Visual exploration of segmentation errors in pelvic structures

Marcelis, F.J.J.

Award date:
2015

Eindhoven University of Technology
Department of Mathematics and Computer Science

Master's Thesis

Visual exploration of segmentation errors in pelvic structures

Freek Marcelis

August 31, 2015

Supervisors: dr.ir. H.M.M. van de Wetering
R.G. Raidou MSc
dr. A. Vilanova

Abstract

Nowadays, the usage of medical imaging data is widely present in both diagnostic procedures and treatment planning. Using Magnetic Resonance Imaging (MRI) or Computer Tomography (CT) scans, sets of two-dimensional slices are obtained, from which three-dimensional models can be constructed using segmentation algorithms. These three-dimensional models are frequently used in addition to the two-dimensional slices, to provide insights in the physical properties of the organs such as shape, size and relative position in an intuitive way.
This information is, amongst others, used by medical specialists in radiotherapy treatment planning to determine the exact location and dose of radiation required to treat cancer patients.

For the inspection of the organs physical properties, it is essential that the results from the segmentation algorithm are accurate and reliable. Using the data originating from the segmentation algorithm, the quality of the resulting meshes can be assessed. On the one hand, this allows users to confirm the quality of a single mesh, ensuring that this mesh is suitable for use by clinical users. On the other hand, quality analysis allows technical researchers such as computer scientists working on the segmentation algorithm to identify weaknesses in the segmentation phase, allowing them to improve the algorithm.

Currently, there is no system available which provides the tools for users to obtain this required information yet. Therefore, the process of studying the available patient data is complex and time consuming.

This document contains the design decisions made in the development of a web-based visualization framework for three-dimensional model inspection and multivariate data inspection and correlation detection in the segmentations of pelvic structues. It documents the requirements for such an application, provides context in terms of the current state of the art, and discusses the resulting framework.

Acknowledgements

I would like to thank my primary supervisor, dr. ir. Huub van de Wetering, for sharing his expertise on the topic of visualization approaches and for his constructive criticism required to guide me through this project. I would also like to thank my second supervisor, Renata Raidou MSc, for her neverending enthusiasm and willingness to answer all my questions. Also, her experience with the available medical data has proven invaluable. Lastly, I would like to thank dr. Anna Vilanova, for sharing her knowledge about the medical world, providing the insights needed to give a practical direction to this project.

Contents

1 Introduction

In the medical world, imaging data is used for a wide variety of purposes ranging from posing a diagnosis to determining and applying treatments. For example, when doctors are unable to diagnose a patients condition using external examinations only, or when exact locations of affected regions are required prior to treatment or surgery, internal images are used to aid the medical experts. One of the most common methods is to use Magnetic Resonance Imaging (MRI).

For the analysis of medical imaging data, often three-dimensional meshes are constructed representing the boundaries of organs in the data set. Regardless of the method by which this is done, the resulting meshes do not perfectly correspond to the actual locations of the boundaries. However, even the smallest imperfections may be of influence. In radiotherapy, for example, segmentation of the structures is important in order to apply the required dose in the tumor, while preserving adjacent healthy structures. In order to improve the use of segmentation of organ structures, research done in this area focuses on two main points. On the one hand, the algorithms with which the imaging data is converted to 3D-meshes is constantly refined, minimizing the likelihood and size of the errors. On the other hand, novel methods are developed to provide insight in the uncertainties and errors in the segmentation.

By providing means to efficiently visualize the data sets resulting from segmenting scan data, users discover both structural weaknesses across segmentations and uncertainties in individual cases. As will be explained in Subsection 1.4, there currently is no framework available facilitating the interactive exploration of the segmentation outcome. This document describes the development process and design decisions of an interactive application for web-based visual analytics which can be used for both of these causes.

The context of the project described in this report, is the visualization of errors in 3D-meshes. In this work, the focus is on visualizing the error in structures in the pelvic region, yet the framework is designed such that an extension to other regions or types of segmentation approaches is straightforward. Using model based segmentation [EPS$^+$08] on MRI scans of patients, 3D meshes are derived consisting of sets of triangles. Additionally, for each of these triangles several metrics are returned containing information on the performance of the segmentation algorithm. These metrics are relevant as the quality of the generated surfaces depends on a variety of parameters of this algorithm. Using visualizations of the quality, either algorithm researchers can alter the parameters of the surface construction algorithm to produce better results or information about the uncertainty can be conveyed to medical and segmentation specialists.

Figure 1.1 shows a schematic representation of this context. For this project, the scope consists of only the visualization step of this process. The data files containing generated meshes and quality metrics are provided, and also altering the parameters of the surface construction algorithm is outside the scope. More information about the data used for this project will be provided in Subsection 1.6.

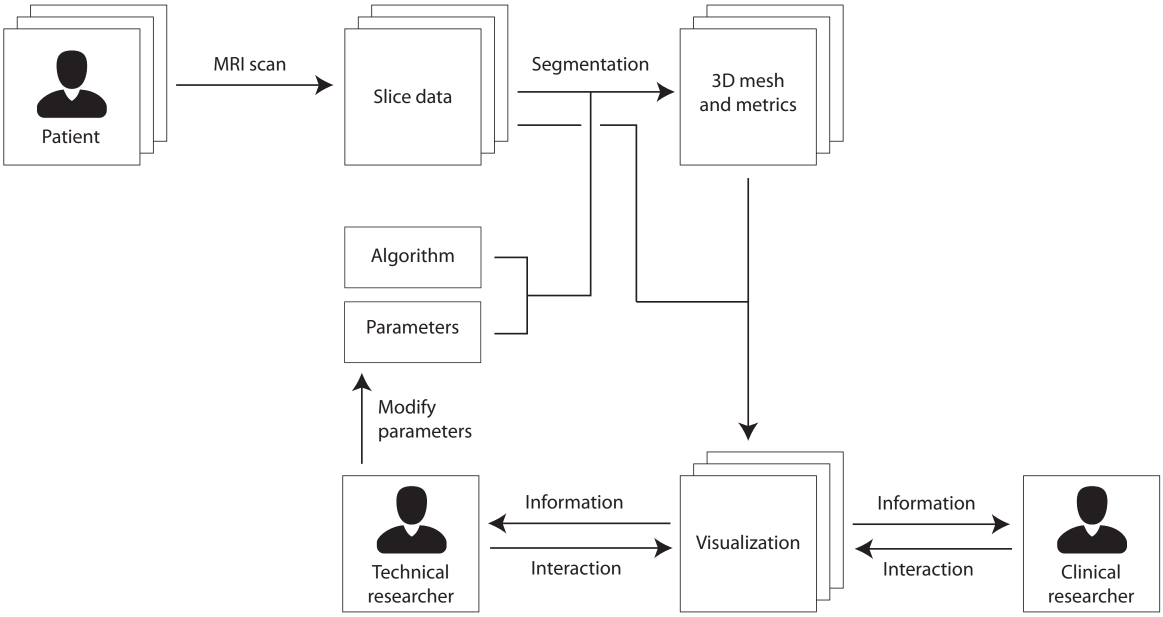


Figure 1.1: A schematic representation of the context of the project. The scope of this project is the content of the 'Visualization' block. In this figure, blocks represent static entities such as data and users, and arrows represent either actions or information flow.

1.1 Problem description

The aim of this project is to provide a visual analytics framework enabling users to explore and analyze interactive visualizations of provided data sets. Detailed information on the accepted types of input data and the intended users of this visual analytics framework can be found in Subsections 1.2 and 1.6 respectively. Using this framework, the user should be able to perform the following actions on the data:

- Data exploration
  - Inspect the three-dimensional shape of the organs
  - Explore the attribute values
  - Explore the physical distribution of attribute values
- Knowledge discovery
  - Discover correlations between values of attributes and profile information
  - Compute new attributes from response profile data (e.g. number of peaks in a profile)
- Data comparison
  - Compare attribute value distributions across multiple data sets
  - Compare physical differences between a 3D mesh and shapes in 3D volume data

From the actions mentioned above, the user is able to generate or confirm hypotheses about the data. Also, technical researchers can make decisions concerning the algorithm used and clinical researchers can make decisions concerning the treatment.

A more detailed list of the goals and questions to answer for the different types of users is provided in Subsection 1.3.

1.2 Users

The set of potential users of this organ mesh visualization application can roughly be divided into three groups;

- Technical researchers, who work on the segmentation algorithms.
- Clinical researchers, who use organ data for scientific research.
- Clinicians, who use the exact position of the organs to determine the details of treatments such as radiation doses.

Each of these three groups has different goals, and uses visualizations in a different way. Because of this, the visualization methods needed to accomplish their tasks are not the same either.

Technical researchers focus on improving the segmentation algorithm. This requires them to investigate the quality of the results of the segmentation, and try to define weaknesses and opportunities for increasing the reliability. Therefore, technical researchers are mostly interested in discovering trends occurring over larger sets of patients. For technical researchers, the application should aid in exploring the data and showing relations between user-defined sets of attributes.

Clinical researchers require accurate information on the uncertainty of the 3D meshes. In contrast to the technical researchers, clinical researchers are not looking for relations between physical properties and the uncertainty. Also, for a clinical researcher, gathering information about a single patient is not uncommon, as for some tasks, the reliability of a single segmentation suffices.

Lastly, clinicians do not need information on the process used for converting imaging data into meshes. Clinicians focus only on using the 3D models for a variety of purposes, and need clear information on the location of the organs. The visualization approaches used by clinicians mainly serve to inform them of the uncertainty of the model, allowing them to make decisions in a well-informed manner.

For the scope of this project, only the first two types of users, technical researchers and clinical researchers, are considered. The application focuses on accurately visualizing the available data and is therefore suitable for technical and clinical researchers.

Clinicians however, also need reliable verdicts regarding the accuracy of the data. Because this reliability can as of yet not be guaranteed, it is not safe to let clinicians base treatments on information provided by the application.

1.3 Tasks

As explained in Subsection 1.2, the application is focussed on two types of users, technical researchers and clinical researchers. The questions users are trying to answer depends on which type of user they are.

Technical researchers are generally posing questions such as:

- What is the shape of the organs in the data set?
- How are the values of each attribute distributed over the value range?
- How are the values of each attribute distributed spatially?
- How accurately does the segmented mesh correspond to the patients organs?
- How do available attributes correlate with each other?
- Is there a relation between organ properties and segmentation accuracy?
- How many potentially correct positions were considered for specific parts of the mesh?

Clinical researchers, on the other hand, are trying to answer questions such as:

- What is the shape of the organs in the data set?
- How are the values of each attribute distributed over the value range?
- How are the values of each attribute distributed spatially?
- How accurately does the segmented mesh correspond to the patients organs?
- How reliable is the segmentation result in specific areas of the mesh?

1.4 Related work

Previously, multiple frameworks have been created for the segmentation of (medical) image data, such as for example by Chen et al. [CM05] and Yezzi et al. [YZK03]. However, none of the frameworks developed concerning image segmentation include the assessment of the quality of the result after the segmentation phase.

Earlier research on visualizing the quality of three-dimensional meshes can be found in the area of mesh (re)construction, for example from point clouds. However, although constructing 3D meshes from point clouds has been a popular research area [F+03], with improving results due to technological advances and increasing hardware performance, visualizing the uncertainty in these meshes is usually either still done using the traditional methods available before these advances, or not considered at all. Therefore, most work related to visualizing the quality of meshes can be found not in the field of surface construction or mesh comparison, but in the field of uncertainty visualization.

In their article “Approaches to uncertainty visualization”, Pang et al. [PWL97] provide an overview of different methods used for visualizing uncertainty both in 2D and 3D. G. Grigoryan and P. Rheingans show how a combination of coloring and surface attributes or glyphs can be used to convey uncertainty information [GR04]. The study by Sanyal et al.[SZB+09] confirms that these methods are considered most clear by end users. For efficiently comparing the shapes of two different meshes, several dynamic approaches exist, as shown by Busking et al. [BBF+11]. They used a combination of transparency, color mapping and glyphs to convey differences with minimal clutter. Additionally, by allowing the user to set a threshold for intermesh distances which modifies the color mapping, they allow the user to explore the size of the distance between meshes at a given point.

However, even though the use of 2D and 3D visualization has increased drastically after the publication of these articles, new methods for the visualization of uncertainty, especially for three dimensional data, are rare. In their article “A next step: visualizing errors and uncertainty” [JS03], C. Johnson and A. Sanderson acknowledge this. Combined with the article “Visual Analytics for model-based medical image segmentation: Opportunities and challenges” from Von Landesberger et al.[vLBK$^+$13], they provide elaborate lists on the most important challenges present in this field of study. Challenges related to data complexity and parameter space analysis, and regarding visual quality assessment and comparative analytics are addressed.

Regarding the use of multiple, possibly linked, views, Jonathan C. Roberts argues in his article “Multiple-View and Multiform Visualization” [Rob00], that using multiple views of the same data set is beneficial to the user’s understanding of the information, and proposes several aspects, such as data coupling and preservation of the hierarchy when the data is divided into subsets, which should be optimized to produce high quality multiple-view visualizations.

Besides the different approaches to visualizing uncertainty in 3D models, visualizing distributions and correlations in multivariate data is essential for assessing the quality of the segmentation result. In contrast to surface quality visualization, multivariate data visualization is studied exhaustively. In their survey “A survey on multivariate data visualization” [Cha06], Chan elaborates on the wide variety of multivariate data visualizations in the categories of geometric projection, pixel-oriented techniques, hierarchical display and iconography.

For the purpose of studying multivariate data sets, elaborate frameworks already exist, as Yang et al. described in their paper “Interactive hierarchical displays: a general framework for visualization and exploration of large multivariate data sets” [YWR03]. However, such frameworks lack the connection between the multivariate data visualizations and the three-dimensional mesh, and are therefore not suitable for the interactive exploration of segmentation quality.

1.5 Contributions

As described in Subsection 1.4, both in the fields of image segmentation algorithms and multivariate data exploration, elaborate frameworks currently exist. However, for the assessment of the quality and reliability of meshes resulting from segmentation algorithms, and the discovery of correlations between computed error and the various attributes resulting from the segmentation phase, no such framework exists yet.

The application described in this document attempts to fill that void, allowing for both the shortcomings of the segmentation algorithm to be detected and the reliability of the results to be determined more easily. Thereby, this work contributes

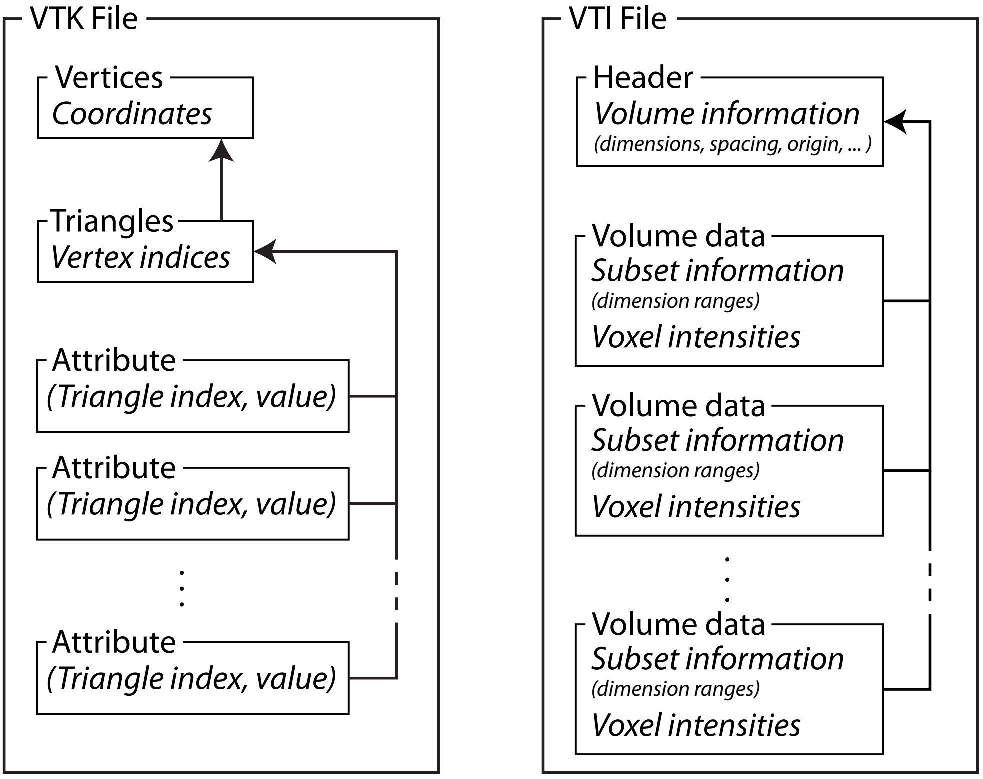
- a combination of visualizations both in three and two-dimensions for the exploration of both meshes and multivariate data
- a low-level interaction mechanism allowing all visualizations to respond dynamically to actions performed on a single visualization

- a sophisticated method for storing and displaying user-defined subsets of the data, allowing for the analysis of specific parts of the provided information.

1.6 Data

To provide a generic visual analytics framework for analysing segmented meshes, the application has been designed to function with all data sets following a fixed structure. To be precise, the framework accepts data files employing the VTK standard, as described in the VTK User's Guide [SMAL98]. Furthermore, for the volume data, the VTI standard, also described in the VTK User's Guide, is used.

Highly simplified, the file structure of VTK and VTI files can be represented by Figures 1.2a and 1.2b respectively. The blocks in these figures represent (numerical) data, and the arrows represent references to these data fields. For example in the VTK structure, the set of vertices is defined as a set of coordinates, and the set of triangles is defined as groups of vertex indices. Therefore, the triangles are represented as a block of indices, referencing the set of vertices.



(a) A schematic representation of a VTK file structure

(b) A schematic representation of a VTI file

Figure 1.2: Schematic representations of the VTK and VTI file structures, as defined in the VTK User's Guide [SMAL98]. In these structures, blocks represent numerical data, and arrows represent references to data specified earlier.

Although the framework is designed to provide a generic approach, such that all data sets using the same format can be visualized efficiently, during the development a fixed set of data is used. The available data can be subdivided into three categories: mesh and attribute data, containing the 3D triangle set and a set of attributes for each triangle, volume data, containing a set of voxels from which slice images can be constructed, and profile data, which contains information on which the decision for the location of a triangle is based.

Across all data sets, the number of triangles in the mesh is identical, and each triangle at index i in a certain data set corresponds to triangles at index i in other data sets. This facilitates comparing subsets of meshes by matching triangle indices, and allows combining the data sets containing a mesh and attribute data with the profile information data sets.

Mesh and attribute data

Files containing the segmented mesh and attribute data follow the VTK structure. This means that the file starts with a set of coordinates representing vertices in 3D space. Next, a set of triangles is defined using the indices of the previously defined set of vertices. After this mesh declaration, a set of attributes is provided as k lists of length n, where k is the number of attributes, and n is the number of triangles in the mesh.

The contained set of attributes differs from file to file, depending on the source of the information. Mesh and attribute data files can provide data on the result of the segmentation process, with attributes representing parameters such as ‘feature response strength’ and ‘target point distance’ for individual triangles. Data files can also contain data on the quality of the segmentation acquired by comparing the segmentation result to a ground-truth mesh constructed by medical experts. Lastly, mesh and attribute data set can represent information averaged over multiple patients, such as average mesh shapes and average errors. This last category is used mainly to explore the characteristics of the performance of the segmentation algorithm, in contrast to the exploration of patient specific information for which the other categories are used.

The available data files using the described structure and attribute types are listed below.

- `model_bprs.vtk` contains an average mesh, accompanied by two attributes per triangle; a label representing the organ of which the triangle is a part, and the surface area in mm^2.

- `training_error.vtk` also contains the average mesh accompanied by two attributes. One is a label representing the organ of which the triangle is a part, and one attribute represents the average error per triangle in millimeters throughout the training phase. This average error therefore provides insight in the performance of the algorithm when averaged over multiple patients.

- `bprs_local_error_per_case.vtk` again contains an average organ mesh. Additionally, this file contains 9 attributes representing the error, in this case the target point distance distance in millimeters, per triangle for the 9 different patients. This target point distance represents the distance from each triangle in the mesh to the corresponding triangle on a ground-truth mesh.

- `#.error.vtk` is a file containing the segmented mesh for a specific patient number #, along with an attribute representing the point to ground-truth surface distance for each triangle as described earlier.

- `#.quality.vtk` also contains the segmented mesh for a specific patient #, though this time accompanied with four attributes; a feature response measure, a weighted feature response measure, the target point distance in millimeters and the triangle area in mm^2.

Slice data

Files containing the volume data follow the VTI structure, which is the image data structure of the VTK standard. At the start of the file, the volume set properties such as the size, spatial resolution and location in 3D space are defined. Next, a (possibly encoded) string is provided representing an array of length $width \cdot height \cdot depth$.

The volume data files provided consist of a set of voxels with dimensions $320 \cdot 320 \cdot 120$, with spatial resolutions of 1.563, 1.563 and 1 mm. From this volume set, two-dimensional slices can be constructed. Figure 1.3 shows an example of one slice in the transverse plane.

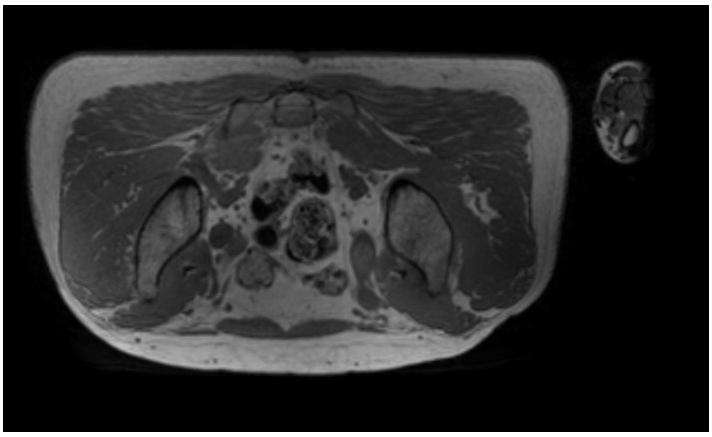


Figure 1.3: A slice of CT scan data in the transverse plane.

Profile data

Response profiles consist of a list of response strengths per triangle, which are used to determine the location of the boundary of an organ. For each triangle, the response is computed for multiple locations, each representing a likelihood that this location is part of the organ boundary, based on the imaging data provided.

The files containing the profile data, named `#.profile.vtk`, follow the VTK structure. Similar to the files mentioned under 'Mesh and attribute data', profile data files start with sets of vertices and triangles representing the segmented mesh. Following this mesh, the profile data file contains a single attribute consisting of an array of values per triangle.

For each triangle, the feature response is computed for 21 points, ten on one side of the triangle, ten on the opposite side and one on the position of the triangle itself. Figure 1.4 shows an schematic representation of a response profile for one edge of a two-dimensional surface, where the values in the profile are represented by the length of the lines parallel to the surface edge.

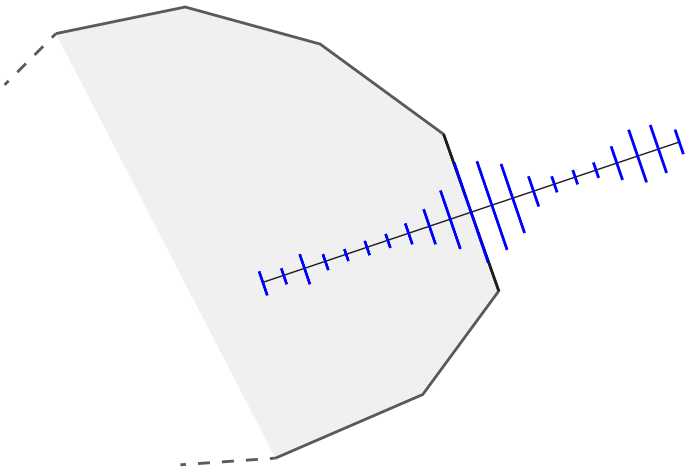


Figure 1.4: A schematic example in two dimensions of a response profile. For each part of the surface boundary (which in 2D is a line, and in 3D is a triangle), there are 21 response values on the line perpendicular to the surface boundary. In this example, the length of the 21 blue lines represents the 21 values in the profile.

1.7 Technology requirements

The data sets used as input for the application are formatted according to the VTK file structure as described in the VTK User's Guide [SMAL98].

To optimize the compatibility and ease of access, the application is developed using WebGL, a JavaScript based 2D/3D renderer. Additionally, the libraries Three.js [Cab10] and D3.js [Bos12] are used to allow the use of higher level code. This results in an application that is executable on all compatible browsers over multiple platforms. Furthermore, developing in WebGL enables a straight-forward inclusion into other WebGL based frameworks.

2 Design

In this section, we describe the analysis of the tasks described in Subsection 1.3 and the resulting design of an application that supports these tasks.

In the application, each visualization type is assigned a fixed position on the screen. Figure 2.1 shows a screenshot of the user interface of the application, with each of the four elements labeled.

A) A small panel at the top, which is used to load data files and select a loaded data file for visualization.

B) A large panel on the left of the screen, containing a three dimensional representation of the organ mesh contained in the selected data set.

C) A large panel on the right, of which the contents can be changed by selecting any of the options listed at the top of the panel. This panel contains several kinds of two-dimensional visualizations and settings with which the three-dimensional view can be modified.

The size of the bottom two panels, B and C, is automatically adjusted such that together they fill the entire screen. This is done because after selecting a data set to visualize the main focus of the user is interacting with the visualizations and does not require loading another data file. Therefore, the available space on the screen is used optimally for the visualizations.

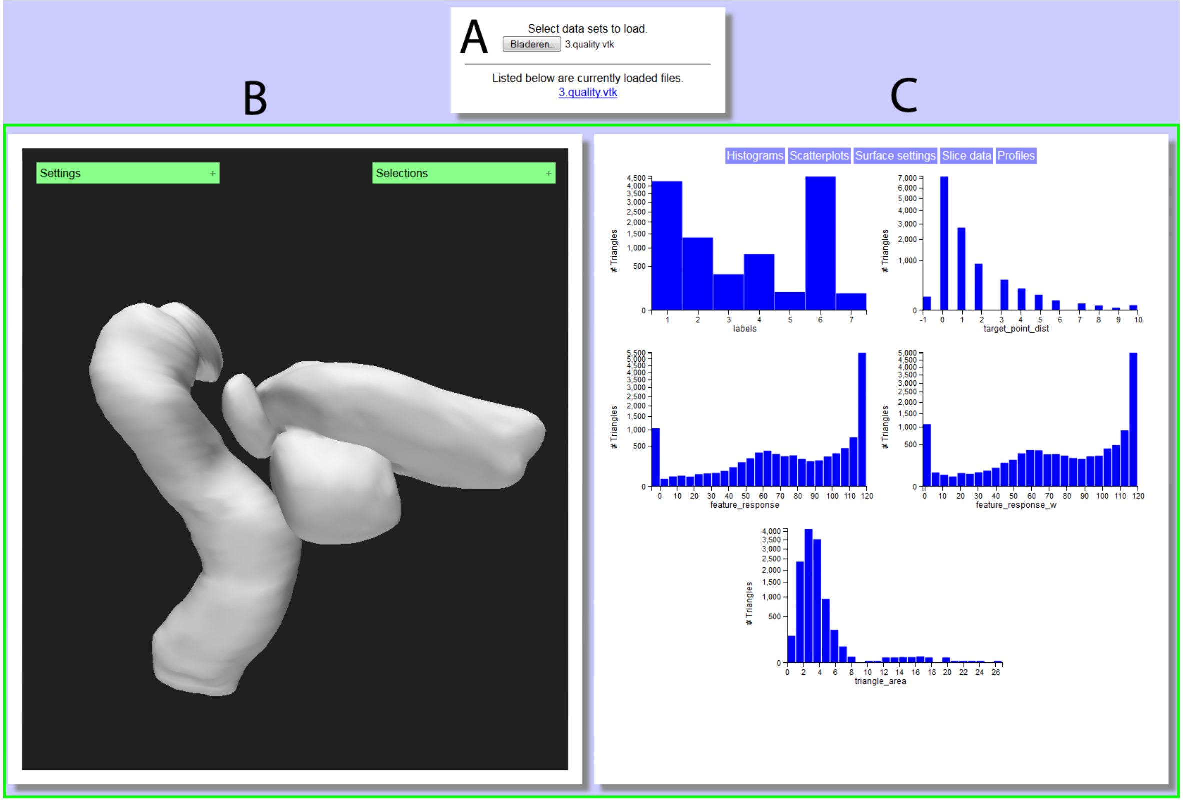


Figure 2.1: An overview of the user interface of the application. The interface is subdivided into three panels. One for data file selection (A), one containing 3D visualizations of the mesh in the data files (B), and one containing two dimensional visualizations and settings (C). The green rectangle illustrates the dimensions of the screen, illustrating how during panel B and C together use all of the available screen space.

2.1 Mesh and data exploration

For both technical and clinical researchers, it is important to get familiar with data sets before starting to execute more complicated tasks such as determining the reliability of the mesh or searching for correlations. To this extent, users will likely start by exploring the three-dimensional shape of the mesh and the value range and distribution of the attributes. To facilitate this, several features are required.

First of all, on the left of the screen (panel B), a 3D representation of the mesh is displayed, allowing the user to inspect the physical properties of the model. The relative positions of the triangles with respect to each other can not be conveyed using a fixed two-dimensional projection, so to explore the mesh, the user will need to manipulate a virtual camera. Interacting with this camera can be done by dragging one of the mouse buttons. Using the left, middle and right mouse button for rotating, panning and zooming the camera, it is possible to explore the segmented mesh from all sides and accurately manipulate the camera to inspect every part of the mesh up close.

The distribution of values per attribute is displayed using a visualization based on a histogram [War12], on the right half of the screen, in Panel C. This method uses a two-dimensional graph displaying the value range of the attribute on the horizontal axis, and the number of triangles with a given value on the vertical axis, as displayed in Figure 2.2.

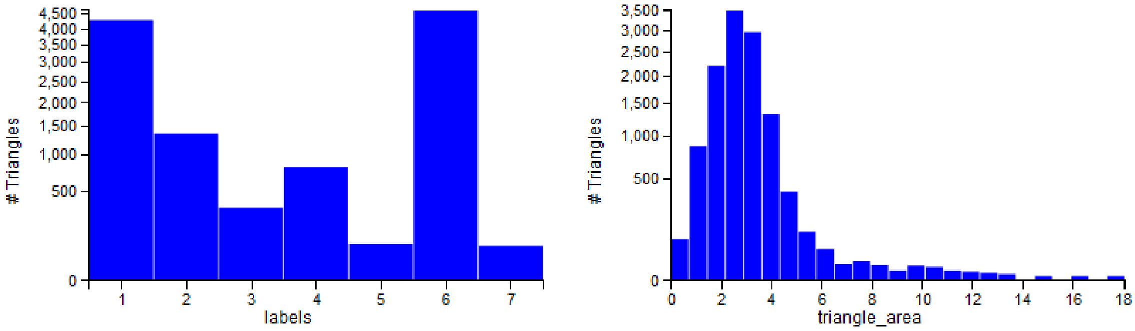


Figure 2.2: An example of a bart chart visualization, mapping the value distribution for an attribute with discrete values (left) and continuous values (right).

Each scalar attribute in the data set can contain either discrete or continuous values. For the latter case, the values are distributed over 'buckets' with equal width, such that for a given minimal and maximal value min and max and number of buckets $|buckets|$, the width of each bucket is

$$width = (max - min)/|buckets| \tag{2.1}$$

and each attribute value v is placed in bucket number

$$bucketID = \lfloor (v - min)/width \rfloor \tag{2.2}$$

If the attribute has a discrete value distribution with $|buckets|$ or less distinct values, no buckets are used and the number of bars in the graph is $max - min + 1$. If there are more than $|buckets|$ distinct discrete values, the same procedure is applied as for continuous values.

The height of each bar in this graph corresponds logarithmically to the number of triangles with a given value. This avoids the presence of tall bars from causing the height difference between smaller bars to become hard to distinguish.

The histogram offers insight in the value distribution per attribute, but ignores the physical distribution of the values. To show the distribution of the values on the mesh, a mapping can be applied to link the values of an attribute onto a range of colors. By default, this color distribution is a modified black-body radiation map, which uses a color range from red via yellow to white, as this is widely considered to be an intuitive and useful color map [BTI07]. For any attribute value v in the range between the minimal and maximal value min and max, by defining the value range as $range = max - min$, the color is defined by color map $color(v)$, represented as three values in the range [0..1] representing the intensities of red, green and blue respectively.

$$color(v) = \begin{cases} [1, \frac{v-min}{range/2}, 0] & \text{if } v - min \leq range/2 \\ [1, 1, \frac{v-min-range/2}{range/2}], & \text{if } v - min > range/2 \end{cases} \tag{2.3}$$

However, even when using an ideal color map, this method is not without downsides. For example, the mesh onto which the attribute values are color mapped is subjected to shading, which may alter the perceived color of a triangle. This is inevitable, as the shading is required in order to perceive the shape of the mesh correctly, and by definition shading alters the displayed colors.

Additionally, estimating the attribute value of a single triangle accurately is made difficult, as the perceived color of an individual triangle may be influenced by the color of neighboring triangles.

Lastly, users suffering from color blindness may find it difficult to judge where to place a triangles color on the legend, thereby influencing the estimated attribute value. Since there is a variety of different types and severities of color blindness, it is difficult to find a universally optimal color map.

Another method to visualize the values of all triangles for a single attribute at once, is to display glyphs on the triangles in the 3D representation. In the application, the glyphs are lines perpendicular to the triangles surface, positioned at the center of the triangle, as illustrated in Figure 2.3. This data-driven glyph placement is especially suitable for outlier detection in large data sets, as described by Ward in his “A taxonomy of glyph placement strategies or multidimensional data visualization” [War02].

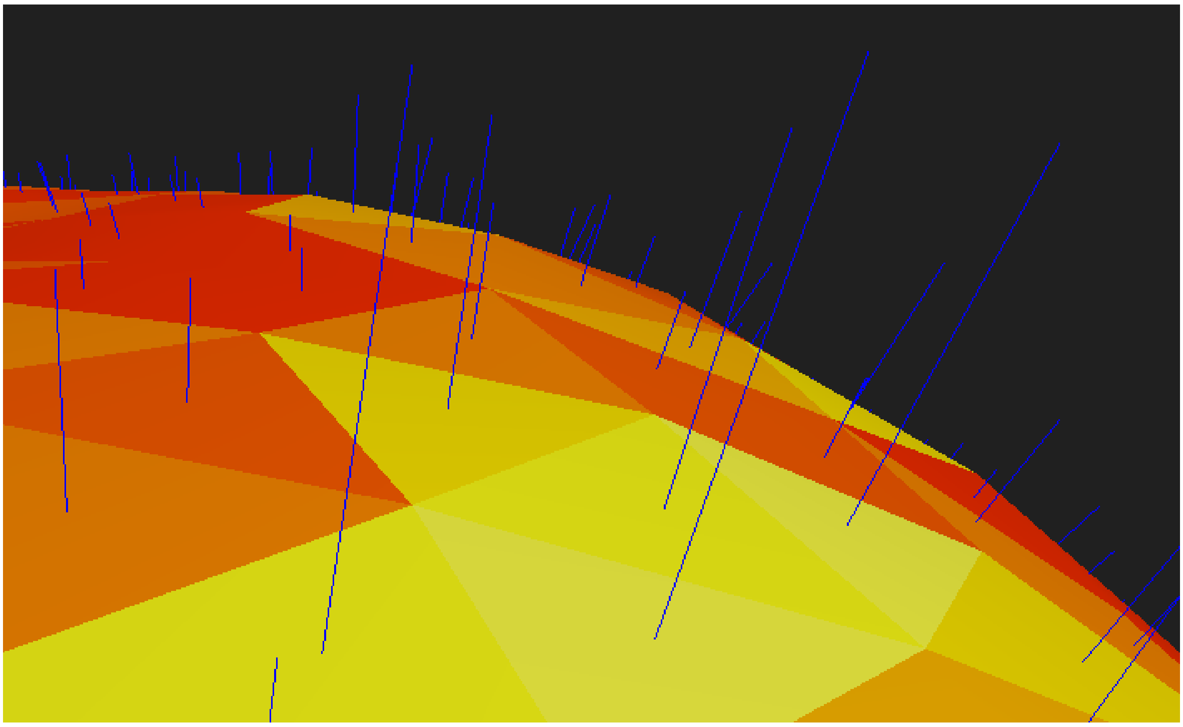


Figure 2.3: A part of the mesh with on each triangle a blue glyph perpendicular to the surface, representing the target point distance attribute. The color of the triangles also represents the target point distance, but is only added to this image to illustrate the position of the individual triangles.

The length of this line can be either the exact value of the attribute, or it can be normalized. In the latter case, for a value v, highest occurring value v_{max} and maximal glyph length l_{max}, the length of the glyph l is equal to

$$l = (v/v_{max}) \cdot l_{max} \tag{2.4}$$

One of the greatest strengths of this type of visualization is that it can be used simultaneously with the color mapping, without one of the visualization influencing how values represented by the other method are interpreted. However, this approach also has two drawbacks. Firstly, the length of a glyph can be misinterpreted due to the projection of a 3D shape onto a 2D canvas. Secondly, glyphs representing outliers smaller than the average glyph tend to be hard to notice due to the magnitude of the number of glyphs, whereas outliers larger than average are easy to detect.

Due to these disadvantages, users will likely prefer to represent values using a color map. Still, the added value of a second visualization approach which can be used in parallel to color mapping is large enough to justify the presence of this feature.

2.2 Correlation detection

Both technical and clinical users need information not only on the distribution of the values for individual attributes, but are trying to find relations between sets of attributes. This allows them to create hypotheses for possible causal relationships between attributes and verify existing hypotheses. To achieve this, users need to identify correlations between the attributes. When exploring the data for correlations, we can distinguish two different types of correlations. We can identify correlations between the values of different attributes, and correlations between the attribute values of a certain triangle and its physical position.

An intuitive way to visualize the correlation between two attributes is by drawing a scatter plot [War12]. Because scatter plots are a well known approach to visualizing correlations between two attributes, the majority of users will already be familiar with this method, which decreases the amount of training users need to discover correlations.

A scatter plot is a two dimensional graph where the two attributes are mapped onto the horizontal and vertical axis. In this graph, each point represents a triangle in the mesh, such that its horizontal and vertical position represent the values of the two attributes for that specific triangle. This property causes sets of triangles with similar values for both projected attributes to be grouped close together, and triangles with dissimilar attribute values to be drawn further away. An example of a scatter plot is presented in Figure 2.4, showing the correlation between the attributes ‘feature_response’ and ‘triangle_area’.

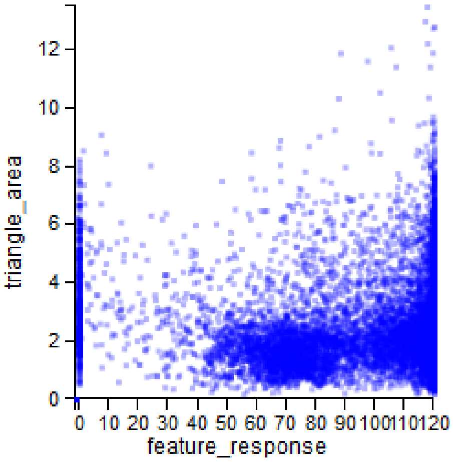


Figure 2.4: An example of a scatter plot visualization, mapping two attributes 'feature_reponse' and 'triangle_area' against each other.

When the number of triangles increases relative to the size of the scatter plot, the chance of overplotting, where a significant portion of the pixels of which the scatter plot consists represents multiple triangles,increases. This is inevitable, as triangles with similar attribute values need to be drawn close together, if not at exactly the same position, but it does cause a loss in comprehensibility, as it is no longer able to identify the number of triangles are included in a specific set of points. Therefore, each point in the graph is drawn semi-transparent, allowing the user to distinguish areas with high point density from less dense areas. This principle is also visible in Figure 2.4. On the right side of this image, there is a distinction between areas with a darker shade of blue (near the bottom of the graph) and lighter areas (further removed from the horizontal axis), indicating a difference in point density.

This distinction between densities could also have been visualized using a kernel density plot as shown in Figure 2.5, but because we already use the color of the points in a scatter plot to convey a different type of information, as will be elaborated upon later under 'Selection visualization', this would become confusing.

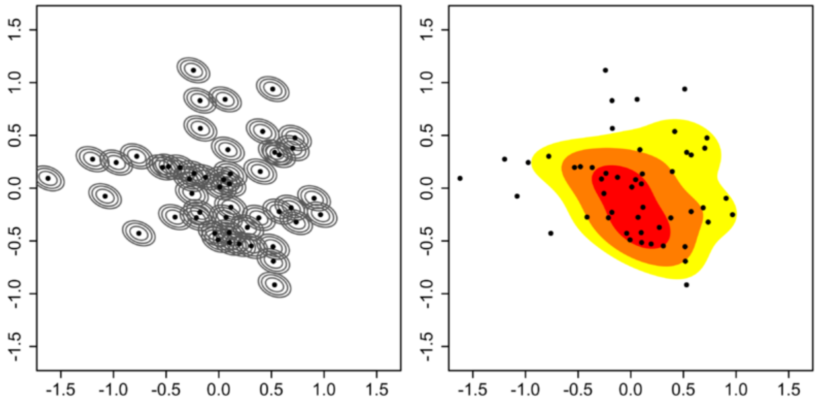


Figure 2.5: An example of a kernel density plot showing synthetic data. On the left, the data points are displayed with their individual kernels displayed as grey lines, and on the right the sum of those kernels is displayed as the kernel density estimate. Source: https://en.wikipedia.org/wiki/File:Synthetic_data_2D_KDE.png

For a data set containing an arbitrary number of attributes n, $n-1 \cdot n-2$ different combinations of two attributes can be made, which means the number of scatter plots grows quadratically in the number of attributes. To present this number of graphs in such a way that the user can still easily deduce what each individual graphs represents, the scatter plots are presented in a two-dimensional matrix, as described in “Scatterplot matrix techniques for large N”, by Carr et al. [CLNL87]. This enables us to order the graphs such that for each row and column the attribute on the horizontal and vertical axis respectively are the same. Figure 2.6 shows an example of a scatter plot matrix for a data set with five attributes.

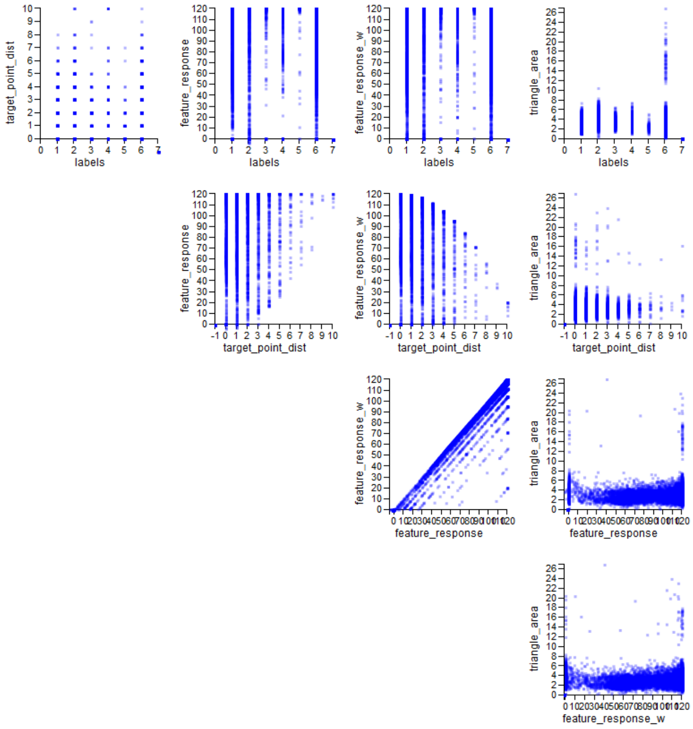


Figure 2.6: An example of a scatter plot matrix, where on each graph on a row share the same horizontal axis and each graph in a column have the same attribute on the vertical axis.

Note that only half of a full matrix is drawn, as the graphs below the diagonal are identical to those above, mirrored through the diagonal. This reduces the number of scatter plots that need to be displayed while still providing the same information, thereby making it simpler for the user to explore the correlations.

A different popular approach to visualizing correlations between attributes is by using a parallel coordinate plot, as introduced by Inselberg [Ins85], where all attributes are represented as parallel vertical axis and each triangle is represented by a line through each vertical axis at the height corresponding to the attribute value for that specific triangle, as demonstrated in Figure 2.7.

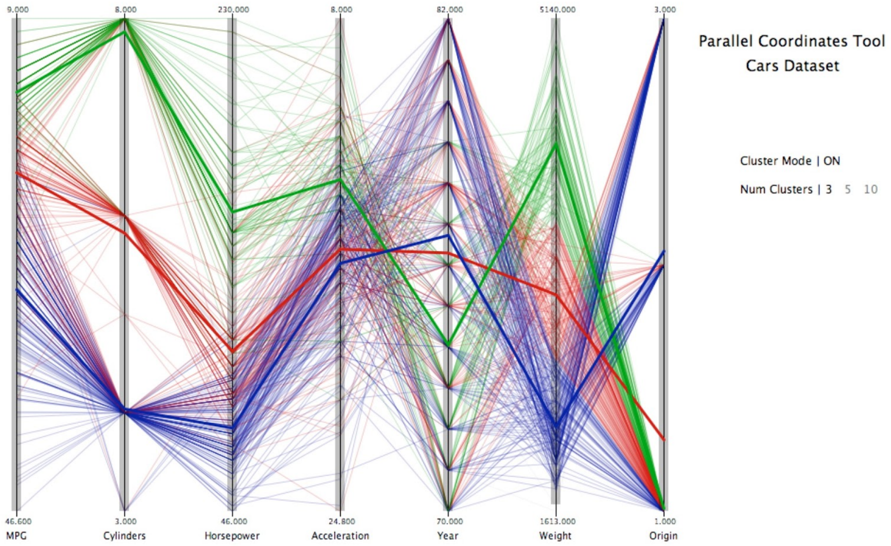


Figure 2.7: An example of a parallel coordinate plots, displaying seven properties for a large set of cars. (Source: [Aie12])

However, discovering correlations using parallel coordinate plots highly depends on the order of the parallel axis [UVW03]. When a set of attributes is correlated, but non-correlating attributes are placed in between, it is difficult to notice the relation. Although parallel coordinate plots allow for the discovery of patterns between more than two attributes, which in a scatter plot is displayed as multiple correlations between two attributes, this does not compensate for the problem with selecting the optimal order of the attributes. Therefore, parallel coordinate plots are not included in the application.

In contrast to correlations between two attributes, displaying correlations between the physical location of a triangle and attribute values is done by displaying attribute values in the three-dimensional image on the left. Using the approaches utilizing color mapping or glyphs, as explained in Subsection 2.1, it is possible to detect local similarities between attribute values. An additional advantage of the usage of the three-dimensional representation, is that when both color mapping and glyphs are employed, three different variables are presented simultaneously with minimal visual clutter.These methods provide less insight in the absolute values of attributes than two-dimensional scatter plots with clearly labeled axis, but are sufficient to provide information on global trends, smoothness of the value distribution and outliers.

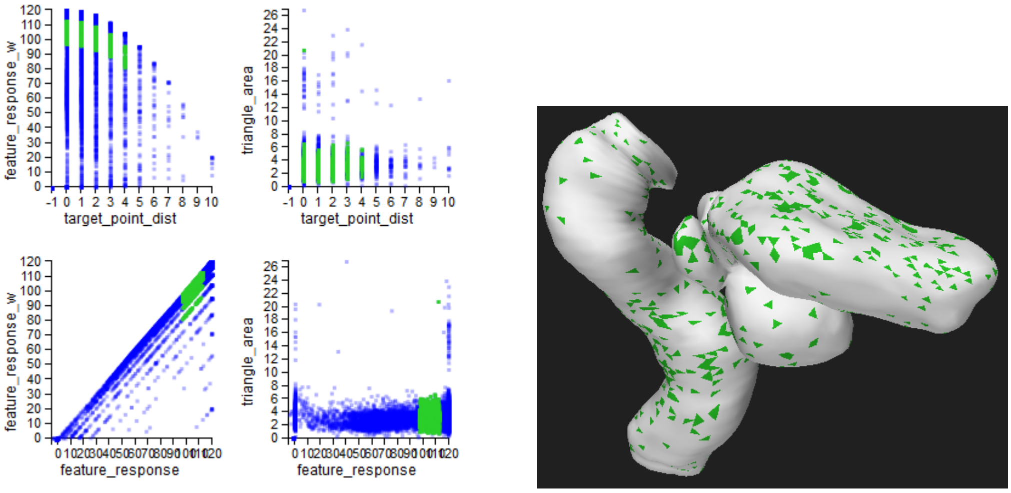
Selection visualization

In addition to the correlation detection techniques mentioned above, which rely on visually inspecting the static representation of attribute values, an added interactive component can prove useful for the efficient detection of correlations. To this extent, users can create selections of subsets of the data set which then influences the way this selection is displayed on all visualizations. This approach is commonly known as brushing and linking, as introduced by Becker et al. [BC87] and further discussed by Doleisch [Dol04], and allows users to explore the distribution of attribute values for a specific subset of the mesh intuitively.

Due to the nature of the data set, each attribute value is linked to a specific triangle in the mesh, and therefore a selection consists of a fixed set of triangles. There are multiple ways in which selections can be made, and selections have a different effect on each of the visualizations.

Each selection is defined as a list of triangle indices, and is assigned a unique color. The list of indices represents the contents of the selection, and the color defines how the selection is represented in the different visualizations. In the scatter plots, each point representing a triangle that is contained in a selection is drawn using the unique selection color, instead of the default color. Because this is done across all scatter plots, it is possible to see the distribution of selected triangles also in the plots where the selection has not been made. This functionality is illustrated in Figure 2.8a.

Similarly, in the three-dimensional representation, each triangle is colored corresponding to the selection it is a part of. Therefore, this representation can be used to visualize the spatial distribution of selected triangles, as is illustrated in Figure 2.8b.

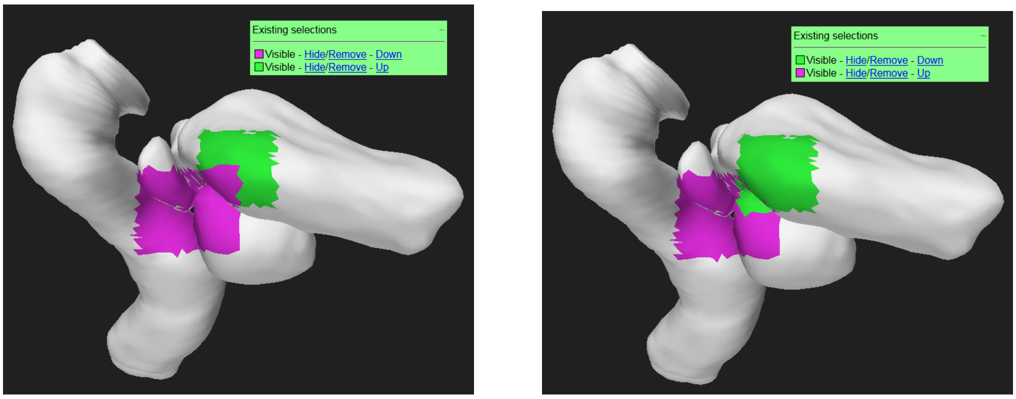


(a) A single selection, displayed in multiple scatter plots simultaneously

(b) A single selection, displayed in the three-dimensional representation

Figure 2.8: A single selection in green, displayed in multiple representations, namely a set of scatter plots (a) and the 3D-representation (b)

To handle cases where multiple overlapping selections exist, an order exists within the existing selections. In both the scatter plots and three-dimensional representation, the selection colors are applied in the order of this hierarchy, meaning that a triangle contained in multiple selections is displayed using the color of the selection highest in the order. Figure 2.9 shows the difference a change in the order can make for the way selections are represented.


Existing selections
Visible - Hide/Remove - Down
Visible - Hide/Remove - Up
Existing selections
Visible - Hide/Remove - Down
Visible - Hide/Remove - Up

(a) Two overlapping selections displayed on the mesh, where the green selection is higher in the hierarchy than the purple selection

(b) Two overlapping selections displayed on the mesh, where the purple selection is higher in the hierarchy than the green selection

Figure 2.9: Two overlapping selections, displayed with two different orders. Notice that in the overlapping section, the color of the selection highest in the order is used.

Creating selections

Making selections in the panel with scatter plots is done by clicking and dragging a rectangle in one of the graphs. This method of brushing and linking [BC87] is also described by Doleisch and Hauser in their paper "Smooth brushing for focus+context visualization of simulation data in 3D" [DH02]. The borders of the rectangle determine the ranges of attribute values included in the selection. Using an inverse of the mapping used for positioning a point in the scatter plot, the selected variable ranges can be computed.

After mapping two diagonally opposite corners of the rectangle back to attribute values, a minimum and maximum for both attributes is derived. These minima and maxima are then used as lower and upper bounds, thereby defining the selected subset.

In the histograms, a temporary selection is created by hovering the mouse pointer over one of the rectangles in the graphs. Alternatively, a permanent selection can be created by clicking on one of the bars.

Similar to creating selections in scatter plots, a rectangle in the histogram can be used to obtain a value range which can be used to select a subset of the triangles in the mesh. For an attribute with minimal and maximal values v_{min} and v_{max}, divided over n buckets, the index of a selected bucket i can be mapped to a value range as

$$(min, max) = (v_{min} + i \cdot (v_{max} - v_{min})/n, v_{min} + (i + 1) \cdot (v_{max} - v_{min})/n) \tag{2.5}$$

Since each histogram shows the distribution of values for a single attribute, the selection created by hovering over one of the bars is, in contrast to selections in the scatter plots, based on the value of a single attribute.

This selection method based on brushing is implemented bi-directionally. Besides creating selections by defining value ranges in one of the two-dimensional graphs, it is also possible to create a selection containing all triangles within certain spatial bounds in the three-dimensional representation. By selecting only specific regions of interest, it is possible to inspect the quality and correlations for only the part of the mesh the user is interested in. This is done similar to the way all points within a rectangle are selected in the scatter plots. By clicking and dragging on the 3D representation of the mesh, a rectangle is defined on the two-dimensional projection of the mesh. All triangles which are contained within this region on the projection form the new selection. This means that the position of a triangles vertices are not used to compute the selection. Using this method for selecting physical regions, it is also possible to define whether all triangles within the region, or only those not obscured by overlapping triangles should be considered. Figure 2.10 shows a mesh with three selections made by defining a rectangle on the 2D projection. Note that due to the nature of the selection method, the selected triangle together form the shape of a rectangle in the projection.

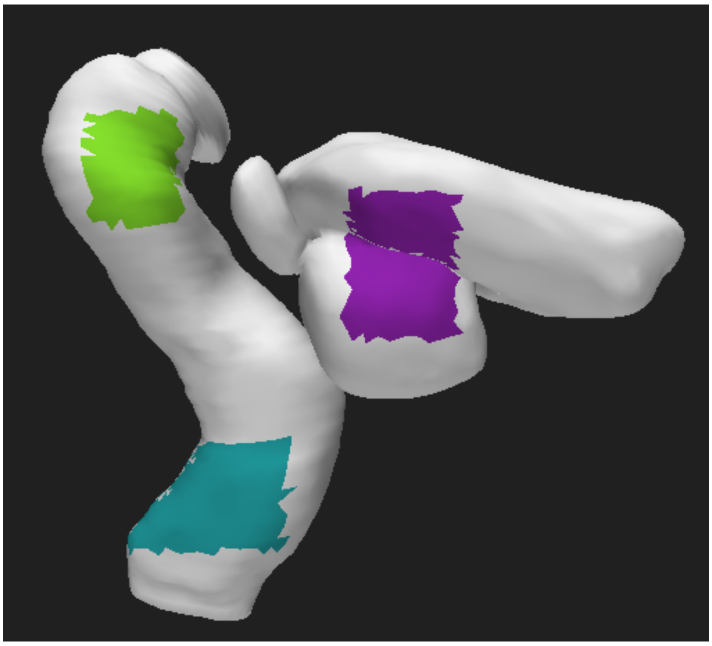


Figure 2.10: An image of the three-dimensional mesh with three selections created by defining spatial ranges on the 2D projection.

2.3 Slice data visualization

The 3D meshes resulting from the segmentation algorithm, are based on the images acquired by scanning the patients body using MRI scanning. This scanning results in a set of ‘slices’, two-dimensional arrays of intensities representing the densities in the patients body, By combining all slices into a single three-dimensional set, a 3D volume is constructed, from which 2D slices can be constructed in any desired plane. These slices are particularly valuable since they provide anatomically context for the displayed mesh. Figure 2.11 shows an example of slices through the sagital, coronal and transverse plane respectively.

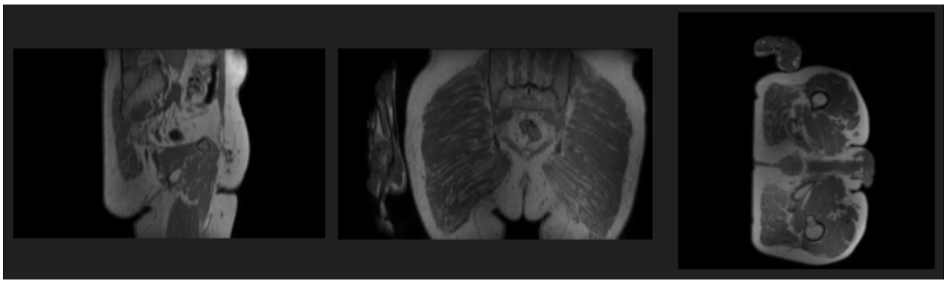


Figure 2.11: A screenshot of the application showing image slices for the sagital, coronal and transverse plane respectively.

In the ideal case, the boundaries contained in the mesh are perfectly aligned with boundaries visible in the image slice data. If they are not, there is a difference between the shape shown in the image and the positions of triangles in the resulting mesh. To facilitate the comparison between the image data and segmentation output, the slice images can be combined with the three-dimensional data into a single visualization. To do so, there are two possible methods.

Firstly, slices can be mapped onto planes in the 3D representation of panel B. When the plane is placed in exactly the same location as represented by the slice data, then the outlines of the mesh should line up with the boundaries visible on the slice plane. Figure 2.12 shows an example of a 3D mesh with three planes representing the coronal, saggital and transverse slices through a specified point in three-dimensional space.

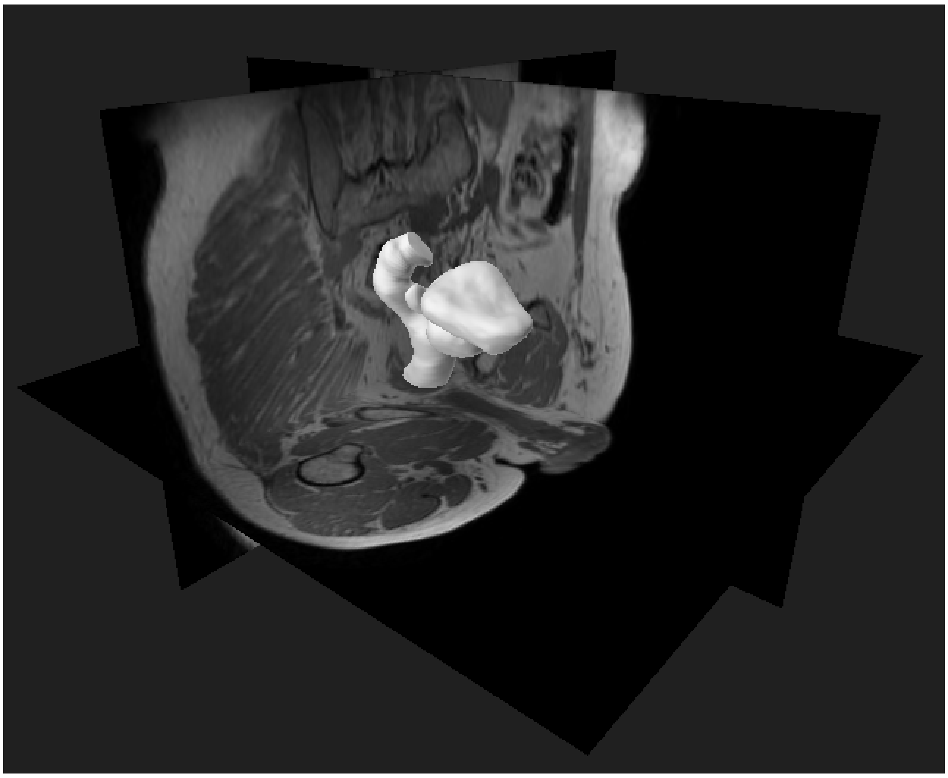
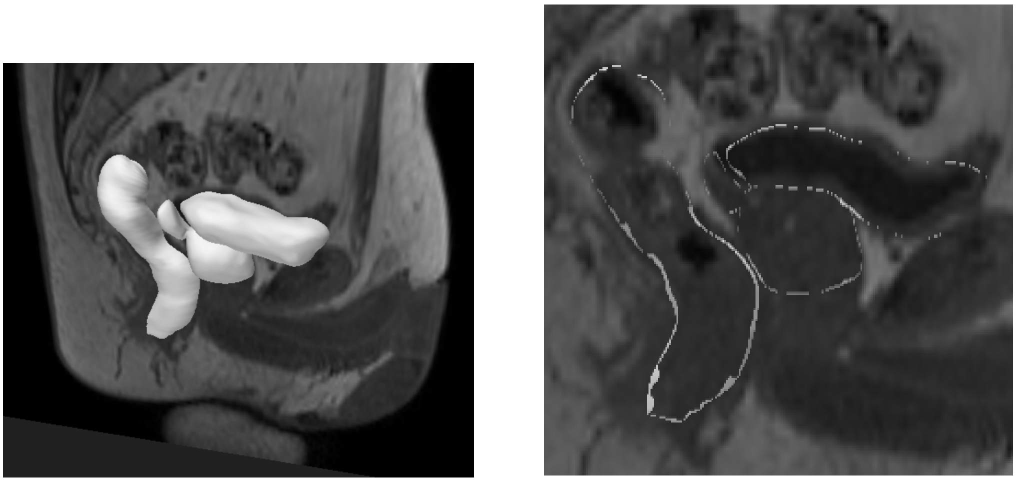


Figure 2.12: A screenshot of the application showing the sagital, coronal and transverse image slices in the three-dimensional representation.

Secondly, to avoid the perspective from distorting the users perception of the image data, the slice data can also be presented in a two-dimensional environment using orthogonal projection. Comparing the mesh with the slice images is done by adding all triangles intersection the slice, as shown in Figure 2.13. This results in showing an outline of the mesh which should ideally correspond to the organ boundaries visible in the slice. Because instead of only the boundary of the intersection, the entire intersecting triangles are shown, information about the orientation of the intersecting surface is preserved.



(a) A screenshot of the three-dimensional representation of the organ mesh combined with a plane with the sagital plane.

(b) A screenshot of the two-dimensional orthogonal representation of the image slice, combined with a projection of all triangles intersecting the slice plane.

Figure 2.13: Two screenshots, showing the three-dimensional mesh and two-dimensional image slice plane both in a three- and two-dimensional representation.

2.4 Profile inspection

As explained in Subsection 1.6, the profile data contains a set of 21 feature response values for each of the triangles in the mesh, for 10 evenly spaced positions on either side of the triangle and the response for the location of the triangle itself. These responses are numeric values indicating how likely, based on the information in the imaging data, it is that a boundary should occur on a given position. In the segmentation phase these response values are used to determine the optimal position of individual triangles in the mesh. Therefore, when a profile consists mostly of weak responses and one significantly stronger response, it is likely that this location indeed is the desired boundary location. However, it is also possible that multiple peaks exist in the response profile, or none at all.

In these cases, the segmentation algorithm either has to rely on the profile information of neighboring triangles, in the case where no peaks exist, or select one of the peaks in the response profile and ignore the others. Since these feature responses are so influential for the shape of the resulting mesh, it is likely that response profiles with not exactly one peak cause inaccuracies in the resulting mesh. However, manually inspecting individual profiles is not feasible due to the large number of triangles in the mesh, and in addition would not provide enough information as the position of a triangle depends not only on its own feature response profile, but also on those of its neighbors.

Peak detection

Therefore, it is desirable to use the number of peaks in the response profiles as an additional attribute in the visualizations. Using an extra attribute allows the user to identify regions with similar properties. For example, a region where the vast majority of triangles has a response profile with a single peak is more likely accurate than a region with multiple feature response peaks per triangle. To compute the number of peaks in a response profile, first a gaussian smoothing step, as described in the "Digital image processing" book by Gonzalez et al. [GW02], is applied to avoid noise from influencing the result. Next, a discrete derivative is used to identify local maximums. The gaussian function used for the smoothing step $gaussian(\sigma, d)$ is

$$gaussian(\sigma, d) = \frac{1}{\sigma\sqrt{2\pi}} \cdot exp(-\frac{d^2}{2\sigma^2}) \tag{2.6}$$

where σ is the standard deviation and d is the distance from the mean μ, which is 0. In this thesis, a standard deviation of $\sigma = 1$ is used. This value for σ is chosen based on the length of the profiles used, where using σ significantly smaller than 1 is not sufficient to remove small variations in the response profile, resulting in a too large number of peaks being detected. Using σ much larger than 1 on the other hand, smooths the profile too strongly, resulting in a too strong decrease in the number of peaks detected. Each feature response r in the profile is replaced by the sum of all responses in the profile weighted using the gaussian function of the distance to r. This gives

$$profile[r] = \sum_{i=0}^{|profile|} profile[i] \cdot gaussian(\sigma, r - i) \tag{2.7}$$

However, the sum of all the weights used for the smoothing of a response value is not the same for all places in the profile. This is because for all responses r, the result of the gaussian weighting function depends on the distance between r and all other responses in the profile, causing responses in the middle of the profile to have a higher sum of weights than responses towards the edge of the profile due to the profile length being finite. To compensate this, for each feature response in the profile the sum of all weights is computed, and used to scale the new feature response value.

$$weightSum(r) = \sum_{i=0}^{|profile|} gaussian(1, r - i) \tag{2.8}$$

$$profile[r] = \frac{profile[r]}{weightSum(r)} \tag{2.9}$$

After applying the smoothing step, we compute the forward finite derivative per response value Δr as

$$\Delta r = profile[r + 1] - profile[r] \tag{2.10}$$

and then use these derivatives to count each local maximum in the smoothed profile as a peak. Since this is a derivative of a discrete function, it is not possible to find the exact location where the derivative is zero, so we simply look for neighboring responses in the profile r_1 and r_2 where the derivative $\Delta r_1 > 0$ and $\Delta r_2 \leq 0$.

Profile clustering

Alternatively, instead of reducing the profile information to a single number representing the number of peaks, another approach to efficiently convey the profile information to the user, is to construct clusters of profiles with a similar shape. These clusters can then be represented by a single profile representative, such that a displayed profile does not correspond to a single triangle, but to a larger set of triangles.

In the user interface, each cluster is depicted in a one dimensional visualization onto which the clusters average profile is color mapped, as depicted in Figure 2.14. In this visualization, the values in the profile are normalized to the range [0..1], and mapped linearly to a color scale from black to white.

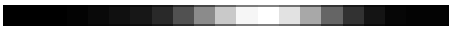


Figure 2.14: An example of a visual representation of a clusters average profile. Profiles of length l, are represented as horizontal bars containing l squares color mapping the profile normalized to the range [0..1] such that black represents a normalized feature response of 0 and white represents 1.

In order to satisfy the needs of all types of users, a hierarchical clustering method is used. This means that initially, the number of clusters is equal to the number of triangles, followed by a phase where iteratively the two most similar clusters are merged. This results in a cluster hierarchy, where most clusters are merely a combination of two smaller clusters.

In the visual representation of the clusters, each cluster can be expanded, revealing the two underlying clusters, again with a similar visualization for both, as shown in Figure 2.15. This allows the user to inspect the contents of the clusters interactively, and the number of clusters or the number of triangles per cluster does not need to be defined prior to the clustering phase.

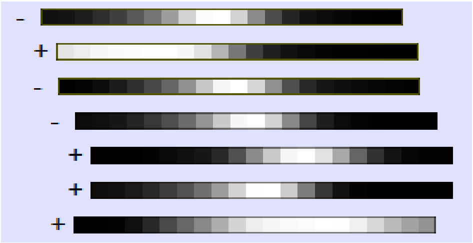
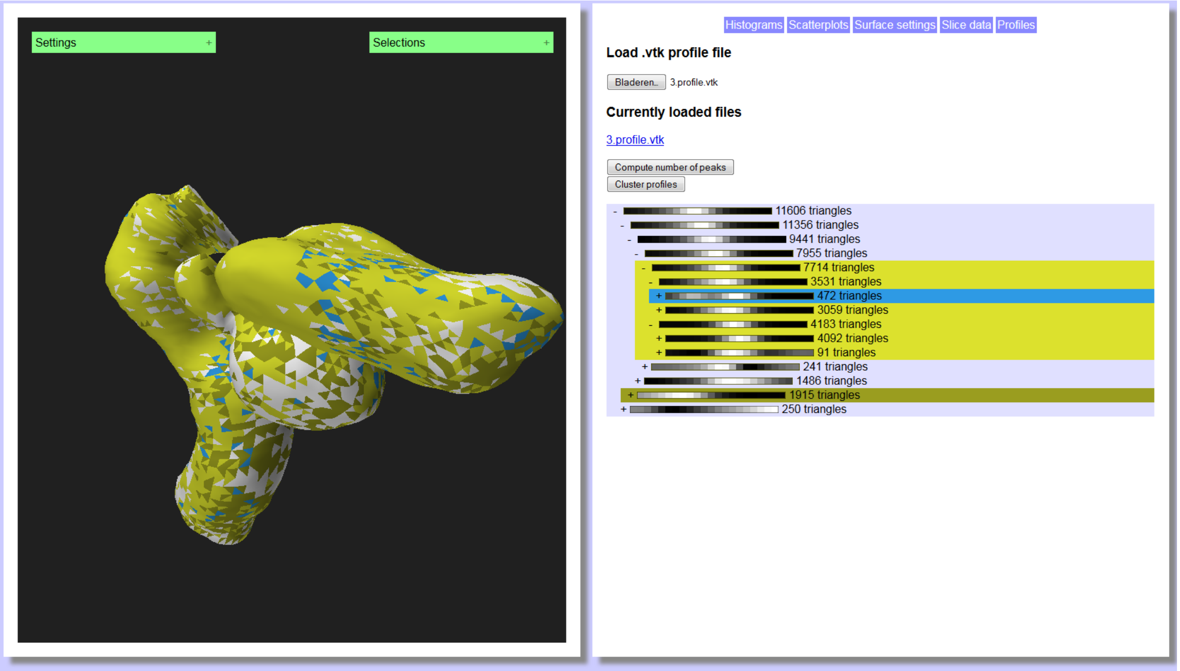


Figure 2.15: A schematic representation of the hierarchical structure of the profile clusters. For profiles of length l, the horizontal bars contain l squares color mapping the profile normalized to the range [0..1] such that black represents a normalized feature response of 0 and white represents 1.

To inspect where the triangles contained in a cluster are located on the mesh, the user can hover the mouse pointer over one of the horizontal bars. Upon doing so, all triangles are colored light blue on the mesh. In order to create permanent selections, used for comparing the distribution of clusters or visualizing the triangles contained in a cluster in the scatter plots, the user can click on the bar representing a cluster. A new selection is then created, and the color used to represent members of this selection is used as the background color for all clusters in the hierarchy of the selected clusters, as illustrated in Figure 2.16.



Load .vtk profile file

3.profile.vtk

Currently loaded files

3.profile.vtk

```
-  11606 triangles
 -  11356 triangles
  -  9441 triangles
   -  7955 triangles
    -  7714 triangles
     -  3531 triangles
      +  472 triangles
      +  3059 triangles
     -  4183 triangles
      +  4092 triangles
      +  91 triangles
    +  241 triangles
   +  1486 triangles
  +  1915 triangles
 +  250 triangles
```

Figure 2.16: A screenshot of the application showing a part of the cluster hierarchy on the right, where three selections have been made, and the mesh showing the physical distribution of triangles in the selected clusters on the left.

The computation of clusters with similar profiles is done using a hierarchical bottom-up clustering method, as introduced by Ward in his paper "Hierarchical grouping to optimize an objective function" [WJ63], which uses a similarity measure to combine low level clusters, representing only several profiles, to higher level clusters until only one single cluster remains representing the entire set of profiles.By storing the hierarchical structure of the clusters, it is possible to interactively browse through the hierarchy.

A pseudocode version of this algorithm is provided in Algorithm 2.1, and will be elaborated upon in the remainder of this section.

Algorithm 2.1: The hierarchical clustering algorithm

```
input  : A set of smoothed profiles $profiles$
input  : A ratio r in [0..1] specifying what part of the profiles should form the initial
         clusters
output: A cluster C which is the root of the cluster hierarchy
for $0 \leq i <$ profiles.length do // Create initial clusters
    if random(0,1) $<$ ratio then
        clusters.add(new Cluster(profiles[i]));
    else
        nonClusters.add(profiles[i]);
while nonClusters.size $> 0$ do // Assign profiles to initial clusters
    p = nonClusters[0];
    bestSim = 0;
    mostSimilarCluster = null;
    for c in Clusters do
        if similarity(p, c) $>$ bestSim then
            bestSim = similarity(p, c);
            mostSimilarCluster = c;
    mostSimilarCluster.add(p);
    nonClusters.remove(p);
for $0 \leq i < j <$ clusters.length do // Compute cluster similarities
    similarities[i][j] = similarity(clusters[i], clusters[j]);
while clusters.size $> 1$ do // Iteratively merge clusters
    x = -1;
    y = -1;
    bestSim = 0;
    for $0 \leq i < j <$ clusters.size do
        if similarities[i][j] $>$ bestSim then
            x = i;
            y = j;
            bestSim = similarities[i][j];
    clusters.add(merge(clusters[x], clusters[y]));
    clusters.remove(clusters[x]);
    clusters.remove(clusters[y]);
    similarities.removeRow[x];
    similarities.removeColumn[y];
    for $0 \leq i <$ cluster.size-1 do // Compute similarities with new cluster
        similarities[i][clusters.size-1] = similarity([i][clusters.size-1]);
return clusters[0];
```

The hierarchical clustering uses the similarity measure described below, which takes parameters p_1 and p_2, which are two profiles of equal length normalized to only contain intensities in the range [0..1]. This normalization is applied because due to the way the actual feature response profiles are used to position the triangles, the absolute value is irrelevant. The features used to compute response profiles differ per triangle, and the segmentation algorithm uses a relative comparison between the response values to determine the target point.

$$similarity(p, q) = \sum_{i=0}^{p.length} 1 - |p[i] - q[i]| \tag{2.11}$$

Because two profiles with peaks at nearly the same, but not identical, profile indices should get a higher similarity score than two profiles with peaks further apart, the similarity between feature response profiles of length l can not be considered as the distance between two points in l-dimensional space. However, even with the similarity measure described above, it is still possible for two profiles with peaks at near identical positions to be assigned a low similarity.

To solve this, prior to computing the similarity between two feature response profiles, the same smoothing step used for computing the number of peaks is applied to both profiles. This causes peaks to be transformed into bell curves spanning multiple indices in the profile, increasing the similarity between profiles with peaks at similar indices.

To compute the similarity between two clusters, the same metric is used. However, instead of the normalized feature response profile, a representative profile is used, which is the average of all profiles in the cluster.

Finally, once the pairwise similarities for all pairs of clusters are computed, the two most similar clusters can be combined into a single cluster, after which the matrix with pairwise distances is updated. When two clusters merge, the profiles both clusters represent are averaged, resulting in a profile representative for the contents of the new cluster. This representative is then also used to compute the similarity between the new cluster and other clusters or profiles.

Instead of using a hierarchical clustering method, which uses a greedy approach, an alternative such as K-means clustering [HW79], which generally performs better when defining homogenous clusters, could also have been used. However, because determining the optimal value of K may be time consuming, and especially because the result of K-means clustering does not provide information on the contents of a cluster, such as the similarity between subsets of the triangles in a cluster, a hierarchical clustering method is more suited for executing the users tasks.

One of the downsides of using a hierarchical clustering method that uses a bottom-up approach, is that in the initial phase there are as many clusters as there are profiles in the data set, resulting in $\frac{n^2}{2}$ pairs for which similarities must be computed in a profile set with n entries. To decrease the computation time required for the initialization, a modification is applied such that only a fraction of the triangles is assigned a cluster. All triangles not assigned a cluster are then merged with the existing cluster to which they are most similar.

This decreases the computation time of the initialization significantly. Although in the commonly used 'big O notation' [Knu76] both versions still take $O(n^2)$ time, the expected number of instructions executed decreases from $\frac{n^2}{2}$ to $((1 - r) \cdot n) \cdot (r \cdot n) + \frac{(r \cdot n)^2}{2}$ where n is the number of profiles in the data set and r is the ratio of profiles initially assigned a cluster.

Because this decreases the quadratic component in the running time and only increases a linear component, this modification drastically decreases the initialization time.

Moreover, after the initialization phase is done, we now only have to execute a hierarchical clustering with $(r * n)$ instead of n clusters, speeding up the remainder of the clustering algorithm as well.

3 Results

All design choices described in Section 2 combined result in a framework for the interactive data exploration and hypothesis discovery and verification for mesh segmentation results. The features contained in this framework have been selected such that the tasks described in Subsection 1.3 can be executed efficiently. In this section, we will elaborate on two simulated examples of real world use cases, showing both how the framework is used and that it indeed allows the user to executed the proposed set of tasks.

3.1 Use case 1

A technical researcher inspects the mesh representing the organs of patient 18, without prior knowledge of the segmentation results. Any anomalies contained in this data set will have to be investigated, as they can potentially indicate problems with the segmentation algorithm.

The user starts by opening a file containing the mesh of patient 18, combined with the attributes resulting from the segmentation phase (‘target_point_dist’, ‘feature_response’, ‘feature_response_w’, ‘triangle_area’). By default, the application shows a three-dimensional representation of the mesh and a set of histograms depicting the value distribution of the attributes, see Figure 3.1.

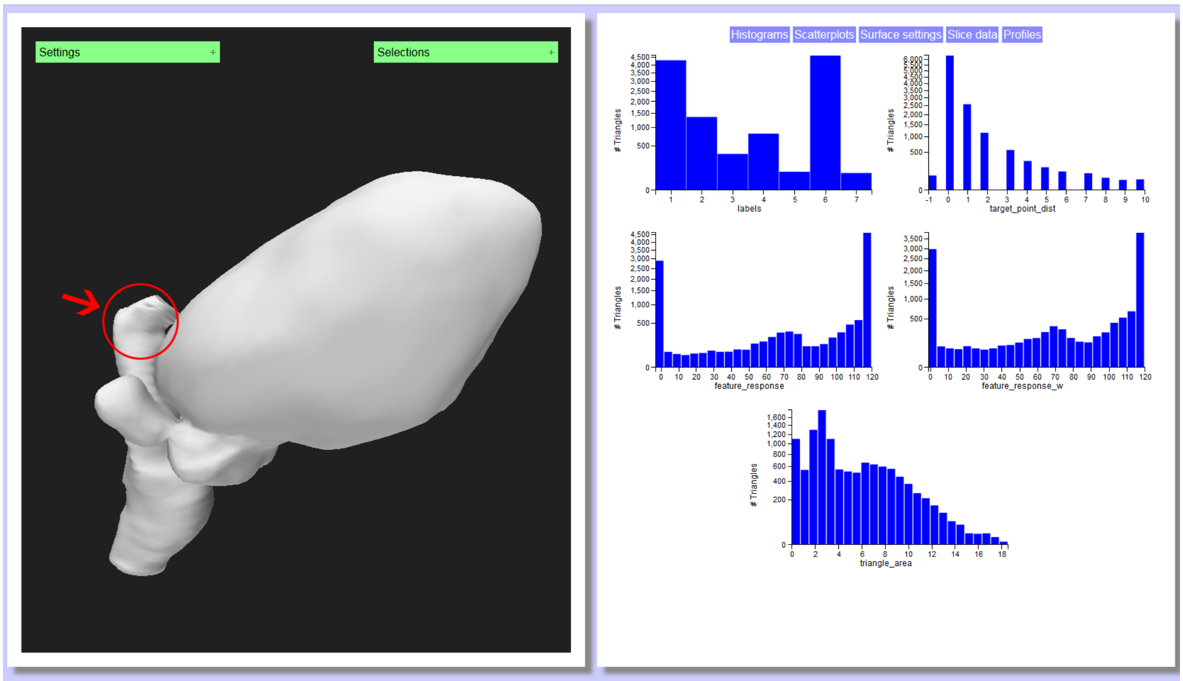
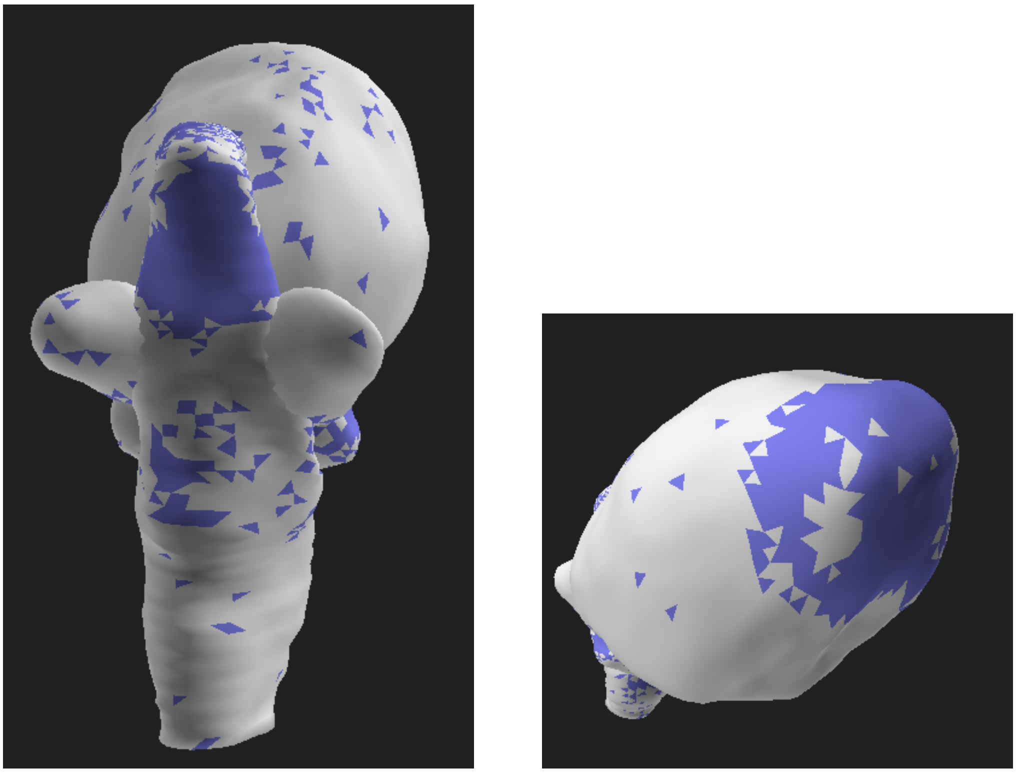


Figure 3.1: Use case 1: a screenshot of the application showing the three-dimensional representation of the organ mesh on the left, and a histogram for each attribute on the right. The red circle and arrow indicating a part of the rectum with irregular shape are added manually, and not part of the original screenshot.

Inspecting the mesh in the three-dimensional representation quickly reveals two unusual properties. Firstly, the bladder is extraordinarily large, and secondly, the rectum is rather irregularly shaped towards the top, where there is a sharp bend (indicated with a red arrow and circle in Figure 3.1).

From the histograms, it can be seen that there is a large peak in the distribution of the attribute 'feature_response' at the bar representing values of 0 and lower. Since a negative response is reported when no suitable feature could be discovered, this is a first indication that there may have been a problem with the features during the segmenting phase. Hovering the mouse over that bar shows that the triangles associated with this bar are indeed located towards the top of the rectum and on the part of the bladder furthest removed from the rest of the organs, as illustrated in Figure 3.2b and 3.2a. This strengthens the hypothesis that the organ set may be shaped as it is due to inaccuracies of the algorithm instead of simply the organs of the patient being shaped as they are displayed.

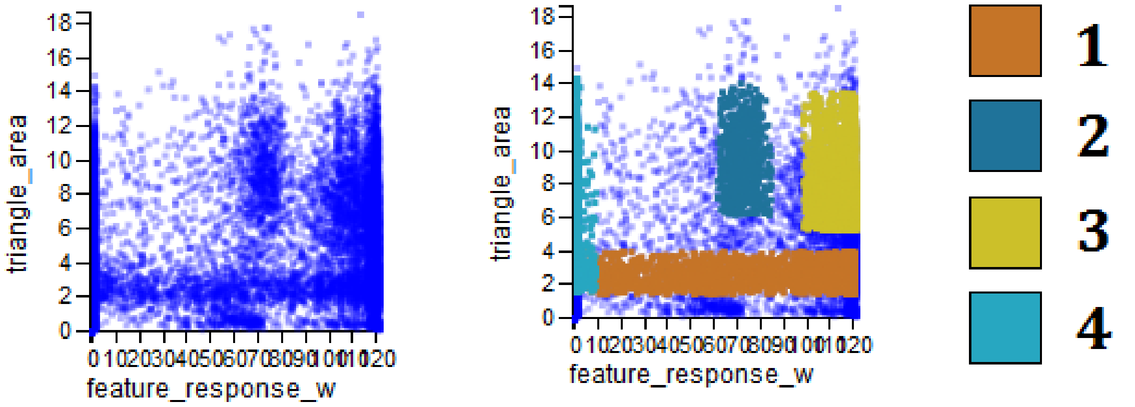


(a) A posterior view of the mesh of patient 18, with triangles with a low feature response highlighted in blue.

(b) An anterior view of the mesh of patient 18, with triangles with a low feature response highlighted in blue.

Figure 3.2: Use case 1: the mesh representing patient 18, with triangles with low feature response highlighted, as seen from two different angles.

Upon inspection of the scatter plot, several clusters of points are most easily identified when plotting the weighted feature response ('feature_response_w') against the triangle area ('triangle_area'), as depicted in Figure 3.3a; one with low response values, one with low triangle area, one with high feature response and one cluster in the middle with average feature response and above average triangle area. Highlighting these clusters by defining rectangles on the scatter plot, as shown in Figure 3.3b, provides insight in the physical location of these clusters, displayed from four angles in Figure 3.4, revealing several counter-intuitive aspects.



(a) The scatter plot mapping the weighted feature response against the triangle area without selections.

(b) The scatter plot mapping the weighted feature response against the triangle area with four selected clusters; one with low response values, one with low triangle area, one with high feature response and one cluster in the middle with average feature response and above average triangle area.

(c) A legend used to map each of the selections to a number.

Figure 3.3: Use case 1: A scatter plot, plotting the weighted feature response ('feature_response_w') against the triangle area ('triangle_area').

The red selection, with number 1, consisting of a set of triangles with an area slightly below the global average, spanning all weighted feature response values, is spread evenly across the rectum, prostate and seminal vesicles. Interestingly, nearly none of these triangles are part of the bladder.

The dark blue selection, number 2, containing triangles with an area larger than average and a weighted feature response between 65 and 90, consists of triangles located mainly on lateral sides of the bladder and on the prostate and rectum. Although having triangles with above average surface area along the sides of the bladder may suggest that the bladder is indeed stretched to a larger size than it usually is, it is not yet possible to verify such a claim based on the evidence discovered so far.

The light green selection, number 3, covers a large portion of the bladder, with exception of the part of the bladder covered by selection 2 and a portion in the part furthest removed from the other organs in the set, and several triangles scattered across the top of the prostate the the seminal vesicles.

Finally, the light blue selection, number 4, contains a set of triangles scattered across all organs, plus two areas where a relatively large set of adjacent triangles are all selected. This is at the posterior part of the rectum located higher than the seminal vesicles, and the anterior portion of the bladder furthest removed from the other organs.

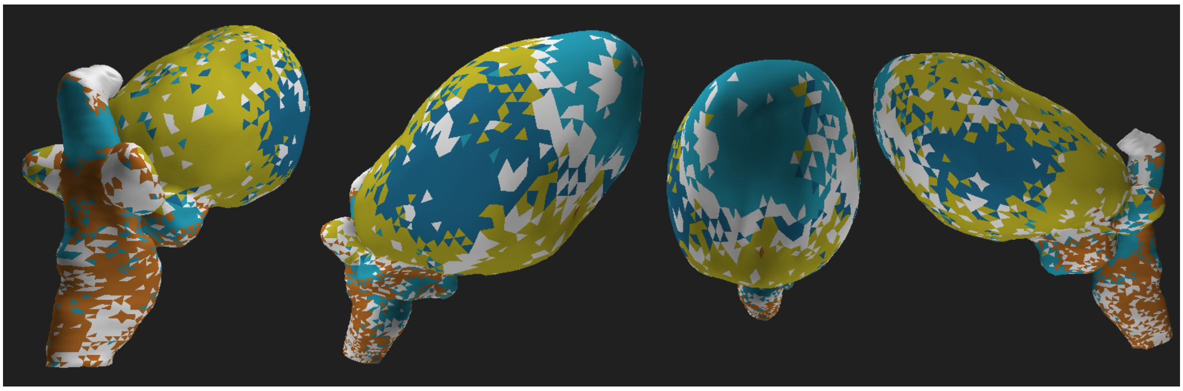


Figure 3.4: Use case 1: The organ mesh, with the color of the triangles representing the selected clusters they are part of. Note the presence of a large area of light blue at both the unusual shapes; near the top of the rectum and the tip of the bladder.

This last selection seems to be related to the errors discovered upon visual inspection of the mesh, as the two problems identified (an irregularly shaped top of the rectum and unusually large bladder) affect exactly the two areas consisting of light blue triangles and the light blue selection consists of triangles with low feature response. This suggests that the lack of strong features in these areas may be responsible for the errors in the mesh.

To investigate this further, the feature response profile information of this patient is incorporated into the visualizations. After loading the profiles, the number of peaks per profile is computed and inspected in a histogram, see Figure 3.5. The number of triangles of which the feature response profile did not contain any peak is remarkably high with just over 2500 triangles, or more than one fifth of the entire mesh. The response profiles are used to determine the optimal location for a triangle, so an absence of peaks in such a profile indicates that no information was available on how to modify the triangles location. Additionally, this means that not only is there no information on how to move the triangle, it also indicates that the current location of the triangle is not supported by any of the features.

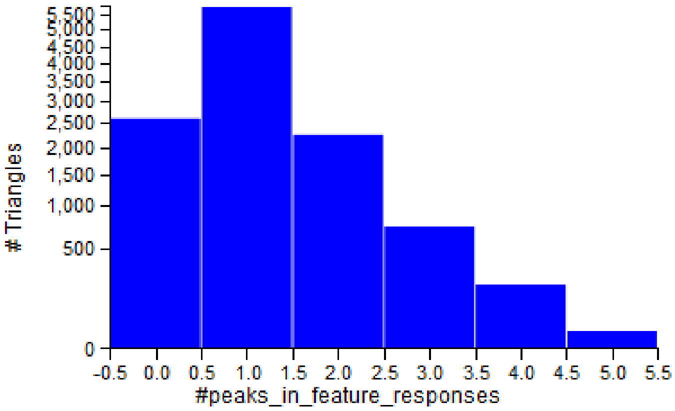


Figure 3.5: The histogram displaying the value distribution for the number of peaks per response profile. The number of profiles with no peak in the response profile, represented by the leftmost bar, is remarkably high, as a profile without peaks suggests that there is little information available to determine where a triangle should be placed.

In practice, it is acceptable if a triangle has a feature response profile without peaks as long as most of the neighbors do not lack response profile peaks, since the position of neighboring triangles can positively influence a triangles position. Therefore, areas where triangles with feature response peaks occur frequently indicate potential accuracy problems.

Upon adding a new selection, containing only triangles with a feature response no higher than 10, and zero feature response peaks, it shows that this new selection overlaps near perfectly with the light blue selection created earlier. This is illustrated in Figure 3.6.

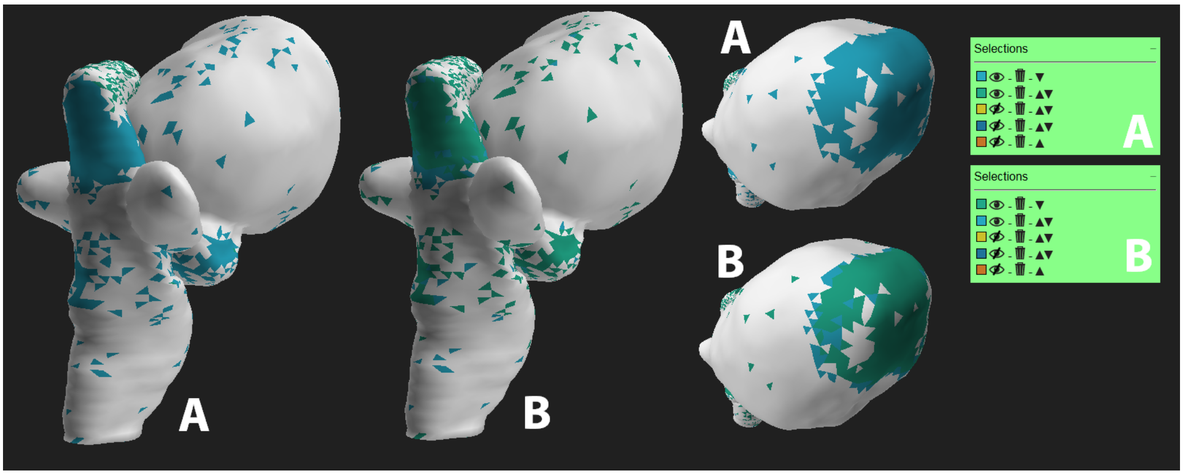


Figure 3.6: The mesh with two selections, one light blue selection containing all triangles with low feature response in light blue, and one containing all triangles with both low feature response and zero feature response peaks, in mint green. In the meshes and menu labeled A, the light blue selection is first in the selection order, causing triangles contained in both selection to be displayed as light blue, and in the meshes and menu label B, the mint green selection is first in the selection order.

From this correlation between the irregularities in the mesh and the absence of feature response peaks, it can be concluded that this patient-specific unusual shape of the organ mesh can most likely be attributed to imperfections either in the provided imaging data or in the segmentation algorithm. Due to the absence of imaging data for this patient, the question which of these two options is correct and, if the latter is the case, how this can be avoided, is outside the scope of the framework.

3.2 Use case 2

A clinical researcher is presented with a mesh resulting from the segmentation algorithm applied to the organs of patient number three, who is being treated for prostate cancer. In order to safely maximize the dose of radiation applied, an estimation of the reliability of the mesh is required. Although the clinical researcher is not the clinician determining the details of the treatment, information about the reliability of the result of the segmentation algorithm is valuable for studying the effects of radiation in patients. In this case, the inspection focuses on the boundaries of the prostate, with the interfaces between the prostate and neighboring organs (bladder, seminal vesicles and rectum) as locations of increased interest.

The user starts by loading the available data, which contains a 3D representation of the patients organs, combined with five attributes. For each triangle, these attribute values specify an organ label, the target point distance, the feature response strength, a weighted feature response strength and the surface area of the triangle. As explained in use case 1 (Subsection 3.1), a low (weighted) feature response can be an indication that the location of a triangle may deviate from the true organ boundary. This can be attributed to the fact that sampled positions with a high feature response are used as guides to determine the optimal location for individual triangles, hence an absence of strong responses signals an absence of guidance. Upon highlighting the bar representing a feature response of five or below in the histograms, it is visible on the mesh that although there are triangles on the prostate with low feature response on the boundary of the prostate, they are only few and fairly evenly spread. This distribution is shown in Figure 3.7.

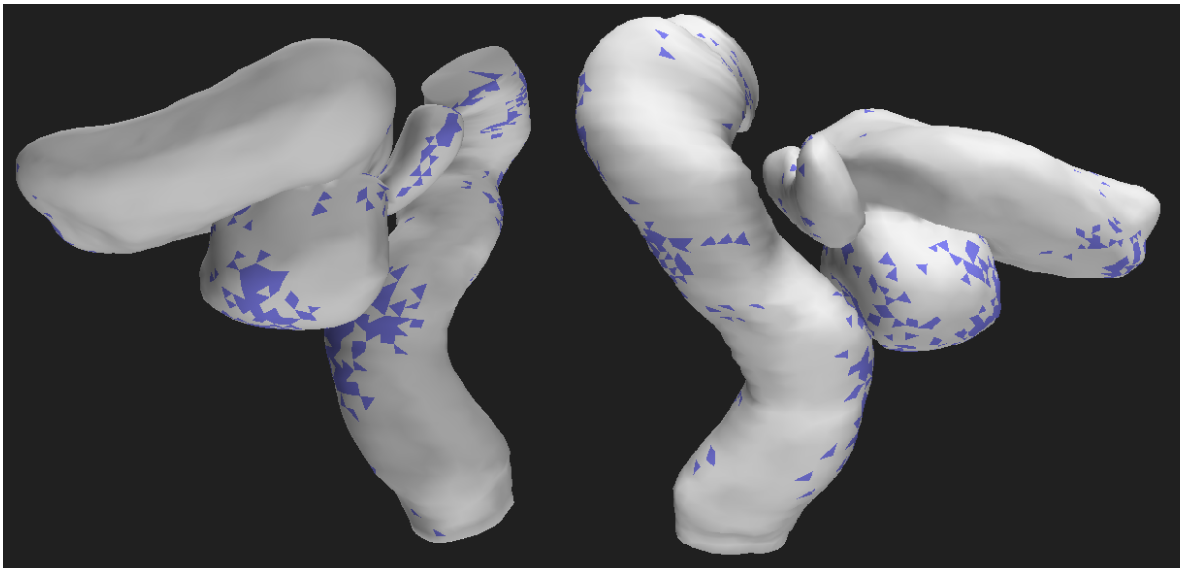


Figure 3.7: The same mesh shown from two different angle, with all triangles having a feature response below 5 highlighted in blue.

Judging by distribution of low and high feature response values, it is not yet possible to estimate the accuracy of the organ mesh. Since both high and low feature responses are spread evenly on the prostate, a reliable segmentation result is possible, but not inevitable. Therefore, the target point distance is also inspected. As the user now is not trying to visualize the distribution of a specific range, but the distribution of all values of an attribute at once, color mapping the value distribution onto the mesh is most efficient. To avoid distracting the user with clutter caused by color mapping organs not relevant for the case, selections are made such that all organs except for the prostate, bladder-prostate interface and seminal vesicles-prostate interface are displayed in a single color. The result of this approach is shown in Figure 3.8, where the mesh is shown from two angles, accompanied by an image of the seminal vesicles-prostate interface and the bladder-prostate interface.

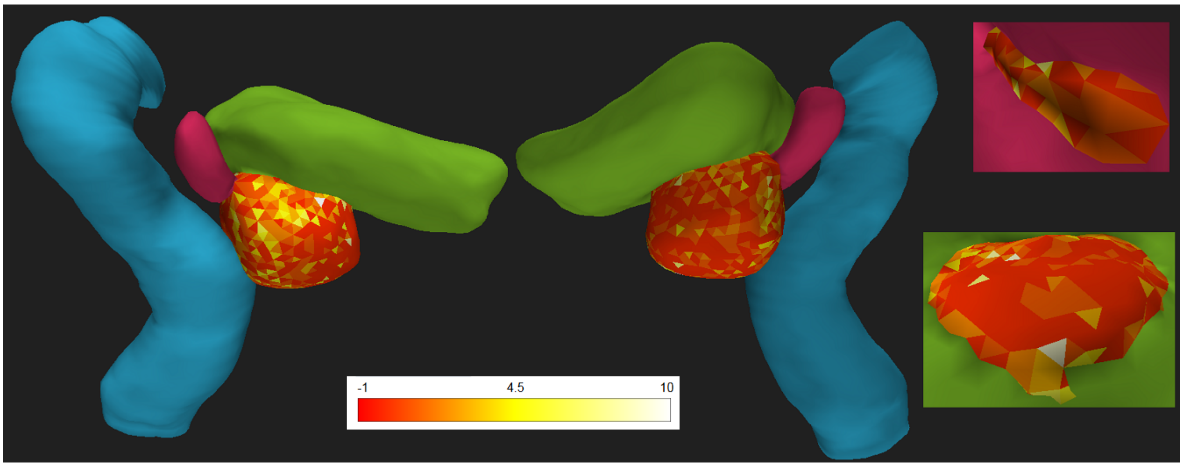


Figure 3.8: The patient mesh with the target point distance color mapped onto the prostate and the rest of the mesh obscured by single color selections. In the top right, the color map on the seminal vesicles-prostate interface is shown, and on the bottom right the bladder-prostate interface. On the bottom of the image, the legend for the target point distance color map is shown, ranging from -1 (red) via 4.5 (yellow) to 10 (white).

From this visualization, it can be concluded that although the vast majority of the triangles composing the prostate has a relatively low target point distance, with values of 4 mm and lower, there are several outliers with significantly higher target point distances. However, the initial exploration of the feature response distribution had revealed that a small part of the triangles in the prostate had a low feature response. Only when there is a reliable guidance for the optimal triangle position available, the target point distance is relevant, so by creating another selection using the scatter plots all triangles with a low feature response are covered by a single color as well, resulting in the image presented in Figure 3.9. The main contrast between these two images is that the majority of outliers with high target point distance, which are the triangles on the prostate with the lightest color, are now obscured. This means that the average target point distance of triangles with a reliable position guide is lower than we initially expected, and that the boundary of prostate is quite accurate.

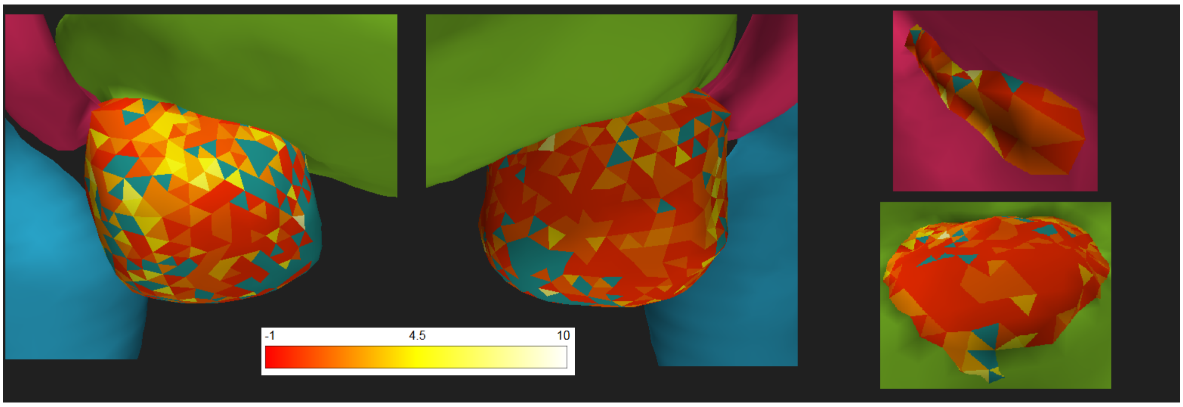
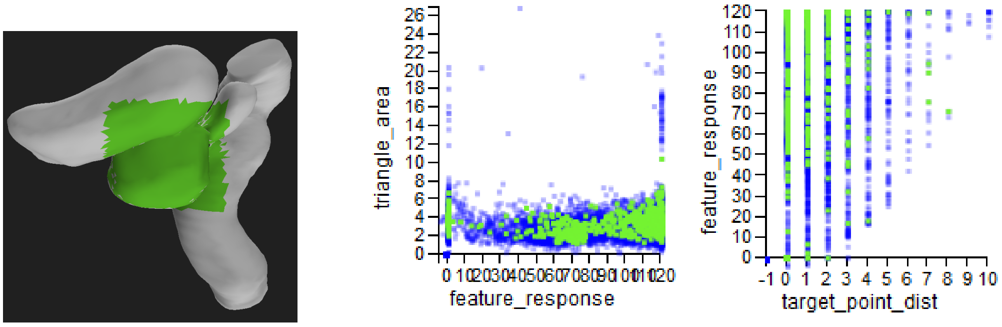


Figure 3.9: The same mesh, image angles and color map as in Figure 3.8, only this time with also all triangles with a feature response below 30 obscured by a color not part of the color map legend. The main difference between the two images is that the majority of outliers with high target point distances are now obscured.

Now that we have a reasonable amount of evidence that the boundary of the prostate is accurate in the segmented mesh, it is relevant to consider the accuracy of the neighboring structures. Besides applying the same technique as before, another approach to investigate the segmentation quality is to select triangles within physical bounds and inspect the attribute value distributions in the scatter plots. Figure 3.10a shows the selection containing the prostate and a small part of the adjacent organs in the three-dimensional representation, selected by brushing the mesh from an angle that allows selecting only the desired triangles, and Figure 3.10b shows the distribution of the selected triangles in the scatter plots mapping the feature response against the triangle area and the feature response against the target point distance. From the first of these two scatter plots it can be deduced that, as is the general trend in the mesh, the majority of triangles has a high feature response, and from the latter of the two scatter plots it can be confirmed that little to none of the selected triangles with a high feature response also have a high target point distance.



(a) A selection containing only triangles physically close to the prostate displayed on the mesh.

(b) The selection from Figure 3.10a displayed in two scatter plots, mapping the feature response against both the triangle area, used to get a sense of the general distribution of feature response values in the selection, and against the target point distance.

Figure 3.10: A selection containing only triangle within close proximity of the bladder, displayed both on the mesh and in scatter plots mapping the feature response distribution against the triangle area and target point distance

Combining the evidence obtained from inspecting the distribution of target point distance and feature response values in the prostate with the information gathered concerning the distribution of the same attributes in neighboring tissue, the user can conclude that the boundary of the prostate presented in the segmented mesh is accurate to a large extent. This allows the clinical researcher to more accurately study the results of radiation therapy on prostate cancer patients.

3.3 Evaluation

To evaluate to which extent the presented framework satisfies the users needs, we will investigate if and how each of the tasks specified in Subsection 1.3 can be executed. In this initial evaluation, we wanted to assess how our framework supports analysis and reasoning about the data and helps to derive relevant knowledge for its intended users. The evaluation, however, is based solely on the tasks from Subsection 1.3, without assessment by the intended users.

The tasks lists were originally presented for both types of users, technical researchers and clinical researchers, separately. However, since they overlap on multiple tasks, in this section those two task lists are presented merged into one list without duplicates.

- What is the shape of the organs in the data set?

  Upon loading a data file, the 3D mesh contained is displayed on the panel with the three-dimensional representation, on the left of the screen. Using the mouse buttons, the user can interact with this visualization by manipulating the virtual camera. With rotating, panning and zooming the camera, the user is enabled to inspect the mesh from all sides, both globally and close-up. This allows the user to inspect the global shape of the organs, and examine specific areas of interest in detail.

- How are the values of each attribute distributed over the value range?

Inspecting the value distribution of individual attributes can be done by inspecting the histograms displayed on the panel on the right. For attributes containing a limited number of discrete values, each bar in the histogram represents one of these values, and for attributes with either a large number of values or using continuous values, binning is applied. These bins are then used to define a limited number of value ranges such that every bar represents a specific range.

An alternative approach to inspecting the value distribution of attributes is to inspect the scatter plots mapping the desired attribute. This way, when the user ignores one dimension of the position of the points, an overview of the value distribution without binning is provided. However, due to the number of data points often exceeding the number of pixels in the scatter plots, the histograms usually are more efficient for estimating the number of data points with a specific attribute value regardless of the binning.

Both these methods can also be used to inspect the distribution of the number of peaks per response profile. After loading a feature response profile data set, the number of peaks per profile can be computed and stored as a new attribute.

- How are the values of each attribute distributed spatially?

  Inspecting the physical distribution of attribute values, there are multiple approaches. Which of these is preferable depends on the specific interests of the user.

  If the user is focussing on the physical position of specific attribute values, the histograms and scatter plots can be used. In the histograms, the user can highlight triangles by hovering over the bars of the histogram, showing the location of all triangles with a value within the range represented by the bar for the attribute the users hovers over. In the scatter plots, the user can create selections by drawing a rectangle on the plots. This provides two value ranges for the two attributes displayed in the scatter plot, and the newly created selection is then displayed in the three-dimensional representation of the mesh by drawing all triangles in the selection in the color of the new selection.

  If the user is interested in the physical distribution of all values of a specific attribute, the user can map the attribute values onto the mesh. This can be done either by color mapping the attribute values onto the triangles of the mesh, or by adding glyphs to the three-dimensional representation, where the attribute value is represented by the length of the glyphs.

  Again, this method can also be used to inspect the distribution of the number of peaks per response profile, by storing the number of peaks per feature response profile in a separate attribute.

- How accurately does the segmented mesh correspond to the patients organs?

  This task is solved in different ways depending on the data available to the user. For some patients, experts have created ground-truths, allowing the exact error to be computed by comparing the mesh to the ground truth. For other users, instead of such a ground truth only the original imaging data is available. In this case the user will have to explore for errors manually. Finally, if none of those are available and the user only has access to the segmentation result, it is not possible to compare the segmentation results to the actual location of the patients organs.

In the first case, when an expert-segmented ground truth is available, it is possible to compute the distance from each triangle to the true boundary of an organ. This error can then be added to the data set as an extra attribute, making it possible to visualize the error in the same fashion as other att leributes. This includes inspecting the value distribution in histograms, exploring correlations using the scatter plots and mapping the error value onto the model by color mapping or glyphs.

In the second case, when there is no ground truth available, but the imaging data is, the user can manually assess the accuracy of the segmentation result by comparing the mesh with the organ boundaries visible in the slices. To do this, the user can both display the imaging data on planes intersecting the three-dimensional representation of the model, and can display the mesh triangles intersecting those planes on the two-dimensional representation of the imaging data. In both approaches, the user will have to manually iterate through the slices to compare the accuracy of the entire model, yet lacking the ground truth mesh, this still is a fairly intuitive and reliable method.

- How do available attributes correlate with each other?

Discovering and verifying correlations between attributes can be done by interaction with the scatter plot matrix. Each of those scatter plots presents the correlation between two of the attributes. Any set of points displayed in close proximity of each other represents a set of points with for both of the attributes on the axis values similar to those of other points in the set.

By selecting set of points in the scatter plots, the user is able to inspect the distribution of the set of selected triangles in the other scatter plots, allowing the user to verify whether the selected cluster also correlates in terms of other attributes or not. Additionally, once a selection has been made, the physical location of the selected triangles can also be inspected, as these are then highlighted on the three-dimensional representation of the organ mesh.

- Is there a relation between organ properties and segmentation accuracy?

Executing this task requires information on the segmentation accuracy. As discussed in the item "How accurately does the segmented mesh correspond to the patients organs?", there are three options: the accuracy is stored in a separate accuracy, the user has to manually inspect the accuracy using imaging data, or no information on the segmentation accuracy is available.

In the first case, the relation between all attributes, including organ properties such as physical location, can be explored in the same way was other correlations are discovered and verified.

In the second case, the user can import the imaging data, and display both the slices on panels intersecting the three-dimensional representation of the mesh, and display the triangles of the mesh intersecting those panels in the two-dimensional representation of the imaging data. When the user then maps attributes onto the mesh, or creates selections of triangles, for example by selecting clusters in the scatter plots or by selecting physical regions by brushing in the three-dimensional representation, these selections are visible in all three- and two-dimensional representations. Using this approach, it is possible to combine the accuracy information retrieved with the information contained in the attributes.

- How many potentially correct positions were considered for specific parts of the mesh?

  During the segmentation phase, for each of the triangles of which the mesh consists a statistically optimal location is computed. This is done by considering a finite set of positions, for each of which a metric is computed expressing how likely it is that an organ boundary should occur there based on the imaging data. This metric is called the 'feature response', and all feature responses for the considered locations are listed in the feature response profiles provided in the profile data sets. If in this profile there is one position with a significantly higher feature response than all other locations, it is likely that this position is the optimal location for a triangle. However, if there are multiple high values, or 'peaks', in the profile, or none at all, this is an indication that the triangle may be positioned unreliably.

  The number of positions with a high feature response can be computed by loading the profile data set, and storing the number of peaks in a new attribute. This way, it becomes possible to show the physical locations of triangles with a specific number of feature response peaks, and find correlations between the number of peaks and other attributes.

  Alternatively, it is possible to generate clusters of triangles with similar response profiles, and show their physical locations. By doing so, the user can inspect whether there are larger physical regions which were all positioned using less certain target points, indicating possibly less accurate parts of the mesh.

- How reliable is the segmentation result in specific areas of the mesh?

  The reliability of specific regions of the segmentation result can be explored in a similar fashion as the accuracy of the complete mesh is inspected, as treated in the item “How accurately does the segmented mesh correspond to the patients organs?”. Additionally, the user can create extra selections to obscure the colors caused by color maps or other selections on areas outside of the scope of the users interest. Moreover, by selecting the specific area the user is focussing on, the triangles in this selection are highlighted in the scatter plots, simplifying the process of discovering correlations applying to a specific portion of the mesh.

In conclusion, it has been proven possible to execute each of the tasks specified in Subsection 1.3 using the presented framework.

4 Conclusion

In this thesis, a framework for the interactive visualization of segmentation results has been presented. This framework utilizes data files containing the resulting mesh along with a number of attributes describing properties of the individual triangles of which the mesh consists, and allows the user to explore the available data, to discover and verify hypothesis and to inspect the reliability of the segmentation result.

Section 1 presents an introduction to the problem the presented framework intends to solve. This section contains amongst others a description of the problem, the intended users and the data types the application is designed to handle.

Section 2 provided elaborate descriptions for all features that have been implemented, along with technical details and motivations for all design choices made.

In Section 3, it has been shown that the framework described in this document indeed does allow the intended users to successfully execute simulated real world use cases.

Finally, in Subsection 3.3, we systematically evaluate whether the framework provides the required functionality in order to execute all tasks lister prior to the development, as listed in Subsection 1.3. This shows that the framework described in this document meets the requirements set at the start.

From this, we conclude that the presented framework allows the intended users to successfully execute all the tasks we have foreseen. In addition, the provided functionality allows for a broad exploration and careful inspection of various types of data, presenting a suitable starting point for future further improvements in the interactive exploration of segmentation quality.

4.1 Future work

As defined in Subsection 1.2, the application described in this document is designed for only two of the three existing types of potential users, namely technical researchers and clinical researchers. The value of the presented framework could increase significantly if functionality would be added to assist clinicians as well, providing one application with which all types of users can work, since this increases the efficiency with which discovered information can be shared. Also, to ensure that the framework remains suitable for the types of users it is designed for even when the provided data changes, the scalability should be increased. Currently, the data sets consist of at most 6 attributes, but in the future, the provided number of attributes could grown drastically, which would especially for the scatter plot matrix be troublesome.

In the development of this visual analytics framework, the main focus has been on providing the user with the ability to explore and inspect the available data. Both correlation detection and hypothesis generation and confirmation rely heavily on the creation of subsets of the data, which are then displayed in a distinguishable way in the existing visualizations. However, this is not sufficient for all possible tasks. In some cases, the user would benefit from a more elaborate set of possible actions to execute with created selections. For example, if the user would be able to interactively choose which part of the data set is visualized, the clutter caused by parts of the data set irrelevant to the users needs could be removed. With that functionality, scatter plots would no longer be crowded with large numbers of points, and the profile clustering could focus on specific physical regions of the mesh.

On the other hand, instead of reducing the amount of information visualized, allowing for a larger number of patients to be visualized simultaneously could improve the efficiency. By providing, possibly simplified, information on multiple patients at once, the user can quickly inspect where possibly unusual properties occur. For example, in their paper "Dynamic Multi-View Exploration of Shape Spaces" [BBP10], Buskin et al. show how a dynamic overview of larger data sets can help quickly comparing multiple meshes and identifying interesting cases.

In terms of efficiency, users may benefit from additional functionalities allowing the automation of the knowledge discovery phase. By including statistical analysis tools into the framework, the application could simplify several aspects.

Firstly, the user can be provided with a list of correlations between attributes, saving the user the time to inspect the scatter plots manually.

Secondly, correlations which are less noticeable, for example because they occur in only small specific subsets of the data, would likely be missed by a user manually inspecting the data. Statistical analysis could prevent this loss of information.

Lastly, verifying whether discovered correlations are statistically significant is difficult using only visual representations of the data. By implementing automated statistical analysis, the user can reliably verify hypotheses, instead of relying on convincing visual clues.

Regarding the visual aspect of the presented framework, currently the user has no control over any of the aspects of the user interface. The application is built in such way that types of visualizations most likely used simultaneously, namely the 3D-representation and the 2D-visualizations, can be shown side by side. However, there may be users who prefer a different layout. For example, it may be useful to display the profile clustering side by side with the scatter plot visualizations. In future work, it might therefore be useful to allow the user to arrange the different parts of the application freely.

Additionally, the colors used currently are fixed. The user cannot manually select colors for the selections (with the exception of deleting and recreating a selection), and the legend of the color map cannot be altered.

Lastly, the evaluation as conducted in Section 3.3 is likely biased, as no external end users have been involved in the process. To truly identify both positive and negative aspects of the presented framework, a more thorough assessment should take place.

Using the aspects mentioned in this section, the current version of the framework could potentially be extended from the application for data exploration and knowledge discovery it is now, to an important tool for all types of medical users in the development of segmentation algorithms and the treatment of patients.

5 References

[Aie12] K Aiello. Assignment parallel coordinates. `http://visthis.blogspot.nl/2012/10/assignment-3-parallel-coordinates.html/`, 2012.

[BBF$^+$11] Stef Busking, Charl P Botha, Luca Ferrarini, Julien Milles, and Frits H Post. Image-based rendering of intersecting surfaces for dynamic comparative visualization. *The visual computer*, 27(5):347–363, 2011.

[BBP10] Stef Busking, Charl P Botha, and Frits H Post. Dynamic multi-view exploration of shape spaces. In *Computer Graphics Forum*, volume 29, pages 973–982. Wiley Online Library, 2010.

[BC87] Richard A Becker and William S Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[Bos12] Michael Bostock. D3. js. `http://d3js.org/`, 2012.

[BTI07] David Borland and Russell M Taylor II. Rainbow color map (still) considered harmful. *IEEE computer graphics and applications*, 27(2):14–17, 2007.

[Cab10] R Cabello. three. js-javascript 3d library. `http://threejs.org/`, 2010.

[Cha06] Winnie Wing-Yi Chan. A survey on multivariate data visualization. *Department of Computer Science and Engineering. Hong Kong University of Science and Technology*, 8(6):1–29, 2006.

[CLNL87] Daniel B Carr, Richard J Littlefield, WL Nicholson, and JS Littlefield. Scatterplot matrix techniques for large n. *Journal of the American Statistical Association*, 82(398):424–436, 1987.

[CM05] Ting Chen and Dimitris Metaxas. A hybrid framework for 3d medical image segmentation. *Medical Image Analysis*, 9(6):547 – 565, 2005. {ITK} Open science - combining open data and open source software: Medical image analysis with the Insight Toolkit.

[DH02] Helmut Doleisch and Helwig Hauser. Smooth brushing for focus+ context visualization of simulkation data in 3d. 2002.

[Dol04] Helmut Doleisch. *Visual analysis of complex simulation data using multiple heterogenous views*. PhD thesis, Doleisch, 2004.

[EPS$^+$08] Olivier Ecabert, Jochen Peters, Hauke Schramm, Cristian Lorenz, Jens von Berg, Matthew J. Walker, Mani Vembar, M. E. Olszewski, K. Subramanyan, G. Lavi, and Jrgen Weese. Automatic model-based segmentation of the heart in ct images. *IEEE Trans. Med. Imaging*, 27(9):1189–1201, 2008.

[F$^+$03] Remondino Fabio et al. From point cloud to surface: the modeling and visualization problem. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(5):W10, 2003.

[GR04] G. Grigoryan and P. Rheingans. Point-based probabilistic surfaces to show surface uncertainty. *Visualization and Computer Graphics, IEEE Transactions on*, 10(5):564–573, Sept 2004.

[GW02] Rafael C Gonzalez and Richard E Woods. Digital image processing. *Prentice Hall*, pages 299–300, 2002.

[HW79] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108, 1979.

[Ins85] Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.

[JS03] C.R. Johnson and A. Sanderson. A next step: Visualizing errors and uncertainty. *Computer Graphics and Applications, IEEE*, 23(5):6–10, Sept 2003.

[Knu76] Donald E Knuth. Big omicron and big omega and big theta. *ACM Sigact News*, 8(2):18–24, 1976.

[PWL97] Alex T. Pang, Craig M. Wittenbrink, and Suresh K. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 13(8):370–390, 1997.

[Rob00] Jonathan C. Roberts. Multiple view and multiform visualization, 2000.

[SMAL98] William J Schroeder, KM Martin, LS Avila, and CC Law. The vtk users guide. kitware, 1998.

[SZB$^+$09] J. Sanyal, Song Zhang, G. Bhattacharya, P. Amburn, and R.J. Moorhead. A user study to compare four uncertainty visualization methods for 1d and 2d datasets. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1209–1218, Nov 2009.

[UVW03] Antony Unwin, Chris Volinsky, and Sylvia Winkler. Parallel coordinates for exploratory modelling analysis. *Computational Statistics & Data Analysis*, 43(4):553–564, 2003.

[vLBK$^+$13] Tatiana von Landesberger, Sebastian Bremm, Matthias Kirschner, Stefan Wesarg, and Arjan Kuijper. Visual analytics for model-based medical image segmentation: Opportunities and challenges. *Expert Systems with Applications*, 40(12):4934 – 4943, 2013.

[War02] Matthew O Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3-4):194–210, 2002.

[War12] Colin Ware. *Information visualization: perception for design.* Elsevier, 2012.

[WJ63] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[YWR03] Jing Yang, Matthew O. Ward, and Elke A. Rundensteiner. Interactive hierarchical displays: a general framework for visualization and exploration of large multivariate data sets. *Computers & Graphics*, 27(2):265 – 283, 2003.

[YZK03] A. Yezzi, L. Zllei, and T. Kapur. A variational framework for integrating segmentation and registration through active contours. *Medical Image Analysis*, 7(2):171 – 185, 2003. Mathematical Methods in Biomedical Image Analysis - {MMBIA} 2001.