

Ontology-Based Visualization of Hierarchical Neuroanatomical Structures

A. Kuß¹, S. Prohaska¹, B. Meyer¹, J. Rybak², and H.-C. Hege¹

¹Zuse Institute Berlin

²Institut für Neurobiologie, Freie Universität Berlin

Abstract

This work presents a method for generating intuitive visualizations for high-level user queries to an hierarchical surface-based neuroanatomical atlas. We combine a spreading activation approach for computing focus and context in an ontology with a specific level-of-detail strategy for hierarchical structures. We demonstrate our method on an atlas of the bee brain.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques – Interaction Techniques; J.3 [Life and Medical Sciences]: Biology and Genetics

1. Introduction

Anatomy on organ, tissue, and cell level is fundamental to the understanding of living systems. Studying and analyzing anatomical structures and relations is part of education and research in biomedicine. Anatomy related tasks in research are, for example, accumulation and fusion of data and establishment of spatial, functional, and genetic relationships. In education, students want to acquire anatomical knowledge that often is difficult to grasp due to complex spatial relationships.

Anatomical atlases combined with semantic information and visualization can support understanding of anatomy. During specimen preparation, for example, biologists want to locate and explore specific structures and understand their spatial neighborhood. For education and presentation of research results, scientists want to create descriptive visualizations. Existing approaches based on semantic information concentrate on retrieving information from knowledge bases. These often require expert knowledge of the concepts used for knowledge representation, such as relation types or specific query languages [LM07]. Visualization approaches, so far, produce only basic results [RDB*06] or require a lot of user interaction [MZG*08].

From a user's perspective, the visualization of anatomy should provide an intuitive user interface that is easy to use and requires little interaction. The visualization should allow

users to display existing anatomical knowledge and it should allow experts to extend the knowledge base available to other users.

This work contributes an automatic method for creating intuitive visualizations of surface-based anatomy atlases. The visualizations facilitate the understanding of the hierarchical organization and the spatial and functional relations. We combine a spreading activation approach for computing focus+context in an ontology with a specific level-of-detail strategy for hierarchical structures. Expert knowledge is formalized in the ontology and in predefined queries, which a user can choose from. We demonstrate the effectiveness of our method on an atlas of the honeybee brain.

2. Related Work

Several educational anatomy browsers (research prototypes and commercial products) have been developed in recent years. Especially the anatomy of the human body with focus on the human brain has been modeled (for an overview see [PB07]). The browsers support flexible and interactive exploration of labeled anatomic structures and also provide knowledge bases representing semantic information. They are very well suited for anatomy education, but have limited importance for anatomy research since they are not intended to let users integrate, analyze and compare new data.

The semantic information underlying anatomy browsers often results from ontology development driven by specialists from the application fields, mainly biology and medicine. Many ontologies have been developed in the last decade. They range from ontologies for cell types [BRA04] and cell function [HRN02] to ontologies for subcellular neuroanatomy [FLG*07], mouse embryo anatomy [BDB04], and human anatomy [RJ08]. Some of the them enrich digital atlases by enabling reasoning and the presentation of additional information [BDB04, MGLD*04]. Those digital atlases serve as reference frames that allow for relating inter- and intra-individual data from different imaging modalities and experiments together with corresponding geometry reconstructions and semantic information [BRR*05, Ju05, PH04, PSR*94]. Currently they are used to accumulate and manage data which can be browsed via standalone applications or web interfaces [All06].

Techniques for the creation of visualizations that distinguish between objects of interest and the surroundings are commonly known as focus+context visualization. To create focus+context visualizations a classification of objects into focus and context is required. This discrimination can either be binary or smooth [Hau04]. While the selection of one specific focus object in atlas-based applications is performed by the user (e.g. by selecting a structure in a list of objects or by directly picking in the visualization), the determination of other important entities has to be implemented by the system. Some distortion-oriented focus+context methods, such as fisheye-views or hyperbolic trees, implicitly assign medial importance values to objects in the spatial vicinity of the focus by drawing them with medium magnification.

Methods to interactively adapt focus+context visualizations of volume data to user input have been presented in [VFSG06, RBG08]. The importance values used to weight object visibility and rendering style are assumed to be given or set to “high” for the selected object and “low” for all others. Assignment of importance values that do not depend on the spatial relations but on semantic information have been described as cue methods by Kosara [Kos01]. The specification of these importance values depends on the application. In our case ontology queries deliver the required ranking. We adapt and extend the spreading activation approach presented in [HSHS02].

3. Problem Statements

When creating query-based visualizations of neuroanatomical structures several problems arise. We here consider the definition of anatomical parts and the interpretation of user queries as the most important problems.

3.1. Defining Parts in Neuroanatomy

Anatomy describes an organism as a number of tissues each hierarchically subdivided into parts. To successfully

integrate this information in an ontology several problems need to be addressed. Bard [Bar08] describes them as “the anatomical problems”: It is necessary to define boundaries between the different parts, although such boundaries are not always obvious from the experimental data available. It is further necessary to organize the parts in a meaningful way; to decide how many parts to include and which relation between them; to handle variability between specimens; and to handle incomplete data.

The *hasPart/partOf* relation, which is the core anatomical relation, is not as easy to define as it might seem. The Foundational Model of Anatomy Ontology (FMA) [RJ08] requires for this relation that “if A *hasPart* B, [then] there is a complement C which together with B accounts for the whole (100%) of A”. Though plausible, it might not be obvious how to strictly fulfill this requirement in practice. For example, it might be desirable to omit small or unimportant parts from anatomical consideration.

In brain tissue, the *hasPart/partOf* relation is particularly problematic on the cell level because neurons usually cross boundaries of anatomical regions. Classification by these regions hence breaks the FMA requirement. The Subcellular Anatomy Ontology (SAO) of the nervous system [FLG*07] addresses this problem by starting the *hasPart* hierarchy from the individual cell and situating cell parts in anatomic regions using a different *locatedIn* relation. This solves the problem on the cellular scale, but similar problems might arise on other scales, for example, for molecular structures inside the cell.

Visualization of neuroanatomy needs to deal with the anatomical problems described. It should, first, intuitively depict the organizational grouping of parts; second, handle structures on different scales, which might introduce different meanings of the *partOf* relations; third, handle incompleteness, that is parts whose sub-parts do not account for the whole.

3.2. Usage of High-level Queries

Different users want to answer different questions by atlases, yet these users have in common that they want to specify their questions in a natural, easy way, that is as *high-level queries*. Examples for such queries are questions about the position of a specific object, questions about the hierarchical composition of an object, or questions about the interaction between an object and its neighborhood. For example, a student might ask to “see a specific neuron” or a researcher might ask to “see the location of presynaptic swellings of a specific neuron”.

We limit the queries supported by the system to high-level queries that can be specified by the user as a structure of interest combined with a query from a predefined set of visualization queries, such as “show overview” or “show loca-

tion". In addition, the user can control the level of detail of final visualization by a single slider.

4. Ontology-based Visualization of 3D Biological Structures

The core steps of our method for the creation of visualizations of neuroanatomical structures for a high-level user query are:

1. An expert develops an ontology with a specific structure, suitable for deriving visualizations. This ontology is linked to the available geometries.
2. An expert defines high-level visualization queries that specify a set of relevant relations.
3. A user selects a focus object and a visualization query.
4. A graph algorithm generates query-dependent importance values for each structure.
5. These importance values are mapped to visualization parameters such as transparency.
6. The user can control the level of detail of the visualization.

Figure 1 illustrates the workflow. Steps one and two form preprocessing steps in which the semantic information is created (see Section 5). The interactive generation of visualizations represented by step three to six takes place at runtime (see Section 6).

5. Definition of an Ontology for Visualization of 3D Biological Structures

Our method uses semantic information stored as an ontology which is linked to geometries. An ontology is a machine-readable formal representation of classes, instances of classes, relationships and attributes. Classes and instances are connected to other classes and instances via relations. An example for a class is *head*, a subclass would be *human head* and an instance of *the human head* would be *head of patient A*. The *head of patient A* would be related to the *brain of patient A* via the *hasPart* relation and the attribute *geometry* would contain the name and file location of the reconstruction of the patient's brain.

The ontology O can be described as a directed labeled graph (V, E) with a set of vertices V and a set of edges E . An edge $e \in E$ is defined between two vertices $(v_i \rightarrow v_j; v_i, v_j \in V)$, where v_i is the *source* vertex and v_j is the *target* vertex. Each vertex as well as each edge can have several parameters (attributes). The parameters can be elements of \mathbb{R} , \mathbb{N} , or of an enumerated set of vertex labels $L_{vertex} : \{a_0, \dots, a_m\}$ or edge labels $L_{edge} : \{b_0, \dots, b_n\}$, with $L_{vertex} \cap L_{edge} = \emptyset$.

We define the parameters *vertex type* t_{vertex} and *edge type* t_{edge} as the functions $t_{vertex} : V \rightarrow L_{vertex}$ and $t_{edge} : E \rightarrow L_{edge}$. From now on, we omit the subscripts *vertex* and *edge* and simply write t because the argument clearly indicates which function we refer to. If two vertices v_1, v_2

are connected by an edge e of type $t(e) = isSomething \in L_{edge}$, then we also say e has edge type *isSomething* and v_1 *isSomething* v_2 .

The *instance graph* I is the set of vertices $V_I = \{v | v \in V \wedge t(v) = instance\}$ and all their edges $E_I \subset E$. We define the *class tree* C as the set of vertices $V_C = \{v | v \in V \wedge t(v) = class\}$ and edges $E_C = \{e | e \in E \wedge t(e) = subClassOf\}$. The vertices of the instance graph V_I are connected to vertices of the class tree V_C via edges of type *isInstanceOf*. If v_i *isInstanceOf* v_j for $v_i \in V_I$ and $v_j \in V_C$, then v_j is called *class* of v_i .

To cover the problems that arise with the *partOf* relation (see Section 3.1), we take the different hierarchies and concepts in anatomical structures into account. Information about scale is included by a scalar that ranks the elements of V_C .

Hierarchic structures. In order to integrate the hierarchical organization of subparts in the ontology, we define the edge types *hasPart* $\in L_{edge}$ and *isPart* $\in L_{edge}$ with following properties:

$$v_1 \text{ hasPart } v_2 \wedge v_2 \text{ hasPart } v_3 \rightarrow v_1 \text{ hasPart } v_3 \quad (1)$$

$$v_1 \text{ hasPart } v_2 \rightarrow v_2 \text{ isPart } v_1 \quad (2)$$

That is *hasPart* and *isPart* constitute two transitive relations that are inverse to each other.

Scale value. We introduce the function $\sigma : V_C \rightarrow \mathbb{N}$, which assigns a *scale value* for a specific vertex v_i . The scale value is set by the expert to ensure the correct ordering of the structures through the different levels of detail. It can for example be set to distinguish between brain regions (the coarse scale) and individual neurons (the finer scale). Note that the scale value is specified on the class tree. Thus, the scale value is independent of a specific individual.

Visibility value. We introduce the function $\phi : V_I \rightarrow [0, 1]$, which assigns a *visibility value* to the vertices. This value encodes if and how structures at a certain hierarchy level are visualized when their substructures are at the focus of the current visualization. This value is set during ontology development and is stored on the instance graph. That way, it can vary between instances of individuals to account for their variability, which cannot be captured in the class tree. It is also useful for the instantiation of individual data with incomplete substructures.

5.1. Definition of Visualization Queries

Besides the ontology, the expert also defines a set of visualization queries Q and provides a mapping of each query to a subset of the edge labels, called the *query mapping* $M : Q \rightarrow \mathcal{P}(L_{edge})$. The query mapping determines which relations in the ontology are significant for a specific query. It can also be stored in the ontology.

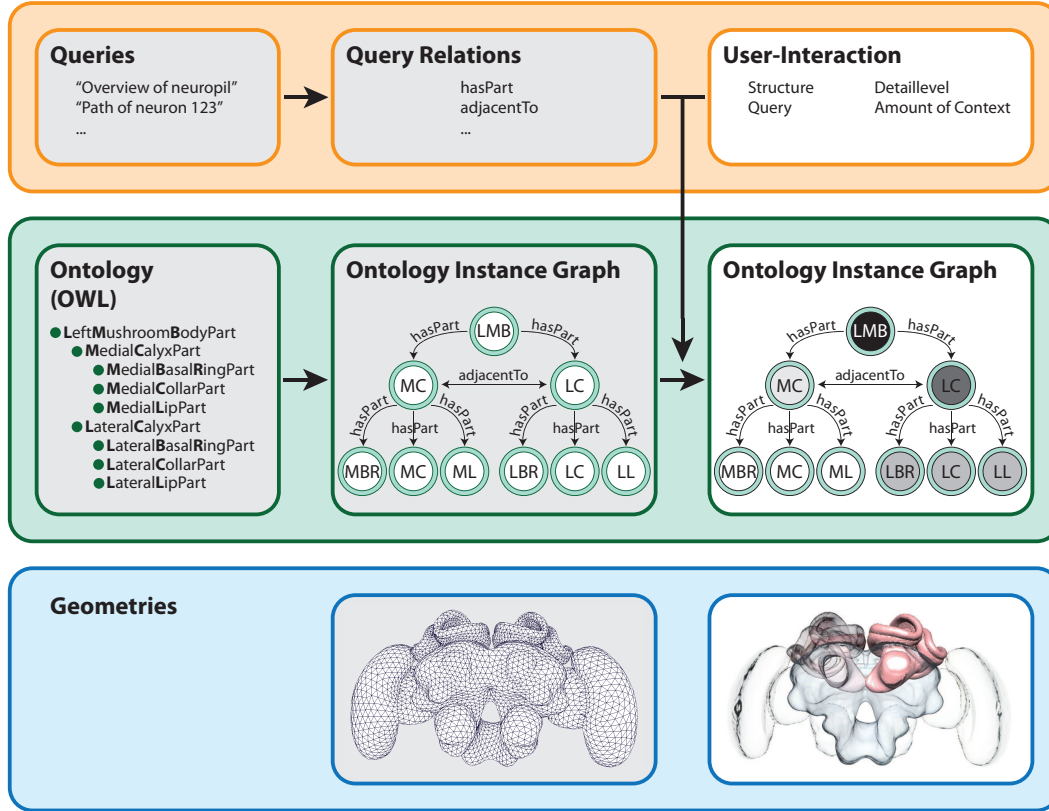


Figure 1: Workflow of our ontology-based visualization approach. First an ontology graph storing semantic information is created. Afterwards high-level queries are defined on the relations of the ontology. The user can ask for a visualization of a structure. This query is evaluated and the vertices of the ontology graph receive query-dependent importance values. Geometries are linked to the graph vertices and are visualized using importance-dependent parameters. Textboxes having a white background show processes during run-time. The example geometries show parts of the beebrain.

6. Query-Dependent Generation of Visualizations

Having defined the ontology and the queries our system is now able to answer visualization queries. The user specifies his query by choosing a predefined visualization query $q \in Q$ and selecting an object of interest $s \in V_i$, which will determine the focus. Later, the user can also select a level of detail (see Section 6.3).

Based on the user input the system computes vertex importance values that are used for focus and context discrimination. The approach first determines the set of focus objects by a hierarchical search, secondly the context is identified by a spreading activation approach. Finally the determined importance values are mapped to visualization parameters.

6.1. Hierarchical Focus Determination

Given the ontology created by the expert and the selected focus by the user, the following algorithm computes the set of focus vertices V_{focus} and a level parameter for each of

the focus vertices that will later determine at which level of detail the corresponding object will be displayed.

The levels are computed based on the relation in the class tree, because the class hierarchy represents the abstract anatomical knowledge that is independent of the individual. Thus the class hierarchy is better suited for capturing the different scales of an anatomy in a robust way. For example, the class tree allows for computing sensibly sorted levels even if certain anatomical structures are missing in the instance graph, which might happen accidentally or on purpose.

Algorithm

Expert Input: ontology O , scale value σ

User Input: selection s

Output: V_{focus} , level : $V_{focus} \rightarrow \mathbb{N}$, maxDetail

1. Start at focus structure s and search for the vertices $V_{focus} \subset V_I$ that are connected through edges $e \in E_I$ of type $t(e) = hasPart$.
2. For each $v_i \in V_{focus}$ compute the maximal path length

$d[v_i]$ from the class $v_j \in V_C$ of v_i to a vertex $v_k \in V_C$ of the class tree with $\sigma(v_j) = \sigma(v_k)$, and insert the maximal path length into the set of path lengths for its scale, i.e. $L[\sigma(v_i)] \leftarrow L[\sigma(v_i)] \cup d[v_i]$.

3. Then $scaleLevels(i) = |L[i]|$ provides the number of different maximal path lengths for each scale and the overall maximum detail level is the sum $maxDetail \leftarrow \sum_i scaleLevels[i]$.
4. Assign consecutive levels to the vertices, scale by scale and for the same scale maximal paths length by maximal path length. Formally: $\iota(v_i) = (\sigma[v_i], d[v_i])$ maps each vertex to a tuple containing its scale and its maximal path length, and ι^{-1} maps such a tuple to one or multiple vertices. Initialize a layer counter $l \leftarrow 0$ and for each $i = 0 \dots$ and for each $d \in sort(L[i])$, assign all $v \in \iota^{-1}(i, d)$ the current level counter, i.e. $level(v) \leftarrow l$, and increase the level $l \leftarrow l + 1$.

6.2. Context Determination

We further determine an appropriate context for a selection s and a query $q \in Q$ by the graph-based algorithm below. It computes an *importance value* T using the concept of *spreading activation* described in [HSHS02]. The idea is to start at selection s in the graph with maximum *importance* and spread a decreased amount of this *importance* along the edges which are defined by the selected visualization query q to incident nodes. From the incident nodes a certain amount of their *importance* is transferred to the next incident nodes and so forth. Structures that are for example directly adjacent to the selection will get a higher importance than structures further away. An additional parameter, *maxdepth*, can be modified to control the spreading range in the graph. A low *maxdepth* will lead to a high importance reduction for distant structures and is useful for scenarios where only structures in a close vicinity are of interest. A higher *maxdepth* will result in a lower reduction of importance and will thus emphasize more structures. This is a useful choice for the creation of overviews.

Algorithm

Expert Input: *instance graph* I , *query mapping* M

User Input: *selection* s , *query* q , *maxdepth*

Output: $V_{context}$,

context importance $T_{context} : V_{context} \rightarrow [0, 1]$

1. Starting at focus structure s search for all vertices $V_{context} \subset V_I \setminus V_{focus}$ that are connected via edges of types that the query maps to, i.e. $M(q)$. We only add vertices with $depth(v_i) \leq maxdepth$, where $depth(v_i)$ is the path length from s to v_i .
2. For each $v_i \in V_{context}$ set $T_{context}(v_i) = \frac{depth(v_i)}{maxdepth}$.

6.3. Mapping to Visualization Parameters

The user can control the visibility of the focus objects in the final visualization by setting the level of detail $D \leq$

maxDetail. The transparency $T_{focus}(v_i)$ of the objects that correspond to the focus vertices $v_i \in V_{focus}$ is set to:

$$T_{focus}(v_i) = \begin{cases} \phi(v_i), & \text{if } level(v_i) < D \\ 0, & \text{if } level(v_i) = D \\ 1, & \text{if } level(v_i) > D \end{cases} \quad (3)$$

with $0 \leq T_{focus}(v_i) \leq 1$.

The appearance of the context objects is controlled by the context importance $T_{context}$. Their importance values can be mapped to different visualization parameters. Parameters that we experimented with are transparency and saturation.

7. Case Study: Bee Brain Atlas

The 3D surface-based averaged atlas used for the case study was reconstructed in AMIRA [SWH05] out of 30 bee brains as described in [BRR*05]. It comprises 22 main structures, the so called neuropils. A neuropil describes a tangle of dendrites, axons and glial processes. Step by step, more detailed structures were integrated into the atlas, ranging from smaller neuropils down to presynaptic swellings (boutons) and postsynaptic spine (microcircuits or microcircuitry) of nerve cells (neurons).

We used the ontology editor Protégé 3.4 to develop our ontology. The result is stored as a Web Ontology Language OWL [SWM04] document, which represents the graph described in Section 4.

We modelled a suitable ontology starting from an existing established formalization. The Foundational Model of Anatomy (FMA) [RJ08] provides the clear and well defined classes *Cell*, *CellPart*, *Organ* and *OrganPart*, which we used as a starting point. Further subclass definitions were taken from anatomy textbooks and refined in discussions with experts.

Scale values were set for the classes *CellPart* (1) and *OrganPart* (0). Appropriate *visibility values* (see Section 5) were set in cooperation with neurobiologists. In our case this value was especially needed to ensure a correct visualization in regions where data was missing. The sum of subparts of a structure should give an impression of the structure's original shape. However sometimes there is not enough data available to fulfill this requirement and the visualization may cause a wrong impression. In these cases it is useful to render a transparent depiction of the structure to convey its complete shape. This transparency can be expressed by the *visibility values*.

Together with neurobiologists who actively use the bee brain atlas, we developed scenarios where a meaningful visualization is desired. For each scenario, we defined relevant structures that should be depicted by the visualization. Formalized queries were derived and mapped to a set of important relations that are required to achieve the desired visual-

Query	Desired Visualization	Relevant Structures	Important Relations
Neuropil Overview	Where can the neuropils and its substructures be found?	Spatial context	adjacentTo isSurrounded
Neuron Path	Show the neuron together with its input and output regions in a spatial context.	Input Region Output Region Tract Spatial Context	hasInput hasOutput hasTermination adjacentTo

Figure 2: Typical examples of visualization queries in a digital neurobiological atlas.

ization. These queries and their relations were also stored in the ontology. Neurobiologists are especially interested in the precise location of neuron parts, neuron pathways and neuron function. For the planning of experiments they show a high interest in the location of neuronal elements relative to gross structures which can support the initial placing of their recording tools. Figure 2 presents two examples for visualization queries and their important edge types.

For our expert the ontology and query creation which comprised the above mentioned modeling steps took about a week. The current version of the ontology contains 1200 edges of 20 edge types and 100 classes coupled to 300 instances. Also contained are the queries. Using the editor the ontology can easily be extended.

The first step in the final application is the parsing of the OWL document to create a representation of the *directed acyclic graph* in AMIRA. Note that this step is only necessary if the ontology was changed. It takes about 10s on a current standard PC. The user then chooses a focus structure from a hierarchical ordered list and one of the pre-defined queries. Our system determines the focus and context vertices and their visualization parameters which takes less than a second. Afterwards the according geometries are loaded using an ID attribute stored on the graph (see Section 5). A detail level slider is available for browsing of substructures. Furthermore, the user can choose between a sparse or a strong presentation of context structures (see Section 6).

7.1. Examples of Visualizations

The path of a neuron, as listed in Figure 2, is for example interesting for the planning of an experiment where a response of a certain neuron type to a stimulus is recorded. Here it is important to know where exactly the neuron starts and gets its input and where it might produce an output. The path of a neuron is described by the location of the neuron's subparts the soma, the axon, and the dendrites. The location of the soma and the dendrites is considered to be the input region of the neuron. Accordingly the output region is the location of the axon, especially the axon terminals. Tracts describe the ways of the neuron between their input and output regions.

The neuron shown in Figure 3 was not completely recorded due to limitations in the imaging method. Nevertheless its

context can be determined by traversing along the edges of type *hasInput*, *hasTract*, *hasOutput*, and *adjacentTo* in the ontology. The context nodes received an importance value which is mapped to transparency. This context clarifies the path of the neuron. It's subparts needed for hierarchy browsing are returned by searching for nodes connected via edges of type *hasPart*.

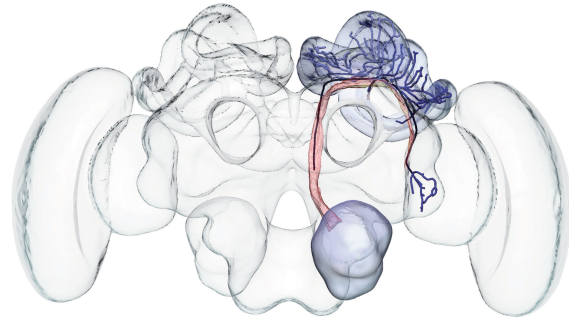


Figure 3: Overview of the path of a neuron. The neuron's input and output regions and its tract are emphasized. A lower transparency is used for the context structures.

In Figure 4 the neuropil *mushroom body* has been selected and displayed with spatial context. The query for this example was "Overview" with *maxDepth* = 2. For an "Overview" of a neuropil mainly the subparts and a spatial context are needed. After determining V_{focus} the edges of type *adjacentTo* are used to compute the context structures and their transparencies.

8. Discussion

Although it requires a domain expert to represent the anatomical and functional knowledge in an ontology, this effort is rewarded by a flexible and easily extendable system. New data can be fed into the ontology by creating new instances of existent classes and linking them to the geometry. To offer the user new visualization queries that emphasize other dependencies and properties of the data, a new set of relevant relations from the ontology has to be defined.

The definition of the *hasPart* relation is an application-dependent problem (see Section 3.1). After analyzing our

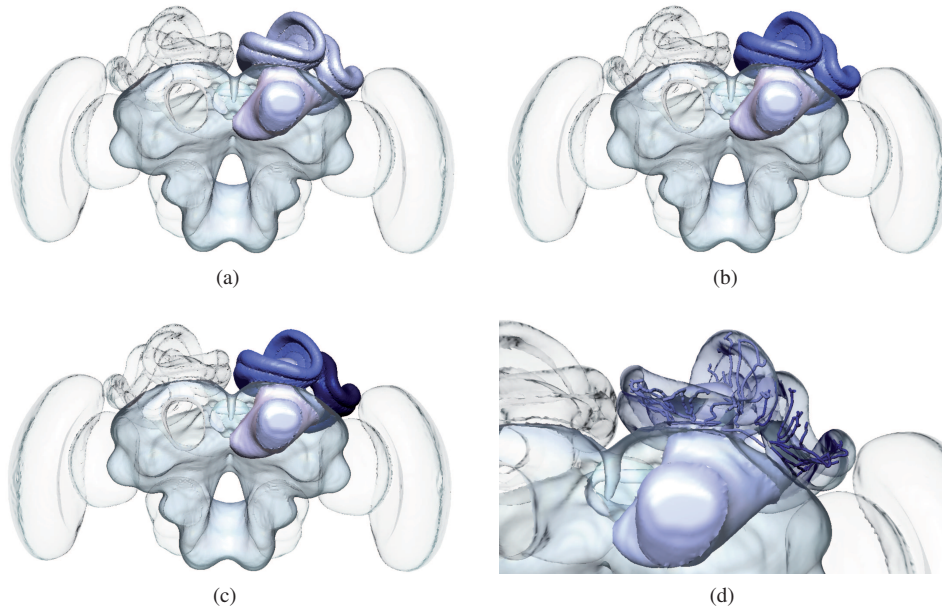


Figure 4: *Mushroom body overview. The user selected the structure "Mushroom Body" and chose the query "Overview". In (a), (b), (c) the subparts of the mushroom body are shown. In (d) the dendrites of neurons are shown while the superstructure still remains visible.*

data we considered the class and *hasPart* definition of the (FMA) to be most suitable for our application. Here the way classes and subclasses are modeled already implies a partitioning. *OrganParts* represent classes for anatomical regions that are composed of cells such as nerve cells. However *CellParts* are not treated as subclasses of *OrganParts*. They form an own class tree which addresses the characteristic of ontologies to be composed of several class trees. This idealized partitioning supports the correct determination of hierarchy levels for instances. Using the *scale value* introduced in Section 5 we can navigate through ontologies (class trees) of different length scales. Further the *visibility value*, also introduced in Section 5, visually supports the understanding of the composition of structures.

Our determination of focus and context structures and their associated visualization parameters is based on the spreading activation method presented in [HSHS02]. For the determination and visualization of context structures we successfully used a simplified version of the original spreading activation algorithm. The algorithm would be more powerful if weights were added to vertices and edges. In this way we could introduce importance differences for context structures.

Due to the fact that the spatial relationships between anatomical structures as well as their size are important features that should be depicted correctly, we use non-distorting techniques for the visual separation of focus and context. In our example we mapped importance values to transparency which resulted in very satisfying visualizations ac-

cording to our cooperation partners. However transparency is not always the best solution: variances in transparencies are hard to recognize and sometimes too much information is lost. Differences in importance could also be mapped to other visualization parameters such as blurring or styles of non-photorealistic rendering [HSHS02]. To further face the occlusion problem techniques such as the estimation of characteristic viewpoints, cut-away views or exploded views [VG05] should be integrated. These approaches could highly improve the quality of the visualization.

9. Conclusion and Future Work

We presented an approach to generate query-dependent visualizations of atlas data. The user interaction needed to create the visualization is reduced to a minimum by use of predefined high-level queries. Our method is based on the evaluation of semantic information from a specialized ontology. The determination of relevant context structures and visualization parameters takes the different scales of biological data into account. This way we provide an intuitive browsing tool for a structure's hierarchy.

A challenge in hierarchy browsing is the integration of geometrical level-of-details. Rendering highly detailed structures from a far viewpoint does not always yield a satisfactory visualization. For these cases geometric approximations of the models and their integration into the focus determination process would be of great use. To carry the automation process on, an intelligent algorithm to find important relations for a desired visualization might be investigated.

Acknowledgements

Anja Kuß is supported by the German Research Foundation (DFG, Grant HE 2948/4-1).

References

- [All06] ALLEN INSTITUTE FOR BRAIN SCIENCE: *Allen Brain Atlas*, 2006.
- [Bar08] BARD J.: Anatomical ontologies for model organisms: The fungi and animal. In *Anatomy Ontologies for Bioinformatics. Principle and Practice*, Albert Burger Duncan Davidson R. B., (Ed.). Springer, 2008.
- [BDB04] BURGER A., DAVIDSON D., BALDOCK R.: Formalization of Mouse Embryo Anatomy. *Bioinformatics* 20, 2 (2004), 259–267.
- [BRA04] BARD J., RHEE S., ASHBURNER M.: An ontology for cell types. *Genome Biol.* 20, 2 (2004), 259–267.
- [BRR*05] BRANDT R., ROHLFING T., RYBAK J., KROFCZIK S., MAYE A., WESTERHOFF M., HEGE H.-C., MENZEL R.: Three-Dimensional Average-Shape Atlas of the Honeybee Brain and Its Applications. *The Journal of Comparative Neurology* 492, 1 (2005), 1–19.
- [FLG*07] FONG L. L., LARSON S. D., GUPTA A., CON-DIT C., BUG W. J., CHEN L., WEST R., LAMONT S., TERADA M., MARTONE M. E.: An ontology-driven knowledge environment for subcellular neuroanatomy. In *CEUR Workshop Proceedings* (2007), pp. 1613–0073.
- [Hau04] HAUSER H.: *Generalizing Focus+Context Visualization*. PhD thesis, Vienna University of Technology, Mar. 2004.
- [HRN02] HUNTER P., ROBBINS P., NOBLE D.: The IUPS human physiome project. *Eur. J. of Physiology* 445, 1 (2002), 1–9.
- [HSHS02] HARTMANN K., SCHLECHTWEIG S., HEL-BING R., STROTHOTTE T.: Knowledge-Supported Graphical Illustration of Texts. In *Proc. of International Working Conference on Advanced Visual Interfaces, AVI 2002* (2002), ACM Press, New York, pp. 300–307.
- [Ju05] JU T.: *Building a 3D Atlas of the Mouse Brain*. PhD thesis, Rice University, April 2005.
- [Kos01] KOSARA R.: *Semantic Depth of Field - Using Blur for Focus + Context Visualization*. PhD thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2001.
- [LM07] LARSON S., MARTONE M.: Rule-based reasoning with a multi-scale neuroanatomical ontology. In *CEUR Workshop Proceedings* (2007), vol. 258.
- [MGLD*04] MACKENZIE-GRAHAM A., LEE E.-F., DI-NOV I., BOTA M., SHATTUCK D., RUFFINS S., HENG Y., KONSTANTINIDIS F., PITIOT A., DING Y., HU G., JACOBS R., TOGA A.: A Multimodal, Multidimensional Atlas of the C57BL/6J Mouse Brain. *J. Anatomy* 204, 2 (2004), 93–102.
- [MZG*08] MARTONE M., ZASLAVSKY I., GUPTA A., MEMON A., TRAN J., WONG W., FONG L., LARSON S., ELLISMAN M.: *Anatomy Ontologies for Bioinformatics*. Computational Biology. Springer, 2008, ch. 13, pp. 267–286.
- [PB07] PREIM B., BARTZ D.: *Visualization in Medicine: Theory, Algorithms, and Applications*. Morgan Kaufmann, 2007, ch. 20.
- [PH04] PEREANU W., HARTENSTEIN V.: Digital Three-Dimensional Models of Drosophila Development. *Curr. Opin. in Genetics & Development* 14, 4 (2004), 382–391.
- [PSR*94] POMMERT A., SCHUBERT R., RIEMER M., SCHIEMANN T., TIEDE U., HOEHNE K. H.: Symbolic Modeling of Human Anatomy for Visualization and Simulation. In *Proc. of SPIE. Visualization in Biomedical Computing* (1994), vol. 2359, pp. 412–423.
- [RBG08] RAUTEK P., BRUCKNER S., GRÖLLER M. E.: Interaction-Dependent Semantics for Illustrative Volume Rendering. In *Proc. Eurographics / IEEE VGTC Symposium on Visualization* (May 2008), vol. 27, pp. 847–854.
- [RDB*06] RUBIN D., DAMERON O., BASHIR Y., GROSSMAN D., DEV P., MUSEN M.: Using ontologies linked with geometric models to reason about penetrating injuries. *Artif Intell Med* 37, 3 (2006), 167–176.
- [RJ08] ROSSE C., JR. J. L. V. M.: The foundation model of anatomy ontology. In *Anatomy Ontologies for Bioinformatics. Principle and Practice*, Albert Burger Duncan Davidson R. B., (Ed.). Springer, 2008.
- [SWH05] STALLING D., WESTERHOFF M., HEGE H.-C.: Amira: A highly interactive system for visual data analysis. In *The Visualization Handbook*, Hansen C. D., Johnson C. R., (Eds.). Elsevier, 2005, ch. 38.
- [SWM04] SMITH M. K., WELTY C., MCGUINNESS D. L.: *OWL Web Ontology Language Guide*, 2004.
- [VFSG06] VIOLA I., FEIXAS M., SBERT M., GRÖLLER M. E.: Importance-driven focus of attention. *IEEE TVCG* 12, 5 (2006), 933–940.
- [VG05] VIOLA I., GRÖLLER M. E.: Smart visibility in visualization. In *Proc. of EG Workshop on Computational Aesthetics in Graphics, Visualization and Imaging* (2005), L. Neumann M. Sbert B. G. W. P., (Ed.), pp. 209–216.