Eurographics/ IEEE-VGTC Symposium on Visualization (2006)
Thomas Ertl, Ken Joy, and Beatriz Santos (Editors)

Automating Transfer Function Design for Volume Rendering Using Hierarchical Clustering of Material Boundaries

Petr Šereda[1], Anna Vilanova[1] and Frans A. Gerritsen [12]

[1]Department of Biomedical Engineering, Eindhoven University of Technology
[2]Research and Advanced Development, PHILIPS Medical Systems Nederland BV

Abstract

Transfer function design plays a crucial role in direct volume rendering. Furthermore, it has a major influence on the efficiency of the visualization process. We have developed a framework that facilitates the semi-automatic design of transfer functions. Similarly to other approaches we generate clusters in the transfer function domain. We created a real-time interaction with a hierarchy of clusters. This interaction effectively substitutes cumbersome settings of clustering thresholds. Our framework is also able to easily combine different clustering criteria.
We have developed two similarity measures for clustering of material boundaries. One is based on the similarity of the boundaries in the transfer function domain and the other on their spatial relation. We use the LH space as the transfer function domain. This space facilitates the clustering of material boundaries. We demonstrate our approach on several examples.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation I.3.6 [Computer Graphics]: Methodology and Techniques I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism I.5.3 [Pattern Recognition]: Clustering

1. Introduction

Direct volume rendering is a powerful and flexible visualization technique. It has the capability to reveal properties and shapes of volumetric objects, as well as their spatial relation. In order to efficiently use this capability, one needs to define a suitable transfer function (TF). In the medical field the increasing size of the datasets asks for 3D visualization as an alternative to the traditional 2D slice-by-slice viewing. However, the cumbersome and nonintuitive design of the TF is an important factor that inhibits its practical use. Especially the need to understand the TF domain and the unpredictability of the resulting rendering are strongly discouraging.

Medical workstations usually tackle the problem of the intuitive TF design by having several preset transfer functions. The user, similarly to the approach of the design galleries [MAB*97], picks the best preview. This approach is, however, not very flexible. If there is no appropriate preset, the TF has to be again tuned manually. There have been several approaches that automate and facilitate the TF design using clustering techniques [HM03,TM04,RBS05]. Interaction with the clusters seems to be more intuitive than a direct interaction with the TF.

The overall performance of the clustering techniques is highly dependent on the separability and compactness of the clusters in the feature space. For the visualization of material boundaries, the space of combination of the scalar value and the gradient magnitude is commonly used. However, in that space the boundaries appear as arches that frequently overlap causing classification ambiguities. Furthermore, it is rather difficult to take into account the shape of such clusters. For clustering, it is a great advantage if the clusters have a more simple shape. It has been shown in previous research [SVSG06] that in the, so-called, LH space the boundaries between materials have substantially less overlap compared to the space of arches. In addition, boundaries appear as blobs instead of arches. This space shows more suitable characteristics for developing a clustering technique.

The usual drawback of clustering techniques is their sensitivity to clustering criteria or parameters. Tuning of such criteria is often a difficult task. Moreover, in order to combine several criteria, one needs to choose proper weighting which is even more difficult. In this paper, we show how hierarchical clustering methods can facilitate the design of transfer functions. Our first contribution is a framework in which the user interacts with a hierarchy of clusters. No predefined clustering threshold is needed. The user interacts with the hierarchy. At any time the user can pick another simi-

larity measure (clusterig criteria) and the hierarchy is automatically adjusted. Our framework enables an easy combination of similarity measures. We define the clusters in the LH space that serves as the TF domain. The second contribution of this paper are two intuitive similarity measures for the clusters. One evaluates the similarity of the boundaries in the LH space and the second evaluates their spatial relation in the volume. We include both similarity measures into the framework.

2. Related Work

There has been an extensive research aiming to facilitate the TF design. For an automatic selection of the important iso-contours Bajaj et al. [BPS97] and Pekar et al. [PWH01] used the contour spectra and Fujishiro et al. [FAT99] investigated the behavior of the contours. He et al. [HHKP96] developed a method that automatically evaluates the quality of rendered images and adapts the TF. The design galleries by Marks et al. [MAB*97] and König and Gröller [KG01] let the user evaluate the image quality and pick the best rendering. Tzeng et al. [TLM05] used a position dependent high-dimensional classification to overcome problems of traditional transfer functions. Lundstrom et al. [LLY05] introduced transfer functions based on automatic detection of tissues. They added an extra TF dimension based on extensive neighborhood analysis in order to separate between overlapping tissues. Kniss et al. [KUS*05] used classification probability in order to automatically visualize the uncertainty of tissue membership.

Visualizing boundaries of objects is crucial for the perception of their shape. Kindlmann and Durkin [KD98] showed that by combining the scalar value and the gradient magnitude one may distinguish between boundaries. In this 2D space the boundaries appear as arches. Kniss et al. [KKH01, KKH02] used this domain for their 2D transfer functions. Although the arches enable some distinction between boundaries, they often overlap causing selection ambiguities. Lum et al. [LM04] used two additional samples on the boundary in order to improve on that. Šereda et al. [SVSG06] went further and traced the boundary profiles in every voxel to determine the start and end of the arches. These two intensity values are the coordinates of the voxel in the so-called LH space. It was shown that in the LH space the boundaries appear as blobs and are easier to separate then the arches.

Several clustering techniques have been developed in order to shield the user from the TF design in the space of scalar value and gradient magnitude. Huang et al. [HM03] generated automatic selections with the help of region growing in the volume. Tzeng et al. [TM04] used a clustering technique based on the K-means algorithm. They designed an interface in which the user can interact with the clusters. Roettger et al. [RBS05] used a clustering method that groups two bins of the histogram if the corresponding tuples are similarly distributed in the volume. These methods require parameters that determine the number or size of the clusters or a threshold for the clustering method. In this paper, we describe a hierarchical clustering method that replaces such parameters with an interactive selection of the clustering level using an intuitive interaction with the hierarchy. Our clusters are defined in the LH space where boundaries have far less overlaps than in the space of scalar value and gradient magnitude.

3. Hierarchical Clustering

Having an initial set of n elements $e_1,...,e_n$, the agglomerative hierarchical clustering [DHS01] describes the order in which the elements join into clusters. In the initialization step each element e_i is in its own cluster C_i. The hierarchy has n levels $k = 1,...,n$. In level $k = 1$ there are n clusters, i.e., no elements are joined. In every following level the two most similar clusters join. The maximal similarity s at level k between clusters is found as

$$s_k = max\{s(C_i,Cj)\}; i, j = 1,...,n-k+1; i \neq j$$

Finally, in level $k = n$ only one cluster remains. The resulting hierarchy can be intuitively represented by a binary tree (supposing no two pairs of clusters have the same similarity) called dendrogram. Figure 1 shows an example of a dendrogram.

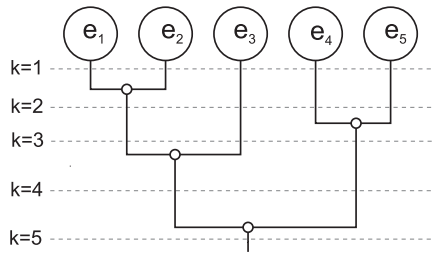

Figure 1: *Example of hierarchical clustering of five initial clusters visualized using a dendrogram.*

In the following text we describe our approach to find the initial set of elements $e_1,...,e_n$. Further we describe two different similarity measures developed in order to evaluate the similarity between each pair of clusters.

4. Similarity Measures

We have developed two different similarity measures that intuitively evaluate the correspondence between clusters. The first measure is designed to group similar boundaries, i.e. boundaries that lie close to each other in the LH space [SVSG06]. The second measure evaluates the spatial connectivity of clusters. In the following text, we first describe how the initial elements e_i are obtained. Then we describe the similarity measures used to generate the hierarchy.

4.1. Initial Clustering

It has been shown in previous research [SVSG06] that the boundaries between tissues form blob-like clusters in the LH space. We use this property to define meaningful initial clustering elements e_i. In the LH histogram, every boundary can be localized as a local peak. Therefore, we label every local maximum of the 2D histogram, see Figure 2 for an 1D illustration. All bins of the histogram that belong to the same peak then form one of the initial clustering elements, e_i.

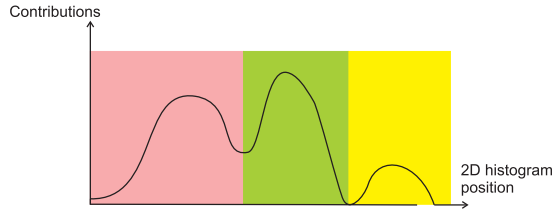

Figure 2: *A 1D illustration of the initial labelling of the 2D peaks in the LH histogram. Every peak becomes one initial cluster.*

Having too many initial clusters is not beneficial for the performance of the clustering technique. The complexity of generating the hierarchy is $O(\frac{n^3}{2})$ where n in the number of initial clustering elements e_i. Depending on the dataset and the resolution of the histogram (in this paper we used 256^2 bins), in some cases there might be several thousands of local peaks. Large number of them would typically represent only very small initial clusters consisting of few voxels. From our observation, reducing the number of initial clusters to several hundreds, allows an interactive generation of the hierarchy.

In order to get rid of very small clusters that contain only few voxels, we perform the following two steps. Firstly, before the local peak detection, the histogram is blurred using a 2D Gaussian kernel of a small σ equal to the bin size. This both removes small noisy local maxima and establishes a direct neighborhood relation between peaks that were separated. Secondly, after the initial clusters have been generated, those consisting of only few voxels (in our method less than 10, note that this is not a critical threshold, other values could also have been used) are joined with their direct neighbors. We can reason that such small clusters have little effect on the visualization and therefore assigning them to their neighbors will not introduce any visible artifacts.

4.2. Similarity in the LH Space

This similarity measure is designed to enable grouping of similar boundaries. In order to achieve that, we use the information given by the positions sizes and shapes of initial elements e_i in the LH space.

In order to evaluate the similarity of two elements, we want to take into account the following criteria:

- *Distance.* Close elements have similar L and H values (see Figure 4a), i.e., they correspond to boundaries between similar tissues.
- *Separation.* A deep valley between two peaks yields a good separation. If there is little evidence of boundary profiles that project in between the two peaks, it is likely that the peaks correspond to different boundaries.
- *Direction of elongation.* Due to changes of the tissue intensities that form the boundaries, the corresponding peaks may be horizontally and/or vertically elongated. Therefore, we preferably want to join the elements in the direction indicated by their elongation.

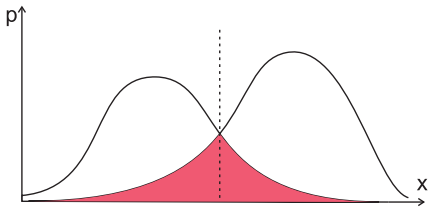

Figure 3: *Two probability distributions. The dashed line is the decision boundary made by a Bayesian classifier. The red area represents the probability of a wrong decision. We use an estimate of the overlap of two 2D distributions as our similarity measure.*

All these criteria can be elegantly combined by applying known techniques from the Bayesian decision theory [DHS01]. If we look at the clustering elements as bivariate (2D) probability density functions (PDF), we can define the similarity between two elements as the overlap of their PDFs. An exact computation of this overlap is not trivial. However, assuming the elements can be approximated by bivariate normal density functions, we can estimate the upper bound of the overlap by using the so-called Bhattacharyya bound. It is the upper estimate of the probability of a wrong decision made by a Bayesian classifier given the PDFs and the priors (Figure 3).

We define the size of an element, as the number of voxels that belong to the element, i.e.:

$$|e_i| = \sum_{b_{xy} \in e_i} |b_{xy}|$$

where b_{xy} is the bin at discrete histogram position (x,y) and $|b_{xy}|$ the number of contributions in the bin. Then, the priors $P(e_i)$ and $P(e_j)$ can be obtained as

$$P(e_i) = \frac{|e_i|}{|e_i|+|e_j|}; \quad P(e_j) = \frac{|e_j|}{|e_i|+|e_j|}$$

The similarity of two elements e_i and e_j is computed using the Bhattacharyya bound

$$s(e_i,e_j) = 2\sqrt{P(e_i)P(e_j)}\,e^{-k}$$

where 2 scales the values into the range $< 0,1 >$ and

$$k = \frac{1}{8}(\mu_j - \mu_i)^t \left[\frac{\Sigma_i + \Sigma_j}{2}\right]^{-1} (\mu_j - \mu_i) + \frac{1}{2} ln \frac{\left|\frac{\Sigma_i + \Sigma_j}{2}\right|}{\sqrt{|\Sigma_i||\Sigma_j|}}$$

where the mean vector μ_i and covariance matrix Σ_i describe the bivariate normal distribution of element e_i. The mean vector of element e_i

$$\mu_i = (\mu_{ix}, \mu_{iy})^t = \left(\frac{1}{|e_i|} \sum_{b_{xy} \in e_i} (x|b_{xy}|), \frac{1}{|e_i|} \sum_{b_{xy} \in e_i} (y|b_{xy}|) \right)^t$$

and the covariance matrix $\Sigma_i = \begin{pmatrix} \sigma_{ixx} & \sigma_{iyx} \\ \sigma_{ixy} & \sigma_{iyy} \end{pmatrix}$

In order to avoid Σ_i to be singular due to clustering elements that are only 1 bin wide, we split every bin into 4

$$b_{xy} \rightarrow \{b_{x\pm\frac{1}{4}\, y\pm\frac{1}{4}}\}; \quad |b_{x\pm\frac{1}{4}\, y\pm\frac{1}{4}}| = \frac{|b_{xy}|}{4}$$

which is equivalent to adding small regularization terms to the diagonal elements of Σ_i, i.e.:

$$\sigma_{ixx} = \frac{1}{|e_i|} \sum_{b_{xy} \in e_i} \left(|b_{xy}|(x - \mu_{ix})^2 \right) + \frac{1}{16}$$

$$\sigma_{iyy} = \frac{1}{|e_i|} \sum_{b_{xy} \in e_i} \left(|b_{xy}|(y - \mu_{iy})^2 \right) + \frac{1}{16}$$

$$\sigma_{ixy} = \sigma_{iyx} = \frac{1}{|e_i|} \sum_{b_{xy} \in e_i} \left(|b_{xy}|(x - \mu_{ix})(y - \mu_{iy}) \right)$$

We compute the similarity of two clusters C_i, C_j by using the single-link (nearest neighbor) method, i.e.

$$s(C_i, C_j) = max\{s(e_i, e_j)\}; \quad e_i \in C_i, e_j \in C_j$$

The joined cluster $C_{i \cup j}$ contains all elements from both clusters.

The presented similarity measure is a semi-metric, since it is positive, symmetric and reflexive but the triangle inequality does not hold. For our method these properties are sufficient. The triangle inequality is not required since we do not perform arithmetic operations on the similarities, we only compare their values.

4.3. Similarity in the Volume

This second similarity measure is designed to group boundaries that are connected in the volume. It is an alternative to the similarity in the LH space. Figure 4 illustrates the use of the spatial similarity on a phantom consisting of four materials. Figures 4d-4f were generated using the similarity in the LH space. In 4f boundaries 1 and 4 are grouped since they are close in the LH space. Combining the LH similarity with the volume similarity helped to group boundary 1 with 2 and 3 with 4 (see Figure 4g). We evaluate the spatial similarity as the number of direct neighborhood relations between clusters. Having the initial clustering elements, we evaluate the number of neighborhood relations for each pair of elements $NR(e_i, e_j)$. We look at all 26 neighbors of every voxel $v_i \in e_i$ and count how many of them belong to element e_j, i.e.:

$$NR(e_i, e_j) = NR(e_j, e_i) = \sum_{v_i \in e_i} \sum_{v_j \in e_j} N_{26}(v_i, v_j); e_i \neq e_j$$

where $N_{26}(v_i, v_j)$ is 1 if v_i and v_j are neighbors and 0 otherwise. Since cluster cannot neighbor with itself we define $NR(e_i, e_i) = 0$. The total number of neighborhood relations of the element is

$$NR(e_i) = \sum_{e_j} NR(e_i, ej)$$

In order to group clusters that belong to the same boundary, we weight the relations between voxels. Then the sum of weighted relations between two elements

$$R(e_i, e_j) = R(e_j, e_i) = \sum_{v_i \in e_i} \sum_{v_j \in e_j} N_{26}(v_i, v_j) r(v_i, v_j); e_i \neq e_j$$

where $r(v_i, v_j) \in < 0,1 >$ reflects the directional coherence of boundaries in voxel v_i and v_j (as used in Šereda et al. [SVSG06]).

Finally, to make the similarity measure independent on the size of the cluster, we normalize $R(e_i, e_j)$ by $NR(e_i)$ and $R(e_j, e_i)$ by $NR(e_j)$. Since generally $NR(e_i) \neq NR(e_j)$ we choose the maximum of the two relations to make the similarity measure symmetric

$$s(e_i, e_j) = s(e_j, e_i) = max \left\{ \frac{R(e_i, e_j)}{NR(e_i)}, \frac{R(e_j, e_i)}{NR(e_j)} \right\}$$

Note that $NR(e_i)$ cannot be zero as long as there are at least two clusters in the data. Further we define $s(e_i, e_i) = 1$ in order to obtain a similarity measure that is semi-metric.

The presented relationships between initial clustering elements are evaluated in one pass through the volume. The initial clusters C_i contain each one element e_i. Therefore $NR(C_i) = NR(e_i)$, $NR(C_i, C_j) = NR(e_i, e_j)$, $R(C_i, C_j) = R(e_i, e_j)$ and $s(C_i, C_j) = s(e_i, e_j)$.

When two clusters C_i and C_j join, we have to recalculate the number of relations of the joined cluster $C_{i \cup j}$

$$NR(C_{i \cup j}) = NR(C_i) + NR(C_j) - 2NR(C_i, Cj)$$

and update its relations with all other clusters C_k; $C_k \neq C_i$, $C_k \neq C_j$

$$NR(C_{i \cup j}, C_k) = NR(C_i, C_k) + NR(C_j, Ck)$$

$$R(C_{i \cup j}, C_k) = R(C_i, C_k) + R(C_j, Ck)$$

$$s(C_{i \cup j}, C_k) = max \left\{ \frac{R(C_{i \cup j}, C_k)}{NR(C_{i \cup j})}, \frac{R(C_{i \cup j}, C_k)}{NR(C_k)} \right\}$$

For both similarity measures the complexity of joining two clusters is $O(K)$, where K is the number of clusters C_k, i.e., clusters remaining in the hierarchy.

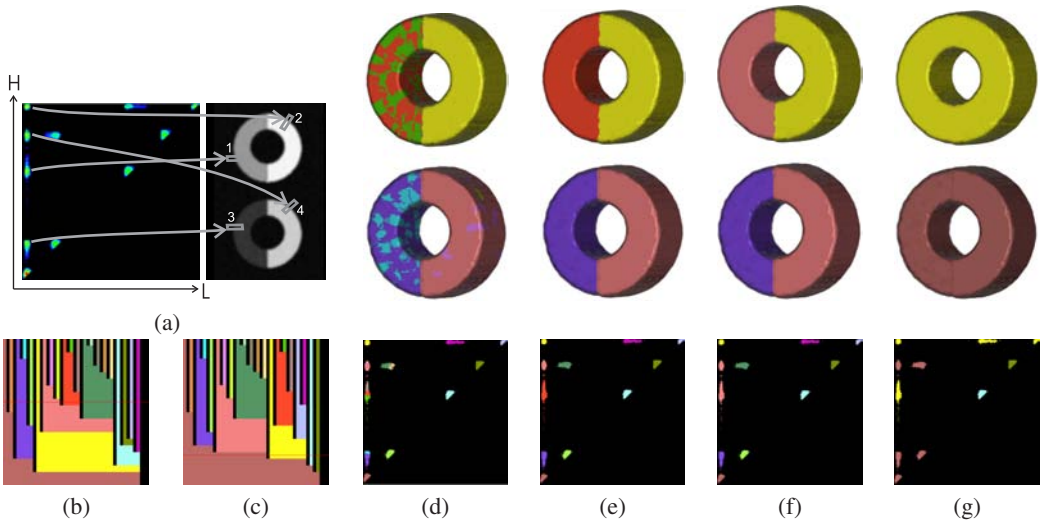

Figure 4: *Phantom data set. (a) shows a slice and the correspondence of four most important boundaries in the LH histogram. (d) shows the initial clustering. In (e) and (f) the hierarchy based on the LH similarity was used (see dendrogram (b)). The level chosen in (e) yields each of the four boundaries in own cluster. In (f) boundaries 1 and 4 are grouped since they are close in the LH space. A hierarchy that combines both similarity measures (c) results in desired clustering (g). The dendrograms (b, c) show the order of grouping. The right child inherits the color from the left child.*

5. Hierarchy Interaction Framework

A major disadvantage of clustering techniques is the need to define a similarity threshold at which the desired partitioning of the elements should exist. Tuning such a threshold by changing its value and restarting the clustering algorithm is rather cumbersome process. Furthermore, this would be assuming that at one level all desired object can be perfectly partitioned. Our framework enables a real-time interaction with the cluster hierarchy. This approach is much more intuitive, since the user does not have to deal with threshold values and gets a better insight into the data set. Moreover, in our method the objects can be selected at different hierarchy levels.

The second weakness of clustering techniques lies in the similarity measure. It is often the case that more criteria need to be combined to cover the crucial aspects of the data. However, the criteria are usually of different nature and their simple combination (such as the most often used weighted sum) have little meaning. In practice these weights are usually tuned for certain data set(s) and their performance for other data is questionable. Again, tuning of these weights is difficult, since different data sets may have distinct sensitivity to certain criteria. In our framework, no weights need to be defined, since only one similarity measure is used at a time. However, the user can pick different similarity measure while moving in the hierarchy. This means that different parts of the hierarchy can be generated by different similarity measures.

We illustrate the use of the framework in the LH space using the two similarity measures described in sections 4.2 and 4.3. However, this is just a concrete implementation of the framework. Using other space and similarity measures or their combination would be possible.

The input to the interaction process (see Figure 5) are the initial clustering elements and a default similarity **s**. After the hierarchy is generated, the user can interactively change the level **k** which results in a new partition of the elements. If moving to the next hierarchy level does not yield the expected clustering, the user might want to change the similarity criteria **s**. In that case a new hierarchy is generated on the

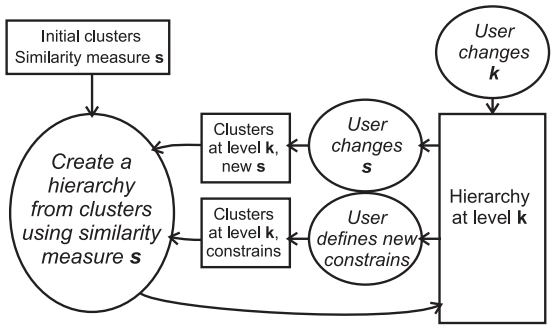

Figure 5: *The hierarchy interaction framework. First, the hierarchy is generated for the initial clusters using a default similarity **s**. Then the user can interact with the hierarchy by changing the level **k** and (de)selecting clusters. The user can also change the similarity measure or apply constrains to selected clusters.*

clusters from level **k** using another similarity measure (see Figure 6).

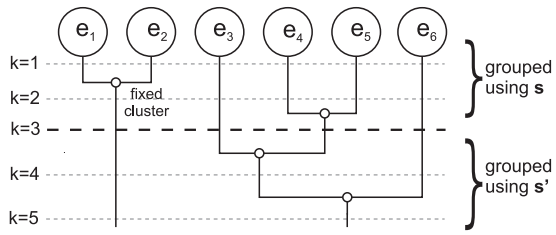

Figure 6: *A hierarchy cut at level* $\mathbf{k} = 3$ *yields clusters that are further grouped using another similarity measure. The hierarchy can be, therefore, a combination of several similarity measures. The cluster consisting of elements* e_1 *and* e_2 *was fixed on level* 2 *and taken out of the hierarchy.*

In general, there might be no single hierarchy level at which the clusters are grouped according to the user's expectations. In order to handle such situations, several constrains can be introduced into the hierarchy:

- *Fixing a cluster.* If a cluster is fixed, it is taken out of the hierarchy so that it is not split or joined with other clusters when the hierarchy level changes. This can be used for situations where certain objects are well selected at different levels.
- *Explicit splitting(joining) of selected cluster(s).* Selected clusters are joined or split and the hierarchy is adjusted accordingly.

For more complex datasets, the user might find it difficult to observe and evaluate the completeness off all objects of interest at the same time. Since it is much easier to interact only with one object at a time, we enable the possibility to visualize only one branch of the hierarchy. The users can then concentrate on one object of interest and after fixing the object at certain level, they can focus on another object.

The hierarchy interaction allows a large interaction freedom based on intuitive actions. However, if an a-priori knowledge is available, it is possible to automatically include certain constrains of the hierarchy without compromising the generality of the framework.

6. Transfer Functions From Clustering

The result of a certain clustering is labelling of the LH histogram. Since the color and opacity are defined for every label, we obtain a 2D transfer function. By default, the clusters are automatically assigned random colors and full opacities. When two clusters join they should appear in the same way, i.e. have the same color and opacity. In our implementation the smaller cluster inherits the color and opacity from the bigger (dominant) cluster. Naturally, the user is free to change colors and opacities of any cluster.

Since our rendering is implemented using the VolumePro1000 board [PHK*99] that allows only 1D TFs, we label the volume and perform a 1D TF on the labels. The labelling, however, needs to be done only once for the initial clusters, therefore the speed of interaction with the hierarchy is not compromised. One of the advantages of using the labels is the possibility to combine the TF with a segmentation (as shown in the previous work [SVSG06]). The disadvantage is that techniques such as fuzzy borders between clusters are not possible. However, the fuzzy borders could be implemented, e.g., in a GPU-based renderer.

7. Results

We demonstrate the functionality of our methods on four different datasets. The first two are the CT data sets of the engine and the carp. The clusters in the LH histogram and corresponding renderings are shown in Figure 7. These data sets contain only few important boundaries. We have only used the LH-based similarity. Visualizing these datasets was fairly easy. All major boundaries can be seen at the same time. Therefore, one can easily see if the cluster should grow more or be fixed.

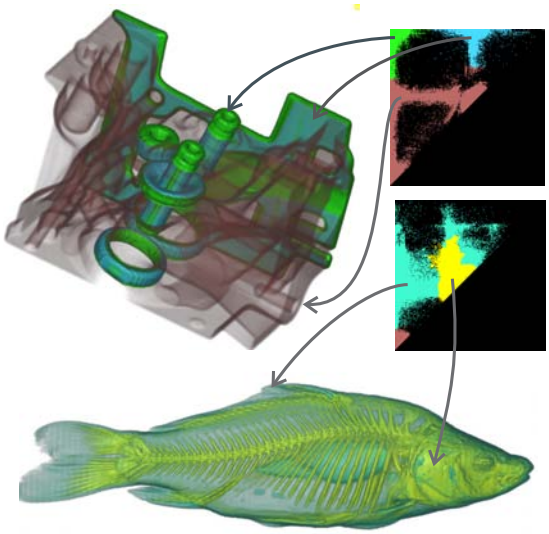

Figure 7: *Renderings of the carp (256x256x512) and engine (256x256x110) data sets. The green and blue boundaries of the engine were visualized at a certain hierarchy level. After fixing them at that level a further change of the level grouped the brown boundary. For the carp we first fixed the bone boundaries at the level where all of them were grouped and then continued joining the remaining air boundaries.*

In the the tooth data set (Figure 8) most of the boundaries could be completely visualized at one hierarchy level. In order to completely select the enamel-air boundary, we had to first fix the dentin-air boundary. In Figure 8d we switched

to the spatial-based similarity in order to group the enamel-air and dentin-air boundaries. In this dataset we had to make certain boundaries (semi) transparent in order to evaluate the quality of grouping in the underlying boundaries.

In Figure 9, we show several combinations of important boundaries in the CTA data set of a head. All the shown boundaries could be selected at the same level in the hierarchy. In Figure 9d we further increased the hierarchy level in order to select and easily remove the remaining boundaries inside the skull.

Note that in the presented results we only used moving in the hierarchy and one constrain, i.e. fixing certain clusters after they were well selected. We did not use other constrains such as the explicit joining/splitting of the clusters. Because of the fast hierarchy interaction, it was easy to select the proper hierarchy level or to realize that no such level exists. If no such level existed, the constrains had to be introduced. For the tested data sets we obtained maximum around 200 initial clustering elements. With such a low number, the hierarchy and its modifications could be generated in real-time without any noticeable delays.

8. Conclusions and Future Work

In this paper we have presented a flexible framework that enables an intuitive interaction with the hierarchy of clusters. The clusters serve as the transfer function for the direct volume rendering. The user works with objects and therefore does not need to understand the underlying transfer function domain. Moreover interacting with the hierarchy replaces the commonly used parameters such as clustering thresholds and weighting constants for the similarity measures. Our framework enables an easy combination of different similarity measures.

We have presented a clustering method based on the LH space. This space seems more suitable for clustering of material boundaries than the commonly used space of scalar value and gradient magnitude. We have introduced two measures that evaluate similarity of clusters in the LH space and in the volume space.

However, the current method is sensitive to the initial clustering that is given by the peaks in the LH histogram. If there are more boundaries in one initial clustering element, in our implementation they cannot separated. Interesting would be to split that element and refine the selection down to the level of single bins. Such operation would fully fit into the framework, since it would only mean replacing the element in the hierarchy by a sub-tree that represents its division.

The presented implementation of the framework used the LH space as the domain for the TF. However, the framework is general and could be applied on any TF domain.

In our current implementation the user interacts with objects by pointing at corresponding clusters in the TF domain. In the future we would like to introduce fully 3D interaction with the rendering. That would further improve the intuitiveness of the interaction since the user would be completely shielded from the transfer function.

Acknowledgment

The work presented in this publication has been funded by Philips Medical Systems in Best, The Netherlands.

References

[BPS97] BAJAJ C. L., PASCUCCI V., SCHIKORE D.: The contour spectrum. *Proceedings IEEE Visualization* (1997), 167–174.

[DHS01] DUDA R. O., HART P. E., STORK D. G.: Pattern classification. *2nd edition. Wiley. New York* (2001).

[FAT99] FUJISHIRO I., AZUMA T., TAKESHIMA Y.: Automating transfer function design for comprehensible volume rendering based on 3D field topology analysis. *Proceedings IEEE Visualization* (1999), 467–470.

[HHKP96] HE T., HONG L., KAUFMAN A., PFISTER H.: Generation of transfer functions with stochastic search techniques. *Proceedings IEEE Visualization* (1996), 227–234.

[HM03] HUANG R., MA K. L.: RGVis: Region growing based visualization techniques for volume visualization. *Proceedings Pacific Graphics Conference* (2003), 355–363.

[KD98] KINDLMANN G., DURKIN J. W.: Semi-automatic generation of transfer functions for direct volume rendering. *Proceedings IEEE Symposium on Volume Visualization* (1998), 79–86.

[KG01] KÖNIG A., GRÖLLER E.: Mastering transfer function specification by using VolumePro technology. *Proceedings Spring Conference on Computer Graphics 17* (2001), 279–286.

[KKH01] KNISS J., KINDLMANN G., HANSEN C.: Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. *Proceedings IEEE Visualization* (2001), 255–262.

[KKH02] KNISS J., KINDLMANN G., HANSEN C.: Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics 8*, 3 (2002), 270–285.

[KUS*05] KNISS J. M., UITERT R. V., STEVENS A., LI G. S., TASDIZEN T., HANSEN C.: Statistically quantitative volume rendering. *Proceedings IEEE Visualization* (2005), 287–294.

[LLY05] LUNDSTROM C., LJUNG P., YNNERMAN A.: Extending and symplifying transfer function design in medical volume rendering using local histograms. *Proceedings IEEE/EuroGraphics Symposium on Visualization* (2005), 263–270.

[LM04] LUM E. B., MA K. L.: Lighting transfer functions using gradient aligned sampling. *Proceedings IEEE Visualization* (2004), 289–296.

[MAB*97] MARKS J., ANDALMAN B., BEARDSLEY P. A., FREEMAN W., GIBSON S., HODGINS J., KANG T., MIRTICH B., PFISTER H., RUML W., RYALL K., SEIMS J., SHIEBER S.: Design galleries: a general approach to setting parameters for computer graphics and animation. *Proceedings SIGGRAPH Conference* (1997), 389–400.

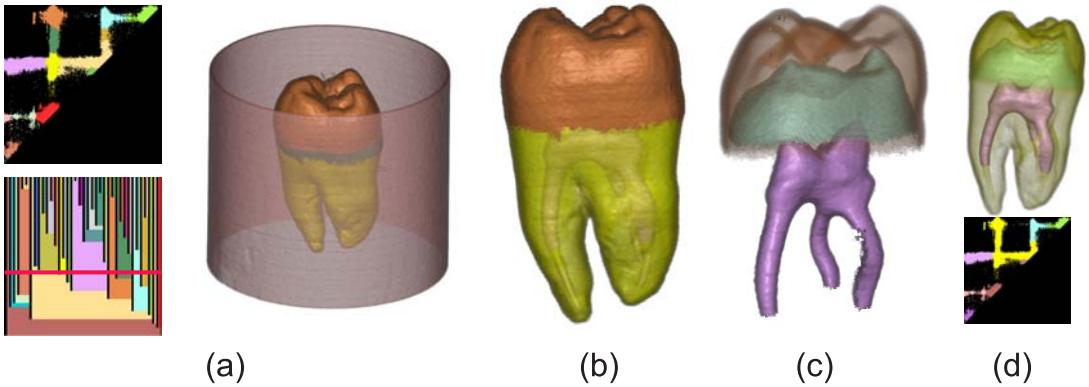

Figure 8: *Tooth data set (256x256x161). In (a) the major boundaries were selected by an easy choice of the hierarchy level (see the red line in the dendrogram). The LH histogram shows corresponding selection. Then the surrounding (pink) boundary was removed (b) and after fixing the dentin-air (yellow) boundary the rest of the enamel-air (orange) boundary was selected. In (c) the inside boundaries are shown after making the outside boundaries to (semi)transparent. In (d) we illustrate the possibility of grouping the outer boundaries (yellow). We achieved that by choosing the spatial-based similarity measure.*

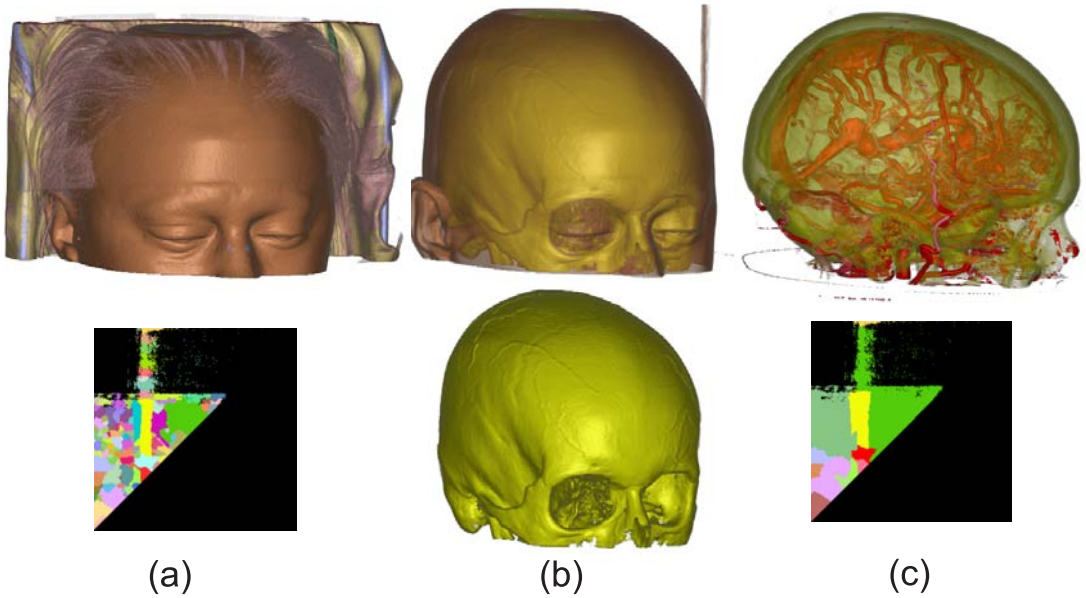

Figure 9: *CTA data set of a head (512x512x286). One level in the hierarchy and corresponding rendering is shown in (a). In (b) only two of the clusters were chosen: the skin and the skull. In (c) the skull and the vessels were selected, fixed and the color of the vessels was changed. Then by changing the level all other clusters inside the head (green) were grouped and removed from the visualization.*

[PHK*99] PFISTER H., HARDENBERGH J., KNITTEL J., LAUER H., SEILER L.: The VolumePro real-time ray-casting system. *Proceedings 26th annual conference on Computer graphics and interactive techniques* (1999), 251–260.

[PWH01] PEKAR V., WIEMKER R., HEMPEL D.: Fast detection of meaningful isosurfaces for volume data visualization. *Proceedings IEEE Visualization* (2001), 223–230.

[RBS05] ROETTGER S., BAUER M., STAMMINGER M.: Spatialized transfer functions. *Proceedings IEEE/EuroGraphics Symposium on Visualization* (2005), 271–278.

[SVSG06] SEREDA P., VILANOVA A., SERLIE I. W. O., GERRITSEN F. A.: Visualization of boundaries in volumetric datasets using LH histograms. *IEEE Transactions on Visualization and Computer Graphics 12*, 2 (2006), 208–218.

[TLM05] TZENG F. Y., LUM E. B., MA K. L.: An intelligent system approach to higher-dimensional classification of volume data. *IEEE Transactions on Visualization and Computer Graphics 11*, 3 (2005), 273–284.

[TM04] TZENG F. Y., MA K. L.: A cluster-space visual interface for arbitrary dimensional classification of volume data. *Proceedings Eurographics/IEEE TCVG Visualization Symposium (VisSym)* (2004), 17–24.