
Visual Analysis of Local Correspondence in Segmentation Quality

Bachelor-Thesis von Dennis Basgier
Mai 2013



Visual Analysis of Local Correspondence in Segmentation Quality

Vorgelegte Bachelor-Thesis von Dennis Basgier

1. Gutachten: Prof. Dr. Ing Georgios Sakas
2. Gutachten: Dr.-Ing. Tatiana von Landesberger

Tag der Einreichung:

Erklärung zur Bachelor-Thesis

Hiermit versichere ich, die vorliegende Bachelor-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 24. Mai 2013

(Dennis Basgier)



"Knowledge is power"

- Francis Bacon

Abstract

The Bachelor-Thesis presents a new interactive system for visual and exploratory analysis of local correspondence in segmentation quality. Segmentations of several samples of one organ are analyzed on the basis of pairwise distances between a reference- and a test- mesh, which is extracted from the organ segmentation. The tool features several views on the data (Coloring, threshold based highlighting, average mesh visualization) and a set of analysis methods (clustering, cluster quality evaluation, dimension reduction) to extract new information, such as reoccurring regions or patterns of low quality segmentation. Segmentation algorithm developer can use the visual information for gaining knowledge on how their algorithms work. This insight can be beneficial to the improvement of the algorithms.

The software is optimized for analyzing medical image segmentation, but can also be translated to countless domains, as it simply operates on the extracted mesh data.

Keywords: Medical Image Segmentation, Segmentation Evaluation, Mesh Comparison, Hierarchical Clustering, Spatial Clustering, Cluster Evaluation, Cluster Validity Index, SD Validity Index , Cluster Analysis, Visual Analysis,

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Goal	3
1.3	Overview	4
2	Definitions and Requirements	5
2.1	Definitions	5
2.1.1	Polygon meshes	5
2.1.2	Average Mesh	6
2.1.3	Reference- and test-mesh	6
2.1.4	Point Correspondence	6
2.1.5	Features, feature Vectors and feature space	6
2.1.6	Over- and under-segmentation	7
2.1.7	Interesting Regions	7
2.1.8	Clustering and clusters	7
2.1.9	Cluster compactness and separation	8
2.2	Requirements	8
3	Related Work	9
3.1	Medical Image Segmentation Evaluation	9
3.2	Clustering	11
3.2.1	Proximity measure (Feature distances)	12
3.2.2	Clustering algorithms	13
3.2.2.1	K-means	13
3.2.2.2	DBSCAN	14
3.2.2.3	Hierarchical clustering	15
3.3	Clustering quality evaluation and knowledge discovery	18
3.3.1	Clustering quality evaluation	18
3.3.1.1	Dunn Index	18
3.3.1.2	SD Validity Index	19
3.3.2	Knowledge discovery	20
3.4	Visualization of multivariate data with Dimension reduction techniques	22
4	Approach	25
4.1	Overview of the approach	25
4.2	Data input	26

4.2.1	Loading data:	26
4.2.2	Distance calculation	26
4.3	Data processing	27
4.3.1	Principle component calculation	27
4.3.2	Clustering	28
4.3.2.1	Clustering algorithm	30
4.3.3	Cluster quality evaluation	32
4.4	Visualization and interaction techniques	34
4.4.1	Interface	34
4.4.2	Distance based coloring	35
4.4.2.1	Threshold based coloring	36
4.4.2.2	Average distance coloring	37
4.4.3	Grid view	38
4.4.4	PCA view	40
4.4.5	Clustering view	42
4.4.6	Cluster analysis views	45
4.4.6.1	Cluster overview plot	45
4.4.6.2	Parallel coordinate plot	46
4.4.6.3	SD*-index simulation plot	49
4.5	Implementation details	50
5	Evaluation	51
5.1	Synthetic data: Clustering Evaluation	51
5.2	Use case: Liver segmentation	57
5.2.1	Explorative evaluation	57
6	Conclusion and Future Work	65
6.1	Summary	65
6.2	Contribution	66
6.3	Future work	66
Bibliography	68	
List of Figures	73	

1 Introduction

This chapter will first present the motivation based on a use case from the medical image segmentation domain, followed by the statement and goal of the Bachelor-Thesis and concluded by an overview of the thesis structure.

1.1 Motivation

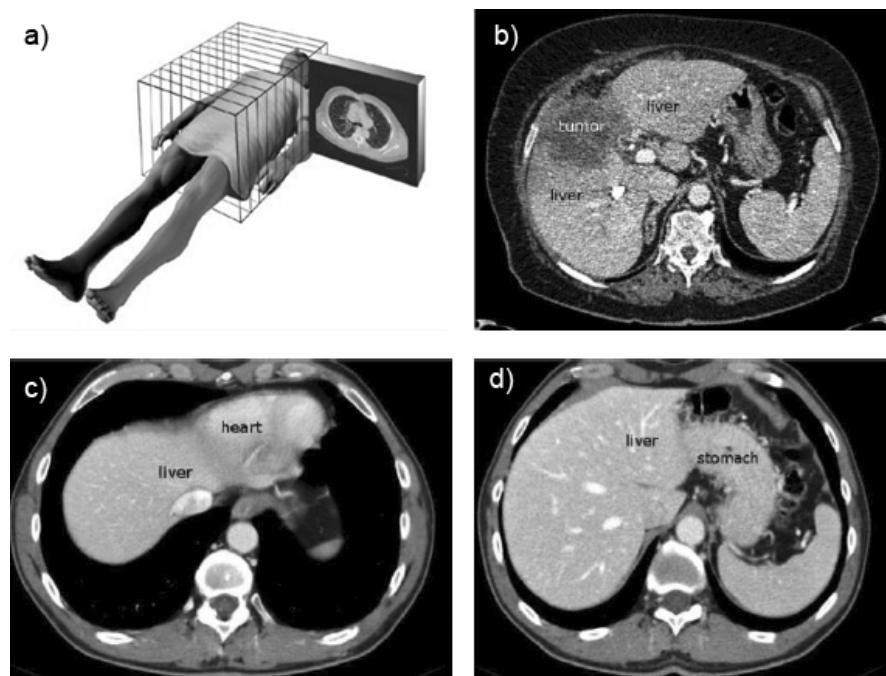


Figure 1.1: a) Dozens of CT- Slides of the Torso b-d) CT images and the problems of segmentation: b) Liver tumor splits the liver in two parts, because of different gray values c) & d) surrounding tissue like the stomach or the heart have identical gray values. Pictures from [HvGS09]

In modern medicine, medical imaging is essential for a large number of use cases, especially in diagnostics it is an inalienable tool to achieve reliable results about the patients condition. It is also crucial in the field of operation planning. The common imaging modalities such as CT (computed tomography), MRI (magnetic resonance imaging) or US (ultrasound) steadily improve in terms of quality and computation time, but the extraction of information from these images is still a challenging task [Doi06].

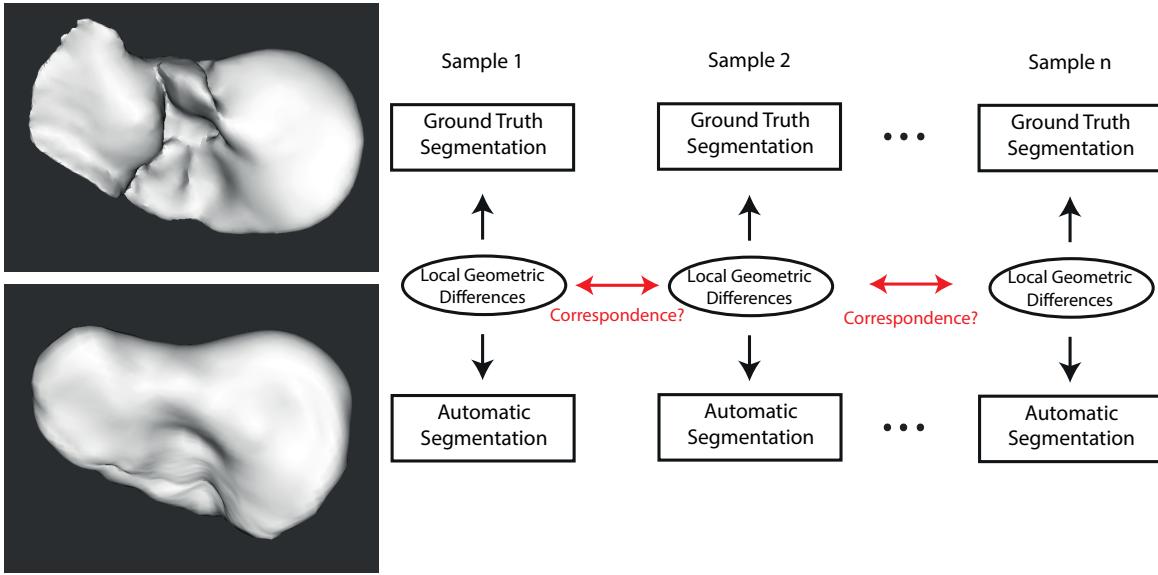


Figure 1.2: The left side shows the extracted ground truth-(top) and automatic segmented (bottom) meshes. The right side presents the problem, which is the missing correspondence of geometric differences over a number of samples (n)

To extract knowledge out of this data, usually a segmentation of the respective organ has to be done to check whether it is healthy or damaged. To find the borders of an organ, especially distinguish it from the surrounding tissue has proven to be challenging. A professional physician (an expert), who has experience in the field and knows the arbitrary shape of the organ needs to be consulted. The physician is capable of detecting abnormalities, which may lead to various diagnoses. The procedure of segmenting a whole organ is very costly and time consuming, as it involves analyzing dozens of images (Figure: 1.1a). The results are the borders of the anatomic structure over hundreds of slides. Therefore automatic segmentation in the medical sector is a rapid developing field with an active community of researchers.

The common approaches are region based, edge-based or threshold based algorithms which identify an anatomical structure based on the difference in the gray values [SK12] . Usually the respective organ has different characteristics, for example a particular density or water content which results in a unique gray value, which defines the specific organ and divides it from the surrounding tissue. A threshold is set and every image pixel which crosses this threshold will be excluded from the organ and will form a border between them. The results of this approach proved to be not adequate for diagnostics, because it is possible that surrounding tissue has identical gray values or abnormalities inside the respective organ (tumors or abscesses) can show different gray values, which prevents a properly segmentation (Figure 1.1).

The state of the art approaches use so called active shape models (ASMs)[KW11] to segment a specific organ. An ASMs uses a statistical shape model (SSM) to constrain the organ

boundary to plausible shapes. SSM is a geometrical analysis from a set of shapes in which statistics are measured to describe geometrical properties from similar shapes or different groups [HM09].

The active shape model approaches have proved to be a fast and accurate way for organ segmentation, however limitations of these methods are, that they tend to be over restrictive in terms of abnormalities in the boundaries such as tumors or damages (Figure: 1.1b).

The “perfect” segmentation algorithm has yet to be developed, but it is important to know the weaknesses of the existing procedures in order to improve them, as failures in the segmentation could lead to misdiagnoses. This can be done through comparing the expert segmentation (ground truth) and the automatic segmentation. There is a set of evaluation metrics (Section3.1) which estimate the overall quality of the segmentation result. However they fail to show whether a bad quality is caused by bad segmentation only in a certain area (locally bad quality) or is spread throughout the whole organ.

To gain geometrical information of the bad segmented regions, a 2D segmentation of every CT- slide is not ideal. To visualize the organ appropriately in 3D space, the surface can be extracted using the marching cubes algorithm [NY06].

The common comparing strategies of 3D surfaces (meshes) (Figure 1.2) involve overlaying ground truth and segmented surfaces one by one to see where the algorithm has failed. Apart from the disproportionate effort, this method only views each segmentation result individually, which prevents comparison of results across the test data set (Figure 1.2) .

1.2 Goal

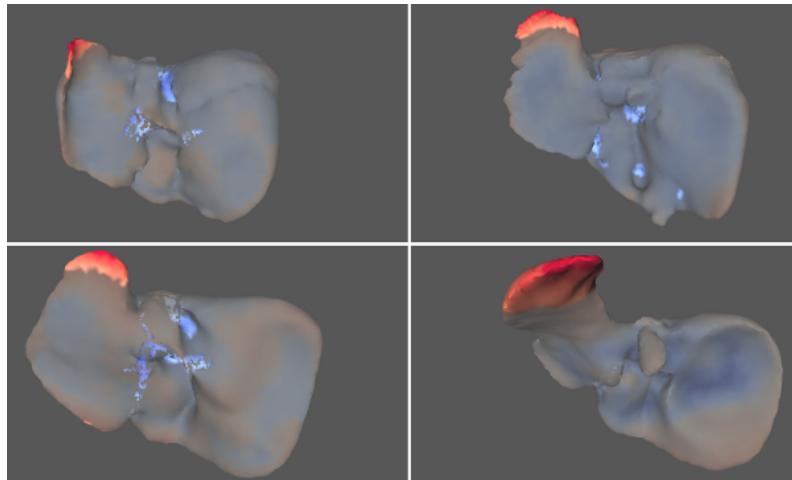


Figure 1.3: The regions of low quality are highlighted across the four data samples. A pattern of the geometrical position of reoccurring regions of interest is visible.

A new tool for visual analysis of local correspondence in segmentation quality over a number of samples has to be developed, to get a direct view of the weaknesses of segmentation

results. Figure 1.3 shows four different samples segmented by the same algorithm, the reoccurring region at the top left is clearly visible. The goal is to extract and analyze these regions of interest.

The analysis should be exploratory and interactive, because the desired result always depends on the use case and on the data set.

1.3 Overview

The work presents an interactive analysis tool for mesh comparison.

In the next chapter I will provide definitions of important terms, which are fundamental to the further progress of the work, followed by the requirements of the software.

In the related work chapter 3, I will start with a closer look at state of the art approaches of medical image segmentation evaluation, clustering algorithms and evaluation methods and dimension reduction techniques.

In chapter 4 I will describe how I altered and utilized these methods in my approach, in order to achieve the goals and solve the presented challenges.

In the following chapter, the implementation of these techniques is presented as well as the visualizations and the interaction possibilities of the system.

Chapter 5 presents an evaluation of the software result on the basis of two different data sets. One data set containing several synthetic data samples and the other one is a data set with 20 different ground truth and automatic-segmentations of a liver data set.

In the final chapter I will summarize my work, and provide possible enhancements of my approach.

2 Definitions and Requirements

In this section I will give definitions of important terms, which are essential for the further progress of this work, followed by the requirements of the system.

2.1 Definitions

2.1.1 Polygon meshes

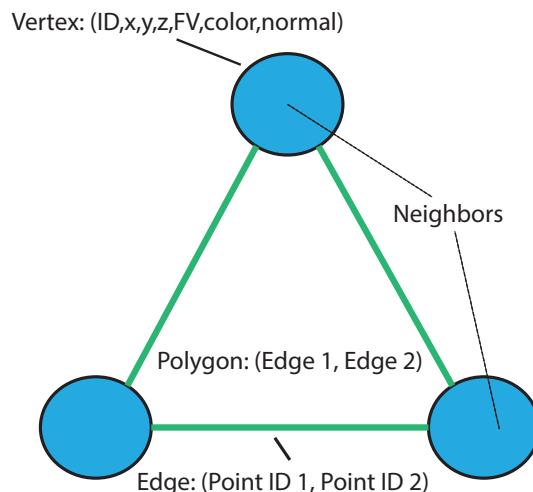


Figure 2.1: Shows a mesh with example information. FV := Featurevector

In computer graphics a collection of connected points via edges defines a polygon mesh. There are many different representation of polygon meshes, which are used for various applications [Got05]. Each one of them has to provide basic information, to represent its data content (Figure: 2.1). These information are:

- x,y,z values to define the position of points in a Cartesian space
- Point ID to define the correspondence and to define edges and polygons
- A list that uses these IDs to specify connected points (**neighbors**). These vertex pairs are connected via edges. In a triangular mesh three edges build a polygons. The edge list can be used to find neighboring vertices
- Miscellaneous information such as point color or surface normals

2.1.2 Average Mesh

A point p with the ID in the average mesh is defined as:

$$p^{ID} = \frac{1}{n} \sum_{i=0}^n x_i^{kID}$$

where n is the number of samples to take the average from and x_i^{ID} is the corresponding vertex to the ID in sample number i .

2.1.3 Reference- and test-mesh

The purpose of this software is to compare two meshes, but mainly to compare one test-model to a reference-mesh. In most cases the reference-mesh is the expert segmentation (ground truth), and the test-mesh is the automatic segmentation. The user is free to load any two meshes (for example to compare one segmentation algorithm to another); Hence it is more precise referring the objects to reference- and test-meshes.

2.1.4 Point Correspondence

The term point correspondence (in this case) means that one point with vertex ID x in sample A corresponds with the point according to the same point ID in sample B . Concluding the corresponding point pairs represent the same geometric position in both meshes, although the meshes differ locally from each other (Figure 2.2).

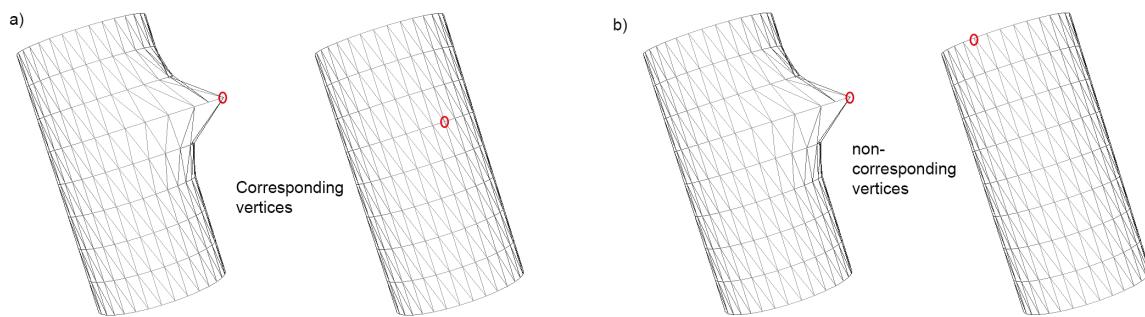


Figure 2.2: Comparison between corresponding and non-corresponding vertex IDs in two meshes: **a)** shows two vertices with the same vertex ID, they geometrically correspond **b)** shows two non corresponding vertices with the same ID

2.1.5 Features, feature Vectors and feature space

A feature is an individual property for each object which allows to distinguish these objects between themselves. A unique feature is for example the vertex-ID for every vertex

in a polygonal mesh. We extend the vertices with a special features: The distance between reference- and test-model (Figure: 2.1). If n is the number of sample pairs and the point correspondence is given we can assign n features to a vertex-ID . The set of these features is a feature vector.

If we talk about distances between vertices, it could either be the distance in the geometric space or in the feature space, which is the distance between feature vectors. How the feature distance is defined will be discussed in section 3.2.1.

2.1.6 Over- and under-segmentation

As previously stated, in medical image processing there is no 'perfect' segmentation; even different expert segmentation will differ from each other. There are two cases of segmentation errors: **Over- and under-segmentation**. Over-segmentation describes a segmented border of the test-model which is 'outside' of the reference-model's border and under-segmentation describes the opposite (Figure: 2.3) . If the mesh is extracted, the under-/over- segmented parts will appear 'inside/outside' of the other mesh.

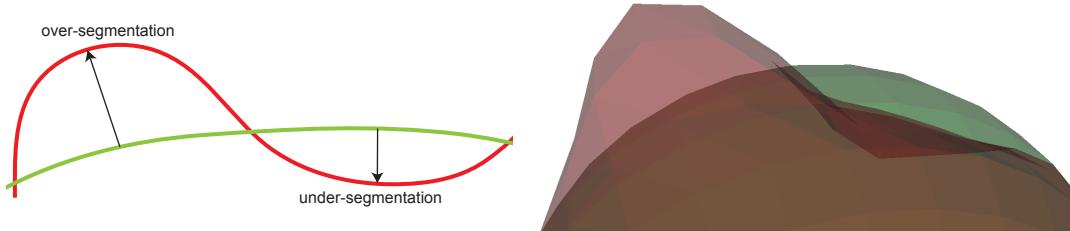


Figure 2.3: Example of over- and under-segmentation. The left picture shows an example of two segmented borders (green: reference-model, red: test-model) in a CT-slide. The right picture shows the resulting mesh.

2.1.7 Interesting Regions

The expression "interesting region" is an ambiguous term, because it is dependend on ones definition of "interesting". In this case interesting regions are areas of the mesh that consists of neighboring points which have similar feature vectors, the definition of similarity in feature space and the conditions to connect these regions is yet to be defined. It will be discussed in Section 3.2.1.

2.1.8 Clustering and clusters

"Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). "[JMF99]

These distinct groups contain a number of data points which are more similar (in a defined sense) to each other than to points in different clusters.

2.1.9 Cluster compactness and separation

The quality of a clustering result is usually determined by rating the compactness and separation of the clusters.

- The **compactness** is a value which determines how close the data points in each clusters are.
- **Separation** on the other hand specifies how well the members of one cluster differs from the members of other clusters.

There are many approaches to calculate these criteria.

Common approaches for compactness are based on the calculation of distances between the cluster center and each member.

The separation measures are usually based on the distance between the cluster centers.

2.2 Requirements

In my system the user can choose any data set to be analyzed, as long as they are in a compatible data format (*.mhd, *.vtk, *.off, *.obj) and meet the three following requirements:

1. All the **test-** meshes have to have the same number of vertices otherwise the different calculations will fail
2. The point correspondence between the test-meshes has to be given
3. The basic shape of the meshes should be similar, otherwise the average visualizations would be confusing in terms of the geometrical landmarks

3 Related Work

3.1 Medical Image Segmentation Evaluation

A similar approach to solve the proposed problem is presented in [GG01]. VALMET is a tool for validation and comparison of object segmentation. The Input for the tool are the original 3D images (MRI, CT, ultrasound), what makes it bound to the medical field. It offers a graphical overlay for the original slices, a 3D view for visual assessment of the locality and magnitude of segmentation variability and several common quantitative evaluation measure, including intra-class correlation and four different shape distance metrics:

- **Volumetric Overlap** is a simple approach for pair-wise comparison of segmentation. The segmented images are analyzed voxel¹ by voxel on the basis of volume overlap which means each voxel is checked whether the containing volume is used by both segmentations or by one. The final measure is calculated by dividing the intersection of both samples by the union $((T \cap R)/(T \cup R))$ or by the reference object² only $((T \cap R)/R)$. Total agreement of the two segmentations results in a score of 1 the opposite in 0. This measure is sensible to the size and shape complexity of the sample, because most of the volume calculation errors occur on the boundaries of the object, which means, if the ratio over the boundary voxels and inside voxels grows, the impact of these errors becomes more vital. There are several other common measures in this category [CBC⁺07] like the Jaccard (JC), Dice Similarity (DS), Tanimoto (TN), and Volume Similarity (VS) , which express different knowledge about the segmentation quality but all of them result in a global measure, which means the information about the local segmentation quality is lost.
- **Probabilistic overlap measure for non-binary segmentations:** Sometimes in medical image segmentation there are no clear boundaries between anatomical structures, which means even manual segmentations can differ from each other (probabilistic segmentation). Therefore a probabilistic overlap measure was developed by deriving the normalized distance between two probability distributions. The new measure is defined as:

$$POV(A, B) = 1 - \frac{\int |P_A - P_B|}{2 \int P_{AB}}$$

where P_A and P_B represent the probability distributions of the probabilistic segmentations and P_{AB} is the pooled joint probability distribution.

¹ Voxel $\hat{=}$ Volumetric Pixel, volumetric element, representing a value on a grid in 3D-space

² T: Test Object, R: Reference Object

- **Surface Distance and Maximum Surface Distance (Hausdorff distance) :** [CRS96, ASCE02] The surface distance is the distance from one point of mesh R to the nearest point (Euclidean distance) in mesh T. This alone leads to a non - symmetrical distance between two points, which means the distance from R to T is not necessarily the distance from T to R. To solve the problem, the maximum distances of both directions of the corresponding points is chosen, the result is the Maximum Surface Distance. The advantage of this method is, that no correspondence between the reference-mesh and the test-mesh is needed. However, this distance measure is not very accurate because the method fails in non-uniform regions. The measure is also sensible to registration³ problems, because a wrong registration could lead to wrong surface correspondence. A detailed description of problems with surface distance is discussed in [Get13]

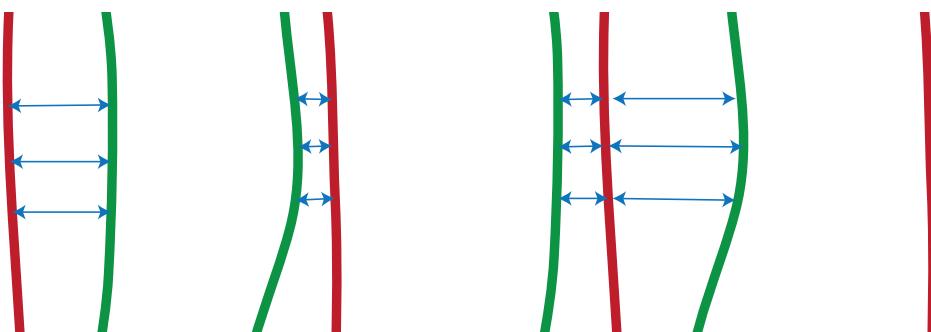


Figure 3.1: A longitudinal section through two tubular meshes (red and green) and their respective contours the arrows show the result of the surface distance. The left picture shows an accurate registration of the two meshes, which results in the right surface distances, whereas the right pictures shows the error, due to a wrong registration.

- The **Mean absolute surface distance** method is quite similar to the volumetric overlap, it traces the surface and integrate the distance values (surface distance) of each point (both over- and under-estimation) and calculates the average. The result is the absolute average distance of both surfaces, which means only domain experts can interpret this value, because one must know the absolute measures of the object. The information of local segmentation quality from the Maximum Surface Distance is obviously lost during the process.
- **Interclass correlation coefficient for assessing intra -and inter-rater reliability** ⁴ : This method produces a reliability measure, which compares different segmentations on the same data set. A mean object is calculated among the data at hand and the variance σ_i^2 relative to this ,so called centroid, of every rater is calculated. To calculate

³ Registration is the aligning of two surfaces to match locally [MV98, ZF03]

⁴ The rater is the respective expert or algorithm, who/which segmented the organ

the overall variance of the object, the mean variance across all raters σ_a^2 is defined. The interclass correlation is then defined as:

$$\rho = \frac{\sigma_a^2}{\sigma_a^2 + \sigma_i^2}$$

If the raters variance is small relative to the total, the result will be close to 1, due to natural variation. There are also several analytical measures, which take the characteristics of the segmentation algorithm into account, but these measures are very specific and not well suited for this work.

Looking at the above mentioned evaluation measures it becomes clear that the use case of VALMET is mainly pairwise analysis of two (reference- and test-mesh) or several segmentations of **one** data set. The possibility to compare different segmentation over different data sets is not given. It is also obvious that the developers did not put too much emphasis on the local segmentation quality, because four of the five measures are global quantitative measures. The main focus of this project is to determine the quality of a segmentation according to a score, which can iteratively be used to improve segmentation quality (for example due to parameter tuning of the segmentation algorithm), but a targeted analysis (for example to improve reoccurring low quality regions) is not possible.

3.2 Clustering

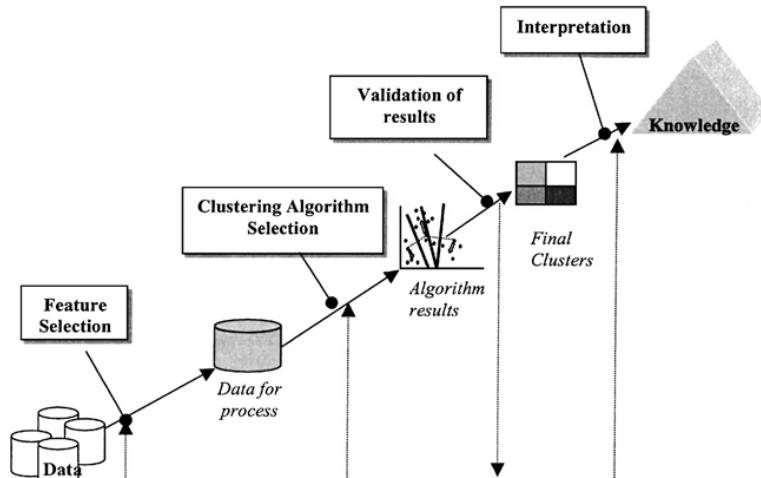


Figure 3.2: Steps of clustering process. Source:[HBV01]

The aim of clustering is to reveal the organization of patterns, to discover similarities and differences and to derive conclusion about them. Clustering in contrast to classification is an unsupervised process, which divides the data set without predefined characteristics, even

more, the extracted information about the clusters should provide rudiment approaches to define characteristics, which are shared by a number of data values. Figure 3.2 summarizes the basic steps of the clustering Process. The feature selection was already discussed in section 2.1.5 followed by the choice of a clustering algorithm.

There are many alternatives of algorithms, **K-means**, **DBSCAN** or **hierarchical methods**[For65, EpKSX96, JD88] are common approaches of clustering, which will be discussed in the next sections. The selection of a fitting algorithm is not the only choice which has to be made in the process, because a clustering algorithm is mainly characterized a proximity measure, which measures the distance between two data points (feature vectors) and by a clustering criterion, which defines, if a data point is qualified to belong to a distinct cluster.

After the clustering, the results correctness has to be verified, which often goes hand in hand with the knowledge discovery or interpretation of the result. Approaches in both fields will be discussed in section 3.3.

3.2.1 Proximity measure (Feature distances)

For all clustering methods it is essential to define a distance measure to determine the distance between feature vectors.

There are dozens approaches to determine the distance between two feature-vectors (a, b) [Cha07]. Two common measures are the Euclidean distance and the Chebyshev distance.

The Euclidean distance is defined as:

$$d(a, b) = \sqrt{\sum_{i=0}^n (a_i - b_i)^2}$$

And the Chebyshev distance:

$$d(a, b) = \max(a_i - b_i, a_{i+1} - b_{i+1}, \dots, a_n - b_n)$$

a_i and b_i are the i-th component of the feature-vectors corresponding to two data points and n is size of a feature-vector.

Each distance metric creates different cluster result, which could lead to several conclusions. The choice is use case dependent.

3.2.2 Clustering algorithms

3.2.2.1 K-means

The K-means algorithm [For65] belongs to the group of **squared error** - partitioning methods. Initially, a parameter K has to be defined and the goal is to assign each data object (N number of data Objects) to K clusters with minimization of the squared error criterion, which is defined as [XW05]:

$$J(\Gamma, M) = \sum_{i=1}^K \sum_{j=1}^N \gamma_{ij} \|x_j - m_i\|^2$$

where

Γ = a partition matrix;

$$\Gamma = \begin{bmatrix} \gamma_{1,1} & \cdots & \gamma_{K,1} \\ \vdots & \ddots & \vdots \\ \gamma_{1,N} & \cdots & \gamma_{K,N} \end{bmatrix}$$

$$\gamma_{ij} = \begin{cases} 1 & \text{if } x_j \in \text{cluster } i \\ 0 & \text{otherwise} \end{cases} \quad \text{with } \sum_{i=1}^K \gamma_{ij} = 1 \forall j;$$

M = cluster centroid(mean) matrix;

$[m_1, \dots, m_k]$ with

$$m_i = \frac{1}{N_i} \sum_{j=1}^N \gamma_{ij} x_j$$

(sample mean for the i -th cluster)

N_i = number of objects in the i -th cluster

To minimize the squared error the steps of the K-Means algorithm are defined as

1. The algorithm starts with a random or knowledge based⁵ positioning of the K centroids in the feature space.
2. The second step is to assign each data object to the nearest cluster centroid.
3. After that, the cluster centroids have to be recalculated based on the current partition
4. The last two steps will be repeated until the partition will not change.

K-means is suitable for large data sets, but the quality of the clustering is heavily dependent on the parameter K and the positioning of the initial cluster centroid. Especially with an inappropriate positioning the algorithm becomes sensible to outliers.

Due to the need of finding the optimum of K (will be discussed in Section 3.3) and positions, the computation time can get very high for finding the optimal clusters.

3.2.2.2 DBSCAN

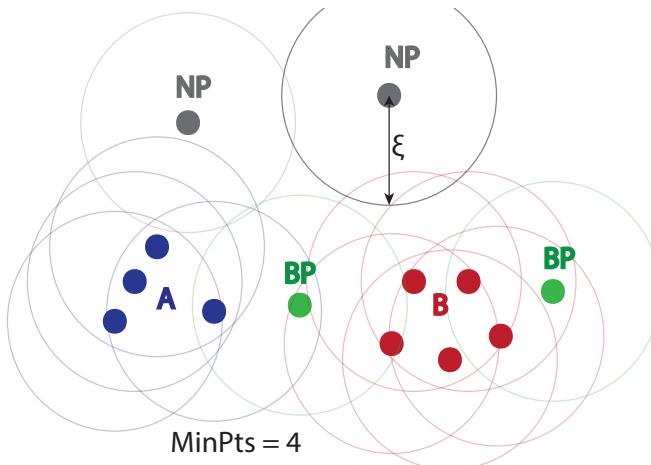


Figure 3.3: A clustering result of DBSCAN. The circles represent the reachability radius of each data value. The blue and red dots are the core points of cluster A respectively B, because they possess more data points in their reachability radius than minPts defines. The gray dots will be marked as noise points and will not be assigned to any cluster. The green dots represent the border points. The right border point will be assigned to cluster B, whereas the left could be assigned to both clusters. Usually it will be randomly assigned to one of them.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise [EpKSX96]) is a **density-based** clustering method. This category of algorithms defines clusters as areas of data values which have higher densities than the rest of the data.

⁵ Many approaches how to position the initial centroids [For65, KR90]

The algorithm uses two parameters: ε , which defines the density **reachability radius** and **minPts** which defines a threshold to decide if a data object is marked as 'dense'.

With the help of these parameters DBSCAN classifies all data points in three categories:

- **Core points** are data objects which possess more than minPts in its reachability radius ε . These points can be defined as 'dense'.
- **Border points** are data values which are in the density reachability ε of a core point, but are not marked as dense.
- **Noise Points** are objects which suit neither of the previously mentioned categories

Chains of connected core points and the adjacent border points form a cluster. It is possible that border points are connected by two different clusters, which means the algorithm has to decide (mostly random) which cluster this point is referred to. Note that in contrary to K-means, noise points will not be assigned to a cluster, what generally makes the algorithm robust against outliers.

However, if the parameters are chosen inappropriately, the worst case can either be that no cluster (all points are marked as noise points) or only one cluster (chaining: all core points are connected) is formed, which is not the desired result.

3.2.2.3 Hierarchical clustering

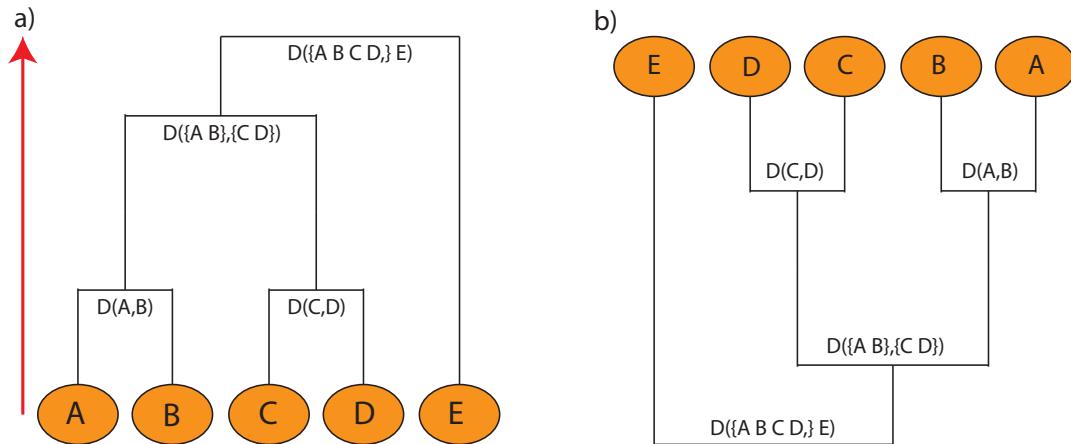


Figure 3.4: Cluster dendrogram: **a)** agglomerative strategy **b)** divisive strategy was used. The Results are the same

Hierarchical clustering also known as **connectivity based** clustering connects points based on their distance to each other.

The result is usually represented in a tree (called dendrogram) in which the leaves represent the data points and the edges contain the hierarchy-level.

There are two basic strategies, which leads to the final form of the Dendrogram [JD88]:

- **Agglomerative**, also called “bottom up approach” initially starts with every data point in its own cluster and moves up the hierarchy as it begins to link cluster pairs with minimal distance from each other until all points belong to the same (Figure:3.4a).
- **Divisive**, also called “top down approach” starts with one cluster, which contains all the elements and splits the groups, so the distance between the resulting them will be maximized as it moves down the hierarchy (Figure:3.4b).

To determine the distance of two clusters a so-called **linkage criterion** needs to be defined. There are several linkage criteria that can be considered [Mur83] , each resulting in different hierarchies. It is difficult to predict the outcome of each criterion, because it strongly depends on the structure of the input data [FS05].

The **complete linkage criterion** is the maximum distance between all elements of each cluster A, B , it is defined as:

$$D(A, B) := \max(d(a, b) : a \in A, b \in B)$$

where $d(a, b)$ describes the chosen proximity measure.

This criterion maximizes the compactness of the cluster, but is sensitive to outliers.

The **centroid-linkage** uses the distance between the two cluster centroids, to determine the center I used the average distance of all points in the cluster. The definition is:

$$D(A, B) := d(\bar{a}, \bar{b})$$

Where \bar{a} and \bar{b} define the cluster centroids of A and B. The compactness with this criterion can be worse, but it is robust against outliers.

There are several other criteria, which are either not compatible with the goal of the system or quite similar to the above two:

- In the **average linkage criterion**, the mean distance of all element pairs of the clusters is computed which will have the same result as centroid-linkage, if the cluster center is the average distance of all cluster points.
- The **single-linkage** is very similar to the complete-linkage, but uses the minimum distance of all elements. The consequences are that clusters may be forced together because of single elements, this problem is called chaining Phenomenon and results in a low cluster separation.
- **Ward's-linkage-methods** compute the compactness when merging two clusters and combines the pair with the minimal aggravation. The compactness will be optimized, but the clusters tend to be the same size, which is unbefitting in our use case.

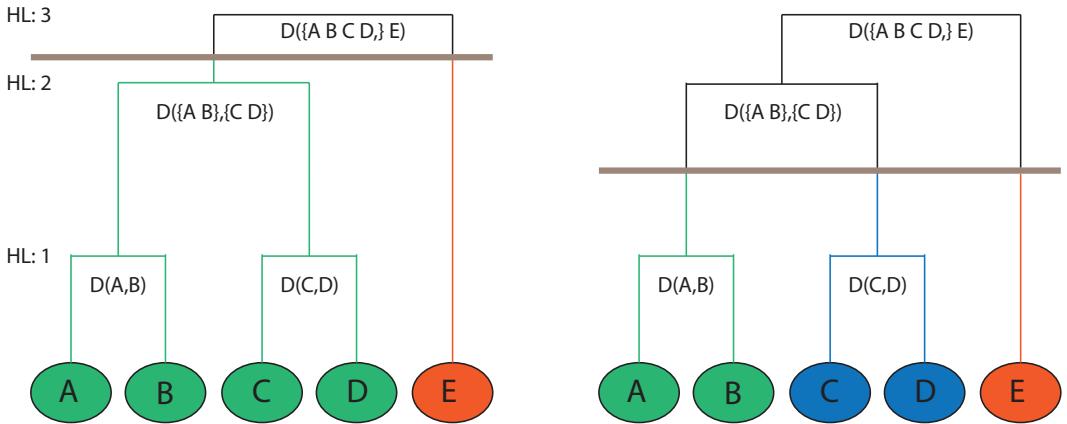


Figure 3.5: Cluster Dendrogram with two different thresholds, the left threshold is set to hierarchy level 3 (HL), which will result in two clusters, whereas the right is set to HL 2, which will form three

After the generation of the hierarchy (dendrogram), the final clusters are yet to be defined. The dendrogram contains information about the relationship between the points as it connects the pairs with the minimal distances between each other. The hierarchy level (HL), which describes the order of how the clusters are merged together, is also saved within the dendrogram. With each merge of the clusters the hierarchy level is incremented.

To form the clusters a **threshold** has to be found, which represent the mentioned hierarchy level. The threshold describes which edges in the dendrogram will be cut. All points which the sliced edges connected will belong to the same cluster.

There are two different trivial cases: If the threshold is set to the maximum hierarchy level, the result will be one cluster containing every point of the data set, in contrary if the threshold is set to the minimum, the consequence will be many clusters, each containing one data set point. All thresholds in between will result in different clusters (Figure: 3.5).

Compared to K-means and DBSCAN no initial parameters have to be defined.

However, a linkage criterion has to be chosen and a threshold has to be defined afterwards, which is not a trivial problem. The solution is similar like the definition of the parameter K in K-means. It will be discussed in Section 3.3.

3.3 Clustering quality evaluation and knowledge discovery

3.3.1 Clustering quality evaluation

The result of each clustering algorithm can be very different, even the result of the same algorithm with different parameter can easily differ from each other. The goal of each clustering is to find the best partitioning that fits the underlying data, usually that is to maximize the compactness and separation of the clusters.

In high dimensional data sets it is nearly impossible to verify the results only through visualization, therefore formal methods are needed. There are three different classifications of formal validation techniques.

*The **External validity methods** evaluate the clustering based on user specific intuition. **Internal criteria** are based on some metrics which are based on the data set and the clustering algorithm. [KLB]*

Both of them use statistical methods.

Relative criteria on the other hand are independent of the underlying data and the clustering algorithm, they simply compare one or more clustering algorithm result, executed with different parameters and measures (relative to the data set) the 'goodness' of the partitioning.

The 'goodness' is always computed by a combination of a value that identifies the clusters compactness and a measure which describes the separation of the clusters.

The best result of the respective method can provide hints for the optimal parameter combination, for example the best K for K-Means or the best threshold for hierarchical clustering. Note, that different criteria could lead to different 'optima'. Choosing an appropriate index is always dependent on the underlying data and the clustering algorithm.

Two widely used indices are introduced in the following sections:

3.3.1.1 Dunn Index

The Dunn index [Dun74] is based on the fact, that well-separated clusters have usually large distances between each other and the diameter⁶ of the clusters is expected to be small. The index is defined as:

$$D = \min_{i=1..n_c} \left\{ \min_{j=i+1..n_c} \left(\frac{D(c_i, c_j)}{\max_{k=1..n_c} (\text{diam}(c_k))} \right) \right\}$$

with

⁶ The maximum distance of data points in a cluster

$$D(c_i, c_j) = \min_{x \in c_i, y \in c_j} \{d(x, y)\}$$

and

$$\text{diam}(c_i) = \max_{x, y \in c_i} \{d(x, y)\}$$

c_i describes the i-th cluster, n_c the number of clusters and $d(x, y)$ the proximity measure of two feature-vectors.

Smaller values of this index describe better cluster configurations.

The Dunn index is very sensitive to noisy data, because the cluster diameter can become very large.

3.3.1.2 SD Validity Index

The SD Validity Index [HVB00] computes the average scattering and the total separation of the clusters and combines them. The scattering of the cluster is calculated with help of the cluster variance and the variance of the whole data set:

Variance of the data set:

$$\sigma_x^p = \frac{1}{n} \sum_{k=1}^n (x_k^p - \bar{x}^p)^2$$

with

$$\sigma(x) = \begin{bmatrix} \sigma_x^1 \\ \vdots \\ \sigma_x^d \end{bmatrix}$$

Variance of a cluster:

$$\sigma_{v_i}^p = \frac{1}{||c_i||} \sum_{k=1}^n (x_k^p - \bar{v}^p)^2$$

with

$$\sigma(v_i) = \begin{bmatrix} \sigma_{v_i}^1 \\ \vdots \\ \sigma_{v_i}^d \end{bmatrix}$$

The average scattering:

$$Scatt = \frac{1}{n_c} \sum_{i=1}^{n_c} \frac{\|\sigma(v_i)\|}{\|\sigma(x)\|}$$

where n describes the number of points, $\bar{x^p}$ the expected value of the p-th dimension, v_i the cluster centroid of the i-th cluster, $\|c_i\|$ the number of elements in the i-th cluster.

The **total separation** is calculated using the distance of the cluster centroids:

$$Dis = \frac{\max_{i,j=1 \dots n_c} (\|v_j - v_i\|)}{\min_{i,j=1 \dots n_c} (\|v_j - v_i\|)} \sum_{k=1}^{n_c} \left(\sum_{j=1}^{n_c} \|v_j - v_i\| \right)^{-1}, i \neq j$$

The SD index can be calculated with:

$$SD = \alpha \cdot Scatt + Dis$$

where α is the total separation (Dis) of the maximum number of clusters, resulting from the clustering algorithm execution with different parameter combinations.

“The number of clusters at which SD reaches its minimum value and also a significant local change (i.e., decrease) in its value occurs, can be considered as the best one”[HVB00]

3.3.2 Knowledge discovery

Usually, if a clustering algorithm is applied to a large and high dimensional data set it is very difficult to analyze why the clusters were formed in the specific pattern. A first approach to get an insight into the algorithm result is an appropriate cluster visualization, e.g. a cluster dendrogram for hierarchical clustering. However, with thousands of data points, resulting in thousands of tree leaves, it is nearly impossible to retrace all the points belonging to one cluster, plus the dendrogram only contains the information about the order of connections (hierarchy level), which means the knowledge about each individual test-subject is lost.

To discover and visualize the knowledge about a cluster several techniques can be used [HG97].

For visualizing high-dimensional geometry and analyzing multivariate data the most common technique are **parallel coordinates** [Ins85].

“The technique capitalizes the abilities of the human visual system to efficiently acquire and process parallel information” [NH06],

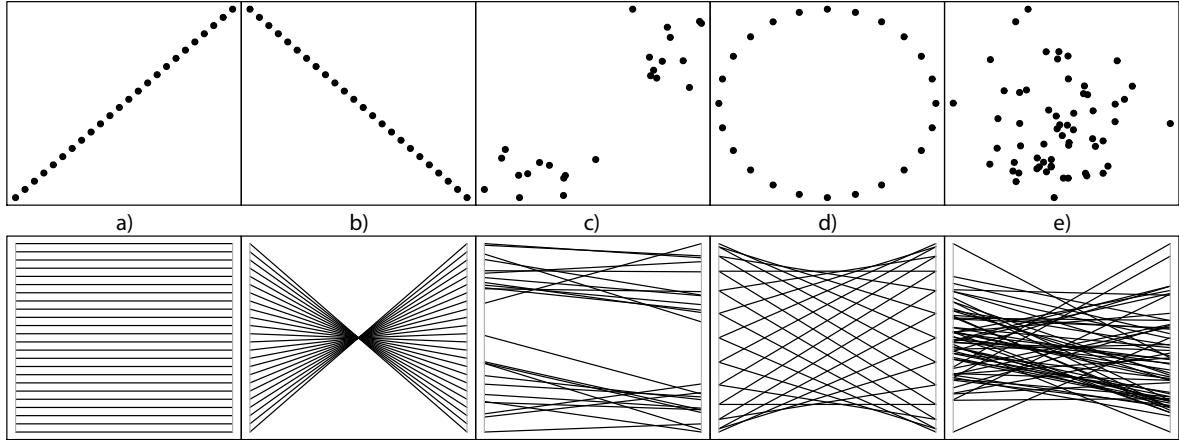


Figure 3.6: Different Correlations between two dimension (top row) in parallel coordinates (bottom row),
Source: [WFI13]

it takes a set of n -dimensional points and aligns them on a coordinate system of n parallel (one for every dimension) lines typically vertical and equally spaced. The data-points are finally presented as a line which connects the values in every neighboring dimension.

If the data points are split into groups (for example clusters, Fig: 3.6c), they can be distinguish using different colors, which enables identification and comparison of the characteristics of different groups.

High numbers of data values can result in a problem in terms of over plotting and it becomes difficult to filter out information. However, assuming that the data is Gaussian distributed (Fig: 3.6c) the center of the values can be visualized using the 0,25 and 0,75 Quartile or the average respectively median of the data.

The positioning of the Quartiles provides information about the quality of the cluster and the size of Quartiles supplies information about the density of clusters, which in combination helps identifying reoccurring regions with same characteristics.

Common problems of parallel coordinates are the scaling and ordering of the axes.

- The consequence of a projection from an n -dimensional to a 2-dimensional space is usually the loss of information, because one data point is exclusively connected to its neighboring dimensions, hence the visible correlation is only linear. Therefore the **ordering** of the axes is crucial for statistical analysis, because an inappropriate ordering could result in a missing correlation. The problem can be solved by using different ordering heuristics or computer aided interactive orderings to find a solution to reduce the leak of information [YPWR03, Weg90], however, the information loss can never be zero.

- The parallel axes are basically a sequence of coordinate axis, which means the maxima and minima of every dimension could be significantly different. Therefore it is necessary to find a common **scale** for every dimension. Different scales of the axis can provide new information.

In the case of visualization and comparing clusters these problems are not as vital as for statistical analysis, because what need to be discovered are distinctive abnormalities, for example a dense and large cluster with low segmentation quality.

3.4 Visualization of multivariate data with Dimension reduction techniques

To get an initial impression of a high dimensional data set, it is beneficial to extract information by reducing the dimension of the data set. Reducing the dimension of a data set always results in an information loss, but the goal of statistical dimension reduction techniques is to minimize this loss. In literature [Cun07] the reduction methods are often distinguish in two types: linear and non-linear. The outcome of linear dimension reduction is a new variable with k components, with each k being a linear combination of the original variables. The most used strategy for linear dimension reduction is Principal Components Analysis (PCA)[AW10, Smi, Pea01]. It is a mathematical procedure to convert possibly correlated variables into a set of linearly uncorrelated variables using orthogonal transformation. The number of these variables (principal components) is equal or less to the number of the basic variables.

To apply PCA, a data Matrix X is needed, which columns consists of p observations and which rows represent the variables over all observations from n :

- The first step of the PCA transformation is to subtract the mean of the column \bar{y}_i from all the X_{ij} values. The result \bar{X} is a data set whose mean is zero.
- The next step is to calculate the covariance Matrix C of X :

$$C = \begin{bmatrix} cov(X_0, X_0) & cov(X_0, X_1) & \cdots & cov(X_0, X_p) \\ cov(X_1, X_0) & cov(X_1, X_1) & \cdots & cov(X_1, X_p) \\ \vdots & \vdots & \ddots & \vdots \\ cov(X_p, X_0) & cov(X_p, X_1) & \cdots & cov(X_p, X_p) \end{bmatrix}$$

with

$$cov(X_a, X_b) = \frac{\sum_{j=1}^n (\bar{X}_{a,j})(\bar{X}_{b,j})}{(n - 1)}$$

where X_i is the i-th column of Matrix X and $\bar{X}_{i,j}$ is the value at the i-th column and j-th row of the mean corrected Matrix. The outcome is a symmetrical matrix $C^{p \times p}$ with p rows and p columns and each entry is the covariance between two separate dimension.

- The third step is the calculation of the eigenvalues and eigenvectors of the covariance matrix and ordering them from the highest to lowest eigenvalue. There are p eigenvectors, which provide the information about the patterns in the data, so with the eigenvectors we can transform the data to a new coordination system that characterizes the data. Note that it is very important that these eigenvectors are unit vectors, which means their length is one.
- The following step is the choosing of the components, which is the actual dimension reduction. The eigenvector belonging to the highest eigenvalue is the principle component of the data set, because it represents the most significant relationship between the dimensions. According to the number d of dimension which have been chosen to extract, the d highest eigenvalues and eigenvectors will be selected. A feature matrix is formed out of these eigenvectors, which will result in a $p \times d$ matrix:

$$\text{FeatureMatrix} = (\text{eigenvector}_1, \text{eigenvector}_2, \dots, \text{eigenvector}_p)$$

- This Matrix is the key to derive the new reduced data set, which is the final step of Principle Component Analysis:

$$\text{FinalData} = \text{FeatureMatrix}^t \cdot \bar{X}^t$$

The result is a transformed data set according to our selected eigenvectors. This transformed data can be used to visualize a high dimensional dataset in a e.g. two or three dimensional coordinate system.

Note that there are several other linear dimension reduction techniques [Fod02] like MDS (Multidimensional scaling, which is basically the same as PCA, if the Euclidean distance is used between data points) or LDA (Linear Discriminant) and non-linear methods like ICA [HO00] or kernel PCA [SSM99], but in the medical field Principle Component Analysis is still the standard method and proved to be reliable to solve those kind of problems.

4 Approach

The goal of the work is to identify and extract connected regions on every sample, which show similar characteristics in terms of distances between the reference- and test-object. The result should provide information about the local correspondence or non-correspondence of these regions over the whole sample data set.

To achieve the goal I developed a system (**VAMCo** - Visual Analysis of Mesh Correspondence) with multiple views and methods to analyze the data without losing the geometric context of the mesh.

In this chapter I will explain how I used the basics methods mentioned in the related work chapter, altered and combined them and put them together into suitable visualization to extract the needed information.

The first section provides an overview of the whole approach.

Section 4.2 describes how the data is loaded into the system and what preparations are necessary for further processing.

The following section explains the pre-processing methods, which includes the PCA- calculation, a new clustering algorithm and a new cluster validity index adapted to the new clustering method.

In the next section I will give a detailed description of the visualizations of the processed data and the interaction techniques used by my system.

Finally in Section 4.5 I will briefly provide information about technical specifics.

4.1 Overview of the approach

In the related work chapter, several common approaches of segmentation quality evaluation are introduced. To extract local corresponding regions over a set of samples, merely the surface distance approaches are relevant

Given the surface distance of every vertex in the samples, my first idea to visualize correspondent regions of the same segmentation quality, was assigning a color and an opacity value to each vertex according to the distance and to under- respectively over-segmentation and render all meshes in a grid.

To get a clear view on corresponding regions over all samples on one geometrical structure and not on each individual object, I developed an average distance coloring method and visualized it on the average mesh.

Regions with large quality deviations could easily be overlooked with the average coloring though, therefore I developed a new clustering algorithm which partitions the data in feature space, but takes the geometrical neighborhood relationships of the vertices into account.

The result is a clustering of the segmentation quality, with a spatial constraint, where only connected regions will be clustered. The clustering result is also visualized with help of the average mesh and can be visually evaluated, analyzed and compared using parallel coordinates. For the formal evaluation I adapted a clustering validity index to the new clustering method.

Clusterings however, can easily be influenced by outliers. Outliers are defined as samples which location patterns (on the mesh) of bad segmented regions differ greatly from the usual.

To identify which samples show the same patterns of interesting regions, I reduced the dimension of the data, using the principle component analysis over the surface distances of all samples and arranged the meshes in a 2D coordination system according to the two principle components. Outliers can now be excluded from the analysis process to increase the local correspondence in segmentation quality of the remaining subset of samples.

4.2 Data input

4.2.1 Loading data:

Like previously stated, the user is free to load any kind of meshes, as long as they are in an acceptable format (*.mhd, *.off, *.vtk, *.obj) and meet the requirements (see 2.2). Firstly the user has to choose all the reference-meshes, followed by the test-meshes.

This is done using a simple file dialog, obviously the number of both models has to match, otherwise a pairwise comparison is not possible. The software is capable of using a marching cubes algorithm to extract the mesh out of image data if necessary, but this is only possible for the reference- models because the extracted meshes will have different number of points and no correspondence between each other. After that, the distance calculation will be set in motion.

4.2.2 Distance calculation

The distance between the two surfaces is the basis for every following analysis algorithm and is calculated in a preprocessing step, therefore it is easy replaceable in the implementation.

The trivial case to calculate the distance between two surfaces is if a total point correspondence exists, which means the vertices correspond over all¹ objects.

Those data sets are rare, because the data preprocessing is quite complicated, so they cannot be considered as the default case.

If no total correspondence is given the software is capable of determining point to point distances, using the surface distance (see 3.1) by default. Like previously discussed the surface distance is error-prone, however it proved to be sufficient for initial results.

¹ Correspondence between all reference- and test-objects across all samples

During the work a new distance measure has been presented [Get13], which provides very precise results and improved the analysis significantly. However, the calculation is quite complex, therefore the data sets must be prepared beforehand and loaded during run time.

The result of the calculation process is a list which holds the distance value for all vertices in every mesh. With this the data analysis processes can be set in motion.

4.3 Data processing

4.3.1 Principle component calculation

To get an initial impression of all the data sets at once, it is important to know which samples correlates with each other in terms of the local geometric differences between reference- and test-model.

The input for the principle component analysis is a matrix $M^{n \times m}$, which is defined as:

$$M = \begin{bmatrix} D_{0,0} & D_{1,0} & \cdots & D_{n,0} \\ D_{0,1} & D_{1,1} & \cdots & D_{n,1} \\ \vdots & \vdots & \ddots & \vdots \\ D_{0,m} & D_{1,m} & \cdots & D_{n,m} \end{bmatrix}$$

where $D_{i,j}$ describes the distance of the i-th vertex in the test-mesh of the j-th sample, n is the number of vertices in the test-mesh and m is the number of samples.

The according covariance-matrix $C^{n \times n}$ (Definition in Section 3.4) is a squared matrix with size n .

After the calculation of the eigenvalues and eigenvectors of Matrix C , the desired number of dimensions after the reduction has to be defined. I chose two dimensions because it can be accurately visualized in a two dimensional coordinate system.

To reduce the dimension, the two largest eigenvalues with the corresponding eigenvectors had to be found.

The two eigenvectors ($eigV_1, eigV_2$, with length n) will be used to create the *FeatureMatrix*:

$$\text{FeatureMatrix}^{n \times 2} = (eigV_1, eigV_2)$$

and the final data matrix FD will be calculated via:

$$FD^{2 \times n} = \text{FeatureMatrix}^t \times \bar{M}^t$$

where \bar{M} describes the mean corrected Matrix M (Definition in 3.4).

The final data Matrix FD contains 2 values for each data sample which represents their position in the principle component space.

Every sample which values are similar will have a high correspondence of distance values over the vertex IDs.

Usually in a training data set, one will expect that most of the samples will have a pattern of locations of bad segmented respectively good segmented regions, while some samples (outliers) will show completely different characteristics. In the principle component space the 'common' samples will be located around the zero point and the outliers will be on different locations, so they can easily be noticed.

For the evaluation of the segmentation both outliers and common samples could be interesting, because outliers provide much information about the conditions when the algorithm could fail and the common samples indicate the reoccurring pattern of low segmentation quality regions.

Another possibility is that the respective organ has two modalities, which means several samples show the same characteristic while the other group shows completely different.

In principle component space these two groups will be on separate locations and they can be easily distinguished.

The position in principle component space will be saved for each sample and used in the PCA-View (Section 4.4.4) .

4.3.2 Clustering

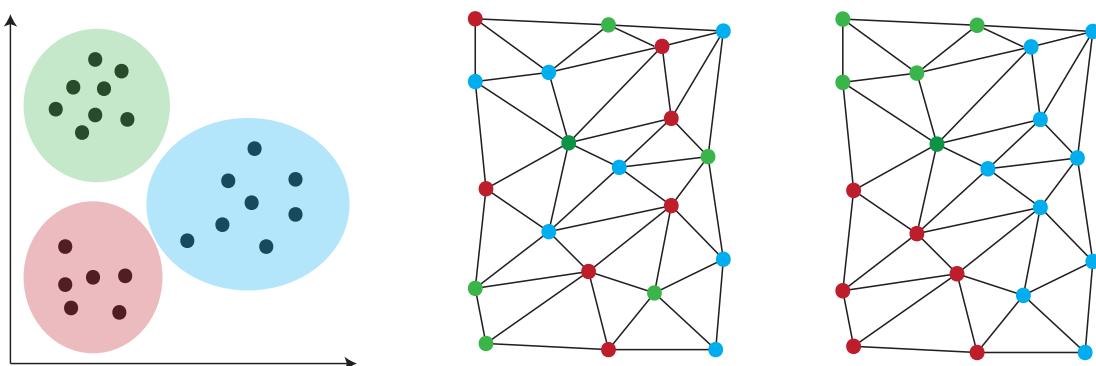


Figure 4.1: **Left image:** Clustering only in feature space. **Middle image:** The corresponding cluster assignment on the mesh, without spatial constraints **Right image:** The desired result. A clustering with spatial constraints, only neighboring vertices are qualified to form a cluster

To partition the data set in groups which consist of points with same characteristics (**local corresponding regions**) I decided to use a clustering method, but most common clustering

algorithms use the feature space for defining the clusters and are not capable of allowing spatial constraints. Spatial constraint means that two points are only qualified to form a cluster if they are neighboring vertices (two vertices which are connected with an edge) in the mesh.

The result of a standard clustering method would be groups of vertices which are geometrically not linked together but rather spread throughout the whole mesh (Figure: 4.1). To extract connected regions a new clustering algorithm has to be developed.

The new clustering method should be able to meet our requirements. Firstly I chose a common clustering method, which is well fit and then enhanced it.

DBSAN was not suited, because of the fact that it will not assign every point to a cluster and it needs appropriate initial parameters to work well. Especially the spatial constraint will work as a filter and will reduce the number of points in the reachability radius drastically, which means that even more data values will be marked as noise points.

It is also possible that wrong parameters will lead to clusters with very drift values

K-means has the same problem with the initial parameters. The solution to find the optimal initial values is a simulation with different parameters and a relative cluster evaluation. In case of K-means the computation would be of high complexity, because on the one hand each computation time of the algorithm would be significantly higher than usual, because of the spatial constraint, that has to be applied and on the other hand a high amount of clustering iterations has to be done to get the optimal combination of initial center position and cluster numbers.

On the contrary, hierarchical clustering algorithms do not need initial parameters. Only a proximity measure for feature distances and a linkage criterion has to be chosen. With the right combination, hierarchical clustering merges similar object very well.

In my approach I used four combinations of proximity measures and linkage criteria to calculate the new distances (Section:3.2.2.3),

- Chebyshev distance and centroid-linkage
- Chebyshev distance and complete-linkage
- Euclidean distance and centroid-linkage
- Euclidean distance and complete-linkage

After the hierarchy is build a threshold to form the final clusters has to be chosen (Section3.2.2.3).

The computation time of hierarchical clustering is higher than K-means, but once the hierarchy is build, the determination of the optimal threshold is merely a repeated tree traversal.

Therefore I chose hierarchical clustering as the basis for the new clustering algorithm and enhanced it with the spatial constraint.

The steps are defined as (see Figure 4.2):

4.3.2.1 Clustering algorithm

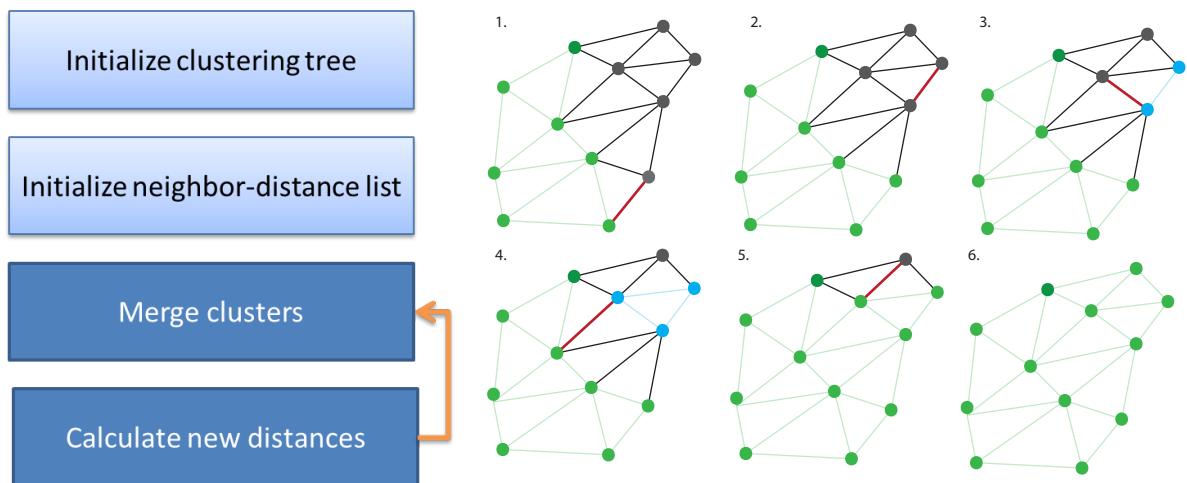


Figure 4.2: **Left image:** Basic steps of the clustering algorithm. **Right image:** Six iterations of the clustering algorithm. The green and blue points represent vertices in the mesh that are already connected in the hierarchy. The gray points show which points are free to connect. The red edge marks the connection which will be merged next, because of the lowest distance. Black edges are still in the neighbor-distance list, whereas the colored edges are already erased.

Step 1: Initialize clustering tree

The basic hierarchical clustering process was described in Section 3.2.2.3.

At the beginning of the new clustering algorithm the clustering tree has to be initialized. This binary tree consists of nodes, which can contain two child nodes (left and right child). There are three types of nodes: The leaf nodes form the basis of the tree and have no children.

Nodes with children are called parent nodes. Parent nodes form the connection inside the tree and each of them are children themselves, except for the root node, which is the top of the tree. All nodes are able to save a number as additional information. In the tree leaves this number is a vertex ID and in the parent nodes it represents the hierarchy level. The hierarchy level is zero at the beginning and will be incremented every iteration.

In my approach I used an agglomerative hierarchical clustering method, therefore at the beginning each vertex belongs to its own cluster. These initial clusters form the tree leaves (the vertexID is saved in the node) and represent the lowest level of the hierarchy.

Step 2: Initialize neighbor-distance list

With a basic clustering approach, it is possible that points, which are geometrically on different positions in the mesh, will be grouped together.

The result would be inappropriate for the user because the goal is to extract connected regions, thus it is obvious that only direct neighbors can form a cluster, so the amount of points which are eligible to merge together are limited by the geometric constraints of the mesh.

In the first iteration every vertex belongs to its own cluster, so the neighbor relationship of the clusters is defined by all the edges which connect every vertex.

A list of all neighboring clusters and the distances between the corresponding points in feature space has to be initialized. This list is called neighbor-distance list.

To find all cluster neighbors initially, I iterate over all edges, calculate the distances according to the chosen proximity measure and add the respective pair to the neighbor-distance list.

Step 3: Merging

Usually the aim of clustering algorithms is to maximize the compactness and the separation of all clusters (Definition:2.1.9 Section).

To achieve that, the two neighboring clusters with the smallest distance will be merged.

Merging in case of hierarchical cluster means that the two clusters will be connected in the clustering tree. The pair with the smallest distance in the neighbor-distance list will be merged. A new node is created with the two mentioned clusters as right respectively left child, plus the hierarchy level.

If two clusters (A and B) are merged, the neighbor-distance list can be used to find the new cluster neighbors. The edge that led to the merging of the clusters and other edges that might connect points inside the cluster as well will be erased from the neighbor-distance list.

The neighbors $N(C)$ of the new cluster C is the union of the neighbors from each cluster $N(A) \cup N(B) = N(C)$.

The neighbor-distance list shrinks every iteration, while the clustering tree grows.

Step 4: Calculate new Distances

The distances between clusters are calculated according to the chosen linkage criteria and proximity measure.

Each combination can provide different clustering results.

The results result of each combination will be discussed in the evaluation chapter (Chapter 5).

These five steps will repeat until all vertices are connected. An example of six iteration of the clustering algorithm is shown in Figure 4.2.

The result is a clustering tree, which can be used to define regions on the mesh which correspond over the defined criteria (proximity measure and linkage criteria).

4.3.3 Cluster quality evaluation

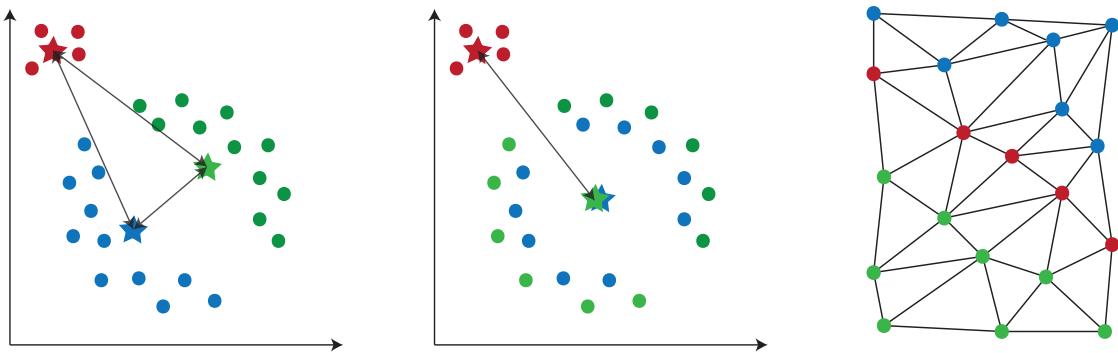


Figure 4.3: **Left image:** Shows a standard clustering in feature space, the clusters are arbitrary shaped. The stars represent the well separated cluster centroids. **Middle image:** The clustering with spatial constraints. The centroid of the blue and green clusters are nearly identical, therefore the separation is significantly worse then in the clustering in feature space. **Right image:** The mesh showing the spatial constraints.

After the clustering algorithm is finished and clustering tree is complete the usual procedure to find out the optimal cluster threshold in hierarchical clustering without spatial constraint is to sample the threshold from the basis of the tree to the top and calculate a chosen cluster validity index on every sample step.

The best result according to a certain threshold will define the optimal parameter combination.

If the clustering is enhanced with the spatial constraint, it is possible that the common indices produce inaccurately results or they cannot be calculated at all, because of the following reason:

The common goal of a data clustering is to maximize the compactness and separation of the clusters.

Most cluster validity indices calculate these two criteria in a specific way and put them in a relation to determine a value which defines the cluster quality.

Usually the separation is determined by accumulating the distances between each cluster center. The standard clustering methods (without spatial constraints) produce arbitrary shaped clusters, in which each cluster center is located in different positions in feature space, therefore all the clusters are naturally 'separate' (Figure 4.3left).

Whereas, if the spatial constraint is taken into account, it is possible that two clusters possess nearly the same characteristics, but are not merged together, because the values have different location on the mesh. Their cluster centers would be nearly identical; hence the distance would be near zero. The standard validity indices would be distorted (Figure 4.3middle).

With the standard cluster validity measures, small distances between clusters mean that the clusters are not well separated, but in our case it is anticipated that several cluster with same characteristics exist, if they are not neighbors.

To solve these problems I adapted a common cluster validity index to meet the requirements. I chose the SD index as basis, because in literature ([HBV01]) it showed better results for arbitrary shaped then the Dunn-based-indices.

I constrain the term for the total separation

$$Dis = \frac{\max_{i,j=1 \dots n_c} (||v_j - v_i||)}{\min_{i,j=1 \dots n_c} (||v_j - v_i||)} \sum_{k=1}^{n_c} \left(\sum_{j=1}^{n_c} ||v_j - v_i|| \right)^{-1}, i \neq j$$

with the adjacent matrix M of the cluster, which values $M_{i,j}$ are defined as:

$$M_{i,j} \begin{cases} 1 & \text{if Cluster } i \text{ and } j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases}$$

$$Dis' = \frac{\max_{i,j=1 \dots n_c} (M_{i,j} \cdot ||v_j - v_i||)}{\min_{i,j=1 \dots n_c} (M_{i,j} \cdot ||v_j - v_i||) |M_{i,j} \neq 0|} \sum_{k=1}^{n_c} \left(\sum_{j=1}^{n_c} M_{i,j} \cdot ||v_j - v_i|| \right)^{-1}, i \neq j$$

which means only the distance of adjacent clusters will be taken into account.

I kept the term for the average scattering:

$$Scatt = \frac{1}{n_c} \sum_{i=1}^{n_c} \frac{||\sigma(v_i)||}{||\sigma(x)||}$$

The altered validity index (SD^*) definition is still the same:

$$SD^* = \alpha' \cdot Scatt + Dis'$$

n_c	Number of clusters
v_i	Centroid of the i-th cluster
$\sigma(v_i)$	Variance of the i-th cluster
$\sigma(x)$	Variance of the whole Dataset
α'	Dis' of the maximum number of clusters

The enhanced SD* index is adapted to the spatial clustering method.

4.4 Visualization and interaction techniques

4.4.1 Interface

The basic interface of the system consists of four major parts:

1. The visualization window
2. The color legend for distance coloring (Section 4.4.2)
3. The sidebar, which adapts depending on the active visualization
4. A menu bar to load data and switch between views



Figure 4.4: Basic GUI of the System - a) Visualization window - b) Color legend for distance coloring - c) Removable sidebar d) Menu bar to load Data and switch between Views

4.4.2 Distance based coloring

As a first approach to visualize the distances between two meshes I assigned a color for each mesh-vertex which represents a specific distance.

To map each distance to a color I chose a bipolar color map (Figure:4.4b). Points which are inside the other mesh (under-segmentation) will create negative distance values and will be mapped to a bluish color whereas outside points (over-segmentation) will receive a reddish color tone. Zero distances are mapped neutral white (Figure 4.5).

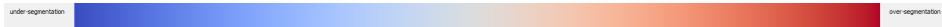


Figure 4.5: Bipolar color map

The greater the absolute distance value (negative or positive) the more intense the resulting color will be.

To map the color, the distances have to be normalized. There are several possibilities to normalize these values:

Local normalization calculates a maximum and minimum for each sample pair (Ground truth and segmentation) and maps the whole color scale to every mesh, which provides an overview of the worst segmented regions in each individual sample. The fact, that the distances of each individual sample are mapped to the same whole color map can mislead the user, because a “good segmented” sample will always look as bad as a “bad segmented” one (Figure 4.6a).

To resolve this problem, a global normalization can be used where the extreme values over all samples are measured and mapped to the color scale. This could lead to another problem, namely, due to outliers it is possible that the whole color scale will not be used and the visual representation for distance is lost (Figure 4.6b).

A basic approach to negate this effect is to use the 0.95 Quantile of all the distance values for the global maximum and the 0.05 Quantile for the minimum. It leads to a coloring, where single outliers will not have grave consequences (Figure:4.6c).

Note that none of these normalizations can qualify to be “the best”; it always depends on the use case and what the users intentions are.

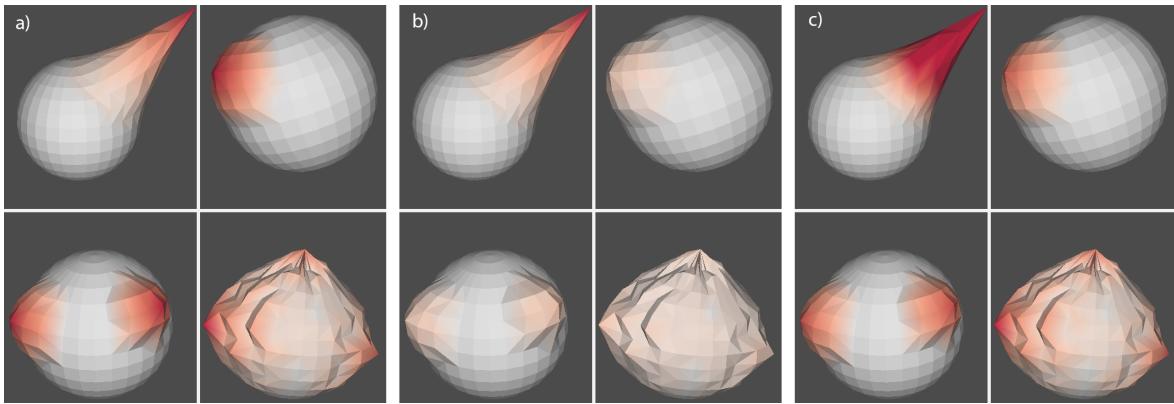


Figure 4.6: Three different distance colorings of 4 samples arranged in a grid: a) Local normalization
b) Global normalization c) Global normalization with Quantiles

4.4.2.1 Threshold based coloring

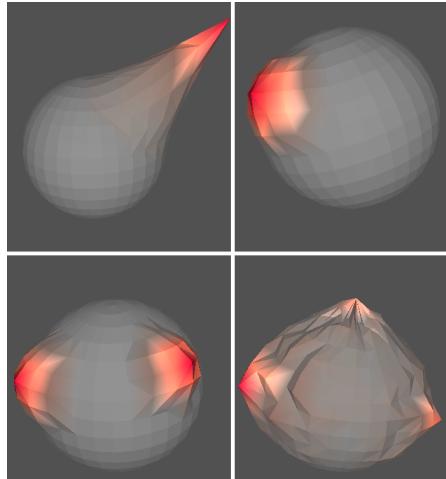


Figure 4.7: Threshold visualization in combination with distance coloring

The threshold- coloring visualization is a variant of the distance coloring, where an opacity value according to a distance threshold is assigned to each vertex.

With a slider the user generates a distance threshold and according to it all points with a lower distance will be faded out (low opacity), whereas all higher values will be highlighted (Figure 4.7). The advantage lies in the better perception of the geometric context. It is a more intuitive way to find reoccurring outstanding regions over every sample. The opacity according to the threshold can be combined with distance coloring and average distance coloring, which will be explained in the following section.

4.4.2.2 Average distance coloring

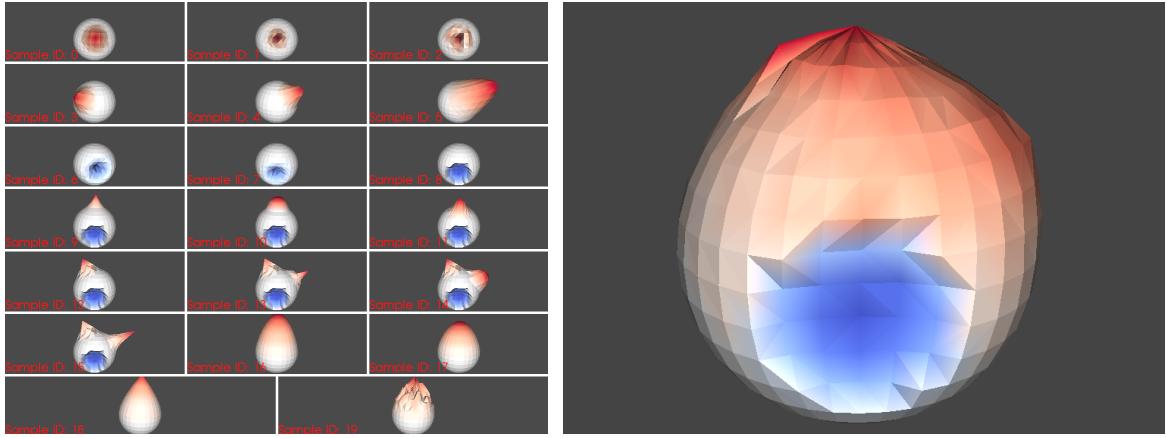


Figure 4.8: **Left image:** Grid visualization of 20 samples, **Right image:** The average distance coloring according to the 20 samples visualized on the average mesh.

If the sample number is too high it becomes difficult to identify reoccurring regions in multiple grid windows, therefore I created an alternative coloring on one single mesh which can be used as an representative for all other samples. The above defined coloring procedure for every vertex is the same, merely the average distances of all samples on a specific vertex is taken for the color assignment. The distance of a respective vertex a^k is defined as:

$$a^k = \frac{1}{n} \sum_{i=1}^n x_i^k$$

where x_i^k is the distance of the vertex k of sample i and n is the number of samples.

The coloring is applied to the vertices of the average mesh and is used in the clustering view, which will be explained in Section 4.4.5

4.4.3 Grid view



Figure 4.9: Grid view with 3x3 samples (Ground truth is green, auto-segmented red) - sidebar provides options like: show/hide mesh, activate distance coloring, change opacity, show global or local error and thresholding

The grid view is the most basic view of the data set. Every sample of reference- and test-mesh are rendered in one window and all samples are arranged in a grid (Figure 4.9). The user can choose between showing a single mesh (reference- or test-model) or both in each window using the sidebar.

For a large number of samples, each grid cell becomes too small to extract any detail of it, so the user is able to resize any cell to make it bigger for the analysis.

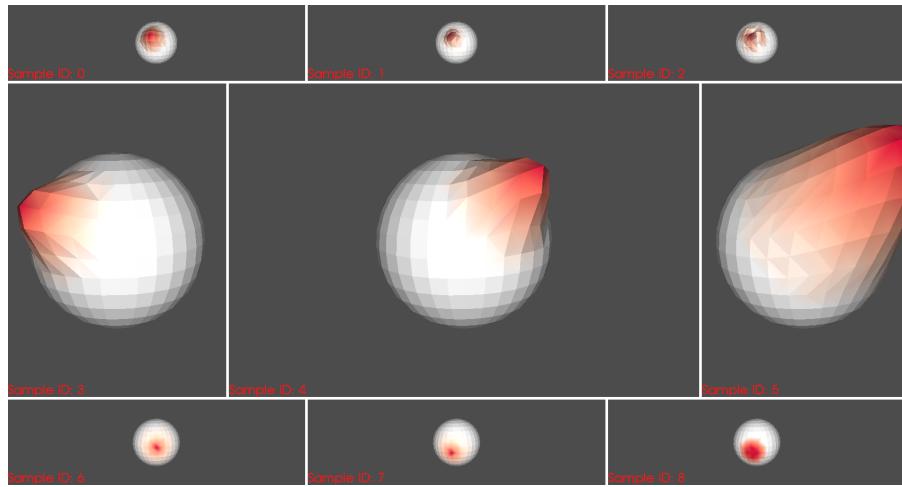


Figure 4.10: Resizable grid, to enable focused view of interesting object.

With the transparency slider the user is able to overlay both meshes and get an impression of the disparities.

The Cameras are synchronized in each window by default, so if you move the camera in one window all the others will follow, this is to keep the orientation of the sample, but the feature can be turned off for expert users.

The grid view can serve as a basic exploration of the geometry of the data set and conveys the impression of how each sample can differ from another.

To get a view of the distances between both meshes, the distance coloring can be activated. The user is able to move the threshold for threshold based coloring, using a slider in the sidebar.

4.4.4 PCA view

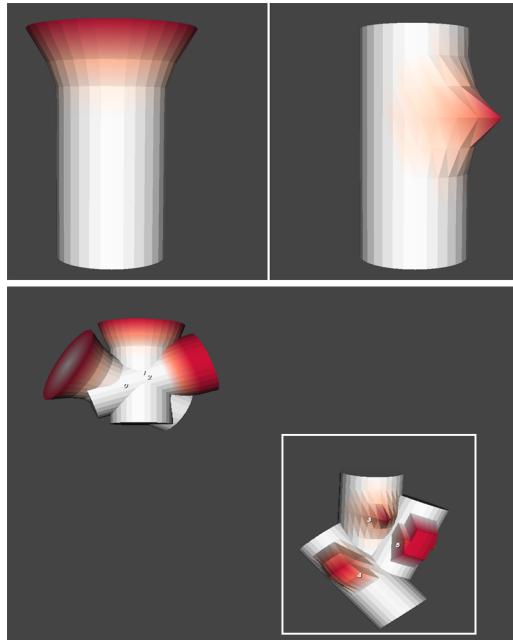


Figure 4.11: The **top picture** shows two modalities and on the bottom there are two groups with 3 samples each, which express the same characteristics as the two modalities. **Bottom picture:** The meshes moved in the principal component space. The two groups are perfectly separated. The right frame represents the selection box, for subset creation.

The PCA window provides a view, where all test-meshes are rendered in one window. They are arranged in a two dimensional coordinate system by moving the meshes according to the calculated PCA values (x-direction for the first principle component and y-direction for the second). If the meshes collide with each other the user can always scale the coordinate system, to move them farther away from each other. The distance coloring with each normalization can also be activated to keep the geometrical context of the interesting regions.

In Figure 4.11 I created a data set which consists of six meshes. Three meshes show a distinct pattern of bad segmented regions and the pattern of the other three differs greatly. The PCA is capable of separating these two modalities perfectly.

The user interactions in the PCA view aside from basic actions to examine the mesh and scale the coordinate system to prevent mesh collisions are to choose the respective groups or outliers (with a box selection in the rendering window, Figure [fig:PCA-two-modalities]bottom) and analyze them separately.

Once a subset of samples is selected, the user can visualize the respective samples in the grid view. This subset will also be used for further processing (clustering, average mesh

coloring). Like previously stated, a well selected subset of samples can have a huge impact on the cluster quality.

The analysis of the subset enables a finer and more precise way to extract the regions of interest.

Note that the Principle Components are calculated on the basis of the features, hence the distance between reference- and test-mesh, which does not necessarily mean the difference is visible in the mesh. It is possible that only the reference-mesh is geometrically standing out, and the test-object does not reflect these geometric characteristics (due to the constraints of the SSM) but the features do.

4.4.5 Clustering view

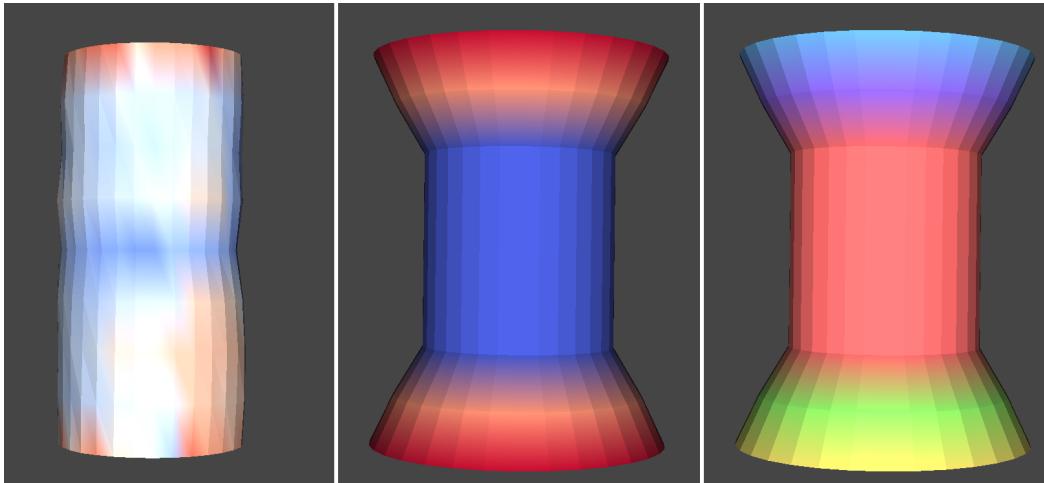


Figure 4.12: The clustering view consists of three widgets. The left one contains the average mesh and coloring over all samples, the middle one contains the average mesh and coloring over a subset of samples (if a selection exists) and the right one is a result of a clustering over the same subset, represented on the subset average mesh .

To visualize global information about (like clustering result or average distance coloring) all the meshes it is helpful to calculate an average mesh to find a representative geometric structure to work on. In this work, if we talk about average mesh, it is the mean of the test-meshes of all data samples, except if a sample subset was selected via PCA. Note that if the number of samples to build the mean is too low, the result becomes vulnerable to outliers and to point correspondence problems. The more samples the smoother the mesh will become.

The view consists of three rendering widgets each containing one mesh: One with a mean mesh with average coloring of all data samples, the second one with a mean mesh and average coloring of a subset of samples (if a selection exists) and the last one contains an average mesh for the clustering representation (Figure 4.12).

After the clustering of the samples is finished and the clustering-trees are available the user is free to choose one of the cluster modalities and start with the analysis.

The first step to get reasonable results from hierarchical clustering is the definition of a threshold to cut the clustering tree and form the clusters. It is possible to move the threshold from the top of the tree (one cluster) to the bottom of the tree (number of vertices = number of clusters) via a slider. After each vertex is assigned to a cluster, the visualization of the clusters can be initiated.

To keep the geometrical context, I decided to visualize the clustering result directly on the mesh and because the clustering represents the characteristics of every sample, I chose the average mesh as representative object.

To distinguish the different clusters I decided to assign a color to every cluster (Figure 4.12right) and every vertex which is a member of the respective cluster will receive the same color. To select a color I chose the HSV-color-space and assigned an angle according to the number of clusters, which means every cluster will receive a different color (Figure 4.13). However, if the number of clusters is too high (>50) several cluster-colors will be visibly identical. The cluster color is not only relevant for the geometrical representation, but also for the cluster analysis, which will be discussed in the next section.

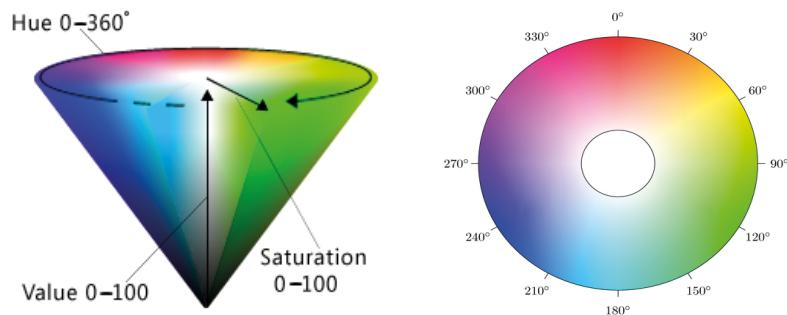


Figure 4.13: The left image shows the HSV-color-space and the right image is a top view of the cone. Different angles will result in different colors, but similar angles will be nearly identical.

To compare the average coloring to the clustering result, I additionally extracted the cluster borders algorithm and offered a combined rendering of distance coloring and clustering. The borders are extracted via a marching squares algorithm, which creates lines segment on the mesh according to the pattern shown in Figure 4.14.

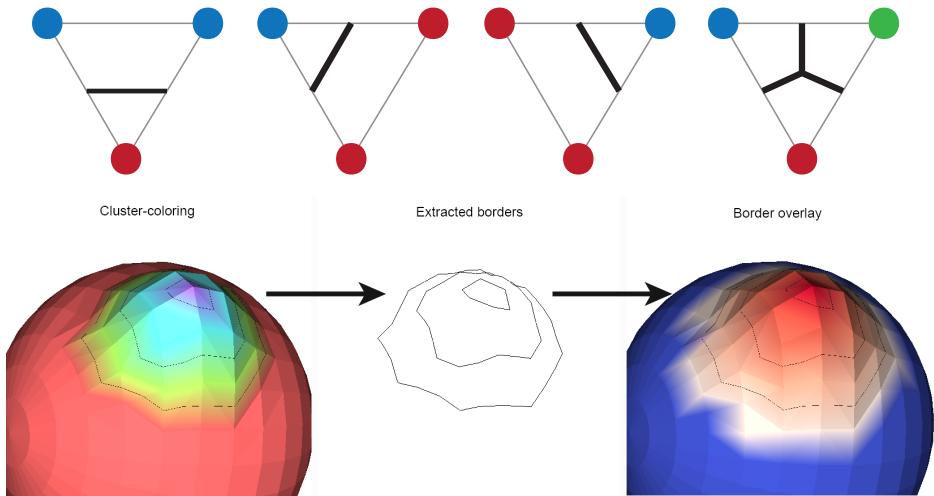


Figure 4.14: The top row shows the marching square algorithm to create the line segments for every triangle of the mesh. Different colors represent the vertexes cluster membership. The bottom row shows how the clustering and the average distance coloring get combined.

Aside from the typical interactions for mesh representation, the user is able to highlight the clusters, by clicking on the respective region on the mesh (Figure 4.15).

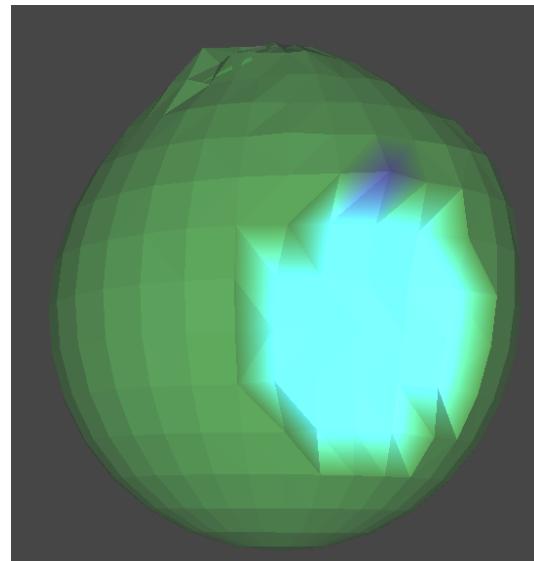


Figure 4.15: Clustering mesh: The blue cluster was highlighted

4.4.6 Cluster analysis views

The cluster analysis view consists of two dimensional plots, but is closely connected to the three dimensional cluster-view.

With the combination of these visualization and interactions (Figure 4.19) it is possible to evaluate all samples from a basic overview to the point of a detailed analysis of the distance values .

4.4.6.1 Cluster overview plot

In the cluster analysis view I created a plot which offers a basic overview of the clusters characteristics. Each cluster is represented by a colored (corresponding to the previously assigned color) circle which is arranged in a 2-Dimensional graph which X-axis describes the average distance of the vertices in the respective cluster and which Y-axis stands for the number of vertices in the group (Figure 4.16). The Y-axis is in logarithmic scale, because in most cases of the range of the maximum and minimum cluster size is too large to scale it linearly.

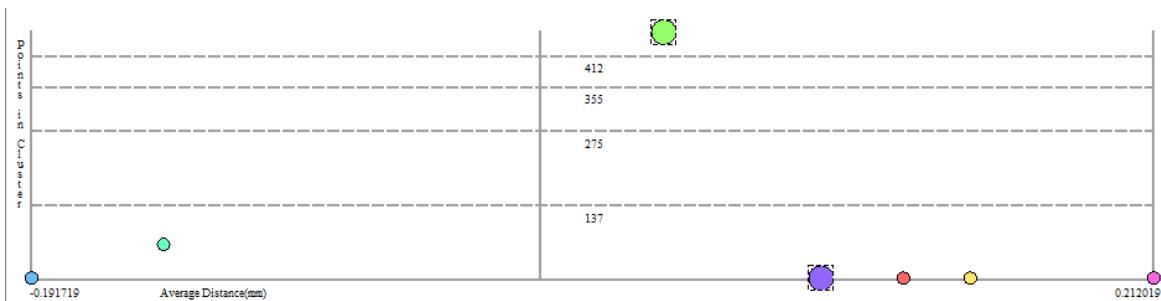


Figure 4.16: Cluster overview plot. Visualization of seven clusters. Two clusters are selected.

The interaction between the cluster overview plot and the mesh-representation is bidirectional, it is possible to select representative circle in the plot and highlight the according region in the mesh and it is possible to directly chose the region on the mesh and see the corresponding selection in the plot. The user is also able to select multiple clusters to highlight.

Every time a cluster is highlighted the according parallel coordinate plot is rendered.

4.4.6.2 Parallel coordinate plot

After the clusters are visualized, the characteristics which lead to the clusters have to be analyzed. I chose parallel coordinates for that matter, because it provides a good overview of the data, for a small set of dimensions.

Each parallel axis stands for a dimension, which in turn represents each sample. The data values in each dimension are the absolute distances between reference- and test-model of the points in the respective cluster.

Because the scale of the distances does not change over the samples, no scaling or normalization is needed, which makes the comparison of data values a lot easier. The colors of the visualized data values, match the colors of the previously assigned cluster colors as well.

The user is able to customize the parallel coordinate graph by ordering the axes and showing/hiding the different data representations.

To prevent over-plotting it is possible to show or hide the 0.75/0.25- quartiles, the absolute values, the mean or median (Figure 4.17).

The ordering of the dimensions can become a problem, especially when comparing three or more cluster. The sorting heuristics can easily become complex and the result is not guaranteed to be good, therefore I ordered the axes according to chosen characteristic (Figure 4.17).

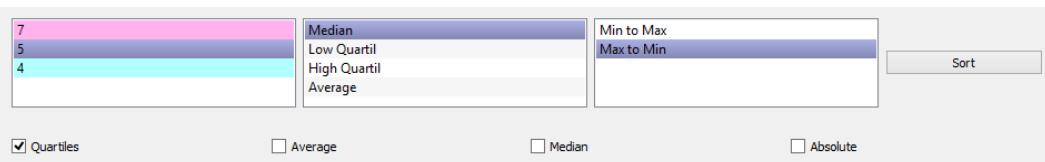


Figure 4.17: Parallel coordinate plot customization. The user is able to order the axis according to the chosen characteristics of the selected cluster (Cluster ID 5 is selected in this example) and it is possible to show or hide different representation of the values (0.75/0.25-quartile, average, median, absolute values)

For example, given a chosen cluster the user is able to order the axes from low to high or vice versa according to the mean of the data values in the cluster (Figure 4.18). I tested this ordering style with 1-30 samples and it proved be suitable to compare several clusters.

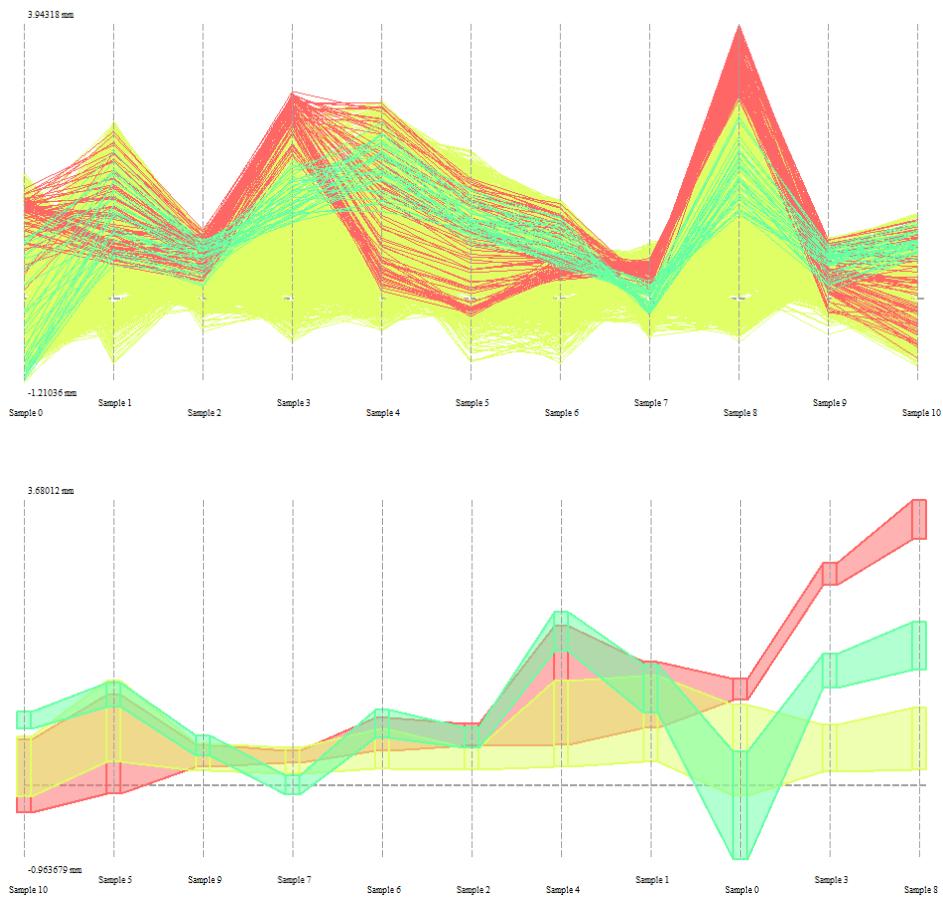


Figure 4.18: Visualization of three clusters using parallel coordinates. The top image shows an unsorted plot, with absolute values, whereas the axes in the bottom image are sorted according to the red clusters average distance and the values are represented as the quartiles.

To optimize the visualization, the horizontal zero axis, which represents a perfect correspondence between the two meshes (distance $\triangleq 0$), moves according to the minimum and maximum value of all visualized data values.

The combination of the mesh representation, the cluster overview plot and the parallel coordinates provides the most detailed visualization of corresponding regions (Figure 4.19).

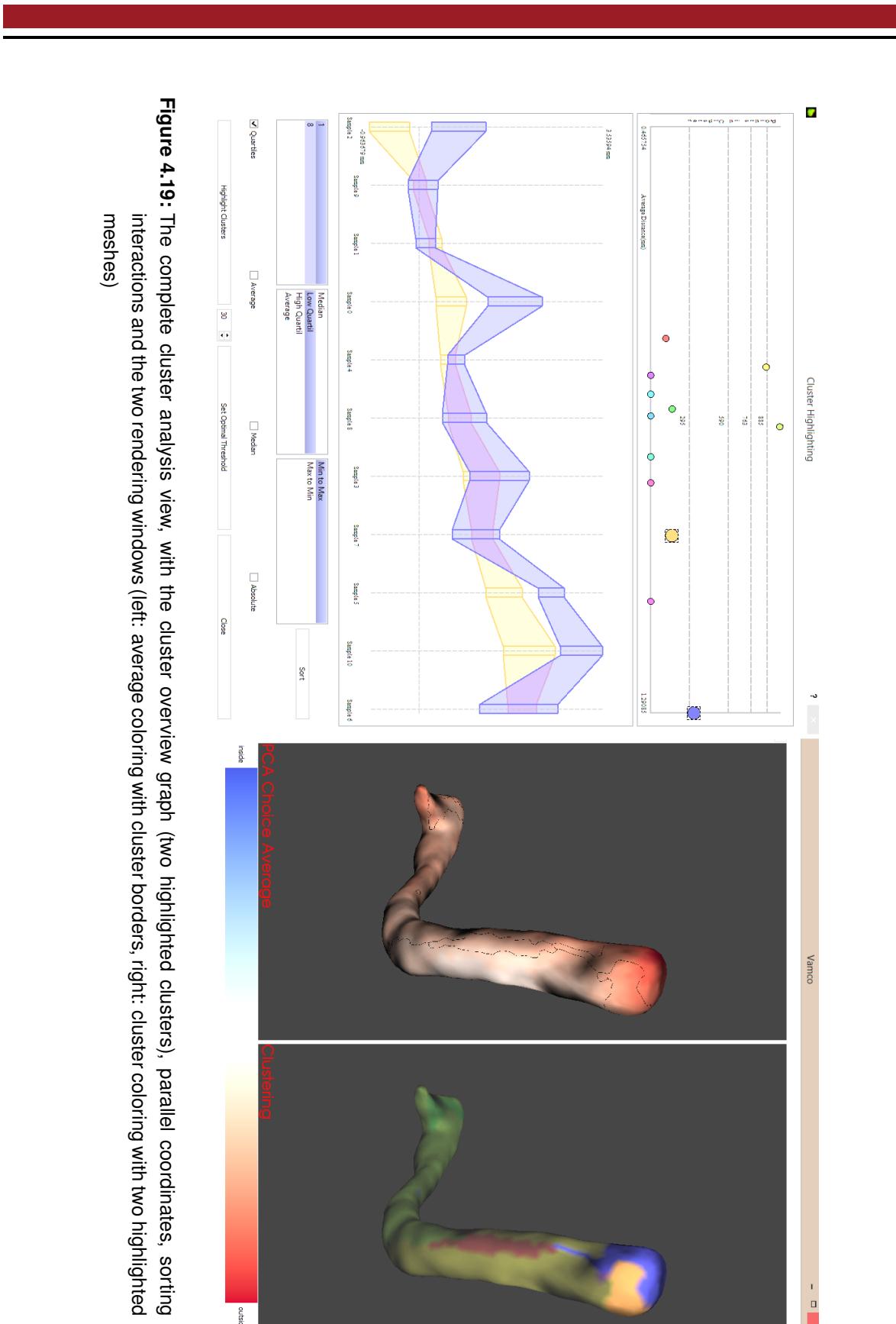


Figure 4.19: The complete cluster analysis view, with the cluster overview graph (two highlighted clusters), parallel coordinates, sorting interactions and the two rendering windows (left: average coloring with cluster borders, right: cluster coloring with two highlighted meshes)

4.4.6.3 SD*-index simulation plot

The user is free to define any desired threshold, but the cluster quality is strongly influenced by this decision.

To help defining a reasonable threshold the user can run a cluster simulation, which calculates the previously mentioned SD*-validity for different thresholds. The result is shown in a 2-Dimensional plot which arrange the SD*-index to the number of resulting clusters (Figure 4.20). The actual threshold is marked with a vertical line in the plot and if the user moves the threshold, the line moves in the graph.

Usually the 'knee'² of the graph or the minimum provides a hint for the location of the optimal threshold.

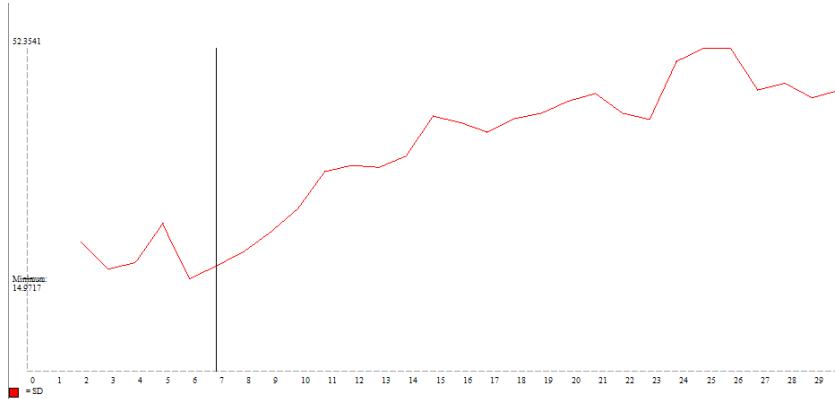


Figure 4.20: SD*-validity index plot: The X-Axis describes the number of clusters and the Y-Axis defines the range of the SD*-index. The vertical line in the plot represents the actual clustering threshold

² significant local change of values

4.5 Implementation details

The system is implemented in C++ with two additional external libraries, Kitware's Visualization Toolkit (vtk) [Kit13] and Qt - Framework [Dig13]. The Visualization Toolkit is the core in the software and a common tool for every data rendering and mesh representation. It provides methods for calculating eigenvalues and eigenvectors, which I utilized for the PCA calculation. It is embedded in the Qt- Framework, which is used for every GUI interaction in the software. It is also used for generating the cluster validation plots, because the graphical elements can handle millions of objects and interactions with reasonable performance. Programming with the combination of those two dependencies is very comfortable and works perfectly.

5 Evaluation

In this chapter I will show the evaluation of my system.

The process is composed of two parts:

The first step is the evaluation of the new clustering algorithm using a synthetic data set. The goal is to demonstrate the behavior of the clustering algorithm with the different combinations of proximity measure and linkage criteria.

The second section presents an explorative analysis of a liver data set with the proposed system.

5.1 Synthetic data: Clustering Evaluation

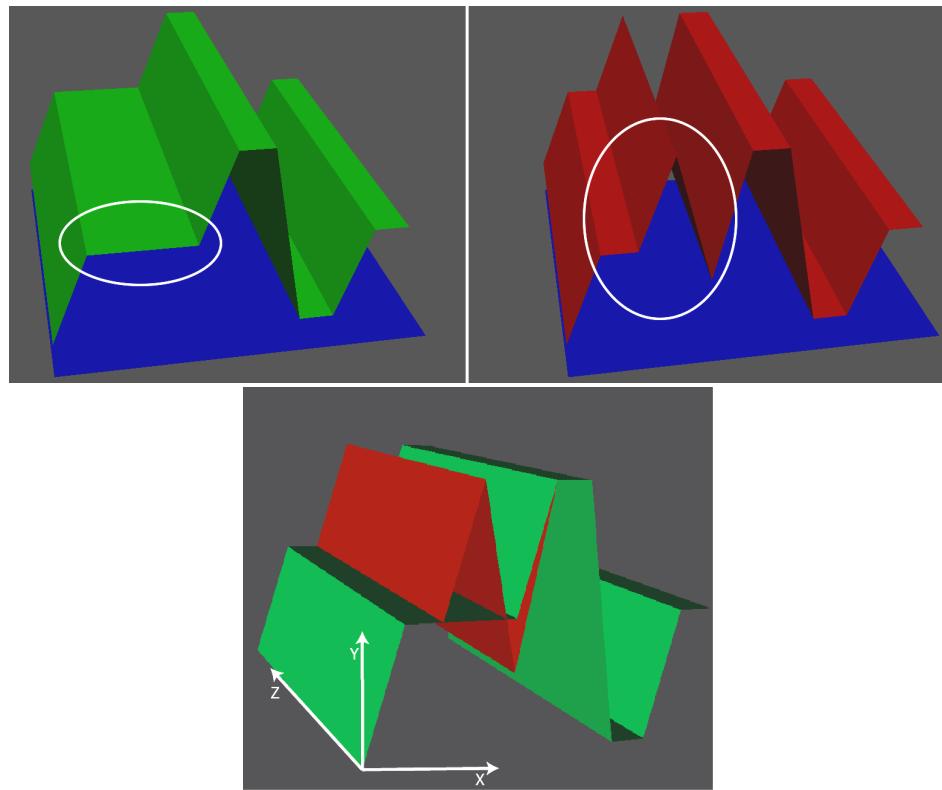


Figure 5.1: Top row: The two test planes to evaluate the clustering algorithms (red and green). The local difference of is marked. The blue plane represents the reference- mesh. Bottom row: The two test planes with the coordinate system.

To validate and analyze the behavior of the four different combinations of the new clustering method I generated a synthetic data set (Figure 5.1) with a simple shape and visualized the resulting trees. I chose a planar mesh, because it is easier to visualize and see the local geometric differences.

The specialty of the two planes is, that they locally differ the location marked in Figure 5.1. The reference-mesh is a plane in the x,z-space, so the distances (features) of all pointsof the test-meshes are their y-coordinates. Another specialty is that the features of the vertices do not change in z-direction so for the clustering process it is only interesting how the features change in x,y-space.

Figure 5.2b shows a view of the x,y-plane of the two test- meshes (Figure 5.1bottom), where they appear as lines. For easy orientation, I named the edge points of the meshes.

Intuitively it is not clear how the points will be clustered.

Therefore I visualized the resulting cluster dendrogram (Figure 5.2a) for clustering with Euclidean-distance and centroid linkage and Euclidean-distance and complete linkage.

At first sight the trees look similar, but on hierarchy level 4 and 5 the resulting clusters differ from each other, as vID 8, 9 and 10, 11 will belong to the same cluster with centroid linkage, but will be in separate ones with complete linkage (Figure 5.3 a), b)).

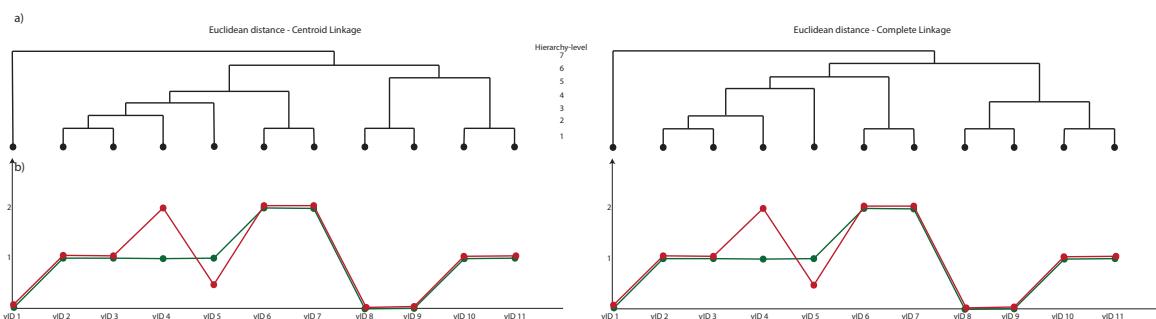


Figure 5.2: The graph (b) shows a view of the x,y- plane of the two test- meshes. The top line (a) shows the resulting clustering tree of the new clustering method with Euclidean distance and centroid-linkage criteria (**left**) and with Euclidean distance and complete-linkage (**right**).



Figure 5.3: The resulting clusters of a threshold set to hierarchy level 5 (a) and 4 (b) with the two mentioned settings. **Left:** Euclidean distance and centroid-linkage. **Right:** Euclidean distance and complete-linkage

In Table I calculated the average scattering, which represents the cluster-compactness, the total cluster separation and the according SD*-validity index (Definitions in Section 4.3.3).

The average scattering defines how similar the data values of each respective cluster are. A lower value means that the clusters are dense, which is the common goal in data clustering.

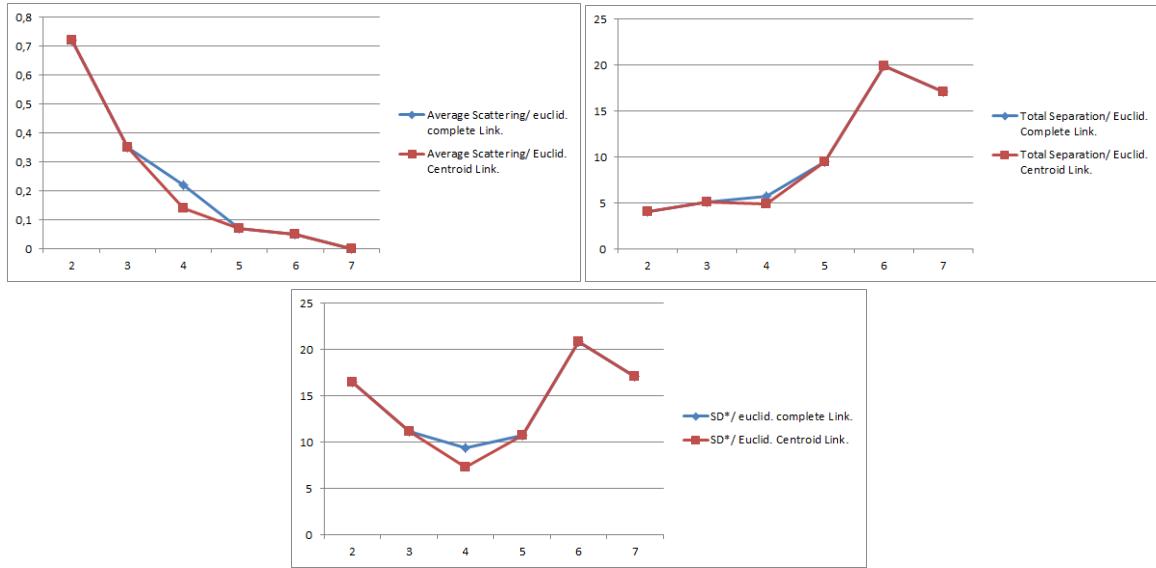
On the other hand, the values in a respective cluster should significantly differ from the values in other cluster. The total separation represents how well separated the clusters are. A lower value means a better separation.

The cluster separation and compactness differs at the partitioning with 4 respectively 5 clusters. The compactness of the centroid-linkage method is higher than with complete linkage.

The reason for this is, that the clustering with the complete-linkage criteria tends to form tighter clusters, but is sensible to outliers. As vID 4 and 5 are both outliers (in the red test-mesh), the cluster quality is lower.

Number of clusters	2	3	4	5	6	7
Average Scattering/ Euclid. complete Link.	0.72	0.35	0.22	0.07	0.05	0
Average Scattering/ Euclid. Centroid Link.	0.72	0.35	0.14	0.07	0.05	0
Total Separation/ Euclid. Complete Link.	4.11	5.07	5.73	9.45	19.95	17.09
Total Separation/ Euclid. Centroid Link.	4.11	5.07	4.9	9.45	19.95	17.09
SD*/ Euclid. complete Link.	16.45	11.12	9.44	10.73	19.95	17.09
SD*/ Euclid. Centroid Link.	16.45	11.12	7.32	10.73	19.95	17.09

Table 5.1: The table shows the calculated values of average scattering, total separation and SD*-validity with different thresholds.



The optimal threshold for both settings is 4 after the SD^* -validity index, but because of the mentioned reasons, the quality of the centroid-linkage method is better. The results on the average mesh is shown in Figure 5.4, both possibilities with Euclidean distance create reasonable results.

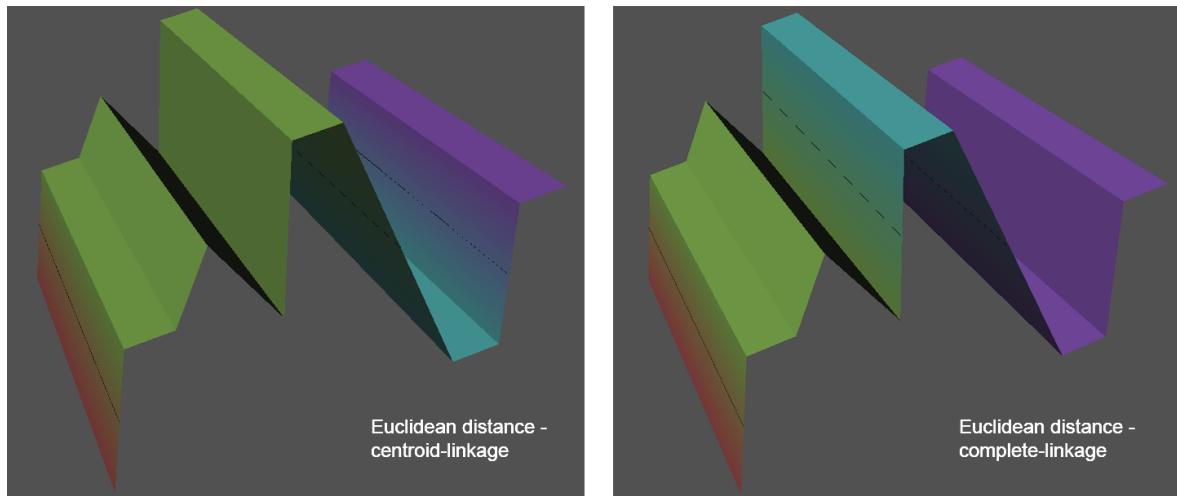


Figure 5.4: Final clustering result of the example plane with two different settings and the same threshold (4 clusters)

If the Chebyshev distance is chosen as proximity measure for this particular example the results will be the same, but to display the disadvantages of the Chebyshev distance I extended the test data set:

This time I took both of the example planes times 10 and added one outlier plane (which is shown in Figure 5.5 top), so the number of samples is 21.

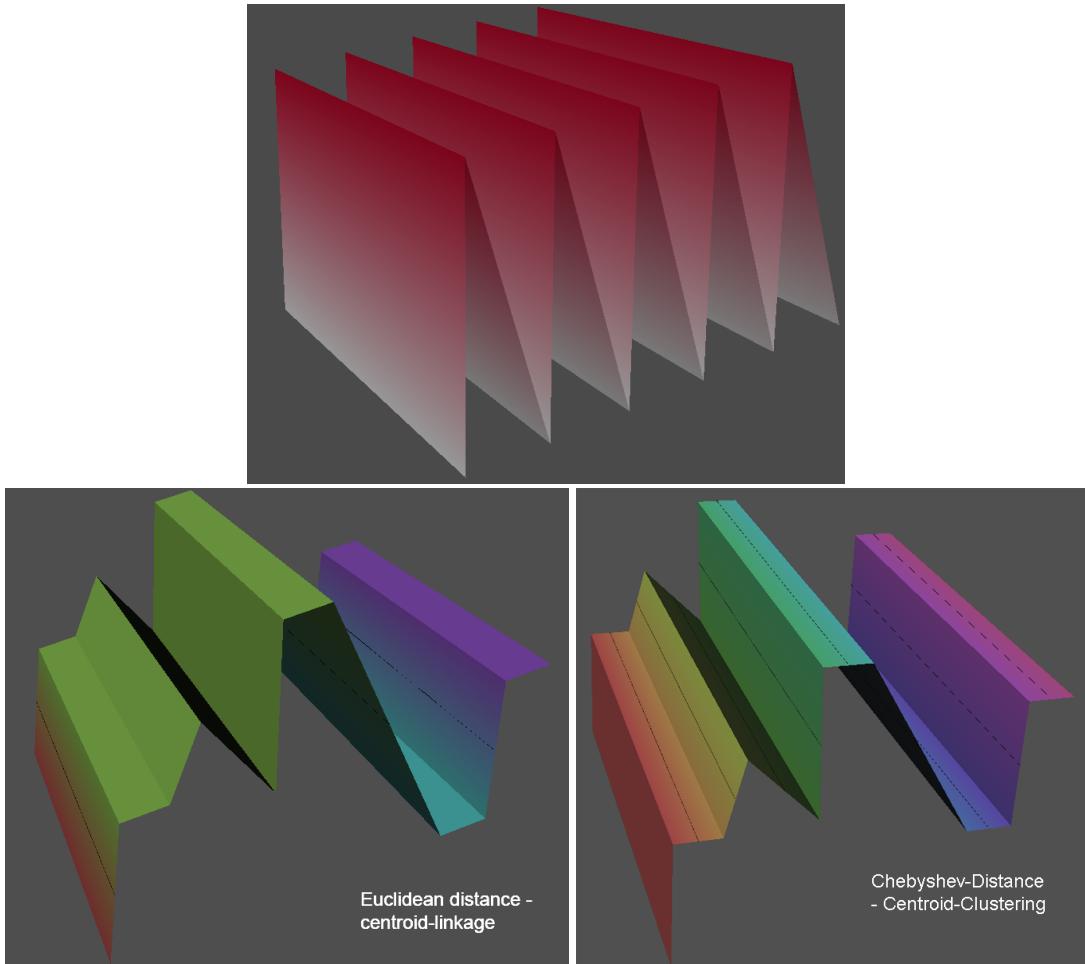


Figure 5.5: **Top row:** Example outlier plane for testing, **Bottom row:** The clustering result based on Euclidean-distance and Chebyshev-proximity measure with their ideal thresholds

The clusterings based on the Euclidean distance proximity measures produced the same clusters as in the prior examples, because the majority of the data showed significant characteristics. The Chebyshev-distance clusterings on the other hand will produce completely different clusters (Figure 5.5 right). The clusters completely correspond to the outlier plane, because the Chebyshev-distance takes the maximum distance of a single feature as feature-vector distance.

The Chebyshev- proximity measure proved to be too sensible to outliers in practice and is therefore too unstable for knowledge discovery.

To get stable results I recommend using the Euclidean distance based clusterings, because they reflect the characteristics of the whole data set and do not get heavily influenced by outliers, like Chebyshev-distance clusterings.

Hence, I tend to use the centroid-linkage, because the cluster quality is usually higher than complete-linkage clusterings.

5.2 Use case: Liver segmentation

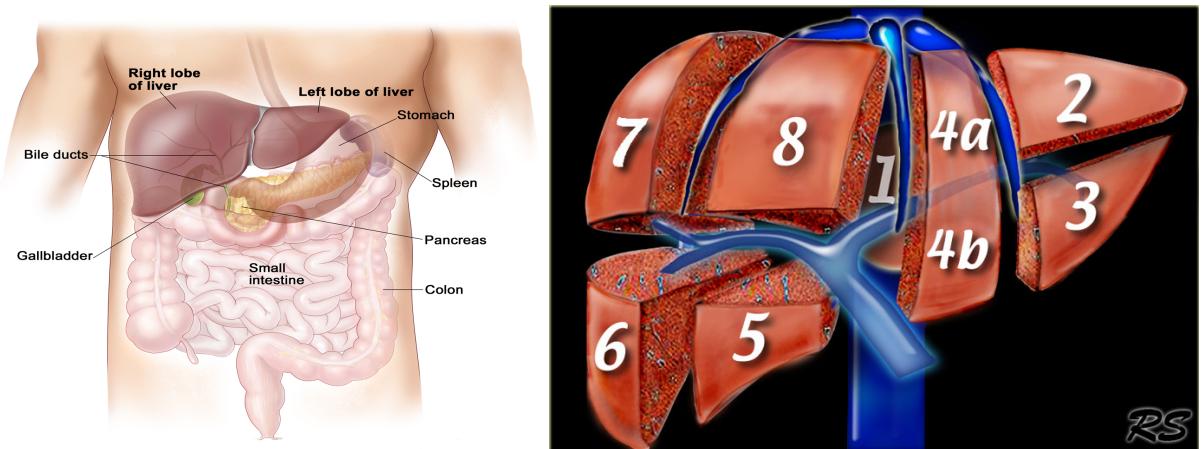


Figure 5.6: **Left image:** Location of the Liver in the human body, **Right image:** Segmental anatomy according to C. Couinaud (1957) classification [Smi06]

In this section I will evaluate how and if the developed system is capable of analyzing the local correspondence in segmentation quality.

Therefore I will analyze the results of an automatic medical image segmentation. The segmentation algorithm was developed by Kirschner et al [Kir13] and is based on an active shape model approach with enhanced CT- scans.

The algorithm was applied on 20 contrast enhanced CT scans of thorax and abdomen to segment the liver. The data set¹ has been made by the IRCCAD research institute against digestive cancer and also includes expert segmentations. The distances between expert- and automatic-segmentation were calculated with the distance measurement described in [Get13].

Figure 5.6 shows where the liver is located in the human body and gives an overview of the major anatomical landmarks, to get a basic orientation of the organ.

5.2.1 Explorative evaluation

Grid view

To get an overview of the data set, it is beneficial to take a look at all 20 samples in the grid view. With the local distance coloring, the areas of over- and under-segmentation of each individual sample become visible. At first sight the regions of bad segmentation seem to have little to no correspondence over the data samples. With the right color threshold, a

¹ which is available under <http://www.ircad.fr/softwares/3Dircadb/3Dircadb.php?lng=en>

5 Evaluation

consequent under-segmentation of the top part of the left liver lobe (segment two) becomes visible, only sample 6 and 7 show over-segmentation in this area (Figure 5.7).

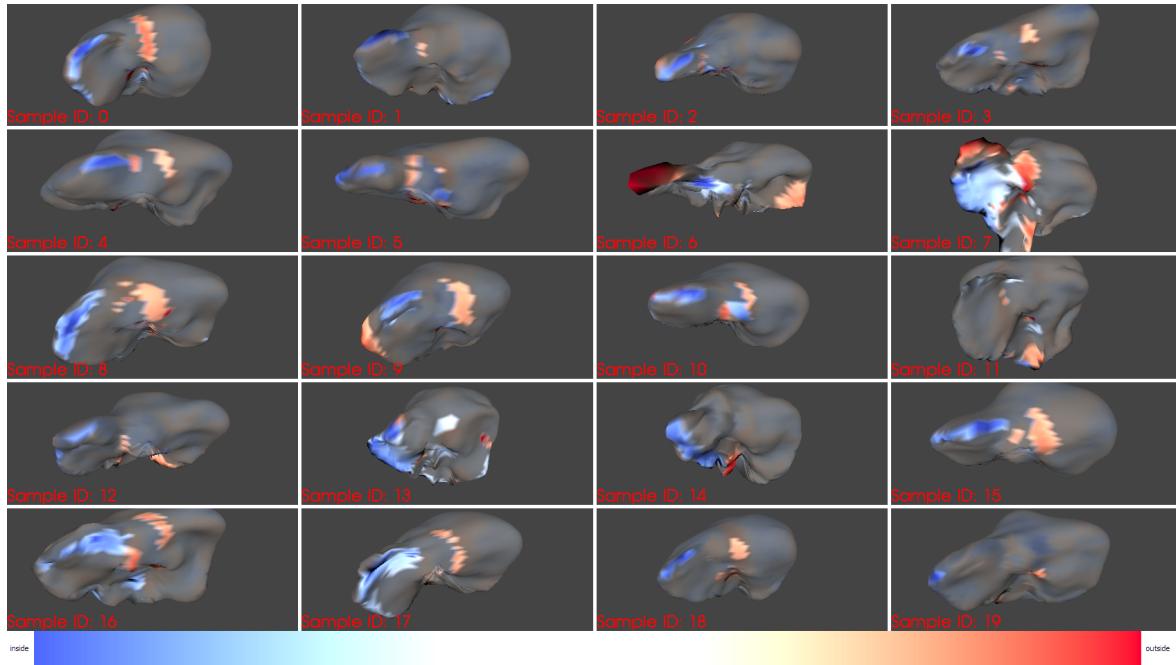


Figure 5.7: Grid View with threshold and distance coloring (blue describes under-segmentation and red over-segmentation) of the automatic liver segmentation from the IRCAD Data set. The top part of left lobes is under-segmented in the majority of the samples. The global threshold is set to highlight the vertices which belong to the 10% with the lowest segmentation quality (over- and under-segmentation).

The global distance coloring provides the information, that the majority of the data is equally bad respectively good segmented, only sample 6 shows remarkable regions, therefore it can be assumed that liver segmentation number 6 is an outlier. To confirm this assumption I continued the exploration with the PCA view.

PCA

To identify outliers and to reveal the distribution of the data, it is recommended to consider the PCA view (Figure 5.8).

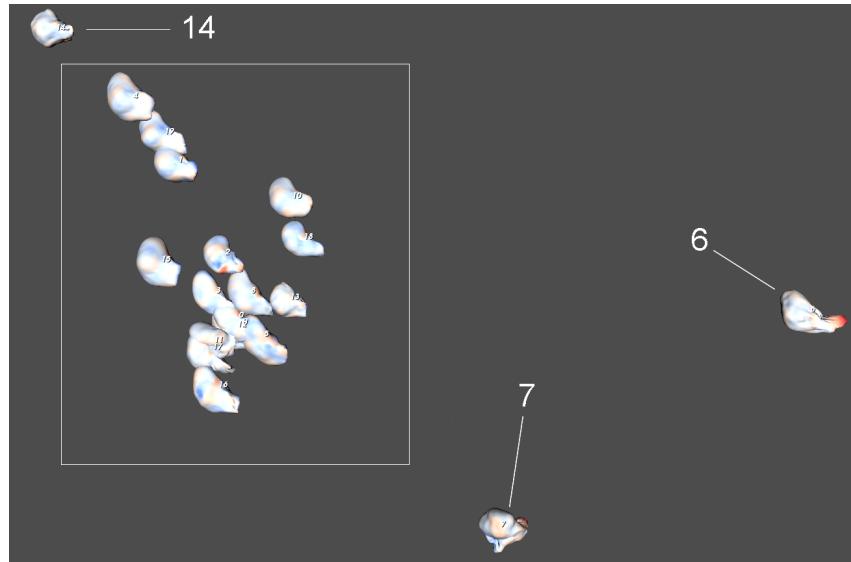


Figure 5.8: PCA view of the whole data set. Sample 6,7 and 14 are easy to identify as outliers. The square shows the selection of the majority of samples for further processing.

In consideration of this visualization the previous assumptions can be confirmed. Samples 6, 7 and 14 do not correlate with the rest of the data set. It is also visible that the three outliers are not correlated to each other, because they are located far away from each other.

The outliers can be visualized and analyzed individually with the grid view or they can be exported and analyzed with the MITK framework.

For the further progression of the evaluation, these outliers had to be excluded, because their regions of interest do not correspond with these of the rest of the data samples, therefore they will lower the quality of the clustering.

I selected the majority of the data (Figure 5.8) and started the calculation of the clustering, based on the selection.

Average distance and clustering analysis

The average distance coloring provides an overview of the frequently under- respectively over-segmented areas and shows if the regions are equally or irregular distant from ground truth.

In this use case it is safe to say, that the algorithm tends to under-segmentation, because the largest part of the average sample surface is covered by blue areas.

5 Evaluation

Also the previous observation, that the top area of the left lobes is constantly under-segmented can be confirmed. Additionally the area in the middle (segment one, where the inferia vena cava is located) is standing out with an average over-segmentation.

The remaining regions represent the overall well segmented areas of the liver. Their average distance to ground truth is very close to zero.

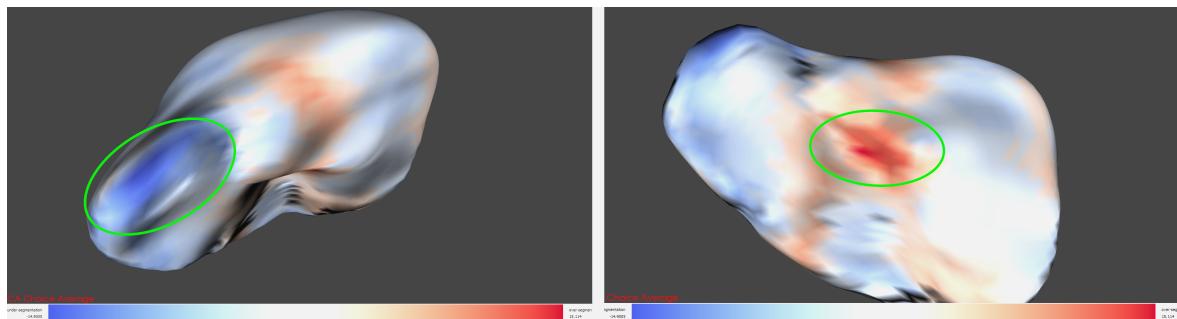


Figure 5.9: The average coloring visualization of the selection from two angles. The green ellipse marks the most significant regions.

The last step is to extract regions using the clustering method and with the help of the parallel coordinate plot I can provide the most detailed information about the regions of interests.

If the clusterings of the chosen subset of samples is complete a proximity measure and linkage criteria can be chosen. As proximity measure I chose Euclidean distance, because as previously stated, the Chebyshev- proximity measure is too sensible to outliers and as linkage criteria I selected centroid-linkage because the resulting clusters had a higher cluster quality according to the SD*-validity index.

The SD* cluster simulation with the centroid-based linkage is an example where the proposed optimal threshold is not suitable for the cluster analysis. The minimum of the SD*-validity index lies at a threshold which forms merely three clusters (Figure 5.10a), two of them containing one vertex each and the other one containing all the remaining points. Since a region consists of more than one point, the result is not appropriate for the analysis.

To get a relevant result I chose the second best threshold (Figure 5.10b). The result is shown in Figure 5.11.

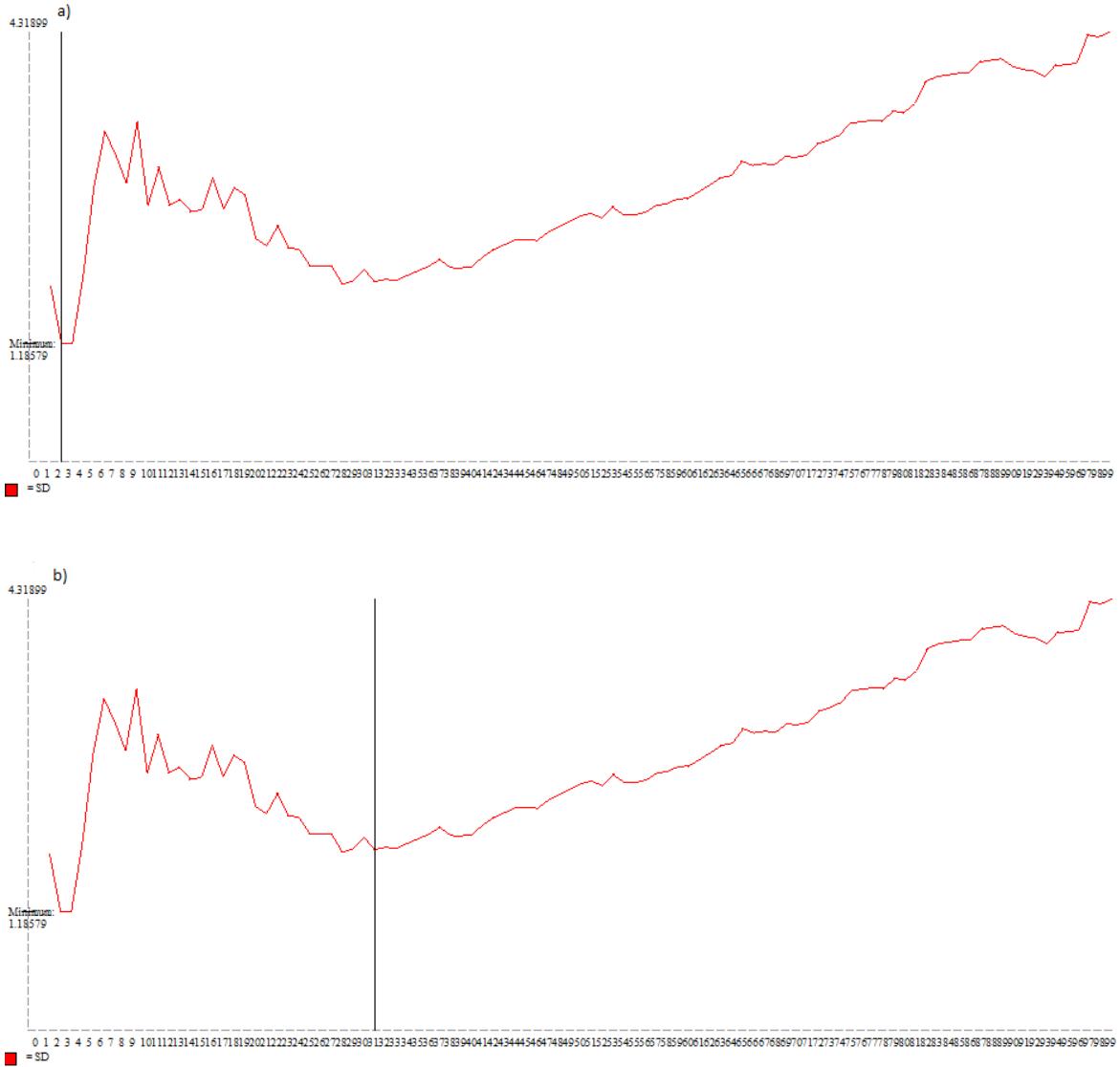


Figure 5.10: The SD*-index plot with two different threshold. **a:** The threshold is positioned at the minimum. The according clustering consists of two clusters **b:** The threshold is positioned at a local minimum, the according clustering consist of 29 clusters.

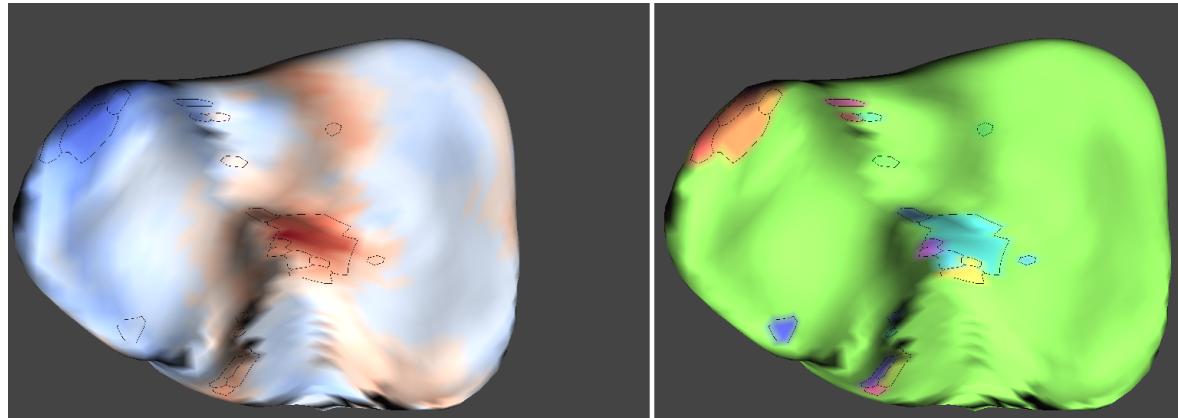


Figure 5.11: The clustering result of the centroid-linkage based method

The clusters correspond well with the average coloring.

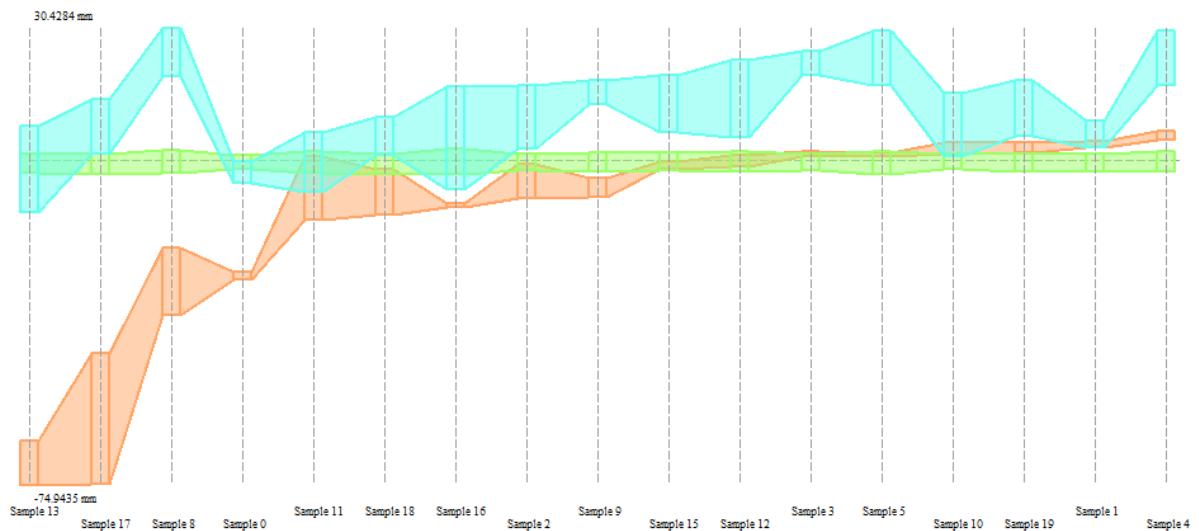


Figure 5.12: Three clusters visualized with the parallel coordinate plot. The orange values represents the cluster in the middle of the organ (segment one), the blue values stand for the top area of the left lobe (segment two) and the green lines represent the majority of the remaining area. The axes are ordered according to the orange cluster.

For further analysis I used parallel coordinate plot (Figure 5.12). Comparing the two clusters with the lowest segmentation quality (the area at the top left and in the middle: segment one and two after Couinaud classification) to the largest region (the green area) showed that the largest part of the liver is almost perfectly segmented.

On the other hand segment two is frequently under-segmented but three samples (13, 17, 8) show exceptional low segmentation quality (orange cluster), whereas the area in the middle (blue cluster) is in most samples equally over-segmented.

It is possible to create even more clusters by moving the threshold to get a finer division of the surface, however in this case the cluster separation will decrease.

The use case example showed, that the system is capable of providing all necessary tools to visually analyze local corresponding region of segmentation quality.

The user is able to customize the level of detail of the process. It is possible to explore the segmentation quality from a small scope of each individual sample to the point of a common context over all samples. Additionally the system is capable of finding outliers and analyzing them separately or excluding them from the majority of the data set (subset creation).

6 Conclusion and Future Work

In this chapter I will firstly recapitulate the whole work and summarize all function of the analysis tool, followed by the contribution of this work to the respective field.

Finally I will discuss possible improvements and enhancement of the software.

6.1 Summary

In my approach I developed a tool for visual analysis of local correspondence in segmentation quality.

The input data is a set of samples, each consisting of a reference- (usually an expert segmentation) and a test-mesh (automatic segmentation). To determine the quality of the automatic segmentation, the surface distance between both meshes is calculated.

A distance value will be assigned to each vertex in the mesh, which will be used as a feature¹. The goal is to find regions with same features, which correspond in the location on the mesh over the majority of data samples.

To achieve this, the system consists mainly of four views, each utilizing different techniques for the analysis:

1. Grid view
2. PCA- view
3. Clustering view
4. Cluster analysis view

Each individual part provides a different view on the data set and the results can be combined to outline the full potential of the software.

- The grid view gives an overview of the data set, by visualizing each sample pair in an own window and arranging every window in a grid. Using the distance coloring, which maps a color to each vertex according to the respective feature, the regions of low segmentation quality can be identified.
- The PCA-view arranges every sample in a single window according to the correlation of the feature vectors . Outlier samples which show different patterns of over- and under-segmented regions will be separated and can be selected and excluded from the analysis process.

¹ Definition in Section 2.1.5

- The clustering view utilizes two techniques, which combine the characteristics (features) of all samples and visualize them on one mesh. The first technique is the average distance coloring, which is similar to the previously mentioned coloring, but assigns the color according to the average distance of the respective vertex over all data samples. The second technique is a clustering of the feature vectors. A new algorithm was developed to cluster data values with same feature vectors without losing the neighbor relationships of the mesh (Definition: 2.1.1). The result are connected regions (clusters) on the mesh which members (vertices) show the same characteristics, in terms of feature similarity.
- These regions can be analyzed using the cluster analysis view. This view contains two plots, one provides general information about the clusters (size and quality) and the other visualizes the members of chosen clusters using parallel coordinates.

In summary, the thesis goal of finding, extracting and analyzing corresponding regions over a set of samples was achieved.

6.2 Contribution

The main purpose of finding the local corresponding regions of same segmentation quality is to improve the automatic segmentation algorithm. With the knowledge of the location of reoccurring regions with low segmentation quality, the algorithm developer is able to directly examine these locations in the raw data (CT-/ MRI- images) and draw conclusions of why the algorithm has failed. These conclusions can be utilized to eliminate the weaknesses.

The improvement of automatic segmentation techniques is beneficial to a large part of the medical field, because the image segmentation results are widely used in diagnostics and operation planning.

Additionally my system is not only restraint to the medical domain, but rather to all domains which can benefit from pairwise mesh comparison.

6.3 Future work

At the beginning of the development the main focus was to implement all methods and visualizations to achieve meaningful results. Therefore the computation time and memory usage is not ideal.

To load more complex and higher numbers of data pairs and to compute it in a reasonable amount of time the memory management has to be improved, plus the algorithms could be optimized to get a higher speedup from parallel computing. Also, the already calculated data (e.g. PCA, or the clustering trees) could be saved and loaded if the user starts the same scenario.

Most of the results are purely visual at the moment, which means it is possible to locate and analyze the interesting region in the software. However to directly improve the quality of

the segmentation algorithm it can be beneficial to give the user the possibility to save the analysis results for further processing.

During the evaluation it became obvious that the different developers of the segmentation algorithms have different requirements to the system and the output. For example, in case of the liver segmentation it is essential to locate the reoccurring regions of interest in the original raw images (e.g. CT slice) to find clues why the segmentation has failed. A possible approach to solve this problem is to give the user the ability to generate an output mesh with a specific color coding to identify and locate the regions of interest.

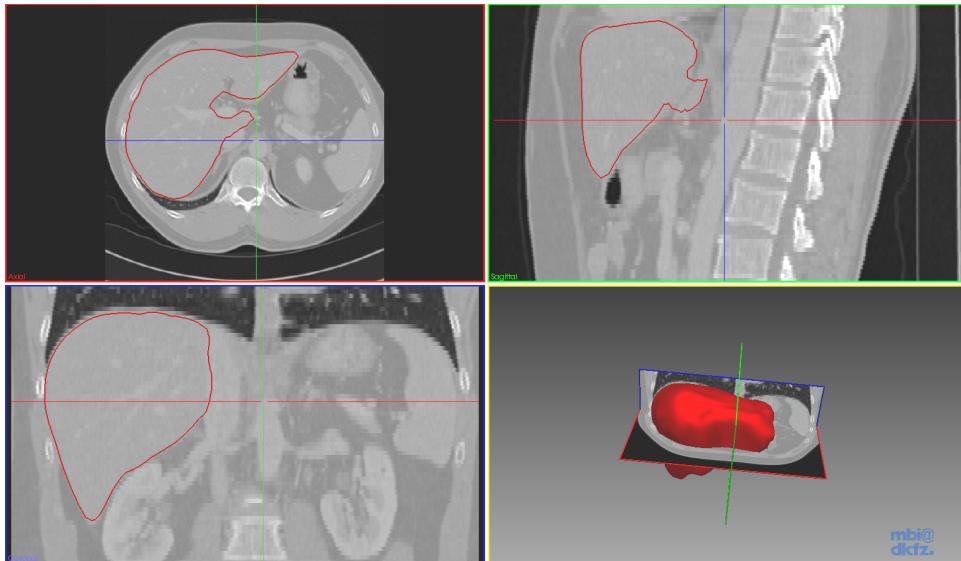


Figure 6.1: Combined visualization of mesh and image data in MITK

A potential toolkit that is capable of a combined visualization (Figure 6.1) of mesh data and image data is the Medical Imaging Interaction Toolkit (MITK: [GCRC13]). A connection or an **embedding of my system directly in MITK** would enhance the visualizations of the result.

The main purpose of the software is to extract region of interest between a reference- and a test-model, mainly an expert and an algorithmic segmentation. Sometimes a cross algorithm comparison can be beneficial, to combine the advantages of both algorithms. At the moment it is only possible to find the corresponding regions over a set of mesh pairs, but for the subject matter it is important to know in which regions one algorithm failed and the other one succeeded or vice versa.

A solution would be to load one reference- and **two** test-data sets and analyze the combination of the distances between each test- to the reference-mesh and the distances between the two test-objects. The result could be beneficial in terms of establishing the advantages and disadvantages of both algorithms and extract this knowledge to develop a new segmentation approach.

In Section 4.3.1 the dimension reduction technique I implemented was PCA, which extracts the linear correlations of the feature vectors, assuming the data is Gaussian distributed. However, it can be argued that the anatomical variance of organs is not Gaussian distributed [KW10], which would mean the correlation between dimensions could be non linear. If a respective segmentation of an organ which shape variance is not Gaussian distributed is analyzed with this evaluation tool, the PCA result could not be sufficient. To solve this problem, an alternative **non-linear dimension reduction** technique could be implemented (ICA, KernelPCA), which would help to find analyze this special cases more accurate.

The software is capable of loading and comparing any kind of surfaces and the analysis features are not only determined for medical data or even segmentation data at all. There are countless domains which could benefit from these mesh analysis examples. Of course all these different fields would need various adaptions of the software to match the requirements, but the further development could result in a steady tool with a wide spectrum of use cases.

Studies showed that organs or other anatomic structures like the brain can change shape during the progress of a disease[MPJ⁺06]. Assuming that the perfect segmentation exists it is also possible to confirm or deeper evaluate these long time studies or even extract new knowledge about various disease and how they affect the shape of particular organs. This knowledge could be used for early diagnoses of long term diseases.

Bibliography

- [ASCE02] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, “Mesh: Measuring errors between surfaces using the hausdorff distance,” in *Proceedings of the IEEE International Conference on Multimedia and Expo*, vol. I, 2002, pp. 705 – 708, <http://mesh.epfl.ch>.
- [AW10] H. Abdi and L. J. Williams, “Principal component analysis,” 2010.
- [CBC⁺07] R. Cárdenes, M. Bach, Y. Chi, I. Marras, R. De Luis, M. Anderson, P. Cashman, and M. Bultelle, “Multimodal evaluation for medical image segmentation,” in *Proceedings of the 12th international conference on Computer analysis of images and patterns*, ser. CAIP’07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 229–236. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1770904.1770936>
- [Cha07] S.-H. Cha, “Comprehensive survey on distance/similarity measures between probability density functions,” *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 4, pp. 300–307, 2007. [Online]. Available: <http://www.gly.fsu.edu/~parker/geostats/Cha.pdf>
- [CRS96] P. Cignoni, C. Rocchini, and R. Scopigno, “Metro: measuring error on simplified surfaces,” Paris, France, France, Tech. Rep., 1996.
- [Cun07] P. Cunningham, “Dimension reduction,” 2007.
- [Dig13] Digia. (2013) Qt-framework. [Online]. Available: <http://qt-project.org/>
- [Doi06] K. Doi, “Diagnostic imaging over the last 50 years: research and development in medical imaging science and technology,” *Phys. Med. Biol.*, vol. 51, 2006.
- [Dun74] J. C. Dunn, “Well separated clusters and optimal fuzzy-partitions,” *Journal of Cybernetics*, vol. 4, pp. 95–104, 1974.
- [EpKSX96] M. Ester, H. peter Kriegel, J. S, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise.” AAAI Press, 1996, pp. 226–231.
- [Fod02] I. Fodor, “A survey of dimension reduction techniques,” Tech. Rep., 2002.
- [For65] E. Forgy, “Cluster analysis of multivariate data: efficiency versus interpretability of classifications,” *Biometrics*, vol. 21, pp. 768–780, 1965.
- [FS05] J. T. Fernanda Sousa, “A validation methodology in hierarchical clustering,” 2005.
- [GCRC13] German-Cancer-Research-Center. (2013) The medical imaging interaction toolkit. [Online]. Available: <http://www.mitk.org/>
- [Get13] R. M. Getto, “Ein neues abstandsmaß für den vergleich von formähnlichen meshes mit lokalen abweichungen,” Master’s thesis, Technische Universität Darmstadt, 2013.

- [GG01] M. C. Guido Gerig, Matthieu Jomier, “Valmet: A new validation tool for assessing and improving 3d object segmentation,” in *Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, ser. MICCAI ’01. London, UK, UK: Springer-Verlag, 2001, pp. 516–523. [Online]. Available: <http://dl.acm.org/citation.cfm?id=646924.709195>
- [Got05] C. Gotsman, “What’s in a mesh? a survey of 3d mesh representation schemes.” in *SMI*. IEEE Computer Society, 2005, p. 2. [Online]. Available: <http://dblp.uni-trier.de/db/conf/smi/smi2005.html#Gotsman05>
- [HBV01] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, “On clustering validation techniques,” *J. Intell. Inf. Syst.*, vol. 17, no. 2-3, pp. 107–145, Dec. 2001. [Online]. Available: <http://dx.doi.org/10.1023/A:1012801612483>
- [HG97] P. Hoffman and G. Grinstein, “Visualizations for high dimensional data mining - table visualizations,” 1997.
- [HM09] T. Heimann and H.-P. Meinzer, “Statistical shape models for 3D medical image segmentation: A review,” *Medical Image Analysis*, vol. 13, no. 4, pp. 543–563, Aug. 2009. [Online]. Available: <http://dx.doi.org/10.1016/j.media.2009.05.004>
- [HO00] A. Hyvärinen and E. Oja, “Independent component analysis: algorithms and applications,” *Neural Netw.*, vol. 13, no. 4-5, pp. 411–430, May 2000. [Online]. Available: [http://dx.doi.org/10.1016/S0893-6080\(00\)00026-5](http://dx.doi.org/10.1016/S0893-6080(00)00026-5)
- [HVB00] M. Halkidi, M. Vazirgiannis, and Y. Batistakis, “Quality scheme assessment in the clustering process,” in *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, ser. PKDD ’00. London, UK, UK: Springer-Verlag, 2000, pp. 265–276. [Online]. Available: <http://dl.acm.org/citation.cfm?id=645804.669820>
- [HvGS09] T. Heimann, B. van Ginneken, and Styner, “Comparison and Evaluation of Methods for Liver Segmentation From CT Datasets,” *Medical Imaging, IEEE Transactions on*, vol. 28, no. 8, pp. 1251–1265, Aug. 2009. [Online]. Available: <http://dx.doi.org/10.1109/tmi.2009.2013851>
- [Ins85] A. Inselberg, “The plane with parallel coordinates,” *The Visual Computer*, vol. 1, no. 2, pp. 69–91, Aug. 1985. [Online]. Available: <http://dx.doi.org/10.1007/bf01898350>
- [JD88] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.
- [JMF99] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999. [Online]. Available: <http://doi.acm.org/10.1145/331499.331504>
- [Kir13] M. Kirschner, “The probabilistic active shape model: From model construction to flexible medical image segmentation (preprint),” Ph.D. dissertation, 2013.
- [Kit13] Kitware. (2013) Visualization toolkit. [Online]. Available: <http://www.vtk.org/>
- [KLB] F. Kovács, C. Legány, and A. Babos, “Cluster validity measurement techniques.”

- [KR90] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, 9th ed. Wiley-Interscience, Mar. 1990. [Online]. Available: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0471878766>
- [KW10] M. Kirschner and S. Wesarg, “Construction of groupwise consistent shape parameterizations by propagation,” in *Medical Imaging 2010: Image Processing. Part One*, 2010, pp. 762.352–1–762.352–12.
- [KW11] ——, “Active shape models unleashed,” pp. 796.211–796.211–9, 2011. [Online]. Available: [+http://dx.doi.org/10.1111/12.877690](http://dx.doi.org/10.1111/12.877690)
- [MPJ⁺06] R. K. McClure, I. Phillips, R. Jazayerli, A. Barnett, R. Coppola, and D. R. Weinberger, “Regional change in brain morphometry in schizophrenia associated with antipsychotic treatment,” *Psychiatry Res*, vol. 148, 2006.
- [Mur83] F. Murtagh, “A Survey of Recent Advances in Hierarchical Clustering Algorithms,” *The Computer Journal*, vol. 26, no. 4, pp. 354–359, Nov. 1983. [Online]. Available: <http://dx.doi.org/10.1093/comjnl/26.4.354>
- [MV98] J. Maintz and M. Viergever, “A survey of medical image registration,” *Medical Image Analysis*, vol. 2, no. 1, pp. 1–36, 1998. [Online]. Available: <http://citeseer.ist.psu.edu/maintz98survey.html>
- [NH06] M. Novotny and H. Hauser, “Outlier-preserving focus+context visualization in parallel coordinates,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 893–900, Sep. 2006. [Online]. Available: <http://dx.doi.org/10.1109/TVCG.2006.170>
- [NY06] T. S. Newman and H. Yi, “A survey of the marching cubes algorithm,” *Computers & Graphics*, vol. 30, no. 5, pp. 854–879, Oct. 2006. [Online]. Available: <http://dx.doi.org/10.1016/j.cag.2006.07.021>
- [Pea01] K. Pearson, “On lines and planes of closest fit to systems of points in space,” *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.
- [SK12] R. Shilpa Kamdi, “Image segmentation and region growing algorithm,” *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, vol. 2, 2012.
- [Smi] L. I. Smith, “A Tutorial on Principal Component Analysis.” [Online]. Available: <http://kybele.psych.cornell.edu/~{}edelman/Psych-465-Spring-2003/PCA-tutorial.pdf>
- [Smi06] R. Smithuis. (2006) Anatomy of the liver segments. Radiology Department of the Rijnland Hospital, Leiderdorp, the Netherlands. [Online]. Available: <http://www.radiologyassistant.nl/en/p4375bb8dc241d>
- [SSM99] B. Scholkopf, A. Smola, and K.-R. Müller, “Kernel principal component analysis,” in *ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING*. MIT Press, 1999, pp. 327–352.
- [Weg90] E. J. Wegman, “Hyperdimensional Data Analysis Using Parallel Coordinates,” *Journal of the American Statistical Association*, vol. 85, pp. 664–675, 1990.

- [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.15.1902>
- [WFI13] Wikimedia-Foundation-Inc. (2013) Wikipedia parallele koordinaten. [Online]. Available: http://de.wikipedia.org/wiki/Parallele_Koordinaten#cite_note-7
- [XW05] R. Xu and D. Wunsch, “Survey of clustering algorithms,” *Neural Networks, IEEE Transactions on*, vol. 16, no. 3, pp. 645–678, May 2005. [Online]. Available: <http://dx.doi.org/10.1109/tnn.2005.845141>
- [YPWR03] J. Yang, W. Peng, M. O. Ward, and E. A. Rundensteiner, “Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets,” in *Proc. IEEE Symposium on Information Visualization*, 2003, pp. 105–112.
- [ZF03] B. Zitova and J. Flusser, “Image registration methods: a survey,” *Image and Vision Computing*, vol. 21, pp. 977–1000, 2003.

List of Figures

1.1	a) Dozens of CT- Slides of the Torso b-d) CT images and the problems of segmentation: b) Liver tumor splits the liver in two parts, because of different gray values c) & d) surrounding tissue like the stomach or the heart have identical gray values. Pictures from [HvGS09]	1
1.2	The left side shows the extracted ground truth-(top) and automatic segmented (bottom) meshes. The right side presents the problem, which is the missing correspondence of geometric differences over a number of samples (n)	2
1.3	The regions of low quality are highlighted across the four data samples. A pattern of the geometrical position of reoccurring regions of interest is visible.	3
2.1	Shows a mesh with example information. FV := Featurevector	5
2.2	Comparison between corresponding and non-corresponding vertex IDs in two meshes: a) shows two vertices with the same vertex ID, they geometrically correspond b) shows two non corresponding vertices with the same ID	6
2.3	Example of over- and under-segmentation. The left picture shows an example of two segmented borders (green: reference-model, red: test-model) in a CT-slide. The right picture shows the resulting mesh.	7
3.1	A longitudinal section through two tubular meshes (red and green) and their respective contours the arrows show the result of the surface distance. The left picture shows an accurate registration of the two meshes, which results in the right surface distances, whereas the right pictures shows the error, due to a wrong registration.	10
3.2	Steps of clustering process. Source:[HBV01]	11
3.3	A clustering result of DBSCAN. The circles represent the reachability radius of each data value. The blue and red dots are the core points of cluster A respectively B, because they possess more data points in their reachability radius than minPts defines. The gray dots will be marked as noise points and will not be assign to any cluster. The green dots represent the border points. The right border point will be assigned to cluster B, whereas the left could be assigned to both clusters. Usually it will be randomly assigned to one of them.	14
3.4	Cluster dendrogram: a) agglomerative strategy b) divisive strategy was used. The Results are the same	15
3.5	Cluster Dendrogram with two different thresholds, the left threshold is set to hierarchy level 3 (HL), which will result in two clusters, whereas the right is set to HL 2, which will form three	17

3.6	Different Correlations between two dimension (top row) in parallel coordinates (bottom row), Source: [WFI13]	21
4.1	Left image: Clustering only in feature space. Middle image: The corresponding cluster assignment on the mesh, without spatial constraints Right image: The desired result. A clustering with spatial constraints, only neighboring vertices are qualified to form a cluster	28
4.2	Left image: Basic steps of the clustering algorithm. Right image: Six iterations of the clustering algorithm. The green and blue points represent vertices in the mesh that are already connected in the hierarchy. The gray points show which points are free to connect. The red edge marks the connection which will be merged next, because of the lowest distance. Black edges are still in the neighbor-distance list, whereas the colored edges are already erased.	30
4.3	Left image: Shows a standard clustering in feature space, the clusters are arbitrary shaped. The stars represent the well separated cluster centroids. Middle image: The clustering with spatial constraints. The centroid of the blue and green clusters are nearly identical, therefore the separation is significantly worse then in the clustering in feature space. Right image: The mesh showing the spatial constraints.	32
4.4	Basic GUI of the System - a) Visualization window - b) Color legend for distance coloring - c) Removable sidebar d) Menu bar to load Data and switch between Views	34
4.5	Bipolar color map	35
4.6	Three different distance colorings of 4 samples arranged in a grid: a) Local normalization b) Global normalization c) Global normalization with Quantiles	36
4.7	Threshold visualization in combination with distance coloring	36
4.8	Left image: Grid visualization of 20 samples, Right image: The average distance coloring according to the 20 samples visualized on the average mesh.	37
4.9	Grid view with 3x3 samples (Ground truth is green, auto-segmented red) - sidebar provides options like: show/hide mesh, activate distance coloring, change opacity, show global or local error and thresholding	38
4.10	Resizable grid, to enable focused view of interesting object.	39
4.11	The top picture shows two modalities and on the bottom there are two groups with 3 samples each, which express the same characteristics as the two modalities. Bottom picture: The meshes moved in the principal component space. The two groups are perfectly separated. The right frame represents the selection box, for subset creation.	40
4.12	The clustering view consists of three widgets. The left one contains the average mesh and coloring over all samples, the middle one contains the average mesh and coloring over a subset of samples (if a selection exists) and the right one is a result of a clustering over the same subset, represented on the subset average mesh	42

4.13	The left image shows the HSV-color-space and the right image is a top view of the cone. Different angles will result in different colors, but similar angles will be nearly identical.	43
4.14	The top row shows the marching square algorithm to create the line segments for every triangle of the mesh. Different colors represent the vertexes cluster membership. The bottom row shows how the clustering and the average distance coloring get combined.	44
4.15	Clustering mesh: The blue cluster was highlighted	44
4.16	Cluster overview plot. Visualization of seven clusters. Two clusters are selected	45
4.17	Parallel coordinate plot customization. The user is able to order the axis according to the chosen characteristics of the selected cluster (Cluster ID 5 is selected in this example) and it is possible to show or hide different representation of the values (0.75/0.25-quartile, average, median, absolute values)	46
4.18	Visualization of three clusters using parallel coordinates. The top image shows an unsorted plot, with absolute values, whereas the axes in the bottom image are sorted according to the red clusters average distance and the values are represented as the quartiles	47
4.19	The complete cluster analysis view, with the cluster overview graph (two highlighted clusters), parallel coordinates, sorting interactions and the two rendering windows (left: average coloring with cluster borders, right: cluster coloring with two highlighted meshes)	48
4.20	SD*-validity index plot: The X-Axis describes the number of clusters and the Y-Axis defines the range of the SD*-index. The vertical line in the plot represents the actual clustering threshold	49
5.1	Top row: The two test planes to evaluate the clustering algorithms (red and green). The local difference of is marked. The blue plane represents the reference- mesh. Bottom row: The two test planes with the coordinate system.	51
5.2	The graph (b) shows a view of the x,y- plane of the two test- meshes. The top line (a) shows the resulting clustering tree of the new clustering method with Euclidean distance and centroid-linkage criteria (left) and with Euclidean distance and complete-linkage (right).	52
5.3	The resulting clusters of a threshold set to hierarchy level 5 (a) and 4 (b) with the two mentioned settings. Left: Euclidean distance and centroid-linkage. Right: Euclidean distance and complete-linkage	53
5.4	Final clustering result of the example plane with two different settings and the same threshold (4 clusters)	54
5.5	Top row: Example outlier plane for testing, Bottom row: The clustering result based on Euclidean-distance and Chebyshev-proximity measure with their ideal thresholds	55
5.6	Left image: Location of the Liver in the human body, Right image: Segmental anatomy according to C. Couinaud (1957) classification [Smi06]	57

5.7	Grid View with threshold and distance coloring (blue describes under-segmentation and red over-segmentation) of the automatic liver segmentation from the IRCCAD Data set. The top part of left lobes is under-segmented in the majority of the samples. The global threshold is set to highlight the vertices which belong to the 10% with the lowest segmentation quality (over- and under-segmentation).	58
5.8	PCA view of the whole data set. Sample 6,7 and 14 are easy to identify as outliers. The square shows the selection of the majority of samples for further processing.	59
5.9	The average coloring visualization of the selection from two angles. The green ellipse marks the most significant regions.	60
5.10	The SD*-index plot with two different threshold. a: The threshold is positioned at the minimum. The according clustering consists of two clusters b: The threshold is positioned at a local minimum, the according clustering consist of 29 clusters.	61
5.11	The clustering result of the centroid-linkage based method	62
5.12	Three clusters visualized with the parallel coordinate plot. The orange values represents the cluster in the middle of the organ (segment one), the blue values stand for the top area of the left lobe (segment two) and the green lines represent the majority of the remaining area. The axes are ordered according to the orange cluster.	62
6.1	Combined visualization of mesh and image data in MITK	67