

Volumetric Segmentation of Complex Bone Structures from Medical Imaging Data Using Reeb Graphs

Vitalis Wiens *

Supervised by: Max Hermann†

Institute of Computer Science II - Computer Graphics
University of Bonn
Bonn / Germany

Abstract

The exploration of medical imaging datasets often requires a segmentation of the images according to different materials or structures. Model-based algorithms excel in finding closed boundary contours enclosing the structure to be segmented. However, porous structures like Spongiosa have a complex topology and do not exhibit a unique single closed boundary contour. In order to enable segmentation of such complex structures we suggest a new algorithmic framework based on a Reeb graph representing the topological information. Each node in the graph corresponds to a connected region of voxels in a specific image slice while edges indicate connected regions between adjacent slices. Starting with a coarse segmentation, the corresponding graph is refined at critical nodes and the resulting connected components of the graph provide the final segmentation. We present two strategies for identifying critical nodes, one solely based on dynamic thresholding and one based on a single user specified pre-segmentation. The approach is evaluated on a dataset of 193 μ CT scans of rodent skulls which are segmented into skull, left and right mandible.

Keywords: Reeb graph, Image segmentation, Computed tomography

1 Introduction

Image segmentation is required for all applications examining individual structures in 3D image data. The increasing trend of statistical approaches (including model-based segmentation) requires simultaneous segmentation of many images. This makes automatic methods highly needed. Automatic classification of different materials based on Hounsfield scale [6] is only sufficient for simple structures. To deal with more complex structures, state-of-the-art algorithms require at least additional manual interaction for initialization, making them cumbersome in use for large databases. This motivated the presented *topo-*

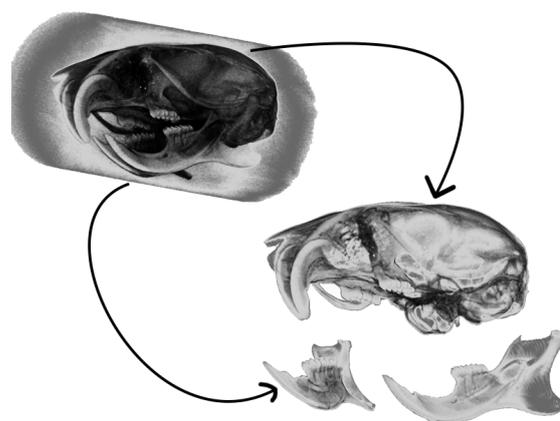


Figure 1: Segmentation of rodent skull μ CT scan into cranium, left and right mandible computed with the presented automatic method.

logical segmentation method, where we keep track of the global topology during the optimization to resolve otherwise ambiguous segmentation problems.

Formally the task of image segmentation is to divide an image volume I into regions T_i (segments) according to different materials or structures. For instance, partitioning of bone structures should ideally produce an anatomical decomposition into individual bones as shown in Figure 1. An image volume is a parallel stack of images, generated with a medical imaging device like CT/MRI scans where each single image represents a slice through the scanned object. The segmentation of bone structures in the rodent skull is considered a particular challenging example [13].

In contrast to simple thresholding, the goal of this paper is to find a segmentation where each bone structure is assigned to its own component T_i . To this end we represent the topology of the bone structures in a Reeb graph [5] which naturally leads to the formulation of the segmentation problem in terms of graph operations like vertex-splitting and computation of connected components. Particularly, each node in the graph corresponds to a connected region of pixels in a specific image slice, while edges indicate connected regions between adjacent slices.

*wiens@cs.uni-bonn.de

†hermann@cs.uni-bonn.de

A segmentation can now be derived from the connected components of the graph by taking the union of the corresponding regions for all nodes in a particular component.

Adopting the topological segmentation method, two approaches for bone segmentation are developed and evaluated. An example-based approach proves its performance on a dataset of 193 rodent skull μ CT scans from which 64% are segmented fully automatic with a very high accuracy as confirmed by a Morphometrics expert (A.C. Schunke, pers.comm.). Considering, that only a single manually segmented left mandible was required to produce this outcome, we consider this a particularly good result. A second approach demonstrates that for moderately challenging datasets, a segmentation can be achieved even without any prior knowledge about the topology (e.g. in form of an example).

The paper is organized as follows: After reviewing related work first, the topological segmentation method and the discrete Reeb graph will be introduced. Based on this method, two approaches will be presented in detail and results for a human skull and a foot dataset are presented as well as an evaluation on a larger rodent skull database.

2 Related work

Reeb graphs were made popular in Computer Graphics by Hilaga et al. [5], who present a multi-resolution variant with applications to shape matching and retrieval. They were previously used for segmentation of point clouds [14], where human body scans are divided automatic into 6 parts: head, body and 4 limbs. The segmentation is obtained from connected paths between critical points in a discrete Reeb graph. The approach presented in Section 4 is inspired by the level of detail hierarchy (LOD) as presented by Hilaga et al. [5]. In contrast to a global LOD hierarchy [5], a local hierarchy of nested segmentation candidates is evaluated in our approach, as similar as to dynamic thresholding and the watershed-algorithm [1].

Model-based segmentation approaches are state-of-the-art in Medical Imaging [4] but usually require a large amount of examples to establish a profound model. In order to support model-building, fully automatic [9], manual [13] or semi-automatic segmentation methods similar to the one described in this paper were developed.

Graph cut approaches comprise another large class of image segmentation algorithms. They are used for object extraction, after an initial user-labeling of some foreground and background pixels. For an overview of those approaches we refer to the survey of Boykov and Funka-Lea [2]. Recently, graph cut techniques have also been successfully applied to bone segmentation [9]. However, the here presented methods explicitly optimize the global topological structure of the segmentation and it is unclear how such global topological information could be included in an energy function for optimization via graph cut [8].

In Medical image processing a classification into different materials can be obtained through thresholding the image with specific intensity values [11] according the Hounsfield range of the corresponding material. An overview of thresholding techniques is given by Sahoo et al. [12]. From such classification a segmentation can be obtained by finding the connected regions via a region growing process [10]. Often these segmentations are topologically invalid and contain severe artifacts since the Hounsfield scale is ambiguous and thresholding is not robust to noise.

The most similar work concerning the field of application by Xiao et al. [13] is where rodent skull μ CT datasets are interactively segmented to build a statistical model of shape variability. The interactive segmentation is highly accurate but manually intensive and does not scale to dataset sizes as presented in this work.

3 Topological segmentation

This paper deals with 3D image volumes given as stacks of 2D images as acquired by a μ CT device. Each 2D image represents a slice $S_i: \mathbb{R}^2 \rightarrow \mathbb{R}$ through the scanned object, while the 3D image $I: \mathbb{R}^3 \rightarrow \mathbb{R}$ is given (in discretized form) by their union $I = \bigcup_{i=1}^n S_i$.

Based on the Hounsfield scale, the contents of a CT scan can be classified roughly according to their density into different materials like air, soft tissue, fat and bones. For our application we are interested in bones which have a higher magnitude in terms of Hounsfield units. The simplest “segmentation” procedure via classification, is to apply a global threshold λ_{min} as follows: Let L_i be all voxels in the image slice S_i which have a higher magnitude than λ_{min} , i.e.

$$L_i = \{v \mid S_i(v) \geq \lambda_{min}\} \quad (1)$$

leading to a thresholded image $I_{\lambda_{min}} := \bigcup_{i=1}^n L_i$. Note that thresholding is only a classification of pixels and does not provide a real segmentation into separate bone structures as we desire. Formally, we define a segmentation of the input image $I_{\lambda_{min}}$ as a decomposition into connected regions $T_n \subseteq I_{\lambda_{min}}$ which are disjunct, i.e. $T_k \cap T_j = \emptyset$ and provide a complete partitioning of the thresholded image as $I_{\lambda_{min}} = \bigcup_{k=1}^l T_k$. Further we will denote a group of simply connected voxels of a segment T_n in a specific slice i as a region R .

A classification of pixels like Eq. (1) induces a segmentation into the connected components of the thresholded pixels. In particular the segmentation induced by Eq. (1) provides an under-segmentation for a small enough chosen λ_{min} which serves as initialization and is further refined by the methods presented in the remainder of this paper. Since this initial under-segmentation is computed by (global) thresholding, it exhibits typical thresholding artifacts. Note that because of image artifacts like fuzzy region borders, a segment T_n of the initial segmentation may contain nodes which require a further separation as

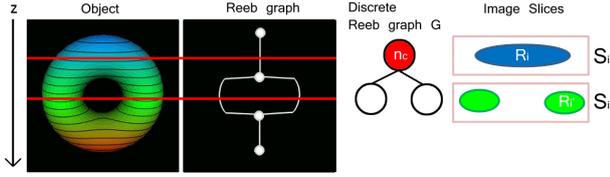


Figure 2: Reeb graph of a torus adopted from [5] and a corresponding discrete Reeb graph.

well as connections between nodes which should not be connected. This is also true for other global initializations [3, 7].

In the presented approach the topological segmentation problem is solved by extracting artifacts and separate bone structures from Eq. (1) via refining the Reeb graph at *critical* nodes, i.e. applying vertex-splitting (in the graph) and corresponding region-segmentation (in the image slice). In the following sections we introduce the Reeb graph construction and present two different strategies for identifying critical nodes, employing different heuristics for automatic and example-based segmentation.

3.1 Reeb graph

In our application we address the segmentation process in terms of graph operations. The structure of a Reeb graph reflects the topological and skeletal structure of an object. It can be defined on an object X via a continuous function $\mu : X \rightarrow \mathbb{R}$. The Reeb graph is the quotient space of the graph of μ in $X \times \mathbb{R}$ [5]. The simplest variant of a μ function, is the function, where the z -coordinate of a voxel $v(x, y, z) \in I_{\lambda_{min}}$ is returned:

$$\mu(v(x, y, z)) = z \quad (2)$$

This choice of μ is naturally related to medical image data which is given as a stack of 2D image slices. It is obvious that the graph for an object changes when μ changes. In Figure 2 an example Reeb graph of a torus is illustrated.

We define a discrete Reeb graph as an undirected graph $G = (N, E)$ where N is the set of nodes representing connected pixel regions on single slices and $E \subset N \times N$ is the set of edges derived from relating adjacent regions across slices. An edge $(n_i, n_{i'})$ is introduced for each pair of regions R_i and $R_{i'}$ on adjacent slices S_i and $S_{i'}$ which are overlapping, i.e. $R_i \cap R_{i'} \neq \emptyset$. Note that E is thereby restricted to pairs of nodes in adjacent slices and there are no edges inside a specific slice. In a Reeb graph critical points are locations where the topology of the object changes. Accordingly a critical node n of the graph G is identified when its in- or out-degree of connections $deg(n)$ is greater than one.

Further we introduce a traversal direction $d^+ = +1$ and $d^- = -1$ in which the graph G is iteratively traversed slice-by-slice. For a specific traversal direction we define $deg^+(n)$ as the number of edges to nodes in the next adjacent slice $i' = i + d^\pm$ and $deg^-(n)$ analogously for nodes

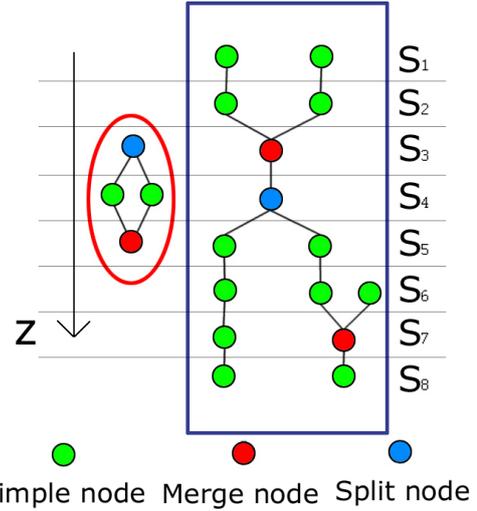


Figure 3: Classification of nodes in a Reeb graph according to a traversal direction z .

in previous slice $i' = i - d^\pm$. According to the traversal direction each node n can be classified in three classes:

1. Simple node: $deg^+(n)$ and $deg^-(n)$ in $\{0, 1\}$
2. Merge node: $deg^-(n) > 1$
3. Split node: $deg^+(n) > 1$

Critical nodes are identified as merge- and split-nodes independent from the traversal direction.

4 Simple automatic segmentation

Starting from an under-segmentation, an object is divided by the connected components of its initial Reeb graph into segments $\{T_n\}$. The segments are refined at critical nodes of their graph by a dynamic threshold process. The largest connected components will contain the structures of interest, while noise and other artifacts can be identified in small connected components of the refined graph.

For the automatic approach only merge nodes are considered as critical nodes, indicating now a change in the topology according to the traversal direction. For the opposite traversal direction, split nodes become merge nodes and traversing the graph in both directions guarantees the investigation of all critical nodes. The automatic segmentation method proceeds slice-by-slice examining all critical nodes and possibly perform vertex-split operations refining the graph. A variant of dynamic thresholding is used to decide whether a critical node can be split within a user specified value range.

The automatic approach starts off with the under-segmentation induced by Eq. (1) and subsequently investigates how the topology changes for all threshold values in the given range $[\lambda_{min}, \lambda_{max}]$. The smallest threshold

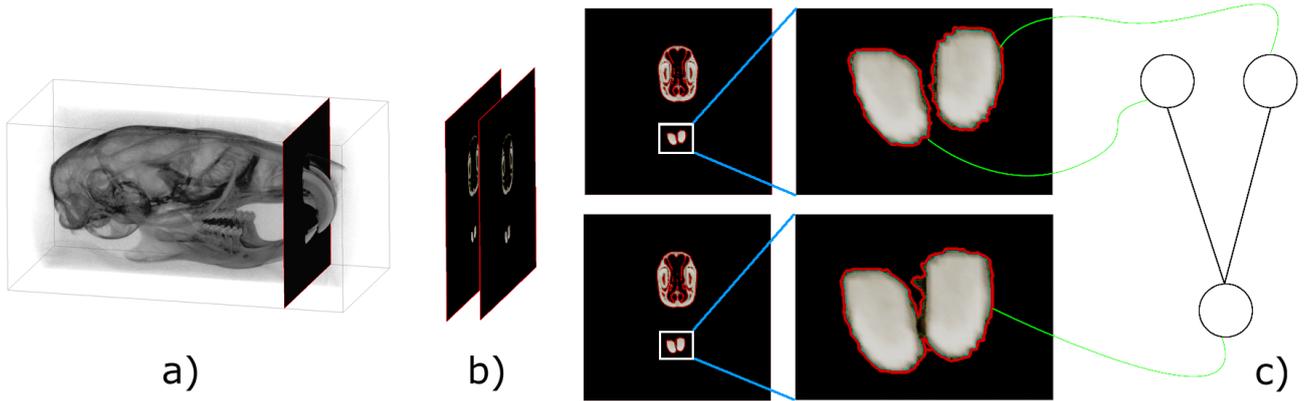


Figure 4: a) 3D volume of a rodent skull, b) Adjacent slices i and i' , side by side view of the slices, red lines indicating the region borders, c) Corresponding nodes and connections.

$\lambda^* > \lambda_{min}$ is applied if the change in topology is considered *plausible* compared to the previous slice. If none such $\lambda^* \leq \lambda_{max}$ is found, the investigated node is considered stable.

The remainder of this section will give the details of the algorithm followed by a short discussion of its parameters and limitations.

4.1 Vertex split and consistent segmentation

In the dynamic thresholding process at a critical node n_c , we assume that the corresponding region R_c is decomposed into k smaller regions $F = \{F_1, \dots, F_k\}$ induced by a certain threshold λ , see Figure 5. In order to decide whether the decomposed regions F provide a consistent topological segmentation, they are compared with the regions of the set of connected nodes N_{ref} on the previous slice i' . We define the change in topology to be *plausible* when none of the decomposed regions $\{F_1, \dots, F_k\}$ is again classified as merge node.

Once a threshold λ^* is found, the regions R_{ref} of N_{ref} are used as an initial mask for region growing to define the final segmentation of n_c . Region growing is started simultaneously from the sets $R_{ref} \cap R_c$ until R_c is completely covered. After the region growing process, we obtain a set of nodes N_{sep} , representing the finally separated regions connected one-to-one to N_{ref} . A vertex-split is performed by replacing the critical node n_c with the separated nodes N_{sep} and the set of affected edges E is updated accordingly.

4.2 The role of λ_{min} and λ_{max}

Note that simply splitting the graph G at each critical node would generate an over-segmentation, while the initial segmentation according to λ_{min} provides an under-segmentation. In order to find a correct segmentation in between those two, an upper bound threshold λ_{max} is required as user input. Usually λ_{min} and λ_{max} can be derived directly from the Hounsfield scale.

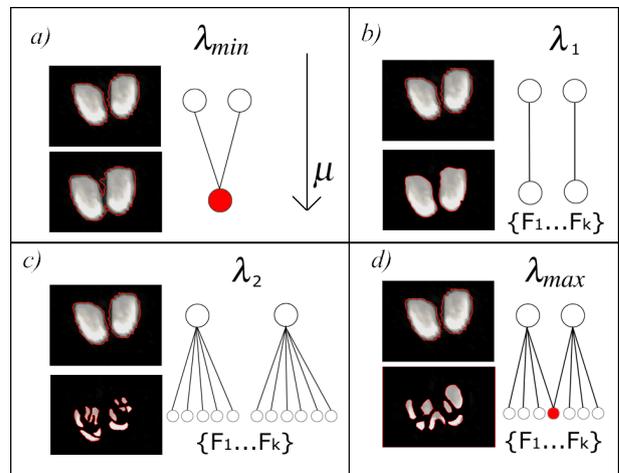


Figure 5: Decomposition of a critical node for different choices of λ as $\lambda_{min} < \lambda_1 < \lambda_2 < \lambda_{max}$. a) Initial region, b) Trivial case of decomposition, c) Over-segmentation but 'plausible', d) Stable node.

The decomposition based on a threshold value λ depends on the scalar values of a critical region and can be classified as illustrated in Figure 5. The trivial case is shown in b), where the decomposition of the region is topologically equivalent to the previous slice. On the other hand, c) illustrates an example decomposition where $\lambda_2 < \lambda_{max}$ and $|F| > |N_{ref}|$. We observed c) to be the typical case in our datasets. Note that in this case performing region growing directly from F as seed regions would result in a false over-segmentation which would in turn corrupt the upcoming classification of nodes in the further slices and finally lead to a false overall result. This shows that performing a consistent segmentation based on a plausible λ^* is vital to acquire correct results for case c) under the assumption that the previous slice was segmented correctly.

4.3 Limitations of the automatic approach

The λ_{max} criterion was introduced to provide a correct segmentation between an over- and an under-segmentation. Connected structures with a high intensity profile can not be separated by a low λ_{max} whereas structures of high intensity experience an over-segmentation by a greater λ_{max} . It has shown to be a non-trivial task determining a good global value for λ_{max} and even for some images no choice of λ_{max} could accomplish a correct segmentation.

A vertex-split is performed at critical nodes depending on the choice of μ Eq. (2) as well as on the traversal direction. Hereby a restriction of orthogonal cuts is induced, since the edges E are introduced for overlapping adjacent regions. Consequently, the choice of the scalar function μ is vital for a successful topological segmentation.

5 Segmentation from single example

In this section we show how the topological segmentation approach can take advantage of already a single example provided by the user. In our application, the example is a manually generated segmentation of a single dataset. Through an alignment method the example segmentation is fitted to their counterparts in each dataset by translation, rotation and scaling. Hereby differences due to position, orientation and size are compensated. This provides a coarse initialization for the segmentation process, which is exploited as described in the following. The particular alignment method will be discussed in the results section.

The presented algorithm requires that the initial coarse segmentation does contain the target structure completely. Although it is allowed to be larger than the target structure, the target structure should be the dominant structure inside, i.e. it should account for most of the non-background voxels and it should not contain whole regions of other structures. In order to guarantee that no voxels of the target structure are missing, a *rough cutout* according to an example segment is obtained by enlarging the example mask e.g. by filtering with a uniform Gaussian kernel of standard deviation σ . We assume that the rough cutout contains then all voxels of the target structure plus perchance some additional voxels from other segments as illustrated in Figure 6e).

In the following, we will investigate only the largest connected component of the Reeb graph corresponding to the dominant structure, while other connected components are assumed to be artifacts produced through the cutout and may belong to other structures. Note that the largest connected component could now be refined with the automatic approach from Section 4, leading in the ideal case to a result as illustrated in Figure 7, resembling a manual segmentation by an expert. But it turned out, that in practice on our dataset of rodent mandibles, the strong shape variation prohibits a correct segmentation in every case. Particularly it showed to be very difficult (and sometimes

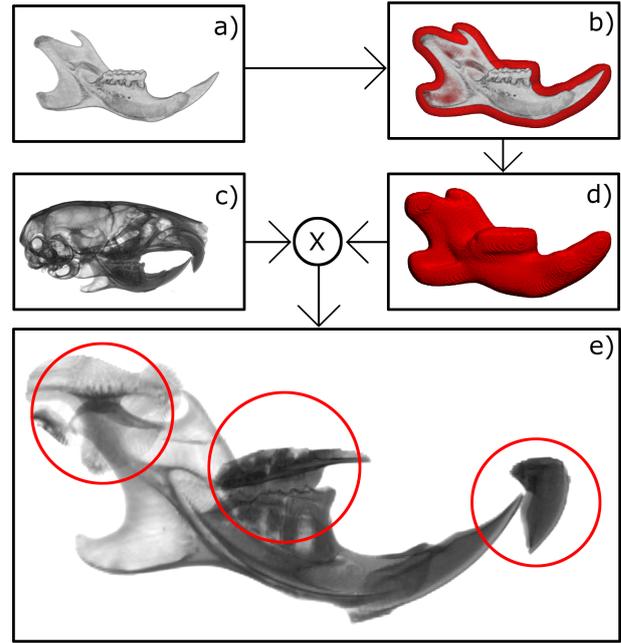


Figure 6: Generating an initial segmentation from an example. a) Aligned example, b) Gaussian filtering, c) Input image $I_{\lambda_{min}}$, d) Binary mask, e) Result of masking where red circles highlight remaining artifacts (see Section 5).

impossible) to choose a good user parameter λ_{max} for the automatic approach. Therefore we will describe now a different method, independent from λ_{max} taking further advantage of the example.

5.1 Example based approach

In the following we consider a separation of an image into two structures. It is obvious that when one structure is found, then the second is induced. For the complete segmentation in this case it is sufficient to create an example for only one structure. Through an alignment method and a rough cutout, a segment is generated whose largest connected component provides a segment T_{invest} for further investigations. The remaining segment is defined as $T_{remain} = I_{\lambda_{min}} \setminus T_{invest}$. We will now describe several operations on the Reeb graph nodes and silently assume the corresponding changes on the edge set E .

In the segment T_{invest} all nodes are labeled as *touching nodes* if they contain voxels which are adjacent to voxels in the other segment T_{remain} . A touching node requires a separation and therefore is also considered as a critical node. The base of this heuristic is that refining the graph in this nodes is sufficient to generate a correct separation of the two structures. Since touching nodes can occur in every position in the graph, first an over-segmentation for the segment T_{invest} is generated. This is done by refining the graph at all critical nodes iteratively slice-by-slice in both traversal directions. Afterwards all nodes except those which are labeled as touching nodes are restored to its pre-

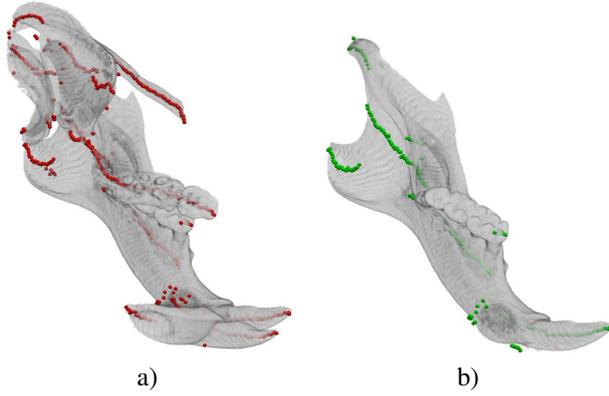


Figure 7: a) Rough cut-out of a left rodent mandible with artifacts, red spheres represent the nodes in the Reeb graph G , b) Result of the automatic segmentation.

vious state yielding the over-segmentation only at touching nodes. After computing the largest connected component, over-segmented regions have to be merged again. This can be done by deleting all regions of nodes which do not belong to the largest connected component. The refined segment T_{invest} induces now also the correct segmentation of the remaining segment.

This approach for two segments can be generalized for more segments in a simple recursive fashion. In the first step the union of all but a single segment T_{remain} form T_{invest} . After applying the above pairwise procedure the correctly segmented T_{remain} is removed from the input set and we repeat until only two segments remain.

5.2 Limitations and heuristics

While evaluating the example-based approach on the rodent skull dataset we encountered some practical restrictions for an axis-aligned choice of the scalar function μ Eq. (2). To this end the following heuristics were developed which can be applied in a post-processing step without changing μ .

Problem 1: The rough cutout may generate regions which contain all voxels of another structure and therefore the corresponding nodes can not be labeled as touching nodes.

Heuristic 1: The graph is traversed slice-by-slice for a specific direction and labeling and splitting process is started. Each node is labeled on the behalf of its predecessor. If a node can be classified as merge node according to the traversal direction and its parents have different labels, this node is separated and the resulting nodes are labeled one-to-one according to the previous node. Such labeling process is shown in Figure 8.

Problem 2: The general limitation of forbidden orthogonal cuts and interlocking areas illustrated in Figure 10 led

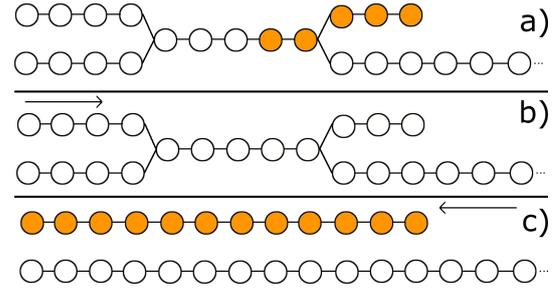


Figure 8: a) Incorrect labeling of touching nodes (orange) in an abstract Reeb graph, b) and c) Labeling and split results for different traversal directions (see Section 5.2).

to the following heuristic.

Heuristic 2: Initially touching nodes are assigned to the structure they are touching. The graph of the investigation segment is traversed in direction towards these nodes. Whenever a region R_{invest} of the investigated segment in slice i has an overlap to a region R_{remain} of the remaining segment in the next slice i' , a dynamic threshold process is started for the region R_{remain} . As described in Section 4, a decomposition of this region is generated for a user-defined upper bound λ_{max} . But here we restrict the decomposition to be correct only when a decomposed region F_r overlaps to the region R_{invest} in slice i and the number of pixels in F_r is less than of R_{invest} . In the case when such F_r is found, it is removed from R_{remain} and assigned to T_{invest} otherwise the process stops. This heuristic is derived from the observation that in the rodent dataset the regions of the tooth become smaller from slice to slice to the peak of the tooth.

5.3 Application to rodent skulls

A manual segmentation was generated only for the left mandible of a single dataset. As a mandible is almost symmetric, the right mandible is generated by mirroring the segmented left mandible. For each example segment a rough cutout and the largest connected component is generated providing the segments T_{left} and T_{right} . In the next step these segments are merged into $T_{mandible}$ which is now the investigated segment. Heuristic 1) with the positive traversal direction is used to obtain a segmentation into $T_{mandible}$ and T_{skull} . For the separation of the mandible in left and right, same heuristic is used but the traversal direction is inverted. The problem of interlocking areas as shown in Figure 6 is solved with Heuristic 2) by assigning the touching nodes at this area to the skull. After the segmentation of the mandible is generated, both graphs of the segments are traversed simultaneously to the direction of touching nodes, which are assigned to the skull (negative traversal direction) and the described technique of Heuristic 2) is performed.

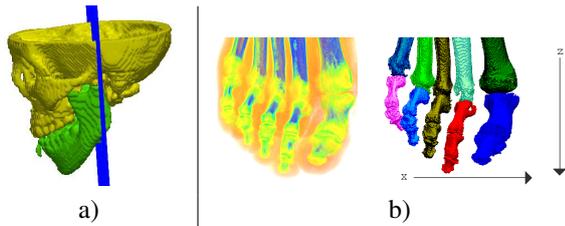


Figure 9: Results of automatic approach: a) Segmentation of a skull in mandible and skull with the slicing direction normal to the blue plane. b) Result for the foot dataset sliced along z . (Datasets courtesy of Kitware and Philips Research respectively.)

6 Results

In this section we will discuss results of the automatic and example based approaches on two test datasets and present an evaluation on a real-world dataset of rodent skull μ CT scans.

6.1 Simple automatic segmentation

Despite its simple nature in design, the automatic segmentation already provides a good segmentation on two tested datasets. Figure 9a) shows the resulting segmentation for a human skull dataset separated into mandible and skull without any prior knowledge besides the Hounsfield range for bone material. In another example we were able to generate a rough segmentation of a human foot dataset, illustrated in Figure 9b). The latter example shows some limitations of the simple automatic approach: Since the μ -Function is chosen axis aligned the approach is only able to find separations along the chosen slicing direction and not orthogonal to it which would be required for some parts of this dataset.

6.2 Example-based segmentation

The example-based approach was applied to separate the skull of 193 μ CT datasets at a resolution of $200 \times 200 \times 400$ into the three parts: skull, left and right mandible. As example we got one manually segmented left mandible which was mirrored to produce a second virtual example for a right mandible. An automatic landmark alignment procedure based on 3D Sift features was employed to find an optimal translation, rotation and scaling of the examples to a particular scan.

The results were evaluated by an expert providing a rate of 64% successful segmentations of very high quality such that the results could directly be used for further applications like model building. In the remaining segmentation results a rate of 8% contained small artifacts as illustrated in Figure 10. For 28% the segmentation was not successful, partly because of bad alignment of the example and partly because the heuristic post-processing routines failed.

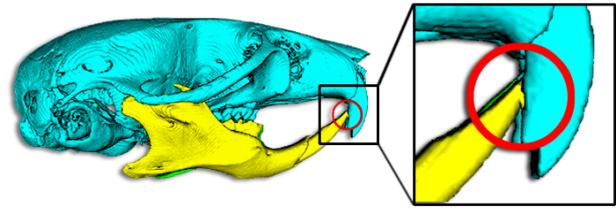


Figure 10: Small artifacts at the interlocking area of the front tooth.

To the best of the authors knowledge this is the first automatic segmentation of a larger rodent skull database. The recent work of Xiao et al. [13] succeeded to segment 16 of similar datasets with a semi-manual procedure which is not scalable to hundreds of scans as can be handled by our approach.

7 Conclusion

We have introduced a topological segmentation method operating on a discrete Reeb graph G . This method enables to formulate segmentation algorithms by graph operations and in particular by performing vertex splits at critical nodes in the graph to refine an intermediate segmentation. Using this technique we have presented two automatic approaches to perform segmentations of topological complex structures. The results are fully automatic decompositions of an input dataset into dominant structures, smaller artifacts, noisy and unstructured parts represented by the connected components of G .

The presented techniques can deal with structures enclosed by a closed boundary contour, also in the case the structures are touching each other or interlocking like for instance at joint locations. Since our method does not rely on a surface model internally it can further deal with structures of complex and irregular topology not accessible with surface models and thus is ideally suited to be used in model building. The segmentation as described in Section 5 can take advantage of a given example. Note that although we used a manual segmentation as input, the input example needn't be a precise segmentation for the algorithm to work. We are confident that a coarse segmentation fulfilling the stated properties can also be derived from a sketch-like user input similar to Xiao et al. [13].

We tried to restrict the algorithms to very sparse user input. The first approach requires only two user parameters $[\lambda_{min}, \lambda_{max}]$ to mediate between an under- and over-segmentation. In the second approach we showed how to take benefit of a single example segmentation allowing the user to implicitly define the target structures for segmentation. Additional post-processing heuristics were developed to overcome some of the limitations encountered during application. This led to a successful automatic segmentation of a large and challenging real-world dataset.

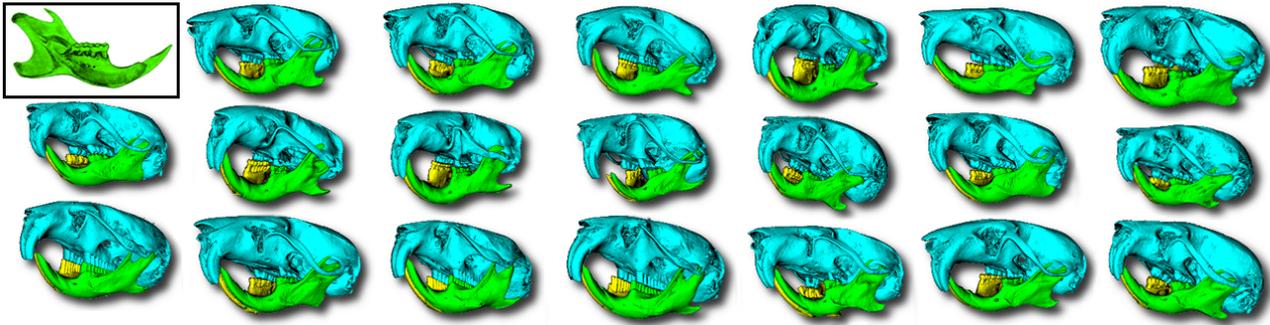


Figure 11: Segmentation results for the rodent skull dataset with a single manually segmented left mandible given as example (upper left). Note the high shape variability and the challenging structures in the area of the incisors, the molar row of teeth and the rear mandible processes which are interlocking with the cranium.

Considering the fact, that the presented Reeb graph approach produced results resembling a manual segmentation by an expert, we believe that there is further potential for topological graph algorithms in image segmentation. Directions for future research include the combination of graph matching and segmentation algorithms by which the image structures could be automatically detected and separated based on prior knowledge. And obvious the choice of problem-dependent scalar functions different from Eq. (2) provides ground for further research.

Acknowledgements

We are grateful to Dr. Anja C. Schunke, MPI Plön, for providing the rodent skull μ CT dataset.

References

- [1] Isaac N. Bankman. *Handbook of Medical Imaging*. Academic Press, Inc., Orlando, FL, USA, 2000.
- [2] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 70:109–131, 2006.
- [3] Yian-Leng Chang and Xiaobo Li. Adaptive image region-growing. *Image Processing, IEEE Transactions on*, 3(6):868 – 872, November 1994.
- [4] Tobias Heimann and Hans-Peter Meinzer. Statistical shape models for 3d medical image segmentation: A review. *Medical Image Analysis*, 13(4):543–563, 2009.
- [5] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *Proceedings of ACM SIGGRAPH 2001*, pages 203 – 212, 2001.
- [6] Godfrey N. Hounsfield. Computed medical imaging. *Medical Physics*, 7(4):283–290, 1980.
- [7] Yan Kang, K. Engelke, and W.A. Kalender. A new accurate and precise 3-d segmentation method for skeletal structures in volumetric ct data. *Medical Imaging, IEEE Transactions on*, 22(5):586 – 598, may 2003.
- [8] V. Kolmogorov and R. Zabini. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):147 –159, feb. 2004.
- [9] M. Krcah, G. Szekely, and R. Blanc. Fully automatic and fast segmentation of the femur bone from 3d-ct images with no shape prior. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, volume 30, pages 2087 – 2090, April 2011.
- [10] Nikos Nikolaidis and Ioannis Pitas. *3-D Image Processing Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 2000.
- [11] Dzung L. Pham, Chenyang Xu, and Jerry L. Prince. A survey of current methods in medical image segmentation. Technical report, Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, 2000.
- [12] P.K Sahoo, S Soltani, and A.K.C Wong. A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing*, 41(2):233 – 260, 1988.
- [13] Mei Xiao, Jung Soh, Oscar Meruvia-Pastor, Eric Schmidt, Benedikt Hallgrímsson, and Christoph Sensen. Building generic anatomical models using virtual model cutting and iterative registration. *BMC Medical Imaging*, 10(1):5, 2010.
- [14] Yijun Xiao, N. Werghi, and P. Siebert. A topological approach for segmenting human body shape. In *Proceedings of Image Analysis and Processing 2003*, pages 82 – 87, September 2003.