

Institut für Computergraphik
und Algorithmen

TU Wien

Favoritenstraße 9-11
1040 Wien
AUSTRIA

Tel: +43 (1) 58801-18684

Fax: +43 (1) 58801-18698

Institute of Computer Graphics and
Algorithms

TU Wien

Email:

jschmidt@cg.tuwien.ac.at

Website:

<http://www.cg.tuwien.ac.at>

TECHNICAL REPORT

Visual Analysis of Volume Ensembles Based on Local Features

J. Schmidt, B. Fröhler, R. Preiner, J. Kehrner, M. E. Gröller, S. Bruckner, and C. Heinzl

TR-186-2-16-2

May 2016

Visual Analysis of Volume Ensembles Based on Local Features

J. Schmidt, B. Fröhler, R. Preiner, J. Kehrer, M. E. Gröller, S. Bruckner, and C. Heinzl

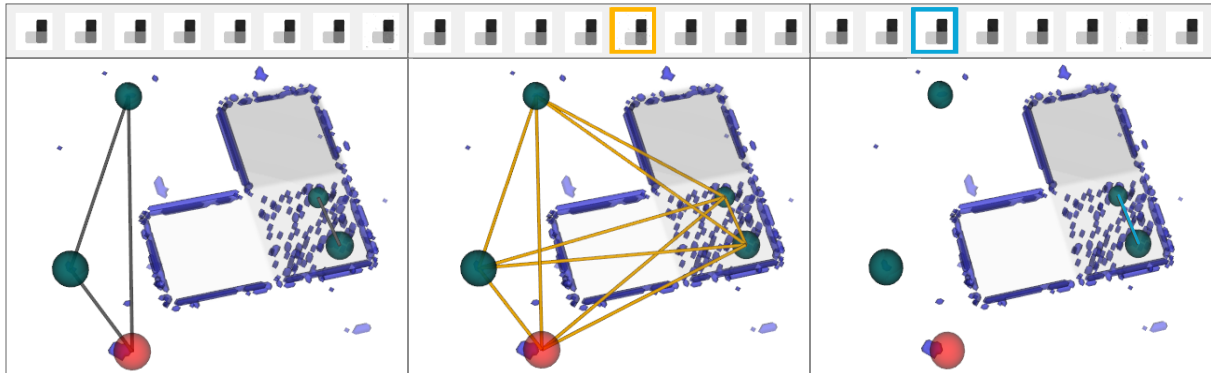


Figure 1: Analyzing local regions in a 3D volume ensemble. This figure shows part of an ensemble of volumetric segmentation masks, and probing widgets placed on positions of interest in 3D. The positions of the probes are used to build a similarity graph and connect nodes that represent similar information. According to the current selection of ensemble members, the graph is constructed differently. This way different aspects of the ensemble can be explored. (*left*) No member is selected, so the graph compares the data at the local positions. (*middle, right*) An ensemble member is selected and the similarity graph connects regions with similar characteristics. This way individual ensemble members can be compared against the rest of the ensemble.

Abstract

Ensemble datasets describe a specific phenomenon (e.g., a simulation scenario or a measurements series) through a large set of individual ensemble members. These individual members typically do not differ too much from each other but rather feature slightly changing characteristics. In many cases, the ensemble members are defined in 3D space, which implies severe challenges when exploring the complete ensembles such as handling occlusions, focus and context or its sheer data-size. In this paper we address these challenges and put our focus on the exploration of local features in 3D volumetric ensemble datasets, not only by visualizing local characteristics, but also by identifying connections to other local features with similar characteristics in the data. We evaluate the variance in the dataset and use the spatial median (*medoid*) of the ensemble to visualize the differences in the dataset. This medoid is subsequently used as a representative of the ensemble in 3D. The variance information is used to guide users during the exploration, as regions of high variance also indicate larger changes within the ensemble members.

The local characteristics of the regions can be explored by using our proposed 3D probing widgets. These widgets consist of a 3D sphere, which can be positioned at any point in 3D space. While moving a widget, the local data characteristics at the corresponding position are shown in a separate detail view, which depicts the local outliers and their surfaces in comparison to the medoid surface. The 3D probing widgets can also be fixed at a user-defined position of interest. The fixed probing widgets are arranged in a similarity graph to indicate similar local data characteristics. The similarity graph thus allows to explore whether high variances in a certain region are caused by the same dataset members or not. Finally, it is also possible to compare a single member against the rest of the ensemble. We evaluate our technique through two demonstration cases using volumetric multi-label segmentation mask datasets, two from the industrial domain and two from the medical domain.

1 Introduction

In an increased number of application areas datasets are created that consist of a collection of individual, but related, datasets. Such datasets can usually become quite large, in terms of data dimensions as well as the number of individual datasets. For the visualization, this means that there is a rising demand for visual analysis tools that move away from visualizing and analyzing single datasets to analyzing a collection of datasets. Such collections of datasets that combine a set of single but related datasets are called **ensembles**. Originating from the simulation domain, where the output results are typically generated by multiple simulation runs [Matkovic et al.(2010)Matkovic, Gracanin, Jelovic, Ammer, Lez, and Hauser], ensembles can also be found in other settings. We refer to ensembles in a more general sense as large datasets covering a certain phenomenon, where the individual data items (i.e., the **members**) show slightly different characteristics. This consideration certainly covers the field of multiple simulation runs, but also includes multi-modal datasets [Wakeman and Henson(2015)], or sets of measurements [Cho et al.(2014)Cho, Kim, Bae, and Seo]. Application areas that rely on the analysis of ensemble datasets involve weather forecast analysis, where temperature curves [Demir et al.(2014)Demir, Dick, and Westermann] or certain weather situations [Rautenhaus et al.(2015)Rautenhaus, Grams, Schäfler, and Westermann] need to be analyzed. In geology, the analysis of different 3D seismic horizon surfaces reveals further information about the underlying geological structures [Höllt et al.(2013)Höllt, Chen, Hansen, and Hadwiger]. Furthermore, the analysis of tensor fields can help to study the effect of different acquisition parameters for magnetic resonance imaging in medical applications [Zhang et al.(2016)Zhang, Schultz, Lawonn, Eise mann, and Vilanova].

Such ensemble datasets exhibit great advantages for a detailed analysis of phenomena: First of all, the members describe the same phenomenon under different prerequisites. The ensemble members are therefore defined in the same spatial (and temporal) domain and typically are of the same data type (e.g., volumetric or mesh data). Furthermore, the members are usually coregistered and of the same size, which allows for comparisons amongst them. Existing approaches for **ensemble visualization** already concentrated on how the multitude of information, and the variability of the data can be visualized in an effective and intuitive way. For example, this can be achieved by displaying statistical parameters of the dataset as an overview (Figure 2(a)). Such methods are particularly helpful to get an overview of the dataset

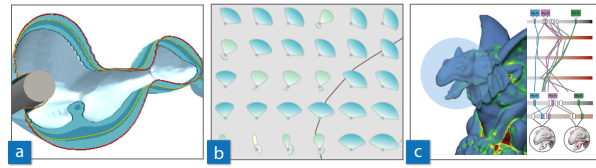


Figure 2: Existing techniques of ensemble visualization. To reduce the multitude of information in an ensemble to a meaningful representation, existing approaches aggregate the data e.g. by using color (a), or statistical measurements (b). Another approach is to allow users to explore local regions in the data, and to compare these regions in a parallel-coordinates plot (c). Images are courtesy of [Höllt et al.(2013)Höllt, Chen, Hansen, and Hadwiger] (a), [Raj et al.(2015)Raj, Mirzargar, Kirby, Whitaker, and Preston] (b) and [Schmidt et al.(2014)Schmidt, Preiner, Auzinger, Wimmer, Gröller, and Bruckner] (c).

and to evaluate its variability, as the complexity of the dataset is reduced to aggregated properties. However, there is also a drawback when using aggregation techniques. When analyzing ensembles, it is often necessary to explore local data regions where the members show slightly different characteristics. Current aggregation techniques do not provide tools to analyze such local regions in the data, to make comparisons or to draw conclusions from the data properties in those regions. In an aggregated view, it is particularly hard to tell whether a high data variance, or variability, is caused by the same or different members. It is also not possible to analyze whether a member producing reasonable results in one region of the data might be responsible for a high variance in another region. Some approaches tried to tackle this problem by using glyphs (Figure 2(b)) to indicate similar local regions, or by arranging local regions in 3D in a parallel-coordinates plot (Figure 2(c)). With such techniques users can locate regions with a high variance. It is further possible to compare those regions, and to track individual members. The existing approaches, however, do not offer the possibility to analyze how much individual members contribute to the local variance in a certain region. They also do not offer possibilities to display local similarities in a 3D volume, since they are suitable for 2D vector data and 3D meshes.

In this paper, we show how local features can be analyzed in 3D volumetric ensemble datasets, and how the information of the local features can be combined into a single global view. For the purpose of this paper, we demonstrate our techniques on ensembles of **volumetric segmentation masks** as exemplary use-cases. Seg-

mentation is applied to the volume data and based on the used parameter settings the resulting *segmentation masks* show slight differences. An overview of the data we used as an illustrative example can be seen in Figure 3. The analysis of the resulting segmentation masks helps to find out which regions of the volume data are critical to segment. Furthermore, parameter settings that lead to unwanted features in the data (e.g., noise) can be excluded from the working process in the future.

In our visual analysis approach, we first provide a general overview of the available ensemble data. For this purpose, we compute the *most representative* member of the ensemble that represents the data best. This most representative member is called the spatial median, or the *medoid*, of the ensemble. We compute the medoid by using the Weiszfeld algorithm [Beck and Sabach(2014)]. The medoid is then used as an entry point for the local exploration. To guide users, we visualize areas of high variance in the data in 3D. These regions can be further explored using interactive *probing widgets*. The probing widgets consist of 3D spheres which can be positioned arbitrarily in 3D space by mouse interaction to investigate regions of high variance. While moving the widgets, the current local characteristics of the data are shown in a separate *detail view*. The detail view lists the outliers at the current position and gives insight into the underlying data. It is also possible to fix widgets at a user-defined position within the volumetric datasets. The fixed probing widgets are arranged in a *similarity graph* in 3D and 2D to indicate whether the local data exhibits comparable characteristics. In this way users can explore local regions and analyze their similarities, without losing the context and the original data. The main contributions of our approach are:

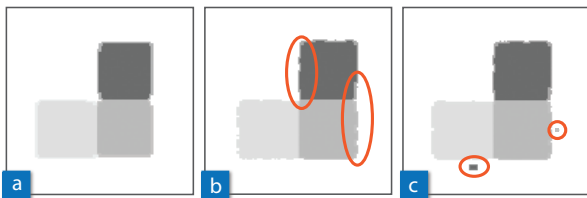


Figure 3: Different segmentation masks of the same volume dataset. The segmentation masks were created using different settings of the same parameter space. Optimal parameter settings result in a reasonable segmentation according to the analysis task (a). Different parameter settings may lead under or oversegmentation, manifesting in uneven surfaces (b), as well as incorporating or excluding regions due to noise in the data (c).

- **Volume probing:** Our proposed interactive, intuitive probing widgets allow users to explore local regions in volumetric ensemble datasets.
- **Similarity graph:** The spatial position and extent of the probing widgets are used to arrange the local regions in a graph according to the similarity of the local data. The graph data can be shown in 3D and in 2D.
- **Multi-level analysis:** With the similarity graph it is possible to analyze the whole ensemble data, and also to compare one or more members against the rest of the dataset.

The paper is organized as follows: The next part (Section 2) describes the related work for this topic. In Section 3 we explain all necessary steps of our approach. Details on the implementation are given in Section 4. Demonstration cases, which are shown in Section 5, demonstrate the effectiveness of our approach. The contributions of the paper and possibilities for future work are discussed in Section 6. The paper is concluded in Section 7.

2 Related Work

The work presented in this paper is located in the area of *ensemble visualization*. This area covers many different sub-topics, like metrics for comparing 3D datasets [Fofonov et al.(2016)Fofonov, Molchanov, and Linsen], as well as storage handling for large datasets [Fogal et al.(2010)Fogal, Childs, Shankar, Krüger, Bergeron, and Hatcher]. This paper focuses in particular on the exploration of local features in ensembles, so work related to this field, and techniques to compare multiple segmentation masks, will be discussed here.

Ensemble Visualization Ensemble datasets may comprise many different types of data [Phadke et al.(2012)Phadke, Pinto, Alabi, Harter, Taylor, Wu, Petersen, Bass, and Healey]. For 2D image datasets, Schmidt et al. [Schmidt et al.(2013)Schmidt, Gröller, and Bruckner] proposed a technique for the analysis of local features. 3D surfaces can be compared by using transparency and glyphs [Busking et al.(2011)Busking, Botha, Ferrarini, Milles, and Post], by representing them in a surface boxplot [Genton et al.(2014)Genton, Johnson, Potter, Stenchikov, and Sun], or by visualizing their similarity [Haidacher et al.(2011)Haidacher, Bruckner, and Gröller]. *Spaghetti plots* [Potter

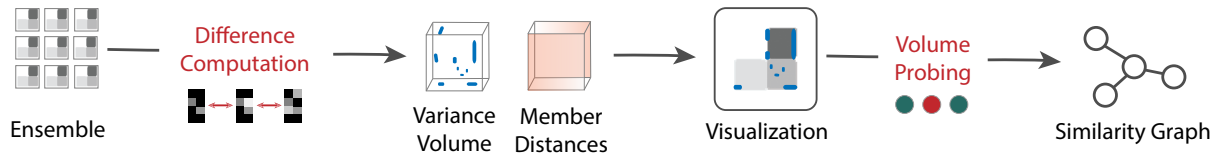


Figure 4: Overview of our visual analysis approach We start with an *ensemble* of volumetric segmentation masks. After the *difference computation* (Section 3.1), we obtain the *variance* and the *distances* of the members in the data. This data is then presented to the users in a *3D view* by showing the most representative member of the dataset (Section 3.2). There users can use *volume probing* to explore local regions (Section 3.3). To compare these regions, they are arranged in a *similarity graph* (Section 3.4).

et al.(2009)Potter, Wilson, Bremer, Williams, Doutriaux, Pascucci, and Johnson] can be used to visualize 2D scalar field ensembles by simultaneously showing an isocontour per ensemble member. The uncertainty in a collection of 3D iso-surfaces can either be visualized by color [Pfaffelmoser et al.(2011)Pfaffelmoser, Reitinger, and Westermann], or by an interactive box-plot visualization [Raj et al.(2015)Raj, Mirzargar, Kirby, Whitaker, and Preston]. To facilitate interaction, Malik et al. [Malik et al.(2010)Malik, Heinzl, and Groeller] decided to compare volume datasets in 2D slices. Schmidt et al. [Schmidt et al.(2014)Schmidt, Preiner, Auzinger, Wimmer, Gröller, and Bruckner] presented an application to compare several 3D meshes and spot local outliers. Alabi et al. [Alabi et al.(2012)Alabi, Wu, Harter, Phadke, Pinto, Petersen, Bass, Keifer, Zhong, Healey, and Taylor] showed how surfaces can be compared in 3D. Jarema et al. [Jarema et al.(2015)Jarema, Demir, Kehrer, and Westermann] proposed a technique to compare 2D vector field ensembles, also based on local regions of interest. Our work presented in this paper is clearly situated in the field of ensemble visualization. We use interaction tools in 3D, and we extend the ideas of Jarema et al. [Jarema et al.(2015)Jarema, Demir, Kehrer, and Westermann] and Schmidt et al. [Schmidt et al.(2014)Schmidt, Preiner, Auzinger, Wimmer, Gröller, and Bruckner] to the field of volumetric datasets, where widgets also need to be placed inside the data.

Segmentation Exploration As an exemplary use case, we analyze collections of volumetric segmentation masks in this paper. Segmentation algorithms typically possess several parameters which require careful tuning. *Tuner*, as proposed by Torsney-Weir et al. [Torsney-Weir et al.(2011)Torsney-Weir, Saad, Möller, Hege, Weber, and Verbavatz], allows the user to analyze different segmentations of 2D images. It requires the definition of at least one objective quality measure, for example defined through difference to a ground truth

image, and concentrates on investigating such derived measures instead of the segmentation masks. Geurts et al. [Geurts et al.(2015)Geurts, Sakas, Kuijper, Becker, and Von Landesberger] proposed a system to analyze different segmentations of medical data. They also require objective quality measures, and provide local analysis methods based on these measures. The *GEMSe* tool by Fröhler et al. [Fröhler et al.(2016)Fröhler, Heinzl, and Möller] can be used to browse an ensemble of 3D segmentation masks to identify parameter settings for a proper segmentation result. It does not require an objective quality measure but provides limited support for local exploration. While segmentation exploration systems mainly focus on browsing the full dataset, our technique allows the detailed exploration and comparison of local features. This is useful to analyze the slight differences of segmentation masks in a subset of the ensemble, where the members have been previously defined as useful. Our work can therefore be seen as an extension to the work by Fröhler et al. [Fröhler et al.(2016)Fröhler, Heinzl, and Möller] and the work by Geurts et al. [Geurts et al.(2015)Geurts, Sakas, Kuijper, Becker, and Von Landesberger], improving these in the aspect of the local analysis. Our technique also operates on datasets without a proper reference solution.

Volume Probing The selection of inner parts in a volume is not a trivial task, since the selection intention behind a mouse-click is not clearly defined. Some parts of the data might be hidden in the visualization due to transfer function settings. Apart from picking visual structures [Wiebel et al.(2012)Wiebel, Vos, Förster, and Hege], it is also possible to select the picked location based on the context [Kohlmann et al.(2009)Kohlmann, Bruckner, Kanitsar, and Gröller]. Some tasks in visualization, e.g., transfer function design [Kniss et al.(2002)Kniss, Kindlmann, and Hansen], can be greatly enhanced by letting users select regions of interest in a volume dataset. The selection of visible structures may help to refine segmentations [Sib-

bing and Kobbelt(2007)]. With seismic data, users can interactively assemble horizon parts by picking them [Patel et al.(2010)Patel, Bruckner, Viola, and Griller]. Selected regions of interest can also be used to display additional information about the volume [Ropinski et al.(2009)Ropinski, Viola, Biermann, Hauser, and Hinrichs]. *Profile Flags* [Mlejnek et al.(2005)Mlejnek, Ermes, Vilanova, van der Rijt, van den Bosch, Gerritsen, and Gröller] are a probing technique to reveal structures in volumetric datasets and display them in an additional view. We make use of probing widgets to allow users to select local regions of interest. The local characteristics of the data are displayed in a linked view.

3 Methodology

To be able to inform users about the differences in the ensemble, the individual members need to be compared against each other (Section 3.1). This leads to the computation of variances in the data, and distances between the members. We further need structures to present the differences to the users in a 3D environment. For this we identify the most representative member in the ensemble using the Weiszfeld algorithm (Section 3.2). This member is then used to display the local differences in a 3D environment (Section 3.3). To be able to compare local regions in the data, we arrange them in a graph and present the results to the user (Section 3.4). An overview of the pipeline is shown in Figure 4.

3.1 Pre-processing and Difference Computation

Before starting the analysis the members of the ensemble need to be compared against each other. We are especially interested in locating regions of high variance in the data, and in how much the individual members contribute to the local variance.

All of the following calculations are not computed on the raw input volume data, but on their **distance transforms**. In a distance transform, or distance map, for each voxel in a volume the distance to the nearest *obstacle voxel* [Saito and Toriwaki(1994)] is assigned. In our case, an obstacle voxel is a boundary voxel of the segmentation mask (see Figure 5 for illustration). We use distance maps instead of the raw data for further comparison, as they provide a more robust measure than comparing the raw pixel values. Therefore, in a pre-processing step, a distance map is created for every input volume. The created distance maps are of the same dimension and spacing as the input volumes.

In the next step additional parameters are extracted from the distance map data. For this purpose we create a new volume of the same dimensions as the distance maps. Then, for every voxel v , the mean value $mean_v$ is computed for all voxel values $value_{n,v}$ in all n members. The per-voxel mean values are stored in a newly created **mean distance map**. The mean values are used in the next step to calculate the variance.

The variance is a very useful tool to measure the variability of a given set. We again compute the variance in a per-voxel process. For every voxel v , the variance s_v^2 is calculated with the standard formula as follows:

$$s_v^2 = \frac{\sum (mean_v - value_{n,v})^2}{n} \quad (1)$$

The per-voxel variance values can be combined into a new volume, which then defines the **variance distance map**. During the computations we also store the overall mean variance mv , which is a single value computed as follows:

$$mv = \frac{\sum s_v^2}{k} \quad (2)$$

The parameter k defines the number of voxels, namely the dimension of the members, computed as $k = width * height * depth$.

In addition to the variance, we would like to keep track of which members are responsible for the high variance in a certain region. We therefore also store the total distance of one member to the other members. This distance $dist_v$ is again computed in a per-voxel process, and is defined as the sum of all $n - 1$ mean-squared Euclidean distances to the other members' voxel values:

$$dist_{j,v} = \sum_{i \neq j}^n (value_{i,v} - value_{j,v})^2 \quad (3)$$

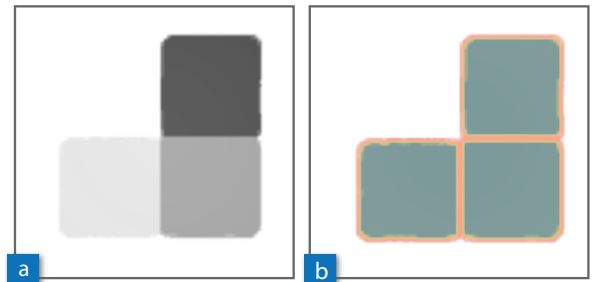


Figure 5: Distance maps. This figure shows an input segmentation mask (a) and its corresponding distance map (b). The distances from the surfaces are highlighted by colors from green (further away) to orange (close). The distance map stores the per-voxel distances to the respective nearest *obstacle voxel*.

We store the per-voxel distances in a new **member distances volume**.

The mean distance map can be deleted after the variance computation is finished to save memory. The variance distance map and the member distances volume are kept to be later used in the visual exploration (Section 3.3). The variance distance map will guide the user to the regions of high variance, and the member distances volume will be used to compute the contribution of the members to the local variance.

3.2 Medoid Computation

At the beginning of the analysis process we need to visualize the input data (i.e., the ensemble), so that users get an overview of the ensemble content and can also start the exploration. There are a lot of possibilities to present ensemble data in an aggregated way, e.g., by calculating the mean ensemble member [Ju et al.(2005)Ju, Schaefer, and Warren]. Using an actual member of the ensemble, though, is generally favorable over a synthetic representative such as the mean [Schmidt et al.(2014)Schmidt, Preiner, Auzinger, Wimmer, Gröller, and Bruckner]. Picking a random ensemble member might give a wrong impression of the actual data, because a member with a lot of artifacts might be picked accidentally. We therefore decided to calculate the most representative member of the ensemble, which we then call the *medoid* member.

Apart from the fact that the mean is not part of the ensemble, it is also known to be sensitive to outliers and can therefore be located far from the cluster of real samples. This is shown in Figure 6, where the presence of a single significant outlier creates a mean (orange) that is a blend between two clusters that does not relate to any of the ensemble members. The figure also demonstrates that a mean-based medoid, i.e., the nearest neighbor of the mean in the ensemble, can also result in a bad representative object.

Therefore, we choose a medoid based on the *spatial median* of the dataset (Figure 6, green). By interpreting each sample volume as a point in k -dimensional-space, its spatial median sm is defined as the k -dimensional point x that minimizes the sum of Euclidean distances to all other n points:

$$sm = \operatorname{argmin}_x \sum_n ||x - value_n|| \quad (4)$$

Equation (4) can be solved using the Weiszfeld algorithm [Beck and Sabach(2014)], which, starting from



Figure 6: Ensemble medoid. The input volumes can be interpreted as points in a k -dimensional space (in blue). The mean (m) is usually prone to follow outliers in the dataset, and the nearest neighbor (arrow) of the mean would not be a good representative of the dataset in this case. The spatial median (sm) minimizes the Euclidean distances to all points, but is generally not a part of the dataset. We therefore find the point that is closest to the spatial median (arrow) and use this as the medoid (med) of the ensemble.

the mean, approximates the minimum using the following fixed-point iteration

$$x = \frac{\sum_n value_n ||x - value_n||^{-1}}{\sum_n ||x - value_n||^{-1}} \quad (5)$$

The median is always a member of the original dataset in the 1D case. This is, however, not the case for the spatial median. The spatial median is generally not part of the dataset, and can therefore not directly be used as the medoid of the ensemble. Therefore, we instead choose the medoid as the k -dimensional Euclidean nearest neighbor of the resulting spatial median sm (Figure 6, dark blue). In the next step, this medoid will be used to start the exploration of the local regions.

3.3 Local Exploration in 3D

Our proposed visual analysis technique provides different views onto the data, which can be influenced by different interaction techniques. The concept of our interaction techniques is described in Figure 7. The individual elements of the concept are described in this Section.

The interaction works mainly in 3D, and therefore the most important view is considered to be the **3D view**. As a starting point for the exploration, we display the medoid volume as a representation of the ensemble dataset in this view. In addition, the regions of high variance are also shown in 3D. The per-voxel variance values have been stored in a separate variance distance map. We decided against standard volume rendering to display the variance values, to make a clear distinction between the medoid and the rendered variance values. To show the regions of high variance, we use Marching Cubes [Lorensen and Cline(1987)] to transform the

variance values into mesh data. The threshold for the Marching Cubes algorithm is set to divide between regions of *low* and *high* variance. We used the mean variance mv (as computed in Section 3.1) as the input threshold for the Marching Cubes algorithm. All values above this threshold are considered to represent a high variance in the data. The mean variance is used, instead of an outlier computation like the Z-score, to capture as many areas with a high variance as possible. This way geometry is created for regions of high variance, and regions of low variance are discarded from the visual representation. Since we operate on mesh data now, it is possible to render it in a way that it is visually easily discriminable from the medoid volume (Figure 8).

In the 3D view users can explore local regions by using our proposed **volume probing**. When activated by a hotkey, a probing widget follows the position of the mouse cursor. Users can also control the movement in the third dimension by keeping the hotkey pressed and using the mouse wheel. Then a clipping plane will be shown and moved along the z-axis of the volume, and the current probing widget will always be positioned on this plane. Users can use the probing widgets to explore the regions of high variance. We opted for using 3D spheres as probing widget due to their intuitiveness and rotation-invariance. The size of the spheres can be individually adjusted by the users. Furthermore, users can fix widgets at the current position by mouse click. The

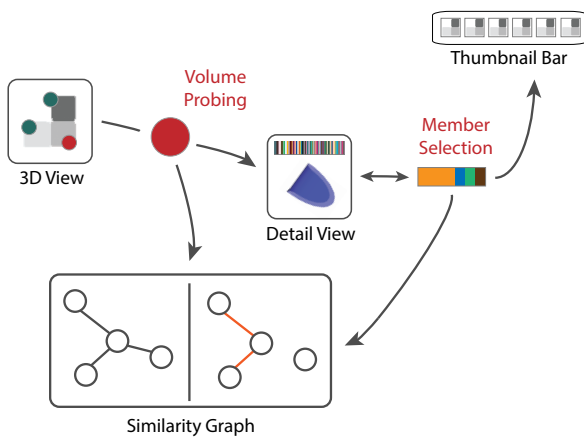


Figure 7: Interaction concept. Starting from the 3D view, the user can employ *volume probing* to explore local regions. The current position of the probe influences the *detail view* and the *similarity graph*. The *member selection* possibility in the detail view has an impact on several other views. The similarity graph has two stages, either for comparing the whole ensemble data, or just one member against the rest.

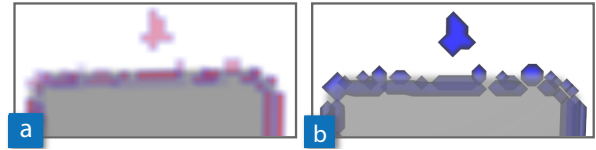


Figure 8: Regions of high variance in 3D. For displaying high variance in the data, we decided against volume rendering (a) to make a clear distinction between the variance information and the medoid volume (b).

widget is then fixed and stored in a sorted list, which is visible next to the 3D view. In this list users can activate fixed widgets again by mouse click.

An important visual channel to convey information in our technique is color. If new members are added to the ensemble, we assign a unique color to each of them that stays the same for the time our tool is running. We call these colors the **member colors**. Whenever a member is selected in the interface (see below), it is important that information related to this member is visible in other views. We decided to indicate these relations with color, and therefore a unique color is assigned to every member. We created a color table of 100 colors by using the *i want hue* tool [Jacomy(2016)]. Whenever a member is added to the analysis, a new color is selected for it from the table. It is also clear that not more than approx. 10-15 colors can be discriminated in one view. It has to be mentioned that it is not the purpose of our technique to compare a multitude of members against each other. So when selecting colors to relate information in the views to the members, it was intended to not compare more than 10-15 different members at once.

The position of the currently active probing widget is used to update the **detail view**. The detail view is placed next to the 3D view and shows the data characteristics at the current probe position. On the top a stacked bar-chart with all members is shown (Figure 9(a)). As described in Section 3.1, we store the Euclidean distances of the members in a separate volume. At the local probe position, we now iterate through all voxels covered by the probe and sum the distances $dist_i$ for all members. The summed values are then displayed in the bar-chart. The larger a segment in the stacked bar, the larger the distance of the member to all others. A large segment thus indicates here that the corresponding member differs from the rest of the members in the current local region. Below the chart, a more detailed view on the local data is presented. This view should reveal more information about the data, which is not visible in the 3D view. We decided to show a 3D rendering of the segmented surface of the medoid (Figure 9(b)). Since

we are dealing with segmentation masks, an additional information we can present to the user (apart from the volume rendering of the medoid) is the shape of the segmentation surface at the current local position. In the case of other volume datasets being analyzed, different information could be displayed here, according to the semantics of the analysis. The local differences in the data are shown as 3D structures in the rendering. If individual members have been selected (this is also explained below), their surfaces are shown next to the medoid surface (Figure 9(c)). The viewing direction of the surface rendering in the detail view is adapted according to the main viewing direction in the 3D view. It is only possible to zoom in/out in the detail view.

In the detail view, the **member selection** tools can be used. It is possible to select individual members in the bar-chart by clicking in the rectangles in the stacked bar. This member selection triggers other views to update, because it has a strong influence on the visualization. First of all, a member selection updates the detail view itself, because if members are selected, their surfaces are shown in the detail view together with the medoid surface. This way the data of several members can be compared to each other, as well as against the medoid.

In our user interface, we also preserve the view on the individual members. The input volumes are represented as thumbnails in a **thumbnail bar** above the other views

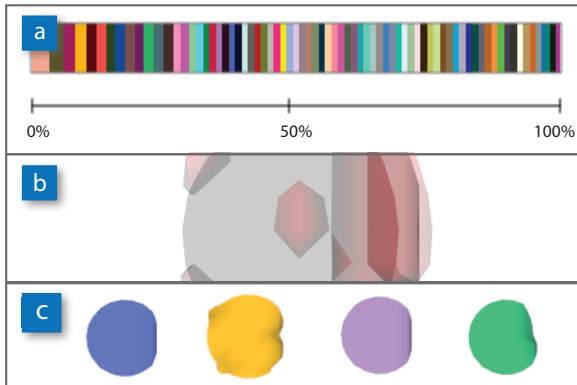


Figure 9: Detail view. This view is updated when the currently activated probing widget is moved. The detail view shows a stacked bar-chart to indicate the distances of the members (a) at the current position. Below a rendering of the cropped surface extracted from the medoid can be seen (b - minimized). The surface is sphere-shaped, since it has been cropped at the current probe position. The high variance regions are indicated on the medoid surface in red color. In the area below, surfaces of members can be shown next to each other (c).

(Figure 1). When hovering over the thumbnail, the filename of the member is shown. The medoid is marked by a grey background in the thumbnail. Users can open a 3D view on the original data by a mouse click on the thumbnails. Members which have been previously selected in the detail view are marked with a colored border. The thumbnail view allows the users to always go back to the original data, if necessary.

3.4 Similarity Graph

An important contribution of our technique is the indication of similarity between different probes. This allows users to compare local regions, and to find similar or distinctive regions in the data.

Users can fix the position of the probes, and all fixed probes are then arranged in a **similarity graph**. In the graph, nodes represent probes and edges indicate their similarity. If two probes are identified to be similar, they are connected by an edge in the similarity graph. The edges are colored accordingly to identify patterns in the graph. The similarity of two probes is computed by analyzing the local outliers at the probe positions.

We make use of the sum of distances for every member, which is automatically computed for every probe position and assign a distance to every member. Then we analyze the distances using statistical methods to identify outliers. First, we compute the median $median_p$ of all distances for the current probe p with the standard formula. Then we compute the median absolute deviation mad_p for the probe, which is a robust estimate for the standard deviation:

$$mad_p = 1.4826 \cdot median(|dist_{p,n} - median_p|) \quad (6)$$

A modified version of the Z-score, as proposed by Iglewicz and Hoaglin [Iglewicz and Hoaglin(1993)], can be used to compute an outlier index $o_{p,n}$ for the current probe as follows:

$$o_{p,n} = \left| \frac{dist_{p,n} - median_p}{mad_p} \right| \quad (7)$$

A member is considered to be an outlier in voxel v if $o_{p,n} \geq 3.5$ holds [Iglewicz and Hoaglin(1993)]. If mad_p is equal to zero, every distance deviating from the median is considered to be an outlier. After the computation, we end up with a list of $1..n$ binary flags for every member n inside the probe. The flags indicate whether a member has been identified as an outlier ($flag = 1$) for the current probe position or not ($flag = 0$). For all fixed probes the flags can be compared to compute the similarity.

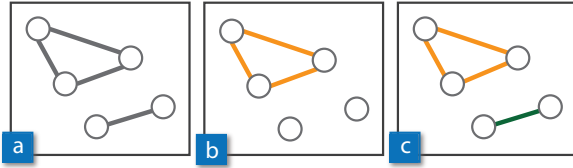


Figure 10: Similarity graph. The visualization of the graph depends on its stage. If the whole ensemble is analyzed, the similarity is indicated by lines (a). If only one member is compared against the rest of the ensemble, the edges between probes where it has been identified as an outlier are colored in the member color (b). If more than one member is compared against the rest of the ensemble, the edge colors indicate which members have been identified as outliers at the respective positions (c).

The nodes of the graph are always colored in the same color, since they refer to the probes in the 3D view. The way edges are colored in the graph, and how the similarity is computed, strongly depends on the current *stage* of the graph. The following three graph stages are possible (see also Figure 10):

- **Analyze all:** *No members have been selected.* In this case the whole ensemble data can be analyzed. The similarity is computed in a way that if all outlier flags for all members are equal, two probes are considered to be similar. The graph edges are all displayed in the same color.
- **Analyze one member:** *One member has been selected.* In this case one member can be compared against the rest of the ensemble. The similarity is computed in a way that only the flags for this member are considered (if the member is a local outlier) - if they are equal, two probes are considered to be similar. The graph edges are colored in the member color, if the member has been identified as an outlier in the probes.
- **Analyze a group of members:** *Between 2 and $n - 1$ members have been selected.* In this case several members can be compared against the rest of the ensemble. The similarity is computed in a way that only the flags for the selected members are considered (if the members are local outliers) - if they are equal, two probes are considered to be equal. The graph edges are colored in the member color of the members that have been identified as outliers.

Since the similarity graph can be set to three different stages, different aspects in the data can be explored.

The layout of the graph is updated when either the stage changes, or when the position/size of the currently active probe changes.

4 Implementation

Our technique was implemented in C++, and the user interface was implemented by using the Qt¹ framework (version 5.5.1 64bit). Loading and rendering of the volume data was done using the VTK² framework (version 7.0.0 64bit). VTK was also used to compute the distance transforms of the volumes, to render the 3D view and the similarity graph, and for handling the user interactions. The application was tested on an Intel i7 CPU 3.07GHz machine with 24 GB RAM and an NVIDIA GeForce GTX 680 graphics card.

To analyze the data, pre-processing and the computation of the differences as well as the outliers is necessary (see Section 3.1). This step is not interactive, so the user has to wait until all attributes are calculated. It is however possible to cache intermediate steps (e.g., the computed distance transform, the medoid index, and the variance outlier volume), so that a repeated load of the same dataset is faster.

A detailed description of the datasets used for demonstration can be found in Section 5. An overview of the runtime of the preprocessing steps for each dataset is shown in the following Table:

Ensemble	Dimension	#	Analysis	Memory
<i>Synthetic</i>	120x120x8	74	47s	1.1GB
<i>Rock Crystal</i>	285x300x216	51	2.6m	14.2GB
<i>Lower Body</i>	512x512x801	13	5.3m	16.9GB
<i>Aorta</i>	512x512x805	21	6.1m	17.3GB

The column *Analysis* shows the time it took to compute the distance maps, the distances and the medoid for every dataset. The computation of the variance and the distances between the members could be greatly sped up by distributing the computations into several threads. Since the variance and the distances are computed in a per-voxel process, the computations are completely independent and can therefore run in parallel. The column *Memory* shows the overall memory consumption during runtime. More discussions on the runtime and the memory consumption can be found in Section 6.

¹ <http://www.qt.io/>

² <http://www.vtk.org/>

5 Demonstration Cases

We applied our technique to volumetric segmentation mask ensembles from different domains. All three datasets have in common that all members are of the same size. Since all input volumes are of the same dimension and spacing, we can rely on the fact that every voxel position is present in every volume, and also represents the same spatial position in every volume.

5.1 Segmentation of Industrial-CT Data

The first two datasets we evaluated our approach with came from the domain of industrial computed tomography (CT) analysis. The segmentation masks were derived from industrial CT datasets by applying the extended random walker algorithm [Grady(2005)] to segment the data. The algorithm was run several times on those datasets with different parameters, which created ensembles of several segmentation runs.

The dataset **Synthetic** consists of 74 different segmentation results of an artificially created industrial workpiece with quite small dimensions (120x120x8). In this dataset we could identify interesting artifacts outside the boundaries of the actual object. Such artifacts usually occur if a segmentation algorithm is incorporating a lot of noise during the calculation. This may happen due to the inherent structure of the algorithm, or due to improper parameter settings. It is therefore important to know which members in the ensemble are responsible

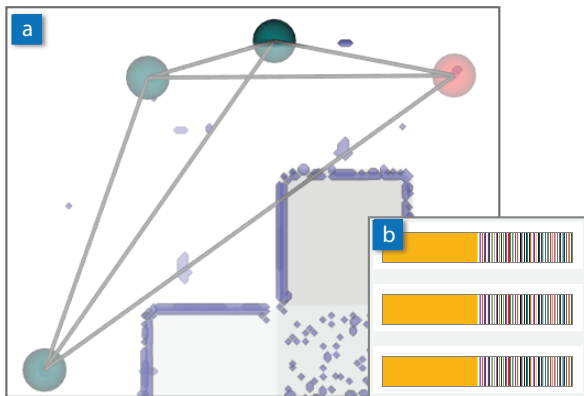


Figure 11: Dataset *Synthetic*. When analyzing the artifacts outside the object, the connected nodes in the similarity graph show that the local distribution of the differences is similar (a). When looking at the detail view, it can be seen that only one member causes the artifacts in the ensemble (b). The probing widget in red is currently active and can be moved by mouse interaction.

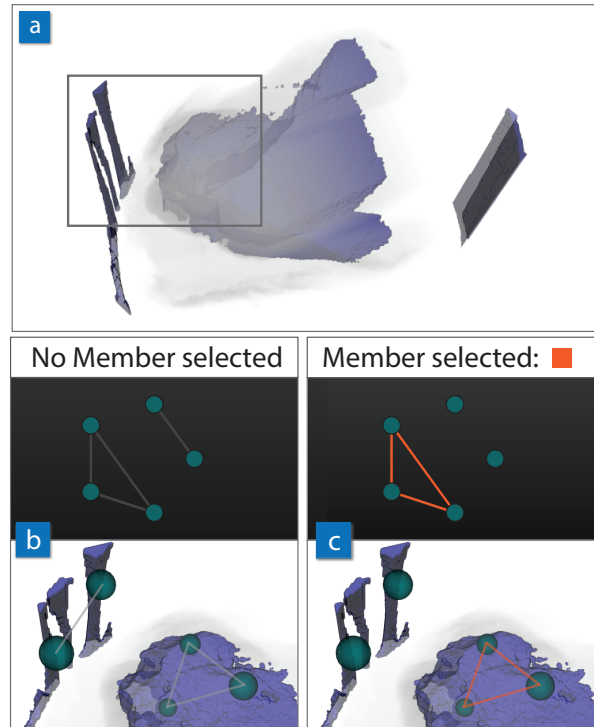


Figure 12: Dataset *Rock Crystal*. This dataset contains results of segmentations of a rock crystal. A view on the complete medoid with the rendered variance regions can be seen in (a). In a close-up of the data (grey rectangle), we placed probing widgets to analyze the local data. Similar regions could be found, which were then connected in the similarity graph (b) When selecting one member, it can be seen that this member only produces different result inside the object (c), but is not responsible for the outer artifacts.

for such artifacts in the data. With our technique such artifacts can be analyzed by placing probing widgets over them. While interacting with the probing widgets, the stacked bar in the detail view clearly indicated that in all cases only one member produced these artifacts. We then fixed the probing widgets at the positions of interest. The probes were then connected by edges in the similarity graph, which means that they are placed in regions with similar data. Figure 11 shows the similarity graph in 3D (Figure 11(a)) and a summary on the stacked bars of the detail views of the fixed probes (Figure 11(b)). In Figure 11 one of the probing widgets is colored in red. This probe has not been fixed yet, so the user can still interact with it. It is also included in the graph, so that the user immediately recognizes when a region is probed which is similar to data already stored in the similarity graph.

The dataset **Rock Crystal** represents a piece of a rock crystal, which has been scanned by an industrial CT scanner. The dataset consists of 51 members with $285 \times 300 \times 216$ voxels each. The segmentation of the inner structures is a non-trivial task, and therefore a lot of regions of high variance could be located inside the object (Figure 12(a)). Some other artifacts are also visible outside the rock crystal. These artifacts appear because some segmentations included parts of the scanning equipment which were also present in the scanned data. To analyze these artifacts, we placed several probing widgets inside the object, and also at the artifact regions in the outer regions. The fixed probes can be further analyzed in the similarity graph. First we had a look at the graph with no member being selected. There we could identify two clusters in the graph, in 2D as well as in 3D (Figure 12(b)). This means that both the regions of the outer artifacts and the inner structures represent similar data. The 2D representation of the graph allows to detect clusters, whereas the 3D representation better reveals the spatial position and size of the nodes. We could identify one member that produced different results inside the rock object than the rest of the ensemble. We selected this member in the detail view. The similarity graph then changed, and the outer probes are no longer connected in the graph (Figure 12(c)). This shows that the selected member produces different results inside the rock object, but is not responsible for the artifacts in the outer regions.

5.2 Segmentation of Medical Data

We applied our technique to two datasets from the medical domain. Both datasets were generated by computed tomography angiography (CTA), and represented segmentations of inner human structures, like bones and vessels.

The first medical segmentation dataset, called **Lower Body**, was created to segment bones and vessels in the lower body of one patient Figure 13(a). The data was segmented using the *AngioVis* ToolBox software [Straka(2006)]. Different levels of Gaussian noise were added to the data, to analyze the impact on the segmentation algorithm. One member of the ensemble corresponds to a noise level in the data. Our technique can be used to compare the noise levels, to see which ones affect the segmentation, and which do not.

The ensemble consisted of 13 members with dimensions of $512 \times 512 \times 801$ voxels each. First, the regions of interest for the analysis can be defined by placing the probing widgets accordingly. In Figure 13(b,c) three

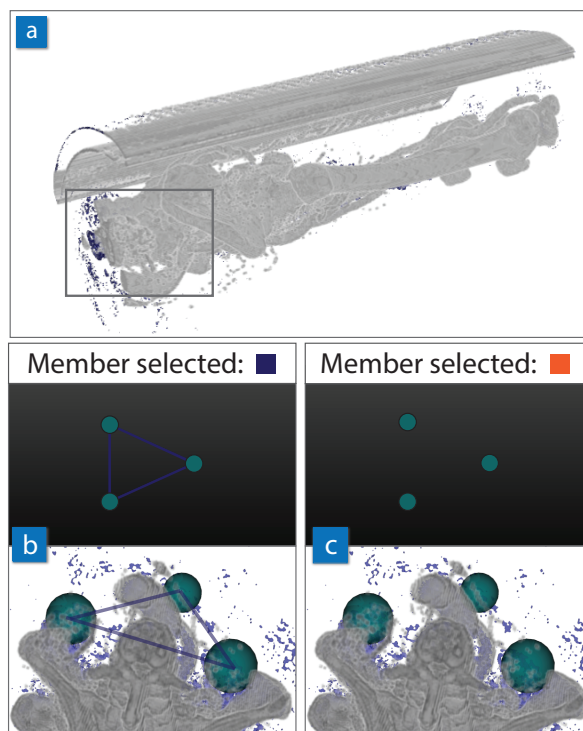


Figure 13: Dataset *Lower Body*. In this dataset bones and vessels of a patient’s lower body have been segmented (a). The members also refer to different noise levels in the data. In the close-up (grey rectangle) three probes have been placed. In the first case (a) the member greatly effects the segmentation, because here the results are significantly different than the rest. In the second case (b), the results do not differ from the existing data, and therefore the nodes in the graph are not connected.

of the placed widgets can be seen. Selecting a member in this case means comparing the results of a segmentation with a certain noise level to the rest of the ensemble. In the first case (Figure 13(b)) the member produces results that are significantly different from the rest of the ensemble. The nodes are therefore connected in the similarity graph. This means that this noise level greatly effects the outcome of the segmentation. In the second case (Figure 13(c)) the member produces similar results like the other algorithms in the ensemble. This can be seen in the similarity graph, since the nodes are not connected. This noise level therefore does not effect the segmentation results. This demonstration case shows that our technique can be used to quickly verify whether a segmentation is affected by certain factors.

The second dataset, **Aorta**, was created to segment the different vessel segments in case of an aortic dissec-

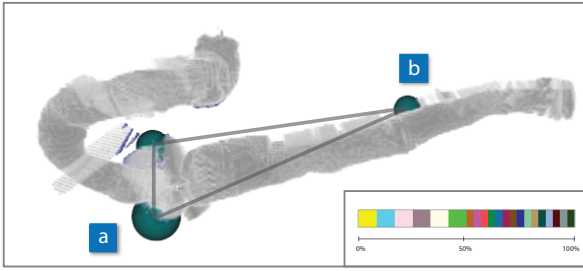


Figure 14: Dataset *Aorta*. In the upper part of the aorta (a), six members could be identified that were responsible for producing artifacts in this region. The corresponding stacked bar-chart can be seen at the lower left. In the lower part of the aorta one region could be identified that contains a dissection, which could not be segmented by all members (b). The six members that produces artifacts in (a) are also the only ones that are able to segment the dissection in this region.

tion. An aortic dissection is a malformation of a blood vessel, caused by a tear of the inner vessel wall. Successively, blood flows between the diverged layers of the arterial vessel wall, leading to the creation of two blood flow channels instead of one. The two channels are called the *true lumen* (TL), the channel where blood is still flowing, and the *false lumen* (FL), the channel where the blood flow has stopped. At slices perpendicular to the centerline a 2D level set segmentation [Chan and Vese(2001)] was applied to segment the blood flow channels. The per-slice 2D segmentations were then combined into a 3D volumetric segmentation mask again. Due to a variety of true and false lumen configurations, a perfect segmentation cannot be expected. A perfect segmentation would separate the TL and FL, if available, in the slice. If the TL and the FL are very close together, it may happen that the TL segmentation flows over to the FL area. If other structures with a similar intensity value like the aorta (e.g., bones) are close to the aorta, the segmentation may flow over into these unwanted parts.

The dataset consisted of 21 members with dimensions of 512x512x805 voxels. The members of the ensemble have been created from the same patient data, with slightly different parameter settings for the level-set segmentation. The segmentation was done for the full aorta, however, our collaboration partners said that they are actually only interested in the lower parts of the aorta. Since our technique operates in local regions, it was possible to concentrate on only the regions of interest. Segmentation in the medical domain is a common problem, and suitable solutions that would work in all

cases do not exist yet. It is common that segmentations have to be edited afterwards before the analysis can be started. One goal for analyzing the segmentation masks was, therefore, to find out which mask would require the least manual editing steps afterwards. We therefore concentrated on the high variance regions in the data, and placed probing widgets there. There we could identify six members where the segmentation of the TL tends to flow into other tissue surrounding the aorta (Figure 14(a)). We then placed a probing widget in another part of the dataset, where, according to what is already known, the aorta is dissected. The variance was high in that region, because not all segmentations were able to capture the two blood flow channels at this position. We then selected the previously identified six members, and compared them to the rest of the ensemble in the local region. Here it turned out that the six were able to capture the two blood flow channels (Figure 14(b)). Our collaboration partners then decided to work with the six members in the further course, because it was important that the lower parts of the aorta are segmented properly.

6 Discussion and Future Work

The demonstration cases showed that our proposed technique can help to identify patterns in an ensemble of segmentation masks. It can also be seen that our technique is scalable to large ensemble datasets. The full complexity of the datasets is only presented to the users in localized regions. This way it is possible to give a comprehensive overview of the data, without getting lost in the complexity of the dataset. We further allow the users to go back to the original data, which is an important task to finalize the analysis.

The **similarity graph** to analyze fixed probes is a complete graph. The graph is shown in 3D and in 2D. There are a lot of existing techniques for visualizing graphs in 2D, especially if the graphs are very large. The 2D representation could be, for example, replaced by a similarity matrix. Such representations are favorable when searching for patterns in the data. We, however, consider the 3D graph to be more important. In the 3D graph the spatial position of the nodes is clearly visible. This information is lost in the 2D representation. The main interactions are carried out in the 3D view, and the graph is updated there at every mouse move. This way users get immediate feedback if they touched a region similar to other nodes in the graph.

Our technique helps users to quickly identify members that produce different results in local regions of the data.

In some cases the **variance** is high, but no local outliers can be detected in the data. This is, for example, the case when segmentations produce slightly different results at the border of an object. In these cases the differences between the members are more or less equally distributed in the stacked bar chart, and no local outliers can be detected (Figure 15). On the other hand, such a case also gives some additional information about the data. It shows that in this area all members are producing slightly different results, so such regions are obviously difficult to segment properly.

Depending on the particular hardware configuration and the size of the ensemble dataset, our difference computations may run into **memory** problems. This is due to the fact that is necessary to calculate a distance map and a surface representation for every input volume, which means that another volume as large as the original one has to be kept on memory. In addition, per-voxel values for the mean, the median and the median absolute deviation have to be stored. New volumes for the variance values and the outliers have to be created. The volume holding the statistical parameters is deleted after the difference computation, though. We therefore implemented a caching mechanism, where only the datasets currently required for calculation are loaded into the memory. This decreases the memory requirements of the applications, but increases the computation time, since additional reads on the hard-disk become necessary.

The runtime for the **medoid computation** highly depends on the dataset size, and on the number of iterations. Since we operate on a very small sample set compared to the available space (comparing the $w \times h \times d$ -dimensional space to the n input volumes), the Weiszfeld algorithm converges very quickly. We therefore discovered that approximately 10 iterations are enough to let the algorithm converge in a point.

In the **future**, we would like to explore further how the graph information can be used for other analysis tasks.

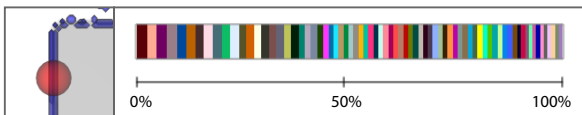


Figure 15: Dataset *Absence of outliers*. In the case no outliers can be detected at a local region, the stacked bar chart shows that the differences among the members are more or less equally distributed. In this case no outlier can be spotted, but it also tells the user that all members produce slightly different results in this region.

We also would like to improve the user interaction possibilities in 3D. Another interesting extension would be to let the user define interesting regions by placing the probes, and then automatically searching for similar regions. It would be necessary to sample the volume to detect similar regions. The sampling has to be dense enough to not miss important parts. In line with this idea, we would also like to find the most significant probe positions automatically, to further help the user to analyze the data. This could be implemented similar to the technique by Obermaier and Joy [Obermaier and Joy(2015)] for automatic slice plane placement. For the probes, however, it has to be considered that they can be placed at any arbitrary position in 3D, and that they can also vary in size. Our techniques are currently applied to segmentation masks, but they are designed in a rather general way. It would therefore be possible to also apply them to general volume data.

7 Conclusion

In this paper we presented techniques for the exploration of local features in an ensemble of volumetric segmentation masks. We calculate the per-voxel variance and outlier of the ensemble data. The spatial median, the so-called *medoid*, as evaluated by the Weiszfeld algorithm, is used as a representative of the ensemble data. The regions of high variance (i.e., variance values greater than the mean variance) are shown in the same view as the medoid. These regions are rendered as mesh data, to clearly separate them from the medoid data. Local regions in the data can be explored by 3D probing widgets. These widgets can be positioned by mouse interaction, and show the data characteristics at the current position in the separate detail view. The widgets can also be fixed at positions of interest. To visualize similarities between local regions, the probing widgets are arranged in a *similarity graph*. The probes are represented by nodes, and the edges in the graph indicate whether local regions are similar. The similarity is calculated by analyzing the statistical distribution of the local regions. The edges in the graph are drawn and colored differently, according to whether the whole ensemble should be analyzed, or whether one or more members should be compared against the rest of the ensemble. With the similarity graph users can interpret local features in a global context.

References

- O. S. Alabi, X. Wu, J. M. Harter, M. Phadke, L. Pinto, H. Petersen, S. Bass, M. Keifer, S. Zhong, C. Healey, and R. M. Taylor. Comparative visualization of ensembles using ensemble surface slicing. In *Proceedings of the SPIE, Visualization and Data Analysis*, volume 8294, 2012. doi: 10.1117/12.908288.
- A. Beck and S. Sabach. Weiszfeld’s Method: Old and New Results. *J. Optim. Theory Appl.*, 164(1):1–40, May 2014. doi: 10.1007/s10957-014-0586-7.
- S. Busking, C. P. Botha, L. Ferrarini, J. Milles, and F. H. Post. Image-based rendering of intersecting surfaces for dynamic comparative visualization. *Vis Comput*, 27(5):347–363, 2011. doi: 10.1007/s00371-010-0541-z.
- T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001. doi: 10.1109/83.902291.
- M. Cho, B. Kim, H. J. Bae, and J. Seo. Stroscope: Multi-Scale Visualization of Irregularly Measured Time-Series Data. *IEEE Transactions on Visualization and Computer Graphics*, 20(5):808–821, May 2014. doi: 10.1109/TVCG.2013.2297933.
- I. Demir, Ch. Dick, and R. Westermann. Multi-Charts for Comparative 3D Ensemble Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 20(12), Dec. 2014. doi: 10.1109/TVCG.2014.2346448.
- A. Fofonov, V. Molchanov, and L. Linsen. Visual Analysis of Multi-run Spatio-temporal Simulations Using Isocontour Similarity for Projected Views. *IEEE Transactions on Visualization and Computer Graphics*, PP(99), 2016. doi: 10.1109/TVCG.2015.2498554.
- T. Fogal, H. Childs, S. Shankar, J. Krüger, R. D. Bergeron, and P. Hatcher. Large Data Visualization on Distributed Memory multi-GPU Clusters. In *Proceedings of the Conference on High Performance Graphics*, HPG ’10, pages 57–66, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.
- B. Fröhler, Ch. Heinzl, and T. Möller. GEMSe: Visualization-Guided Exploration of Multi-channel Segmentation Algorithms. *Computer Graphics Forum*, 35(3), 2016. accepted for publication.
- M. G. Genton, Ch. Johnson, K. Potter, G. Stenchikov, and Y. Sun. Surface boxplots. *Stat*, 3(1):1–11, 2014. doi: 10.1002/sta4.39.
- A. Geurts, G. Sakas, A. Kuijper, M. Becker, and T. Von Landesberger. Visual Comparison of 3D Medical Image Segmentation Algorithms Based on Statistical Shape Models. In *Lecture Notes in Computer Science*, pages 336–344. Springer, 2015. ISBN 978-3-319-21069-8. doi: 10.1007/978-3-319-21070-4{-}34.
- L. Grady. Multilabel Random Walker Image Segmentation Using Prior Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 763–770, June 2005. doi: 10.1109/CVPR.2005.239.
- M. Haidacher, St. Bruckner, and M. E. Gröller. Volume Analysis Using Multimodal Surface Similarity. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1969–1978, 2011. doi: <http://doi.ieeecomputersociety.org/10.1109/TVCG.2011.258>.
- T. Höllt, G. Chen, Ch. D. Hansen, and M. Hadwiger. Extraction and Visual Analysis of Seismic Horizon Ensembles. In *Eurographics 2013 - Short Papers*. The Eurographics Association, 2013. doi: 10.2312/conf/EG2013/short/069-072.
- B. Iglewicz and D. Hoaglin. *How to Detect and Handle Outliers*, volume 16. ASQC/Quality Press, 1993.
- M. Jacomy. i want hue - Colors for data scientists. <http://tools.medialab.sciences-po.fr/iwanthue>, 2016. [Online; accessed 24-March-2016].
- M. Jarema, I. Demir, J. Kehrer, and R. Westermann. Comparative visual analysis of vector field ensembles. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*, VAST ’15, pages 81–88, Oct. 2015. doi: 10.1109/VAST.2015.7347634.
- T. Ju, S. Schaefer, and J. Warren. Mean Value Coordinates for Closed Triangular Meshes. *ACM Trans. Graph.*, 24(3):561–566, Jul. 2005. doi: 10.1145/1073204.1073229.
- Joe Kniss, Gordon Kindlmann, and Charles Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):270–285, Jul. 2002. doi: 10.1109/TVCG.2002.1021579.

- P. Kohlmann, S. Bruckner, A. Kanitsar, and M. E. Gröller. Contextual picking of volumetric structures. In *Proceedings of the IEEE Pacific Visualization Symposium*, PacificVis '09, pages 185–192, Apr. 2009. doi: 10.1109/PACIFICVIS.2009.4906855.
- W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, Aug. 1987. doi: 10.1145/37402.37422.
- M. M. Malik, C. Heinzl, and M. E. Groeller. Comparative Visualization for Parameter Studies of Dataset Series. *IEEE Transactions on Visualization and Computer Graphics*, 16(5):829–840, Sep. 2010. doi: 10.1109/TVCG.2010.20.
- K. Matkovic, D. Gracanin, M. Jelovic, A. Ammer, A. Lez, and H. Hauser. Interactive Visual Analysis of Multiple Simulation Runs Using the Simulation Model View: Understanding and Tuning of an Electronic Unit Injector. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1449–1457, 2010.
- M. Mlejnek, P. Ermes, A. Vilanova, R. van der Rijt, H. van den Bosch, F. Gerritsen, and M. E. Gröller. Profile Flags: A Novel Metaphor for Probing of T2 Maps. In *Proceedings of the IEEE Conference on Visualization*, VIS '05, page 76, Los Alamitos, CA, USA, 2005. IEEE Computer Society. doi: <http://doi.ieeecomputersociety.org/10.1109/VIS.2005.81>.
- H. Obermaier and K. I. Joy. An Automated Approach for Slicing Plane Placement in Visual Data Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 21(12):1403–1414, Dec. 2015. doi: 10.1109/TVCG.2015.2414455.
- D. Patel, S. Bruckner, I. Viola, and M. E. Gröller. Seismic volume visualization for horizon extraction. In *Proceedings of the IEEE Pacific Visualization Symposium*, PacificVis '10, pages 73–80, Mar. 2010. doi: 10.1109/PACIFICVIS.2010.5429605.
- T. Pfaffelmoser, M. Reitinger, and R. Westermann. Visualizing the Positional and Geometrical Variability of Isosurfaces in Uncertain Scalar Fields. *Computer Graphics Forum*, 30(3):951–960, 2011. doi: 10.1111/j.1467-8659.2011.01944.x.
- M. N. Phadke, L. Pinto, O. Alabi, J. Harter, R. M. Taylor, X. Wu, H. Petersen, S. A. Bass, and Ch. G. Healey. Exploring Ensemble Visualization. In *Proceedings of the SPIE, Visualization and Data Analysis*, volume 8294, 2012. doi: 10.1117/12.912419.
- K. Potter, A. Wilson, P.-T. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. Johnson. Ensemble-Vis: a framework for the statistical visualization of ensemble data. In *Proceedings of the IEEE International Conference on Data Mining, Workshops*, pages 233–240, 2009.
- M. Raj, M. Mirzargar, R. M. Kirby, R. T. Whitaker, and J. S. Preston. Evaluating Alignment of Shapes by Ensemble Visualization. *IEEE Computer Graphics and Applications*, 2015. doi: 10.1109/MCG.2015.70. accepted for publication.
- M. Rautenhaus, C. M. Grams, A. Schäfler, and R. Westermann. Three-dimensional visualization of ensemble weather forecasts - Part 2: Forecasting warm conveyor belt situations for aircraft-based field campaigns. *Geoscientific Model Development*, 8(7):2355–2377, 2015. doi: 10.5194/gmd-8-2355-2015.
- T. Ropinski, I. Viola, M. Biermann, H. Hauser, and K. Hinrichs. Multimodal Visualization with Interactive Closeups. In *EGUK Theory and Practice of Computer Graphics*, pages 17–24, June 2009.
- T. Saito and J. Toriwaki. New algorithms for euclidean distance transformation of an n-dimensional digitized picture with applications. *Pattern Recognition*, 27(11):1551–1565, 1994.
- J. Schmidt, M. E. Gröller, and S. Bruckner. VAICO: Visual Analysis for Image Comparison. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2090–2099, Dec. 2013. doi: 10.1109/TVCG.2013.213.
- J. Schmidt, R. Preiner, T. Auzinger, M. Wimmer, M. E. Gröller, and S. Bruckner. YMCA - Your Mesh Comparison Application. In *IEEE Visual Analytics Science and Technology*, VAST '14. IEEE Computer Society, Nov. 2014.
- D. Sibbing and L. Kobbelt. Fast Interactive Region of Interest Selection for Volume Visualization. In A. Horsch, T. M. Deserno, H. Handels, H.-P. Meinzer, and T. Tolxdorff, editors, *Bildverarbeitung für die Medizin 2007*, volume 12, pages 338–342. Springer Berlin Heidelberg, 2007. doi: 10.1007/978-3-540-71091-2_68.
- Mat'us Straka. *Processing and Visualization of Peripheral CT-Angiography Datasets*. PhD thesis, TU Wien, Institute of Computer Graphics and Algorithms, Vienna, Aug. 2006.

- T. Torsney-Weir, A. Saad, T. Möller, H.-C. Hege, B. Weber, and J.-M. Verbavatz. Tuner: Principled Parameter Finding for Image Segmentation Algorithms Using Visual Response Surface Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1892–1901, Dec. 2011. doi: 10.1109/TVCG.2011.248.
- D.G. Wakeman and R. N. Henson. A multi-subject, multi-modal human neuroimaging dataset. *Scientific Data*, 2, Jan. 2015. doi: 10.1038/sdata.2015.1.
- A. Wiebel, F. M. Vos, D. Förster, and H. C. Hege. WYSIWYP: What You See Is What You Pick. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2236–2244, Dec. 2012. doi: 10.1109/TVCG.2012.292.
- C. Zhang, T. Schultz, K. Lawonn, E. Eisemann, and A. Vilanova. Glyph-Based Comparative Visualization for Diffusion Tensor Fields. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):797–806, Jan. 2016. doi: 10.1109/TVCG.2015.2467435.