Interactive Illustrative Visualization of Hierarchical Volume Data

Jean-Paul Balabanian*
University of Bergen

Ivan Viola†
University of Bergen

Eduard Gröller‡
Vienna University of Technology
University of Bergen

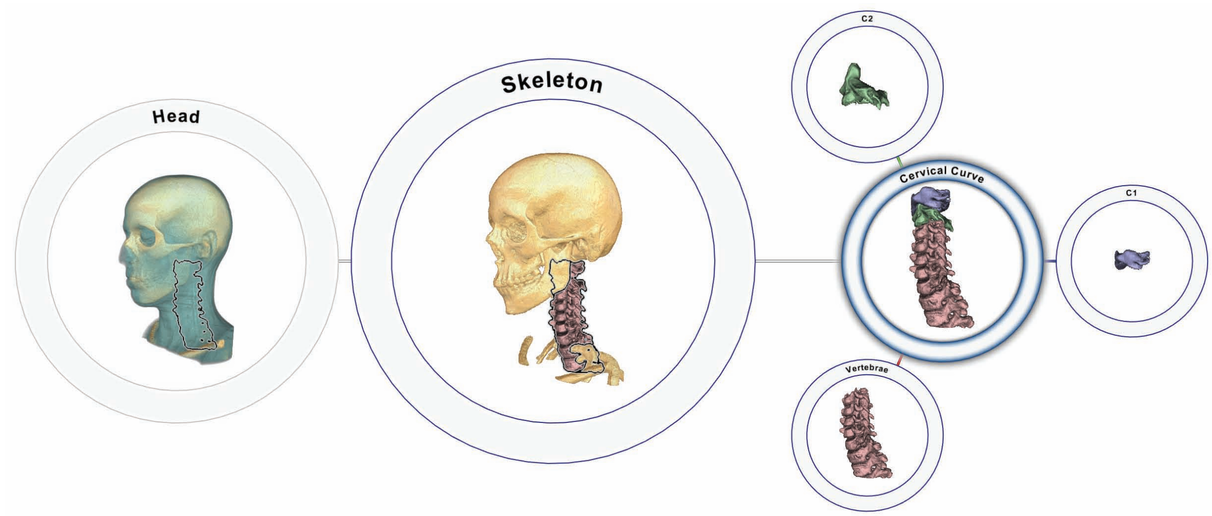


Figure 1: Hierarchical visualization of segmented head and neck. The cervical curve is focused by showing its relative position in the neck and highlighting its substructures.

Abstract

In scientific visualization the underlying data often has an inherent abstract and hierarchical structure. Therefore, the same dataset can simultaneously be studied with respect to its characteristics in the three-dimensional space and in the hierarchy space. Often both characteristics are equally important to convey. For such scenarios we explore the combination of hierarchy visualization and scientific visualization, where both data spaces are effectively integrated. We have been inspired by illustrations of species evolutions where hierarchical information is often present. Motivated by these traditional illustrations, we introduce integrated visualizations for hierarchically organized volumetric datasets. The hierarchy data is displayed as a graph, whose nodes are visually augmented to depict the corresponding 3D information. These augmentations include images due to volume raycasting, slicing of 3D structures, and indicators of structure visibility from occlusion testing. New interaction metaphors are presented that extend visualizations and interactions, typical for one visualization space, to control visualization parameters of the other space. Interaction on a node in the hierarchy influences visual representations of 3D structures and vice versa. We integrate both the abstract and the scientific visualizations into one view which avoids frequent refocusing typical for interaction with linked-view layouts. We demonstrate our approach on different volumetric datasets enhanced with hierarchical information.

*e-mail: jean-paul.balabanian@uib.no
†e-mail: ivan.viola@uib.no
‡e-mail:groeller@cg.tuwien.ac.at

Graphics Interface Conference 2010
31 May - 2 June, Ottawa, Ontario, Canada

1 Introduction

Datasets coming from scientific domains are usually defined with respect to a spatial frame of reference. Examples are volumetric data acquired using computed tomography, seismic acoustic measurements of geological structures, or climate simulation runs. These datasets represent phenomena in reality and they are analyzed with respect to their spatial structural arrangement. Increasingly, additional data is available for such phenomena in an abstract space, for example, depicting relationships between various structures contained in the data. Essentially, the same real-world phenomenon can be studied in two entirely different spaces.

A good example is the structure of the human body. The human anatomy can be given as 3D volumetric data. On the other hand, the body consists of various hierarchically organized sub-systems such as nervous, muscular and vascular systems. These systems define abstract relationships between body parts. The relationships are crucial to better understand processes in the human body. In the human motor system, for example, it is very important to analyze both, the relationships and the shape of skeletal structures.

Current visualization technology enables the user to study the spatial arrangement of scanned human anatomy using techniques from volume visualization. Structures can be visually represented using slicing or volume rendering. To analyze these structures, visualization technology offers various interactions such as defining which and how data values are shown (e.g., by using transfer functions), or from which viewing angle they are shown (e.g., by defining the viewpoint position). With such visualization approaches, the data which is defined in both spaces, in the spatial and the abstract domain, will be *projected* to the spatial domain and only the spatial characteristics will be visually conveyed. In this paper we use the terms *space* and *domain* interchangeably when we refer to the spatial and abstract origins of data, interaction and visualization.

Abstract data visualization is another way to represent this type of data. Structures can be depicted by techniques developed over the years in information visualization, for example, through graphs given as node-link diagrams. For each specific category of graphs, various layouts have been proposed with well defined interactions thereon. Such a representation clearly communicates information about processes and relationships. However, the spatial aspect of the data is missing due to the *projection* into the abstract space only.

To convey both aspects in visualization, i.e., the spatial arrangement of structures and the abstract relational information, one possibility is to employ linked views. In such a visualization setting, both spaces are shown in separate views, and both spaces are analyzed with separate interactions. The views are linked in the sense that manipulating one view will affect the other view as well. Linking and brushing is an example where the interactively selected subset in one view will also be highlighted in the other view. The separate views, however, require switching between domains and require refocusing of the user from one space to the other even if linking is present.

We believe that a stronger integration of spatial and abstract domains can lead to a better overall understanding of the studied real-world phenomenon. We display a graph as a guiding structure for understanding relationships and integrate the spatial characteristics of the data within the graph. The main contribution of this paper stems from this static illustration concept and develops an interactive integrated visualization approach. We define a set of interactions and visualizations that tightly integrate the distinct domains the data is defined in. In this process we utilize illustrative visualization concepts such as stylized volume rendering and structure outlines to convey both abstract and spatial data simultaneously. An example of using such an illustration concept can be seen in Figure 1.

2 Related Work

For visualizing the spatial characteristics in our integrated approach, we rely on existing technology developed over the last decade in volume visualization. The GPU-based rendering approaches that we build on are described by Engel et al. [4] and Krüger and Westermann [9]. The illustrative results in our work are produced utilizing style transfer functions as proposed by Bruckner and Gröller [2]. In addition to volume rendering, we visualize the spatial data by slicing. We augment the slicing with extensions of LiftCharts [15]. Alongside the slicing, a chart is visualized that shows the extent of segmented structures in the slicing direction. This gives a good indication of structure location and relation to the slicing plane and other structures.

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. [16] supports this statement. 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. Herman et al. [8] provide a broad survey on trees and hierarchy interaction.

The problem of integrating data from different spaces is one of the topics that focus+context research [7] has addressed in visualization. Such integration is mostly addressing visualization of data originating from essentially the same domain. An example could be data at different scales or from different acquisition modalities. Our approach, as compared to focus+context techniques, aims at the integration of quite different domains.

For volumetric datasets, the relationship between structures is increasingly being studied using visualization. Recently, a relation-aware volume exploration [3] approach has been proposed. It defines region-connection calculus and builds for each tagged volumetric dataset a set of relations into a relation graph. The paper is focusing on data similar to ours, but the approach is realized through linked views unlike our integrated visualization approach.

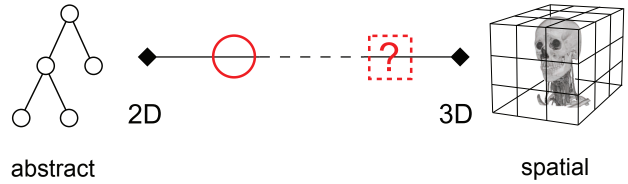


Figure 2: "Interpolation" between the abstract domain and the spatial domain. The circle indicates where our work is contributing.

Integrating abstract information into 3D spatial rendering has been proposed by Pommert et al. [14]. They integrated popup menus into the 3D rendering. Another approach to visualize 3D structures using abstract data was proposed by Li et al. [10]. They describe an exploded view visualization that relies on hierarchical information derived from the 3D spatial structuring. Integration of abstract visualization and spatial visualization using graph rendering and volume rendering has been proposed before [1]. They created a simple integrated visualization of a fixed graph layout with volume rendering inside the nodes. Compared to this previous design study, the contributions of this paper are a completely dynamic tree layout adaptable to any hierarchy, hierarchical information about structure intersection and slices, visualization of structure occlusion, interactive modification of structures through tree pruning, and usage of statistical visualization.

3 Integrated Visualization and Interaction Space

To effectively convey information about datasets defined over a spatial and an abstract domain, both domains have to be present in the visualization. In our work we focus on a strongly integrated visualization of spatial and hierarchical information. Unlike traditional approaches, where in one view only one domain is represented, we propose a tightly integrated display. In the process of merging these two domains we have chosen to use the abstract-domain representation through a graph as the guiding structure. We augment each hierarchical node with spatial information and aggregated information from both domains. This integrated visualization requires new visual and interaction means to effectively realize the visual dialog between the two *merged* domains.

In this work the integration is steered by the graph drawing. The abstract data is used to create a structure to present both the abstract and spatial data. It would also be possible to envision an approach that uses the scientific-visualization space as the embedding space. In Figure 2 we have sketched the "interpolation" between the two spaces that are part of the visualization, i.e., the abstract and spatial space. The circle indicates where this work is located. Using the scientific visualization as the embedding space would result in a visualization located in the dashed square. Such an integrated view might be an exploded view in 3D space where the abstract hierarchical relationships are indicated through arrows.

The proposed visualization inherits visualizations and interactions used previously for each respective domain separately. Essentially, we can now classify three categories of visualizations and interactions: abstract, integrated, and spatial. An abstract visualization is, for example, the display of a graph using a space-filling layout. An interaction on this abstract data representation is focusing on a node which invokes a change in its size or color. Similarly, a purely spatial domain visualization is direct volume rendering. A spatial interaction will be a manipulation of the viewpoint for example.

Apart from visualizations and interactions defined exclusively for one particular domain, our integrated approach especially focuses on integrated visualizations and interactions. An integrated interaction means that a particular interaction invokes visual

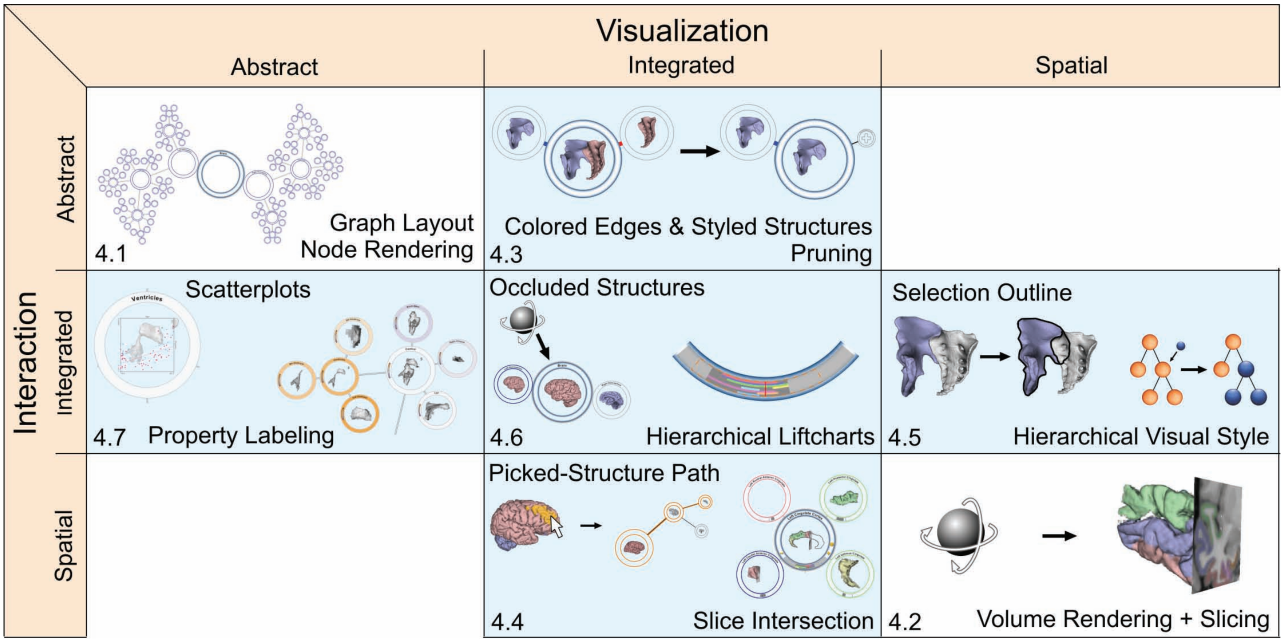


Figure 3: Matrix depicting combinations of interactions and visualizations defined for the abstract, the integrated, and the spatial domain.

changes in both, now integrated, spaces simultaneously.

We give an overview on possible combinations of spatial, integrated and abstract visualizations and interactions in the matrix in Figure 3. The traditional single-domain visualization and interaction approaches are shown in the top-left and bottom-right cells. More interesting are the new integrated visualizations and interactions depicted in the blue cells. The numbers in the matrix cells in Figure 3 correspond to section numberings where each *cell* is discussed.

The matrix contains two empty cells. These represent abstract or spatial interactions that result in visualizations exclusively in the other domain. We do not provide examples of these types of interactions because an interaction in one domain will naturally lead to a visualization in the domain of its origin.

4 Integrating Abstract and Spatial Domains

The following subsections describe the different techniques and approaches created to generate an integrated visualization of abstract and spatial data. We first describe interactions and visualizations that apply to one domain only. The rest of this section is dedicated to the description of the integrated visualization space.

4.1 Abstract Interaction and Abstract Visualization

The category of abstract interaction and abstract visualization corresponds to visualization and interaction possibilities typical for graphs and trees in the information-visualization domain. The abstract data is rendered as a node-link diagram. We utilize standard graph layouts such as force-directed layouts and Balloon trees [11]. The nodes are rendered as circles with the name of the structure as a label on the top half of the circle. The color of the node can be changed to convey state-change information to the user. With the same intent in mind, the edges between nodes can also be colored. Nodes can be focused, selected, or resized. Selecting a node other than the root makes the chosen node the new root and removes all other nodes that are not part of the sub-tree below this node. In addition the path to the original root is included. Figure 1 shows the result of selecting the cervical curve as the new root. Removing specific sub-trees is possible by collapsing a node. Transitions between interactions with the abstract data are animated. The interaction possibilities on the abstract data will be integrated with the spatial domain in the following subsections.

4.2 Spatial Interaction and Spatial Visualization

Spatial interaction and visualization corresponds to a straightforward visualization of the spatial data with typical interaction possibilities like rotation, etc. We display the spatial data using volume rendering and slicing. The volume rendering is aware of segmentation data and individual visual styles can be applied to the different segmentations. In the spatial domain the viewpoint for volume rendering can be relocated, the visual style can be changed, the slicing plane can be moved along the three main axes, and the structure located under the mouse cursor can be identified.

4.3 Abstract Interaction and Integrated Visualization

This category of interaction and visualization consists of interactions typical for the abstract domain, such as node focusing, that leads to visualizations in both domains.

Colored edges and styled structures: Navigating the abstract space and focusing a node in the hierarchy results in the volumetric structure being automatically visually emphasized using a set of predefined styles and colors. To increase overview locally, the edges between nodes are also colored. The same colors applied to the volumetric structures are assigned to the edges. The edge between the node and its parent is colored in black (see Figure 4).

This technique falls into abstract interaction / integrated visualization as the interaction is only with the tree layout, e.g, focusing a node. The result is visualized in both domains, i.e, styling of nodes, edges and volumetric structures.

Pruning: Volume rendering of structures that spatially enclose interior objects results in occluded features. Changing the visual representation of the occluding structures to transparent enables a clear view of otherwise occluded parts. The possibility to remove occluding structures has been realized through interactions with the graph. Typical interaction operations with trees are collapsing or pruning of sub-trees. For the graph display this means to remove from the layout all nodes included in the sub-tree. For volume rendering this means complete removal of the associated 3D structures. By collapsing a node, the sub-tree is effectively removed from the display in both visualization domains.

When a sub-tree is collapsed, the sub-tree root is replaced by a small node with a *plus* symbol. It enables a future expansion of the sub-tree. This interaction operation allows the user to create a specific, desired subset of the entire structure. For example, studying the cortex of the brain, it is possible to remove all of the sub-cortical structures. An example of pruning is shown in Figure 5 where specific bones have been removed from the foot. This makes it easier to study the interface between bone segments and neighboring bones in context.

The interaction in this technique also applies only to the tree layout but results in visual changes in both domains. The sub-tree that was pruned is effectively removed from the display. This produces also the side-effect of removing for all ancestral nodes the spatial structures associated with the pruned sub-tree.

4.4 Spatial Interaction and Integrated Visualization

Spatial interactions that influence integrated visualizations is the category located in the bottom center cell of Figure 3.

Picked-structure path: Picking in the spatial visualization of complex volumetric structures is a straightforward interaction for selecting a sub-structure. This operation is realized by casting a ray through the volume. When a particular structure is selected, visual prominence is given to this structure. The corresponding graph node is emphasized to effectively indicate its hierarchical location. The structure is highlighted under the mouse cursor and the path from the focused node to the graph node representing just the picked structure is highlighted. Figure 6 shows a mouse pointer picking a specific structure and the structure is emphasized with an orange color in the volume rendering. The path to the structure itself, is highlighted with orange outlines on edges and nodes in the graph.

This is an integrated visualization since the nodes and edges that include the picked structure are emphasized while the picked structure in the spatial domain is highlighted as well. It is a spatial interaction only because the structure is associated with a single segmentation and no hierarchy information is necessary to identify it.

Slice intersection: In a medical environment slicing is an often used technique of visualizing and interacting with volumetric data. A slicing interaction shows a cross-section through the structural information, and partitions the volume into two sub-volumes. Our integrated visualization represents this partitioning on the graph. The slicing plane's relative position to a structure is visualized through node coloring. The spatial extent of a structure is defined as the structure's minimum and maximum coordinates in the slicing direction. If a structure's maximum extent is less than the slice position the node is colored green. This can be interpreted as the slicing plane being in front of the structure. If the slice position is less than the minimum extent of the structure, the node is colored red. This is interpreted as the slicing plane being behind the structure. When the slicing plane intersects the structure, i.e., the current slice position is between or equal to the structure extents, the node is colored blue. This visualization can be seen in Figure 11(a). It provides a useful and fast way of getting an overview on which structures are part of the current slice. The visual impact of this technique can be seen in Figure 7. Changing the zoom level from overview to focus, a later described technique (hierarchical liftcharts), provides much more detailed information about the relative positioning of the slice.

The interaction approach in this technique is changing the slice position and is in the spatial domain only. Visualizing the result affects both domains. The slice is displayed together with the volume rendering and the node color changes based on the relative slice position.

4.5 Integrated Interaction and Spatial Visualization

Integrated interaction and spatial visualization results in visual output only in the spatial domain.

Selection outline: In the spatial domain a high level structure may be composed of several substructures that occlude each other. It may be difficult to see where a specific hierarchical substructure is spatially located. To help the user to locate a selected feature and indicate which parts of the structure may be occluded, an outline of the structure is visualized. This interaction takes advantage of visual motion cues to better convey the shape of the analyzed structure. The outline is applied to the whole structure and also indicates the border between the visible part and the occluded part. This is shown in Figure 8(a). In Figure 8(b) the occluded parts of the Coxa have been revealed.

This technique is an integrated interaction because it relies on hierarchical and spatial information to identify the structure to outline. A list of segmentations which belong to a hierarchical structure is used to identify the voxels that are part of the outline. The resulting visualization applies to the volume rendering only.

Hierarchical visual style: Taking advantage of the information in the abstract space creates an intuitive way to change the visual representation of structures in the spatial domain. Changing the visual style of a higher level structure, results in the new style being propagated down in the hierarchy to all lower level features. This results in increased efficiency to refine the visual appearance of the visualized structures. For example, it is possible to first select a visual representation that displays all structures in the same color and then refine for substructures. In Figure 9 this is illustrated by changing the style of all dense tissues to a bone-like visual representation. Afterwards, the remaining soft tissue structures are refined by individually changing their color and style. This approach is increasingly efficient for larger hierarchies.

It is again an integrated interaction. Changing and applying the style is a typical interaction approach in the spatial domain. For the style applied to a structure to propagate to all child nodes, information about the hierarchy is necessary. The resulting visualization only applies to the volume rendering of the structures.

4.6 Integrated Interaction and Integrated Visualization

In the most general case interactions are performed in both domains to invoke a specific visualization which is applied to both domains simultaneously.

Occluded structures: Manipulation of the viewpoint is a frequently used interaction in 3D with structural volumetric information. A chosen viewpoint also determines which structures are visible and which are occluded. This information can be extended to the hierarchical visualization by color coding those nodes and edges which are visible from a particular viewpoint and which are occluded.

Looking at a hierarchical structure that is composed of several substructures, one or more of the substructures may be occluded. In this situation we indicate to the user which of the substructures cannot be seen from the given viewpoint. Visibility is defined as the ratio between the number of pixels rendered for a substructure and the total number of pixels for the complete structure. If a structure is completely occluded, this is conveyed to the user by changing the color of the node. The color of the node is gray when the structure is less than 1% visible. Otherwise if the structure is less than 5% visible the color is interpolated between gray and blue. If the visibility is 5% or more then the node is rendered in blue. In Figure 10 this effect is demonstrated on an overview of the brain. The left hemisphere is completely occluded by the right hemisphere. This is easily perceivable as all nodes on the left part of the image are shown in gray. Some structures in the right hemisphere are also not visible from this viewpoint. Thus some nodes on the right part of

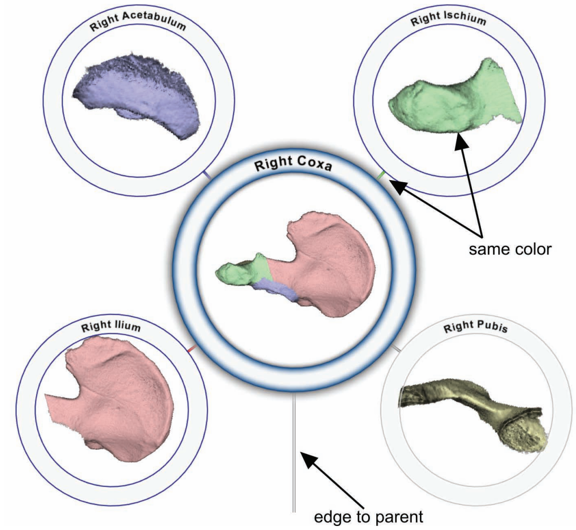


Figure 4: Colored edges with direct relation to structure color. The gray color of the pubis node indicates that this structure is not visible from the current viewpoint of the selected node.

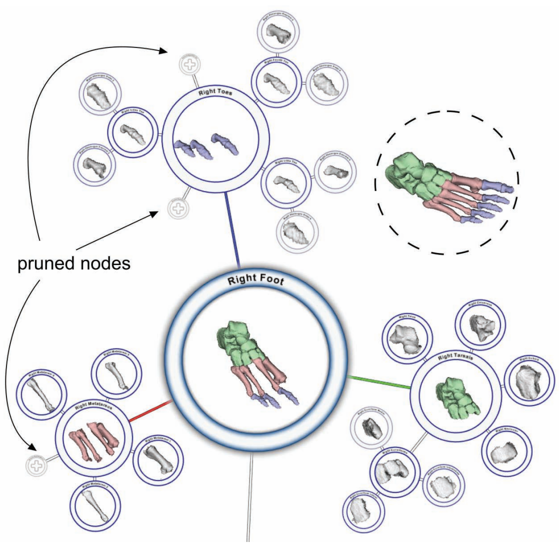


Figure 5: Pruning of big toe, middle toe and metatarsal. Collapsed nodes are shown as circled plus symbols. The dashed circle shows the foot before pruning.

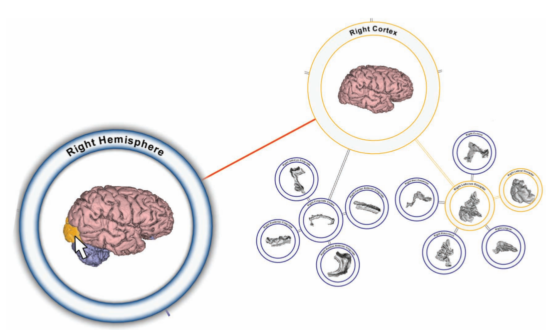


Figure 6: Picked-structure path with orange highlight on edges and nodes. The picked-structure is highlighted in orange in the selected node. The mouse cursor is exaggerated in size.

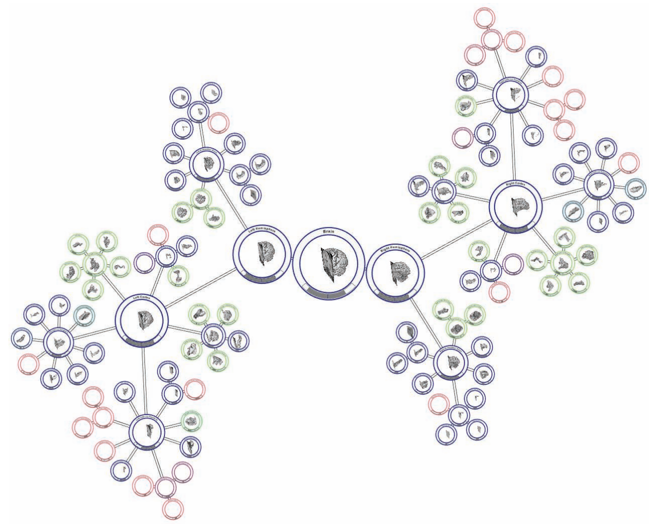
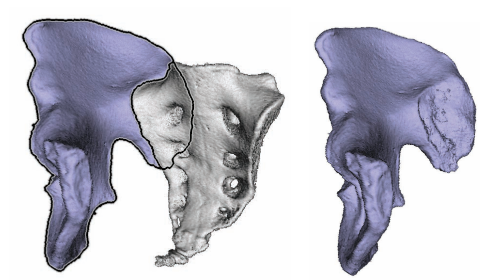


Figure 7: Visual impact of slicing. The node color indicates the relative position of the slicing plane with respect to the node structure. Green, red, blue means the slicing plane is in front, behind or intersecting the node structure.



(a) (b)

Figure 8: (a) Coxa occluded by the sacrum. The outline indicates the extent of hidden structures. (b) Coxa with no occlusion.

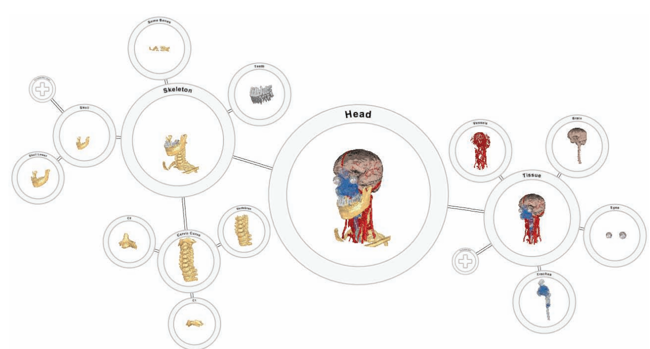


Figure 9: Different style transfer functions applied to structures. Upper skull (left ⊕) and skin (right ⊕) are removed to expose inner structures.

the image are gray as well. Another example can be seen in Figure 4 where the right pubis, i.e., the yellow structure, is occluded by the rest of the coxa.

The spatial interaction for this technique is changing the viewpoint through a rotation and the abstract interaction is focusing on a node of interest. The focused node is used to determine which segmentations to check for occlusion at all levels of the hierarchy. The resulting visualization is using the new viewpoint for the spatial data while indicating the level of occlusion with color on the node outline.

Hierarchical liftcharts: In addition to showing the slice plane, we provide additional information about the structures on the node representation. In the bottom half of the node we introduce a so called slice bar that represents the full extent of the entire volume in the slicing direction. It is labeled with (1) in Figures 11(b) and 11(c). In these figures the extent of the structure represented by the node is shown as a gray ring sector labeled with (3) and the extent of the parent structure is labeled with (2). The current slice position is rendered as a black or red line labeled with (4). If the node is selected, the extents of all child structures are indicated in the slice bar using the same colors as for the volume rendering and for the edges. The extents are labeled as (5) in Figure 11(c).

Hierarchical liftcharts are integrated interactions because they require information about the current slice position and also whether the node is focused or not. A focused node results in a different visualization than an unfocused node. The visualization consists of rendering the current slice and the slice bar which is depicted in the bottom part of the node.

4.7 Integrated Interaction and Abstract Visualization

Integrated interactions that result in abstract visualizations use hierarchical and spatial data but provide visualizations that only apply to the abstract domain.

Property labeling: Let us assume that a whole series of data sets is available, e.g., from a longitudinal study. It might be interesting to see how a specific dataset deviates from the average of the series. Figure 12(a) illustrates our approach in this respect. The structure sizes (voxel counts) for several segmentations in the brain have been measured in a certain population. We compare the visualized brain against the average of the series.

Figure 12(a) shows a part of the brain and gives a comparison with the average structure size. The distance from the average is indicated in color. The color scale is from orange to purple, where orange encodes an above average situation while purple encodes a below average situation. The relative deviation can also be read off from the deviation legend shown in Figures 12(b) and 12(c). The legend shows the color scale and with a black line the location of the color applied to the node. The deviations from the average are aggregated hierarchically. The averages are calculated for all structures and compared hierarchically.

The hierarchical aggregation of a chosen statistics requires both abstract data and spatial data. The resulting visualization is a color change in the abstract domain only.

Scatter plots: In a longitudinal study it is common to also record more than one metric. To visualize such information, scatter plots have been included inside the node rendering. This can be seen in Figure 13. The scatter plot shows the relationship between a patient's age and the number of voxels for a structure. Blue dots represent males, pink dots represent females and a green dot is the current subject. The displayed lines are separate linear regression lines for each sex. The scatter plots are aggregated hierarchically, similarly to the property labeling.

The hierarchical aggregation of a chosen statistic requires both abstract data and spatial data. The resulting visualization is a new abstract visualization of the statistical data composited on top of the volume renderings.

5 Implementation and Performance

The implementation of the rendering system has been done in Java. OpenGL and the OpenGL Shading Language have been used for the graphical rendering. An off-the-shelf graph layouting library is used to position the nodes according to the Balloon placement algorithm. Rendering the node tree has been implemented as a multi-pass algorithm using the visitor pattern [6]. Every pass renders one layer of the final image and the layers are composited together.

The algorithm calculating the selection outline described in Section 4.5 and shown in Figures 1 and 8(a) is a pixel based approach for finding edges of structures. In a separate raycasting pass over ancestral structures of a focused node, a buffer is filled with values that represent one of three cases: 0 if the ray does not hit any segmentations associated with the focused structure, 1 if the ray hits a focused segmentation and 2 if the first segmentation hit is of the focused structure. The resulting buffer is then processed to identify two types of edges by checking gradients: from segmentation hit (1 and 2) to no segmentation hit (0) and first segmentation hit (2) to segmentation hit (1). The identified edges are colored in black. The resulting lines are then dilated to increase thickness and a halo is added to increase visibility. Finally the outline is overlaid on top of the volume raycasting image.

The occluded-structures algorithm described in Section 4.6 and shown in Figures 4 and 10 assumes that structures are opaque. In the pass which calculates the selection outline the identity of the first segmentation hit is stored. The number of pixels for each segmentation is counted using the OpenGL extension ARB_occlusion_query. The total number of pixels for a hierarchical structure is summed up and the relative size of the substructure is calculated. In Figure 4 the focused structure is the right coxa. Since the number of pixels from the pubis that contribute to the image is zero, the pubis from the point of view of the coxa is completely occluded.

On a Dell Precision T5400 using a single thread with NVidia 280 GTX the system performs at interactive speeds. For example, rendering the image seen in Figure 10 at a 1000×1000 pixel resolution we achieve a performance of approximately 11 fps. Zooming in to only render the sub-tree shown in Figure 11(a) increases the performance to 40 fps. The increase is mostly due to the reduced number of visible structures. The largest performance bottleneck of the system is the volume resolution. Larger volumes increase the processing times of the raycasting, selection-outline and occlusion-testing algorithms.

6 Results

The results presented in this paper are based on several different datasets. Figures 1 and 9 use a CT scan of the head and neck with contrast enhanced sinus veins, at a resolution of $512 \times 512 \times 333$ voxels, with several anatomical structures segmented. Figures 4, 5, 8, and 14 are generated with a segmentation of the right leg of the Visible Male CT dataset [12] in full resolution cropped to a resolution of $268 \times 243 \times 1136$ voxels. Figures 6, 7, 10 and 11 use the Bert dataset provided by FreeSurfer [5]. Finally, the dataset used in Figures 12 and 13 are from the OASIS brains database [13] consisting of more than 400 segmented brains with associated meta information, such as age, gender, education and so on. Both Bert and OASIS volumes have a resolution of $256 \times 256 \times 256$ voxels.

In this section we compare our approach to techniques that are also used to visualize data defined over multiple spaces. We have specifically chosen linked view visualizations as these would be a natural choice for visualization of such multi-domain datasets. A mock-up has been constructed in such a way as to not rely on integrated visualizations. In addition the visualizations have the same area available, this means they have the equivalent number of pixels at their disposal. The mock-up is presented in Figure 14. This example illustrates slicing and the linked-view example, shown in

Figure 14(a), conveys this through four different views. The first view shows the labeled hierarchy associated with the data where four interesting labels have been color coded. The color coding is used in the three adjacent views as well. The second view is a LIFTCHART showing the positions of the selected structures in relation to the complete dataset and the current slice position as a red line. The third view shows a coronal slice with the intersected structures color coded. The fourth view shows a volume rendering of the structures with color coding as well. Our approach is shown in Figure 14(b) where the hierarchy is indicated through the node-link graph and every node has hierarchical liftcharts. Every structure that intersects the current slice has the slice integrated into the corresponding volume rendering. The benefit is that the substructures show enlarged localized slices and the spatial position of the slice is easy to comprehend. In addition the node-outline color indicates the relative position of a slice.

From this and similar mock-ups we hypothesize a major advantage of our approach compared to linked views, i.e., having a singular focus of attention. As all the information is presented in one view, the attention of the user does not have to shift between various images. There is also redundancy in the presentation of the information. For example the graph clearly indicates the child-parent relationship between structures. The coloring of substructures and connection lines also conveys this information. Since we are only using one view another advantage is that we can present more information than in a linked-view setup. As the number of views increases in linked views the available space for an individual view decreases there.

7 Conclusions and Future Work

In this paper we have introduced an integrated visualization that bridges spatial and hierarchical domains. We have proposed a classification of visualizations and interactions that can be organized in a 3×3 matrix depending on whether they are of abstract, integrated, or spatial nature.

The increased occurrence of quite heterogeneous data sources for the same real-world phenomenon requires integrated visualization approaches. The currently available algorithms are mostly tailored to a specific data space, e.g., abstract or spatial. The increasingly prevalent heterogeneous data sources make it necessary to develop new algorithms.

In the case of integrated visualizations invoked by interactions in one of the domains, a useful approach in developing new techniques is realized as a two-stage process. First we identify a basic interaction metaphor in one domain, such as *select* or *show*, for example. Then we seek for a specific visualization in the other domain that realizes the respective meaning of this interaction. This way an interaction can result in visualizations that expand beyond the borders of its original domain. Examples of this type of techniques can be found in Sections 4.3 and 4.4. For integrated interactions we have not found such a systematic integration approach and their discovery was rather stimulated by practical needs and experiments.

Illustrative visualization covers many techniques which mimic the approaches that illustrators use. These techniques include exploded views, cut-away views, peel-aways, labels and many other techniques to achieve visualizations with an illustrative presentation. Illustrations that present a scientific topic often have to convey hierarchical information, but only a few of the mentioned techniques are directly applicable in the hierarchical context. We see our work as one element of interactive direct volume illustrations which specifically addresses the hierarchical aspect of scientific data. With this in mind it would be interesting to take the idea of interactive illustrations one step further and to combine several of these illustrative techniques in establishing a visualization toolbox for interactive poster generation. Our work can be one tool to show the hierarchical characteristics within the 3D structures.

8 Acknowledgments

We would like to thank S. Bruckner, D. Patel, P. Rautek, M. Termeer and M. Haidacher for helpful suggestions and comments. This work has been supported by MedViz network, Bergen, Norway.

References

[1] J.-P. Balabanian, M. Ystad, I. Viola, A. Lundervold, H. Hauser, and E. Gröller. Hierarchical volume visualization of brain anatomy. In *VMV 2008*, pages 313–322, Oct. 2008.

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

[3] M.-Y. Chan, H. Qu, K.-K. Chung, W.-H. Mak, and Y. Wu. Relation-aware volume exploration pipeline. *IEEE TVCG*, 14(6):1683–1690, 2008.

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

[5] Freesurfer. http://surfer.nmr.mgh.harvard.edu, 2009.

[6] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

[7] H. Hauser. *Scientific Visualization: The Visual Extraction of Knowledge from Data*, chapter Generalizing Focus+Context Visualization, pages 305–327. Springer, 2005.

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

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

[10] W. Li, M. Agrawala, B. Curless, and D. Salesin. Automated generation of interactive 3d exploded view diagrams. *ACM Trans. Graph.*, 27(3):1–7, 2008.

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

[12] The visible human project. http://www.nlm.nih.gov/research/visible/, 2009.

[13] Open access series of imaging studies (OASIS). http://www.oasis-brains.org/, 2009.

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

[15] C. Tietjen, B. Meyer, S. Schlechtweg, B. Preim, I. Hertel, and G. Strauß. Enhancing Slice-based Visualizations of Medical Volume Data. In *EuroVis'06*, pages 123–130, 2006.

[16] Y. Wang, S. Teoh, and K.-L. Ma. Evaluating the effectiveness of tree visualization systems for knowledge discovery. In *EuroVis'06*, pages 67–74, 2006.

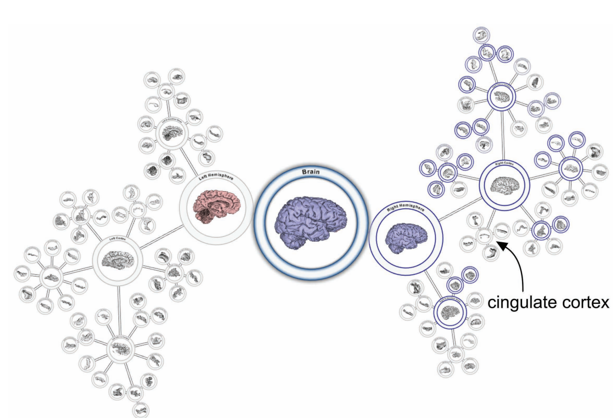
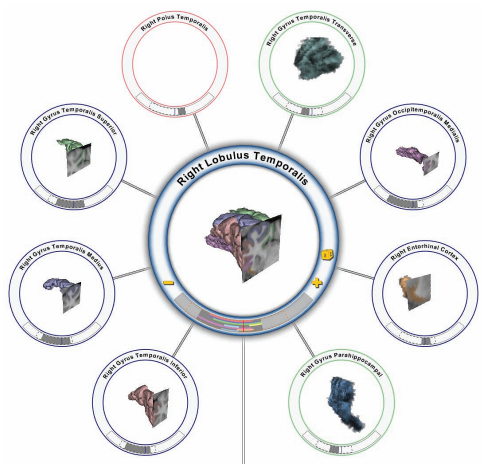
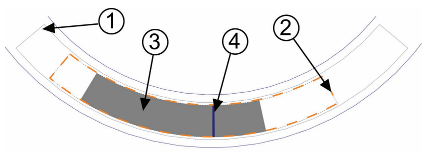


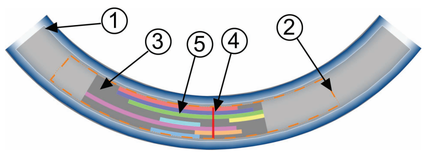
Figure 10: Gray nodes indicate occluded structures. The left hemisphere and several structures in the right hemisphere are occluded.



(a)



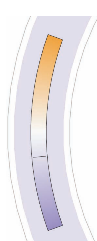
(b)



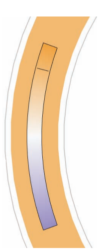
(c)

Figure 11: (a) Node rendering with selected node. A green node indicates the slicing plane is in front of the structure, a red node indicates the slicing plane is behind the structure and a blue node indicates a slicing plane that intersects the structure. Closeup of (b) unselected slice bar and (c) selected slice bar. In (b) and (c) the bar shows the bounding volume (1), parent extent (2), structure extent (3), slice position (4) and extent of child structures (5).

(a)



(b)



(c)

Figure 12: (a) Several nodes colored based on the deviation from an average structure. Orange is above, purple is below and white equals the average. (b) Legend indicating structure is below average. (c) Legend indicating structure is above average.

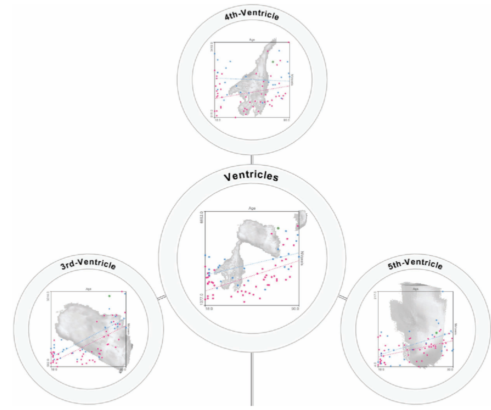
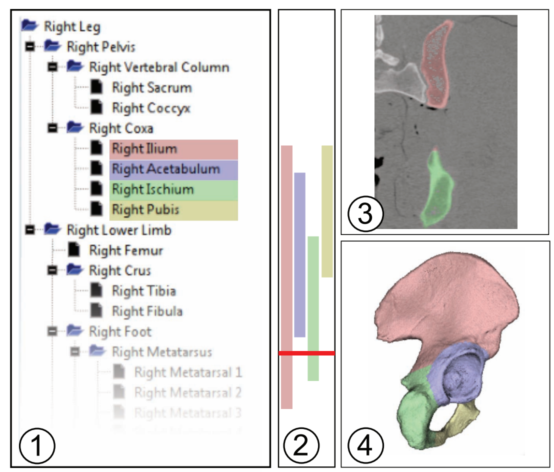
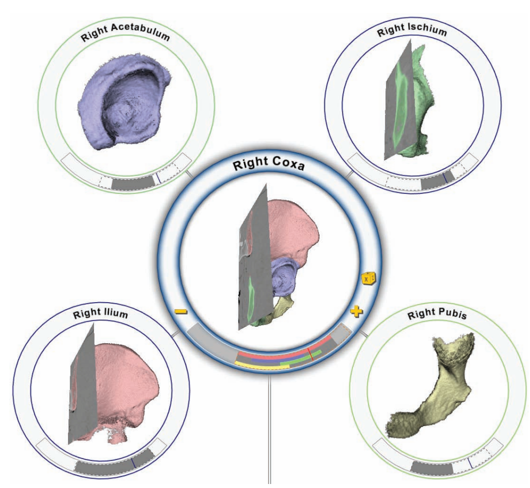


Figure 13: Several nodes with a composite of volume rendering and scatter plots.



(a)



(b)

Figure 14: (a) Linked view showing (1) hierarchy with labels, (2) LIFTCHART of the segments of the coxa, (3) coronal slice and (4) volume rendering. (b) An integrated visualization conveying similar information as the visualization in (a).