nodes and child nodes. The line is drawn from the border of the ring. In the case of replicated nodes, dashed lines are drawn between the original node and the replicated node. This is shown in Figure 1 and Figure 4.

An intuitive way of interaction with the context view is the possibility to focus on nodes. There are two types of direct interaction which result in focusing, i.e., hovering over a node and selection. When the mouse hovers over a node the node will initiate *auto-styling*, described in Section 3.4. Selection is done through clicking on a node. This centers the view on the node and enables manual selection of the style of that structure (Section 3.5).

3.3 Raycasting Portals

In the center of every node a volume rendering of the associated structure from the hierarchy is shown. This depiction of the volume data is generated by a GPU-based volume raycasting technique [8]. A proxy geometry is used to render the cubical shape of the volume data. We use the 3D texture coordinates to render colors that we use as a map into the volume data. The structures to visualize usually occupy only a sub part of the volume. If we only render the voxels included in the segmentation mask, we can significantly improve rendering time. We calculate the bounding boxes for all segmentation masks and the bounding boxes for all nodes higher in the hierarchy and use this to reshape the proxy geometry. We offset the texture coordinates so that they map to the coordinates of the bounding box of the structure. The aspect ratios of the proxy geometry are adjusted to match the aspect ratios of the bounding box. An example of the reshaped proxy geometry for the result seen in Figure 1(b) is given in Figure 5 where the raycasting starting position values are depicted.

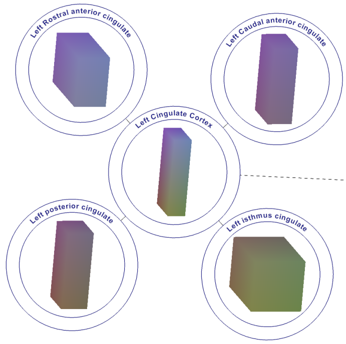


Figure 5: A visualization of rendered proxy geometries with the starting positions for the raycasting.

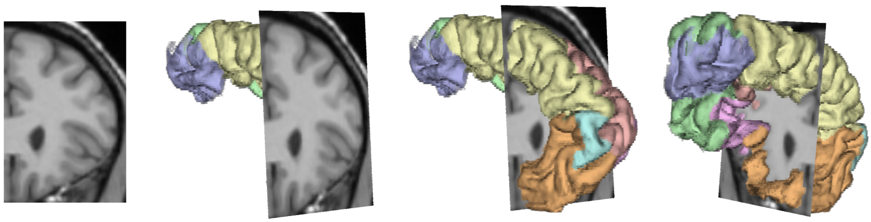
To change the visual representation of the raycasted structures, we use style transfer functions [2]. Style transfer functions utilize lighting information as acquired from orthogonally projected lit spheres. This technique makes it simple to achieve view-aligned lighting. With a style transfer function it is easy to switch from simple diffuse Lambert shading to Phong like shading by using different lit spheres. This technique enables us to simulate the non-photo realistic illustration style of medical anatomy illustrations like the ones by Sobotta [16]. See Figure 7 for examples.

We also need a mechanism to control what styles the raycaster should use for a specific substructure. Our solution to this is to use raycasting *portals*. Usually for GPU accelerated volume raycasting, a full-screen quad is rendered to the screen which is also a trigger for the GPU program to generate pixels. Instead of creating a full-screen quad we render a quad for every structure that we need raycasting for. A quad is centered on every node and the size of the quad is equal to the bounding square of the node drawn at that node. We call these quads raycasting portals. The main feature of these portals is that we can now communicate portal-specific rendering parameters to the GPU program via a shader uniform variable. The uniform variable contains a list of styles for the structures that should be visible and highlighted, and a zero reference to all structures that should be invisible.

To further increase the amount of information that is associated with the hierarchical elements and their volumetric nature, we have additionally implemented axis-aligned slicing. This visualization gives the user the opportunity to study the underlying MRI data and not just the raycasting of the structures. Figure 6 illustrates the combination of 3D and 2D information. Some of the possible combinations of slicing and volume rendering are illustrated in Figure 6.

3.4 Hierarchical Styling

The hierarchical organization of the data is used when we convey the hierarchical arrangement between sub-volumes. We change the visual representation of structures to clearly illustrate the hierarchical relationships. We have implemented two ways of interacting with the visual representation of structures to achieve two different goals. The



(a) (b) (c) (d)

Figure 6: Axis-aligned slicing of the frontal lobe: (a) direct view of a slice in the z direction. (b) structure and slice, features in front of the slice have been removed. (c, d) structure and slice from two different view-points (no removal of structures).

first goal is to visualize the hierarchical relationship between the parent and the child. This is done by showing how the parent node is composed of the child nodes by uniquely coloring each structure. The second goal is to show how a single structure or multiple structures are spatially located in all applicable hierarchy levels. This is achieved by the user interactively setting the style for a structure. The first approach is shown in Figure 1(a) and the second one is shown in Figure 1(b).

Our goal is to provide the user with multiple ways of seeing how the hierarchical structures are organized. One way of doing this is to color each structure uniquely so that they are easy to differentiate from each other. When the mouse hovers over a node, a feature called auto-styling is initiated. Auto-styling is the visual result of applying predefined style transfer functions to structures. This feature simplifies the navigation through the hierarchy. In addition it enhances the mental image the user has of the 3D structures of the brain. We have defined eight perceptually different styles based on pastel versions of red, green, blue, cyan, magenta, yellow, orange, and a darker blue. These colors have been selected based on contrast and lightness. The pre-defined styles are customizable.

If the mouse hovers over a group node, all its children are set to one of the predefined styles. The children are assigned styles in the same order as they are defined in the hierarchy. The style applied to a child is also used in the visual representation of the group node. This can be seen in Figure 1(a) and in Figure 8(a). In Figure 7 it is possible to see where the right medial temporal gyrus is located in the right temporal lobe using this technique.

If the mouse hovers over a child node, this node is displayed using the same pre-defined style as in the previous case, but the group node is displayed differently. The group node is displayed using the group's default style (defaults to grey). The selected structure is the only structure that is displayed with a different visual representation. This can be observed in Figure 8(b) where the selected structure can be seen in the group in light blue color.

Tracking a specific sub-structure in several different hierarchical levels is also possible. The user can change the style of a structure and then move up in the hierarchy to observe where the structure is located relative to higher levels in the hierarchy. Figure 9 illustrates this visual enhancement. The user has changed the style of the medial gyrus part of the temporal lobe. Moving up the hierarchy, the structure is now highlighted with the user selected orange style.

3.5 Interaction and Navigation

We enable the user to navigate by panning the data, hierarchically guided navigation, seamless zooming from the overview to the volume rendering, and by rotation of volume geometry. The user can hover over nodes to initiate auto-styling or manually set the style of interesting structures so they can easily be tracked. When exploring the data, the user can move the entire view or zoom in and out. Exploring the hierarchy in a zoomed-in manner can be a tedious task. A lot of dragging with the mouse is necessary to move around. Therefore we have implemented guided navigation that helps the user to move between nodes. If the user selects a node, the view centers on that node. This also means that zooming in and out is now centered on this

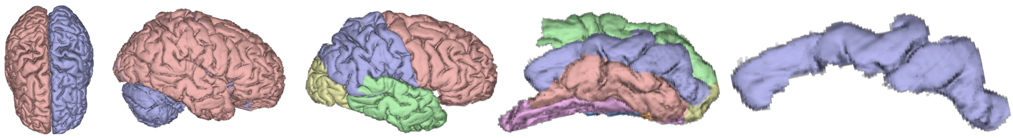


Figure 7: Auto-styling of hierarchically linked structures. From left to right: entire brain, right hemisphere, right cortex, right temporal lobe and right medial temporal gyrus.

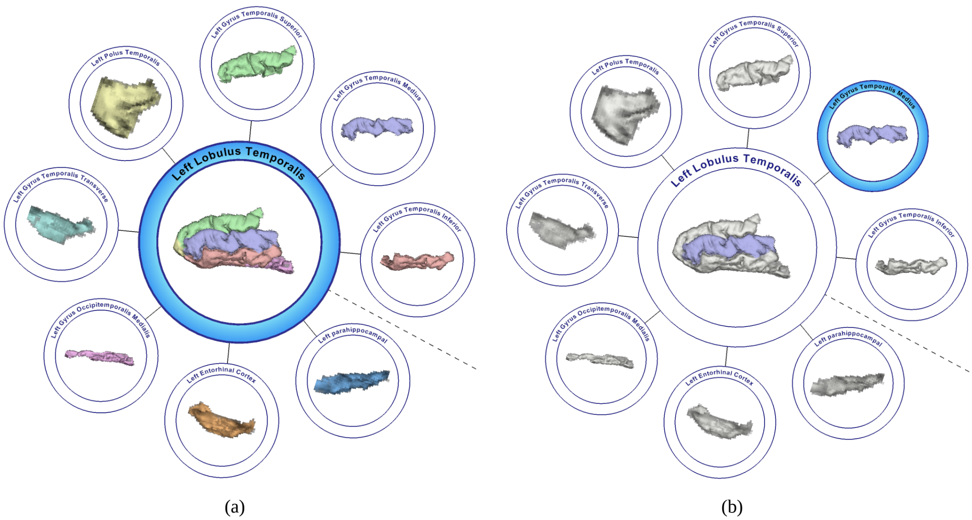


Figure 8: (a) Auto-styling of nodes in a leaf-cluster when the group node is selected. (b) Auto-styling of nodes in a leaf-cluster when the leaf node is selected.

node. If the node is a leaf node then the guided navigation will move the focus to the parent node. If the user initiates guided navigation on a replicated group node then the focus changes between the group node and its replica.

When the user is at an abstraction level, which shows volume rendered structures, rotation of the structures is possible. Rotation applies to all structures so that there is always a coherency between the views of the volumetric data. The user interface also lets the user change the visual representation of a selected structure so that it is possible to see where a feature is located in higher level structures.

4 Results

Seamless zooming from the contextual overview down to the data at the lowest hierarchical level can be observed in Figure 1 and Figure 4. In Figure 4(a) the complete overview is shown. The left part of this overview represents the left hemisphere and the right part represents the right hemisphere. Figure 4(b) is the left hemisphere only. At this level it is possible to see thumbnail sized volume raycastings of the structures. As the labels on the right hemisphere in the figure indicate, the upper part is the cortex and the lower part represents the sub-cortical areas. Zooming into the dashed orange rectangle we get the information as shown in Figure 1. In this figure we start to see details of the volume renderings and labels are readable. It is for example possible to see where the lingual is located in the occipital lobe (green structure in Figure 1(a)).

During inspection of the volume data at the lowest level of data exploration, it is possible to enhance the volume rendering with slicing so that the original data might be inspected. Figure 6 shows this concept. The user can choose to only show the

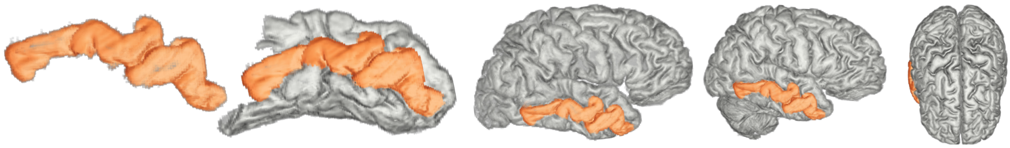


Figure 9: User-specified styling of a selected structure. From left to right: right medial temporal gyrus, right temporal lobe, right cortex, right hemisphere and entire brain.

slices. It is also possible to mix the two types of visualization and in this way get a higher level of understanding of the 3D nature of structures relative to 2D slices.

Auto-styling is highlighted in Figure 7. This Figure shows how the styling looks like at the different hierarchy levels if a user would start at the brain and navigate down to the medial gyrus of the temporal lobe. Going the other way, from the gyrus, and seeing how this structure is positioned relative to the higher level structures is illustrated in Figure 9.

We have implemented our approach in Java using OpenGL on a GeForce 8800 GTS. We used the OpenGL Shading Language extensions, texture arrays and integer texel look-ups, provided in the ShaderModel 4.0 specification.

The `Bert` data-set that we have visualized to create the results in this work is T1-weighted 3D MRI data at an isotropic resolution of 256^3 with 1 μL (microliter) voxels. This data-set has been run through the automated segmentation workflow of FreeSurfer. The surface reconstruction and volume labeling has produced approximately 80 subcortical and 68 cortical segmentations. We have in addition created a data-set that describes the hierarchical semantics of the segmentations. Based on these three data-sets we have generated a contextual overview that illustrates the hierarchy of the data. We render the MRI data using volume raycasting.

The context view without any volume raycasting renders at around 50 frames per second. Adding volume raycasting reduces the rendering speed to around 10 frames per second. The performance depends on how many pixels a structure covers, the number of voxels in the rendered structure and the amount of transparency defined in the style transfer function. To keep the interaction fast the raycasting is speeded up during interaction by doubling the sampling distance. This effectively doubles the rendering speed.

5 Discussion and Future Work

The presented approach to hierarchical brain visualization has an immediate use in education. Learning neuroanatomy and neurophysiology has always been regarded as challenging for medical students. This is due to the inherently complex and abstract structure of the nervous system, and to the intricate three-dimensional organization of the human brain. Our approach allows to visualize how the brain's components fit together, both in a strictly anatomical setting, and also in a functional-hierarchical manner. A daunting task is to understand which parts of the brain are in connection with which others and how they function together. The here presented approach has the potential to increase the efficiency of learning and ease the process of comprehension. Furthermore in radiology, students are required to relate planar images in different orientations to recorded volumes from modalities such as Computed Tomography and MRI. Our approach provides a way to investigate how the cut-planes actually represent parts of the volume. This will help students to understand the process of planar imaging, which is something many find difficult, and help themselves relate the two-dimensional image to the actual volume in real-time.

In a clinical setting it is important to convey subject-specific visualizations from the data. In dementia research, for example, it is of interest to compare volumetrics between a statistically normal brain and probable dementia patients. This comparison can be implemented in the hierarchical visualization to draw attention to structures that deviate, or not, from the normal population. We also foresee that including data obtained from other imaging sources such as functional MRI and diffusion tensor imaging can have a potential use in a clinical setting. Further research and evalutation of these approaches is left to future work.

6 Conclusion

We have presented an approach to visualize hierarchical volume data and enable hierarchy-based interaction. We have described how to generate the context view that enables a seamless navigation from the abstraction to the data. The visualization of structures in focus is automatically changed to reflect the hierarchical nature of the data.

Based on feedback from the medical side, our new visualization concept is considered promising and useful for medical education, especially for teaching radiologists. They usually look at 2D slices only and it is very hard for students to grasp the 3D structure of the structures observed on planar slices. With our approach they can select a portion of the data according to the hierarchical structure, and avoid that they are overloaded with the entire data-set. The integrated slice rendering gives the student a correspondence between 2D and 3D.

In general we do not see any difficulties to adapt our approach to other hierarchical structures such as the bones of the human hand. We think that there are other scientific domains with data containing hierarchies that would benefit from such an approach, also. An outlook for the future in hierarchical brain visualization comes from our medical collaborators. They observe that the anatomical structure of the brain is rather artifical and in many cases does not fit the functional hierarchy. Visualizing functional hierarchies will move the applicability of this tool from educational use to clinical use.

References

[1] K. Andrews and H. Heidegger. Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs. In *LBHT, IEEE InfoVis '98*, pages 9–12.

[2] S. Bruckner and M. E. Gröller. Style transfer functions for illustrative volume rendering. *CGF*, 26(3):715–724, Sep 2007.

[3] K. Engel, M. Hadwiger, J. M. Kniss, C. Rezk-Salama, and D. Weiskopf. *Real-time Volume Graphics*. A. K. Peters, 2006.

[4] Freesurfer. http://surfer.nmr.mgh.harvard.edu.

[5] I. Herman, G. Melancon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE TVCG*, 6(1):24–43, 2000.

[6] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE TVCG*, 12(5):741–748, Oct 2006.

[7] T. Keahey. The generalized detail-in-context problem. *IEEE InfoVis '98*, pages 44–51.

[8] J. Krüger and R. Westermann. Acceleration techniques for GPU-based volume rendering. In *IEEE Visualization 2003*, pages 287–292.

[9] G. Melancon and I. Herman. Circular drawings of rooted trees. Technical Report INS-R9817, CWI, Amsterdam, Netherlands, 1998.

[10] T. Munzner. H3: laying out large directed graphs in 3D hyperbolic space. *IEEE InfoVis '97*, pages 2–10.

[11] P. Neumann, S. Schlechtweg, and M. Carpendale. Arctrees: Visualizing relations in hierarchical data. In *EuroVis '05*, pages 53–60.

[12] A. Pommert, Schubert, Riemer, Schiemann, Tiede, and Höhne. Symbolic modeling of human anatomy for visualization and simulation. In *Vis. in Biomed. Comp.*, volume 2359, pages 412–423. SPIE, 1994.

[13] G. Robertson, J. Mackinlay, and S. Card. Cone trees: animated 3D visualizations of hierarchical information. In *Proc. CHI '91*, pages 189–194. ACM.

[14] B. Shneiderman. Tree visualization with treemaps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, 1992.

[15] B. Shneiderman and M. Wattenberg. Ordered treemap layouts. *InfoVis '01*, pages 73–78.

[16] J. Sobotta. *Sobotta Atlas of Human Anatomy*. Lippincott Williams & Wilkin, 2001.

[17] J. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. *IEEE InfoVis 2000*, pages 57–65.

[18] S. T. Teoh and K.-L. Ma. RINGS: A technique for visualizing large hierarchies. In *GD '02*, pages 268–275.

[19] Y. Wang, S. Teoh, and K.-L. Ma. Evaluating the effectiveness of tree visualization systems for knowledge discovery. In *EuroVis '06*, pages 67–74.

[20] J. Wijk and Nuij. Smooth and efficient zooming and panning. *InfoVis '03*, pages 15–23.

[21] J. Yang, M. Ward, and E. Rundensteiner. Interring: an interactive tool for visually navigating and manipulating hierarchical structures. *IEEE InfoVis 2002*, pages 77–84.