

GEMSe: Visualization-Guided Exploration of Multi-channel Segmentation Algorithms

B. Fröhler^{1,2} and T. Möller² and C. Heinzl¹

¹University of Applied Sciences Upper Austria, Campus Wels, Austria

²Research Group for Visualization and Data Analysis, University of Vienna, Austria

Abstract

We present GEMSe, an interactive tool for exploring and analyzing the parameter space of multi-channel segmentation algorithms. Our targeted user group are domain experts who are not necessarily segmentation specialists. GEMSe allows the exploration of the space of possible parameter combinations for a segmentation framework and its ensemble of results. Users start with sampling the parameter space and computing the corresponding segmentations. A hierarchically clustered image tree provides an overview of variations in the resulting space of label images. Details are provided through exemplary images from the selected cluster and histograms visualizing the parameters and the derived output in the selected cluster. The correlation between parameters and derived output as well as the effect of parameter changes can be explored through interactive filtering and scatter plots. We evaluate the usefulness of GEMSe through expert reviews and case studies based on three different kinds of datasets: A synthetic dataset emulating the combination of 3D X-ray computed tomography with data from K-Edge spectroscopy, a three-channel scan of a rock crystal acquired by a Talbot-Lau grating interferometer X-ray computed tomography device, as well as a hyperspectral image.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications—I.4.6 [Image Processing and Computer Vision]: Segmentation—Pixel classification

1. Motivation

Many scientific disciplines such as material sciences, medicine or geosciences attempt to gain novel insights by acquiring and analysing two- or three-dimensional datasets from a specimen of interest [CCK*14, GFL04, KLF*15]. In most cases, image analysis techniques focus on data acquired by one scanning modality returning a single data value for each spatial location. However, data from single scanning modalities may return unclear or ambiguous responses, or may not deliver all information that is needed. Material scientists, for example, often require the exact topological details of a specimen along with minuscule structural deficiencies as well as information on its chemical composition. To address these demands, the analysis strategies are increasingly adapted to measuring the same scene or specimen using multiple techniques, resulting in a so-called multi-modal dataset, where multiple measurement values are available per dataset location. At times a single scanning modality can also result in multiple measurement values per dataset location. Here, we will use the term multi-channel to refer to any dataset or image where multiple values are available per spatial position.

Image segmentation, the process of partitioning an image into regions sharing common properties, is a very important step of the image analysis workflow in many domains. In industrial applica-

tions it is for example essential in quantifying features of interest, such as voids or defects, or constituent components of a material system like fibers, particles or their surrounding material. Many different segmentation algorithms have been developed, typically tailor-made for a specific target domain. Adapting and parameterizing segmentation pipelines to new modalities or to different inputs is time-consuming and non-trivial. Typically no segmentation expert is at hand to lend her expertise to the domain scientist. Often, there is also no objective measure available and therefore no automatic optimization can be performed. Thus, domain experts can only determine whether a segmentation result is meaningful through visual inspection.

Considering multiple channels of information during segmentation leads to complex algorithms with many input parameters which require careful tuning. Domain scientists, being no segmentation experts, typically have no guidance available for this tuning procedure. The traditional way to determine the influence of the segmentation pipeline's parameters on the result is to manually perform iterative trial and error steps.

We therefore see a strong need for approaches to evaluate the input parameter space and the result space of segmentation algorithms for multi-channel images. A tool implementing these methods will allow domain experts to arrive at optimal segmentation

results for their multi-channel images without ground truth information available and without having advanced expertise in image segmentation. We developed and implemented GEMSe to address this problem. The main contributions of our work are found in the following points:

- Methods for exploring and analyzing the input parameter and result space of a segmentation pipeline.
- Implementing these methods in GEMSe, which guides users towards choosing adequate segmentation parameter values leading to optimal results for a specific kind of dataset.
- Evaluation of the usability of GEMSe in various case studies of different application domains.

2. Related Work

Visually supporting segmentation algorithms of 3D data has been one of the most researched topics in the visualization community. The typical approach is through defining a transfer function [HKRs*06]. For an overview of different transfer function approaches we refer the reader to a recent survey [AD10]. However, transfer functions suffer from two problems. On the one hand they are not very intuitive to use. On the other hand, they are not able to properly segment very complex modalities.

The area of computer vision in general and medical imaging in particular have been focused on developing very sophisticated segmentation algorithms. One of the most widely used approaches today is the random walker algorithm [Gra05]. Hence we are using this algorithm in our approach. However, we are not limited to this algorithm and it could be exchanged with any other. Visually supporting such a segmentation pipeline has also been studied in the visualization community. Saad et al. [SMH10, SHM10] were some of the first to provide visual support to explore the capabilities of the random walker algorithm. Their focus was to use the probabilistic features of the algorithm in order to find anomalies in the segmentation that might uncover tumor tissue. Pražni et al. [PRH10] as well as Top et al. [THA10] introduced approaches to use the segmentation uncertainty to query the user for input where the labeling is too uncertain.

However, in this work we are not focusing on the uncertainty information of the algorithm thus far. Instead we are dealing with the problem of finding good parameter settings for the segmentation algorithm to perform well. Systematically exploring the parameter space of algorithms has become an important research topic in the visualization community over the last few years. Sedlmair et al. [SHB*14] present a recent overview and taxonomy of the approaches in the literature. Using this approach, Tuner [TWSM*11] as well as Paramorama [PBCR11] are possibly the two approaches most closely related to our work.

Tuner was developed for medical image segmentation experts to help them fine-tune a particular energy model. Hence, it required a ground-truth segmentation or at least one objective quality measure in order to function properly. However, in our approach we aim to support domain experts (that are not necessarily algorithmic experts in segmentation) and we do not require any ground truth or objective measure. Instead we navigate the user through segmentation space and rely on visual inspection to help find proper

segmentations. Hence, the interfaces of Tuner and GEMSe have no resemblance (although they provide similar functionality).

Paramorama [PBCR11] on the other hand lets the user browse through a large collection of segmented images. A specialized layout algorithm allows users to manually compare the label images in one or more clusters. While they require using uniform sampling on a fixed cartesian grid, which does not scale well, we can employ any sampling method. They cluster by parameter values whereas we cluster by result similarity.

The major difference of our approach to all these previous work is that we (a) provide a general segmentation framework for many different image modalities, and (b) that we are focusing on multi-channel data. To the best of our knowledge, none of the previous work has these capabilities.

3. Problem Characterization

The main challenge this work addresses is analyzing the input parameter space of segmentation algorithms for multi-channel data and the set of potential label images that they can produce. This problem can be categorized as visual parameter space analysis, as systematized by Sedlmair et al. [SHB*14]. The input is a multi-channel, two- or three-dimensional dataset with an arbitrary number of channels per location. We will use the word *image* to denote conventional 2D images as well as 3D volumes. Our segmentation pipeline furthermore takes a parameter set as input, consisting of various numeric or categorical parameters. A detailed description of the particular parameter set we investigated is given in Section 4.1. The output of one segmentation run is referred to as label image, which is a single-channel image. As derived output, we calculate the count of connected components in the label image, and record the time required by the segmentation algorithm. We used the count of connected components, or *object count*, mainly because it was used by the domain scientists we worked with, but also because it was used in the Paramorama tool [PBCR11]. Measuring the performance is intended for a trade-off analysis between result suitability and algorithm speed.

We sample over the parameter space of this model, resulting in a collection of label images, henceforth referred to as ensemble. Each label image is linked to the parameter set that produced it and its derived output. We provide a global-to-local approach to explore this ensemble.

3.1. Tasks

Through collaborating with domain scientists working on segmenting industrial multi-channel datasets and analyzing their current workflow and analysis needs, the following tasks to be solved were identified:

- **T1:** Get an overview over the possible results produced by the algorithm.
- **T2:** Analyze the influence of input parameters on the output images and their correlation to derived output.
- **T3:** Find stable parameter combinations which produce suitable segmentation results as determined by visual inspection.

In terms of Sedlmair et al. [SHB*14], T1 and T2 fall into the partitioning category, as they are applied to get an insight into what variation of results can be achieved. T3 is a combination of optimization and sensitivity analysis – users first want to arrive at ranges of parameter values producing suitable results, and then identify those regions where the results are insensitive to slight parameter changes.

4. GEMSe Design

For a given multi-channel image we first perform preprocessing as schematized in Figure 1. The analysis work is subsequently done through the main interface of GEMSe as shown in Figure 2.

4.1. Preprocessing

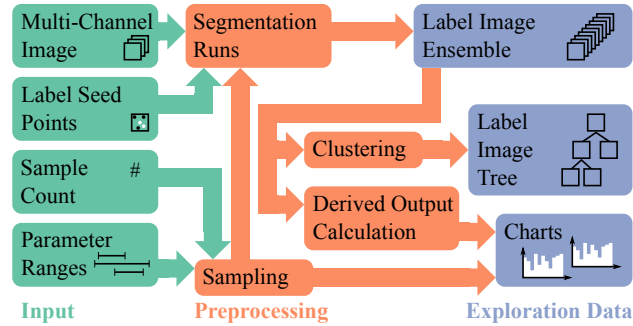


Figure 1: Overview over input, preprocessing pipeline and the data explored by GEMSe.

As depicted in Figure 1, preprocessing starts by sampling the parameter space of a segmentation pipeline. For this purpose users can either use default parameter ranges or refine ranges if they already have a preconception of where suitable results are more likely to occur, for example from a previous sampling run. We assemble parameter sets according to the chosen sampling method and parameter ranges. The result of a single run of the segmentation pipeline is referred to as label image. We perform one segmentation run for each parameter set, resulting in an ensemble of label images. We calculate the object count as the count of connected components of all resulting label images, a simple derived output which can help in determining the segmentation quality, as will be shown in our case studies. On the ensemble we finally perform hierarchical clustering using maximum linkage [FLPZ51]. For that we initially set up each label image as leaf cluster and calculate pairwise similarities as sum of the dice metric [Dic45] calculated for each label. These preprocessing steps typically take a long time (in the range of several hours to days depending on the used hardware) and are therefore performed in advance and detached from the result exploration.

For segmentation, we have adopted a multi-channel segmentation framework around the Support Vector Machine (SVM) classification algorithm and the Extended Random Walker (ERW) segmentation algorithm [Gra05]. Our current sampling tool uses this SVM+ERW-based segmentation pipeline but could be easily adapted to run any other segmentation pipelines or algorithms.

common		
SVM	C_{svm}	soft classification penalty
	γ_{svm}	Gaussian RBF kernel width parameter
	n_{svm}	number of channels to consider
ERW	β_{erw}	normalization neighborhood weight
	γ_{erw}	weight of prior model vs. neighborhood
	m_{erw}	max. number of iterations in linear solver
modality-specific		
	w_i	channel i weight in neighborhood information
		$i \in [1..N]$, N = number of image channels
	pca_i	number of PCA components considered for channel i
	$dist_i$	Distance metric used for channel i

Table 1: Parameters to our custom segmentation framework.

We have chosen both algorithms because they are successfully used in many application domains, for example in medical sciences [GFL04] and the geosciences [KLF*15]. In this setup, there are six common parameters and two to three additional parameters per data modality in use to our segmentation pipeline, which are summarized in Table 1. In addition to input parameters, the SVM takes seed points for each label as input. We use the same set of seed points, predetermined by the user, for each segmentation run of the same dataset at the moment.

4.2. Result Exploration Interface

Figure 2 shows the main interface of GEMSe. Its layout is partially fixed, meaning that the single views cannot be moved around, but they can be resized to enlarge areas of interest. We initially considered a vertically panning screen similar to Bederson and Meyer’s Pad++ system [BM98], so that we would not be limited to the size of a single screen, and to better separate the different analysis layers. We found the downsides of such an approach, namely the lack of overview and the confusion of the user having to switch through many screens, to outweigh the benefits and therefore discarded that idea.

In our interface there are seven linked views. A *cluster tree view* (Figure 2a) displays the hierarchy resulting from our clustering, the *cluster example view* (Figure 2d) displays characteristic images from the selected cluster, the *detail view* (Figure 2b) shows a large version of the currently selected image, the *slice view* (Figure 2c) enables switching between the three possible axis-aligned slice views as well as changing the currently shown slice, the *histogram view* (Figure 2g) shows histograms for input parameters and derived outputs, the *scatterplot view* (Figure 2f) correlates input parameters with derived outputs, and finally the *favorite bar* (Figure 2e) keeps track of preferred images. All interactions in an image view are instantaneously propagated to all other image views for better comparison. Label images are color-coded using a qualitative scheme from ColorBrewer [HB03], the datasets we studied so far only required 2-7 labels, so we do not see a problem in the limitation to a maximum of twelve colors.

The *cluster tree view* (Figure 2a) shows an interactive tree view of the cluster hierarchy. Visual appearance and interaction methods resemble those from the folder view in a file explorer such as Windows Explorer. For each cluster a node with the number of con-

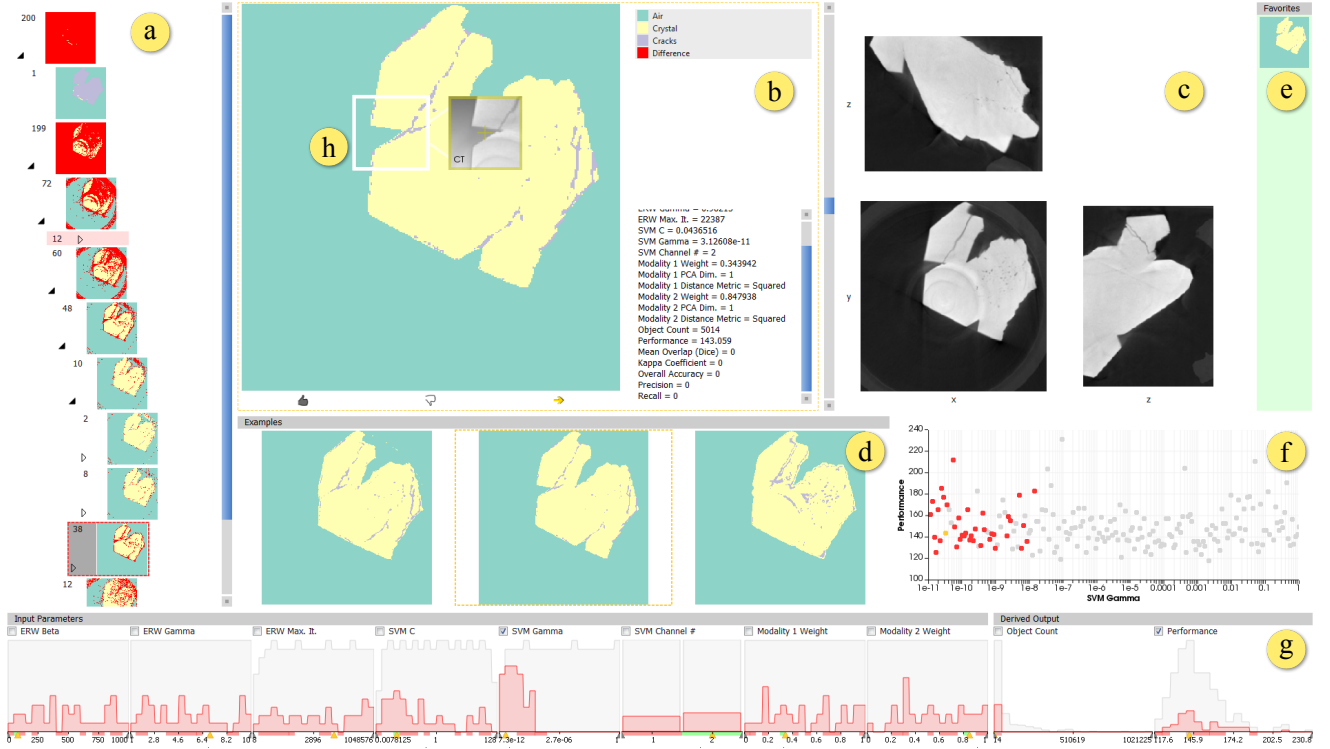


Figure 2: Main interface of GEMSe: (a) cluster tree view, (b) detail view, (c) slice view, (d) cluster example view, (e) favorite bar, (f) scatterplot view, (g) histogram view, (h) magic lens showing original image

tained samples is shown alongside a *cluster representative* image. For a leaf node in our hierarchy, the representative is the label image contained in it. For internal nodes, this image should give an overview over amount and location of differences in the label images of this cluster. Our first idea was to show the medoid of the label images in the cluster. Another option we considered was to calculate an average image from all the masks. A medoid, being a single label image, same as an average image, can not give an adequate impression of the range of images contained in the cluster. An average *label image* is furthermore not meaningful, since it does not necessarily represent any single member of the cluster. We therefore implemented a visual metaphor following similar ideas as the homogeneity view introduced by Malik et al. [MHG10]. All pixels which have the same label in all samples retain that labeling, and all other pixels are replaced with a marker color not used for labeling. This directly visualizes the regions of the image in which a cluster contains variation. The system tries to minimize the space used by the tree. It automatically hides the preview images of parent and sibling nodes when expanding a tree node as well as nodes that are marked as unsuitable. The cluster tree view is mainly addressing task T1. Since selections of a node in the cluster tree view result in linked updates in all other views, it also acts as main interaction hub. Task T3 is mainly solved through the interaction of all views, therefore the cluster tree view plays a vital role in addressing it.

The *cluster example view* (Figure 2d) shows a number of exemplary label images from the currently selected cluster. These are se-

lected so that they represent the variation in the cluster, supporting the overview task T1. The number n of label images to be shown is determined based on the available space. Following the idea of Krishnamachari and Abdel-Mottaleb [KAM99], we choose leafs from a parent cluster in the same relation as the number of leafs in each child cluster. In case we need to choose a single item from a cluster which is no leaf, a random child is picked. This is executed recursively, and will lead to n images, or less if the number of leafs in the selected cluster is smaller than n .

Depending on the previous interaction, the *detail view* (Figure 2b) shows an enlarged version of either the cluster representative selected in the cluster tree view, or a label image selected in the cluster example view. The border color of the view indicates the current state, red indicating a cluster representative, yellow a single label image. Next to the image additional detail information about the parameter sets producing the label images in this cluster, and the cluster ID are displayed as can be seen on the right side of Figure 2b. Comparison to the original images is enabled through a magic lens (Figure 2h). Additionally, users are presented with interaction elements to positively or negatively rate a segmentation result, and to navigate towards a specific label image, as can be seen in the bottom of Figure 2b.

In the *histogram view* (Figure 2g), we followed the idea of scented widgets [WHA07] and implemented histograms for all parameters and derived outputs which are connected to filter sliders on the axes. The view is designed to address task T2, the analysis

of correlations between clusters of label images and input parameters or derived output. By default our histograms show the result count on the vertical axis over the respective parameter or derived output on the horizontal axis. The horizontal axis is scaled linearly or logarithmically depending on the way the parameter was sampled. The data of the whole result space is shown in gray, data of the selected cluster is shown in the same color as the highlight of a selected cluster in the cluster tree view. When a single example image is selected, its parameters are marked with yellow arrows on the horizontal axis.

The *slice view* (Figure 2c) is intended for navigation through the whole image. The original image is shown for comparison with label images in the form of three axis-aligned slicer images, as domain experts are used to such slice views. Selecting one of the three small slicers changes the slice axes for all previews. Providing a 3D view was considered in the design process of GEMSe, but we observed that the domain scientists we collaborated with tended to rarely use 3D visualizations of the dataset, and preferred slice views due to their better interpretability. The *favorite bar* (Figure 2e) is used for collecting intermediate results for later reference. With it we address the issue that when analyzing any large collection, it is easy to get lost or to loose track of previous findings. The *scatter-plot view* (Figure 2f) shows a correlation of two attributes marked in the histogram view, intended for addressing task T2. The data points of the currently selected cluster, potential filter ranges and example image selections are highlighted here as well.

Ensemble members or whole clusters can be marked as suitable or unsuitable. This results in a color-coding on the x-axes of the parameter histograms, which enables the correlation of input parameter ranges to segmentation quality.

5. Implementation

We implemented GEMSe as a standalone application in a custom C++ application framework built on top of the Qt framework, using VTK [SML06] and ITK [YAL*02] libraries for visualization and image processing. The sampling of the segmentation parameter space, resulting in the ensemble of segmentation masks, was implemented inside the same framework. For the segmentation framework we used the libSVM library [CL11], our extended random walker calculation is built on top of the Eigen v3 linear algebra library [GJ*10]. The source code of our tool will be made available on <https://github.com/3dct>.

6. Evaluation

We applied two different types of evaluation to our prototype. First we conducted usability interviews to refine visualization and interaction methods. This led to an improved version of our tool, addressing the main concerns raised during these interviews. We subsequently used this improved tool in several real world as well as synthetic scenarios.

6.1. Usability Evaluation

GEMSe was developed in constant collaboration with domain experts in material science. Our design was therefore heavily influenced by their input. To evaluate and improve its usefulness on

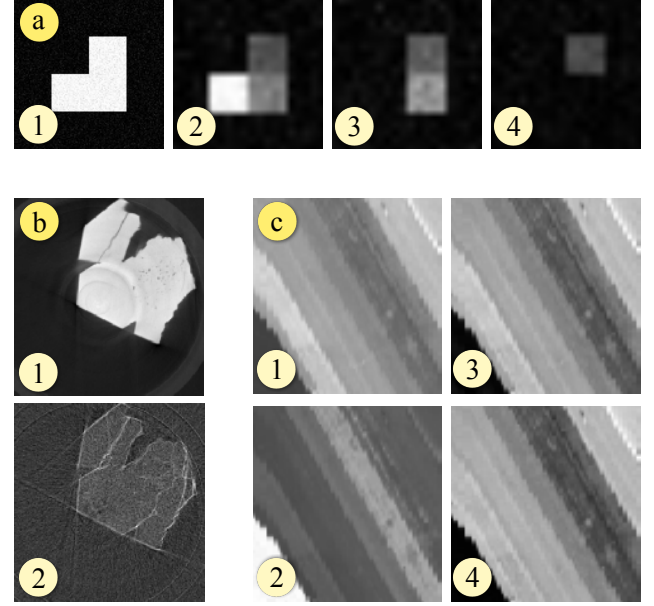


Figure 3: Datasets used in our case studies: Attenuation image (a1) and three K-Edge element maps (a2-a4) of the synthetic K-Edge dataset. Attenuation image (b1) and dark field image (b2) of the middle slice along x-y-axis of the rock crystal TLGI-XCT scan. Channels 26(c1), 46(c2), 125(c3) and 176(c4) from the SalinasA hyperspectral dataset.

a broader scale, we asked five visualization professionals to take a look at our tool. Following the best practices established by Nielsen [Nie00], we performed approximately one-hour long interviews over the course of four weeks. We used the small synthetic dataset as shown in Figure 3a for the evaluation. We first introduced each user to our tool. Then we encouraged the users to interact with the tool to get acquainted with the modes of interaction. During this phase, our users implicitly addressed task T1 from Section 3.1, namely to get an overview over the possible results. Subsequently we posed two explicit exercises for them to solve through the help of our tool: The first exercise consisted of finding out what was the determining factor for the small amount of obviously unsuitable label images in the synthetic example as shown in Figure 5(a5), a specific instance of task T2. The second, more open-ended exercise was to find the visually most satisfying segmentation result and deducing reasonable and stable parameters for segmenting this dataset, basically task T3. We observed our users in performing these tasks and took notes on how they were doing. Additionally, we encouraged them to comment on their current thoughts. For all sessions we recorded screen captures and the conversation. We used this feedback to improve our software prototype, in order to arrive at the most effective ways of visualization and interaction.

Users started their exploration with the root node of the hierarchical clustering selected. To get an overview over the possible results, three of our users navigated down in the tree hierarchy, checking cluster representatives and continuing in the direction of the result which matched their expectation most. Interest-

ingly, the benefits of the cluster example view were controversial. Two users reported that they found the view tremendously helpful and used it as their main navigation cue, chose the label image that they found most suitable, and navigated towards that image. Two users did not use it at all and commented that they found it confusing. We think this suggests different interaction preferences in our users, both of which our tool supports. For the first exercise, finding the parameter responsible for a cluster of unsuitable label images, most of our users observed the histogram view while switching between clusters. Within at most one minute all of our users spotted the parameter γ_{erw} mainly responsible for the difference. Most users required encouragement by the interviewer to employ filtering, facilitated through the slider handles below the histograms, which could also have helped solving this exercise. Once aware, our users commented very positively on this feature, especially the immediate feedback in the cluster hierarchy. There, the number of masks matching the current filter is shown alongside the number of all masks in that cluster, and clusters with all contained label images outside of the filter range are minimized. The scatterplot view, which would allow detailed correlation analysis, was also only used after a reminder by the interviewer. It was not perceived as that helpful. It might not be required at all, but might be used more if it had permanent visibility and deeper integration into the linked views.

For our second exercise, users continued their exploration as they did while getting an overview. They typically arrived at a broad range of parameters suitable for the dataset within a few minutes. We received as common feedback that it was perceived as hard to arrive at a final stable parameter region. A part of these users missed methods to compare the parameter ranges in which favorites were located. The second remark in this direction was that it is hard to spot whether there are any bigger differences left in a cluster. Some users commented that it would be useful to zoom in on a particular subtree, for example when it has become clear that good results are only to be found in one particular cluster. As during these usability sessions we still had one large original image next to the detail view, one common suggestion was to allow a more direct comparison between the label images and the original image. The cluster representative in general was perceived very helpful in determining the variation inside a cluster. When comparing siblings in the hierarchy, we made the observation that the cluster representative can have a slight misleading effect. Users tended to interpret a representative showing many differences as generally worse than one with less differences. The full picture only unravels however when considering the number of label images inside the cluster, as well as the location of differences and the number of label images that contributed to one particular difference. Two of our users suggested to provide filtering capabilities directly in the result space. Their argument was that from looking at the original they already formed a mental image of how the resulting label image should look like. They thus wanted to restrict the shown results to the ones having specific label values at certain positions, and excluding those results which exhibited, for example, noise artifacts.

Based on the feedback from these interviews, we introduced several improvements into GEMSe. To allow better comparison to the original, we implemented a magic lens for overlaying the original data in the detail view directly, as shown in Figure 2h. To im-

	Rock Crystal	Synthetic	Salinas
Size	285x300x216	120x120x8	83x86
Channels	3	4	214
Labels	3	4	6
# of Samples	200	100	500
SVM C	0.01..100 ^l	-	10 ⁻⁵ ..10,000 ^l
γ_{svm}	10 ⁻¹¹ ..1 ^l	-	10 ⁻¹¹ ..0.01 ^l
SVM η	1..2	-	1..224
β_{erw}	0.1..1,000	0.1..1,000 ^l	0.1..10,000
γ_{erw}	1..10	0.1..10	1..10
m	10..1,000,000 ^l	-	10..10,000
w	0..1	0.25..1/0..0.75	-

Table 2: Properties and sampling ranges of datasets used in our case studies. ^l indicates a logarithmic sampling scale.

prove the navigation towards a final suitable parameter range we introduced a coloring of the parameter axes in regions of good and bad label images. Clusters can be marked as suitable or unsuitable through interaction elements in the detail view. These markings determine a coloring of the histogram axes in the histogram view. We calculate the ratio of these suitable to unsuitable label images for each histogram bin. If there are only suitable label images in a bin, then it is marked green, and red if there are only unsuitable ones, as shown in Figure 4(l, o). In between we map to grey and reduce the opacity, so that an equal distribution will result in no coloring.

6.2. Case Studies

We have successfully used our tool on several datasets from different domains. In the following paragraphs we will present three of them. The details of these datasets are shown in Table 2.

6.2.1. Talbot-Lau grating interferometer X-ray computed tomography data

Our main source of test datasets is Talbot-Lau Grating Interferometry X-ray Computed Tomography (TLGI-XCT), which generates three data channels: An attenuation image as known from conventional X-ray computed tomography, the differential phase contrast is related to the index of refraction and the dark-field contrast reflects the total amount of radiation scattered at small angles. We supplied our tool to two domain experts working with phase contrast datasets. We analyze the *Rock Crystal* dataset here, which depicts a quartz structure with several cracks and pores inside. In this case we used attenuation and dark field image, a slice of each is shown in Figure 3b. The attenuation image alone can not resolve all cracks. The dark field image can, but typically over-emphasizes them and also shows high signal near the border of the crystal. The main goal was to segment all cracks and pores in the crystal.

Starting with getting an overview according to task T1, the root cluster as seen in Figure 2a shows variations of segmentation labels for nearly every pixel of the image. The second child cluster of this root is the minimized top cluster in Figure 4a. Its first child a1, currently selected, looks promising, where most of the air is correctly labeled (in green), as well as a big portion of the crystal is labeled as such (in yellow). Its child node a2 shows solid round structures labeled as cracks, which are also visible in the cluster

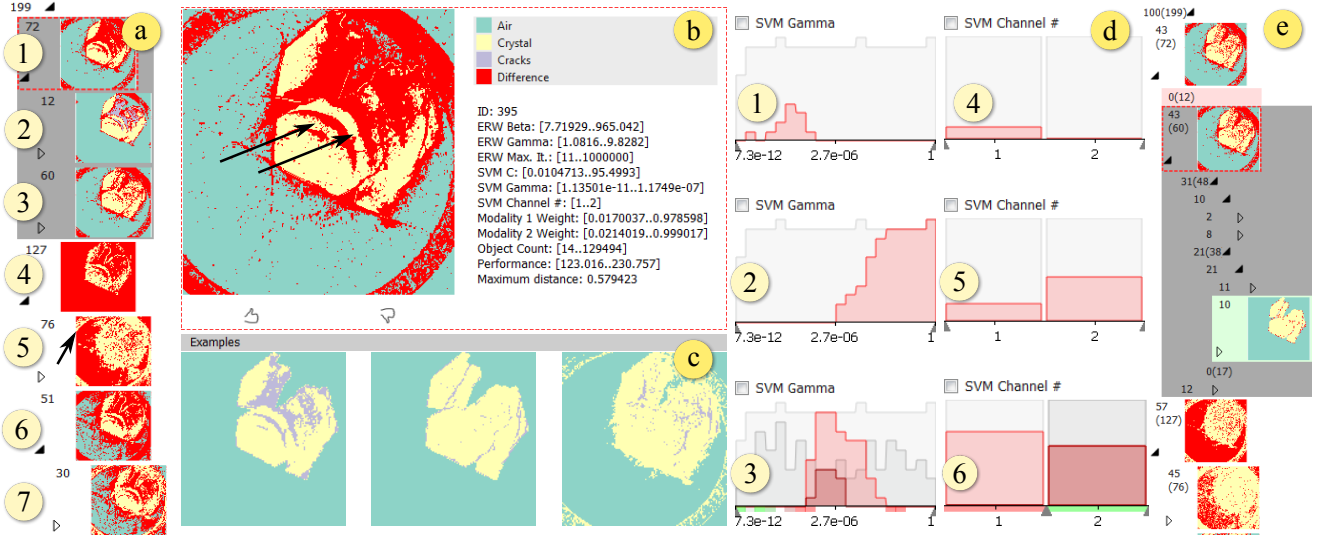


Figure 4: GEMSe with the rock crystal dataset loaded. Clusters (a1-7) show an overview result ensemble. For the selected cluster a1, the enlarged representative is shown in the detail view (b) along with exemplary label images (c) from that cluster. Histograms d1-d3 show the distribution of γ_{svm} for clusters (a2, a5, a6). Histograms d4-d6 show the distribution of the number of channels used in SVM for clusters (a2, a3, a4). The filter set in d6 affects the cluster tree view (e), also clusters rated suitable and unsuitable are highlighted here, and their parameter ranges are color-mapped accordingly in histograms d3 and d6.

representative in the detail view Figure 4b and marked there with arrows. These are not actual crack structures but unwanted artifacts. We therefore want to determine the parameter causing these to be wrongly labeled, an instance of task T2. A look at the histogram for the number of considered SVM channels d4 for cluster a2 reveals that all masks in it were produced using only one data channel. This could be a hint that considering all datasets for SVM will lead to results without these artifacts. Cluster a3, the sibling of a2, does not show these artifacts. Its SVM channel histogram d5 indicates that there were some masks in that node produced also with only one data channel, but significantly more used data available in both channels. To determine this correlation more closely, we filter for a high SVM channel number in histogram d6 of cluster a4. The cluster tree view in Figure 4e shows the filtered cluster representatives, which are calculated only from matching label images. There, the artifacts are not visible anymore.

When looking at the node a5 we see that it contains masks which all over-segment the crystal label, and in addition show a large ring artifact which should be labeled as air, marked with an arrow. The corresponding γ_{svm} histogram d2 shows high values in this cluster. γ_{svm} influences the width of the bell-shaped surfaces in the Gaussian RBF kernel used for separation in SVM. The underlying gray plot representing the distribution in the whole result set shows that the majority of masks with an SVM gamma parameter in the upper third of the range are contained in this cluster. Its sibling a6, where the ring artifact is still visible but less pronounced, shows values in the middle of the range with the corresponding histogram d3. Histogram d1 for node a2, whose representative does not show the large ring artifact, was produced using low γ_{svm} values. For suitable results, γ_{svm} must therefore be selected from the lower end of

the range. The threshold between suitable and unsuitable results, as determined through interactive filtering, is at approximately 10^{-9} .

For further refinement of the parameter ranges, as required by task T3, we rate clusters using the interaction elements shown below the detail image in 4b. Rated clusters are highlighted in the cluster tree view in Figure 4e and on the axes of histograms d3 and d6. For γ_{erw} , the weight factor of the prior model in the ERW, we determine through filtering that very low values are producing noisy results. Values higher than 1.1 result in well-suited label images. Higher values of the parameter C_{svm} , allowing soft margins but penalizing wrongly classified points in the SVM, tend to slightly over-segment the crystal. For the other parameters the situation is less clear, with this sampling we could not determine a direct correlation with a suitable segmentation outcome. One possibility would be to re-sample these with fixed values for the most influential parameters γ_{svm} and n_{svm} . The best label image we could find had parameters $\beta_{erw} = 85.38$, $\gamma_{svm} = 6.98$, $C_{svm} = 0.044$, $\gamma_{svm} = 3.13 \times 10^{-11}$ and $n_{svm} = 2$.

6.2.2. Synthetic K-Edge Data

K-Edge absorptiometry is a new scanning modality in X-ray computed tomography devices, which is enabled by a new generation of X-ray detectors in computed tomography (CT) devices [FNMA06]. The X-ray attenuation image from a conventional CT device can provide high contrast, but typically limited information on the elemental composition. K-Edge absorptiometry delivers elemental concentrations at the cost of a higher noise level and reduced resolution. In our synthetic dataset shown in Figure 3a, the whole specimen is modeled to have the same average attenuation value, and three K-Edge absorptiometry images each give the concentration

of a single element. The sampling here was done with a limited segmentation framework using a fixed prior model derived from thresholding instead of SVM, thus the SVM parameters do not have ranges listed in Table 2.

In the cluster representative of the root node as shown in Figure 5(a1) every pixel has the difference marker color. The first overview conclusion, solving task T1 is therefore that inside this cluster there is not a single pixel with a consistent label throughout all images. Looking at the two child nodes, we see a division between one larger, homogeneous and well-segmented cluster of images in a2 and another node a5 showing highly varying results. Starting a correlation analysis task T2, the γ_{erw} parameter histogram b2 for node a5 reveals that those images were all generated with low γ_{erw} values. For node a2 on the other hand the γ_{erw} values are all higher as shown in the histogram in b1. Apparently low settings of γ_{erw} deliver invalid results for this kind of dataset. The exact threshold value is determined using the filtering controls. Figure 5(b3) illustrates this with a filter interval which masks out all images in node a5 while still matching all label images of node a2, as can be seen in the updated cluster tree view 5(c).

Inspecting the object count histogram 5(d) for node a1 shows that while most label images contain four objects, some have a significantly higher object count. Filtering for a high object count, and using the slice view to change the slice axis, it turns out that in some label images noise lead to separate objects. The histograms reveal that these result from using one PCA dimension and a low γ_{erw} value near the threshold defined before. Adjusting the γ_{erw} filter to a threshold of approximately 1 results in the images with high object count being filtered out. This conclusion can be drawn from the scatter plot in 5(e). For T3, we have already identified that γ_{erw} is the parameter with most influence on the suitability of a segmentation in this case. The PCA dimension turns out to have no direct correlation with the noise – there are label images using only one PCA dimension and no noise. When completely filtering out the noise we arrive at a threshold for γ_{erw} of 3.3. Further analysis shows that a minor influence on noisy pixels also came from β_{erw} . For the modality weights, we could not determine a clear correlation to the segmentation quality for this dataset. Very low values of weights were however only sampled in regions of a problematic γ_{erw} value. Re-sampling with suitable γ_{erw} values might provide additional information.

Using this dataset, we also performed a detailed comparison to Paramorama [PBCR11]. For this specific test we employed uniform sampling. This is the only sampling strategy supported by Paramorama, as its interface is tailored to it through their parameter tree, which is splitting the parameter space hierarchically. In contrast to the test datasets provided for Paramorama, we are facing a higher number of parameters for our synthetic dataset: Six for our overall algorithm and three for each of the two modalities. Even when only two different values per parameter are sampled, $2^{12} = 4096$ segmentation masks are generated. When loading this ensemble in Paramorama, even with the relatively large amount of samples there is no chance to determine a specific threshold between suitable and unsuitable results for the γ_{erw} parameter. The only thing to be determined is that the lowest value results in unsuitable masks, while the highest value produces suitable masks.

With our tool and using the latin hypercube sampling strategy, we could easily determine a narrow range for this threshold with only 100 samples.

6.2.3. Hyperspectral Data

We also tested our tool on the hyperspectral *SalinasA* dataset [VMV] (Figure 3c). It can demonstrate the functionality of our tool even for a high number of channels, and when ground truth is available. The main goal here was to verify the robustness of our segmentation pipeline with ground truth data. For that we calculate objective measures (dice coefficient, overall accuracy and kappa coefficient) as additional derived output for each of our sampling result and show it in the histogram view as shown in Figure 6b-d. By filtering for the highest dice metric scores, we can easily navigate to the best segmentation result. The magic lens proves to be very useful in this case by showing the result in comparison to the ground truth, as can be seen in Figure 6a. Our best result achieves an overall accuracy of 0.8 and only slightly lower kappa coefficient of 0.79 in our ensemble, using an approximate 2 % of the image pixels randomly chosen as seed points. As a general rule values over 0.8 are regarded as strong classification results [CG08]. Considering that we are using a general segmentation framework for multi-channel data not finetuned for hyperspectral data, and without even having refined the sampling to the most suitable ranges, this is a very good result. A very recent segmentation framework also based on the ERW and tuned for hyperspectral images by Kang et al. [KLF*15], achieves an overall accuracy of 0.88 and a kappa coefficient of 0.87 applied on a larger part of the same dataset, using 2% of the pixels as training points.

Figure 7 shows a summary of the ranges used for the most influential parameters in all three case studies. In each instance 5-10 positive and negative examples were ranked, the axis shows the resulting coloring. For the SVM channels, we can see that considering a higher number typically causes better results for both *Salinas A* and rock crystal dataset. Considering the mixed coloring at the higher end for the *Salinas A* dataset, we can see that highest values however do not necessarily improve the results. An SVM C somewhere above 0.354 needs to be chosen for the *Salinas A* dataset, where for the rock crystal dataset the best value is clearly closer to the minimum of 0.0078, values around 1 are not well-suited. γ_{svm} values around 10^{-7} produce suitable results for *Salinas A*, while for the rock crystal dataset best results from choosing it nearer to 10^{-12} .

7. Conclusion and Outlook

We presented GEMSe, a tool for the exploration of the parameter- and result space of multi-channel, multi-dimensional image segmentation algorithms. Three tasks were identified in collaboration with domain experts: getting an overview over the space of potential results, analyzing correlations between input parameters and output, and finding stable parameter regions producing suitable label images. Our case studies conducted together with domain experts show the suitability of our methods for these tasks. Our cluster tree view was perceived as very helpful in solving the overview task. While the cluster representative image was shown to provide

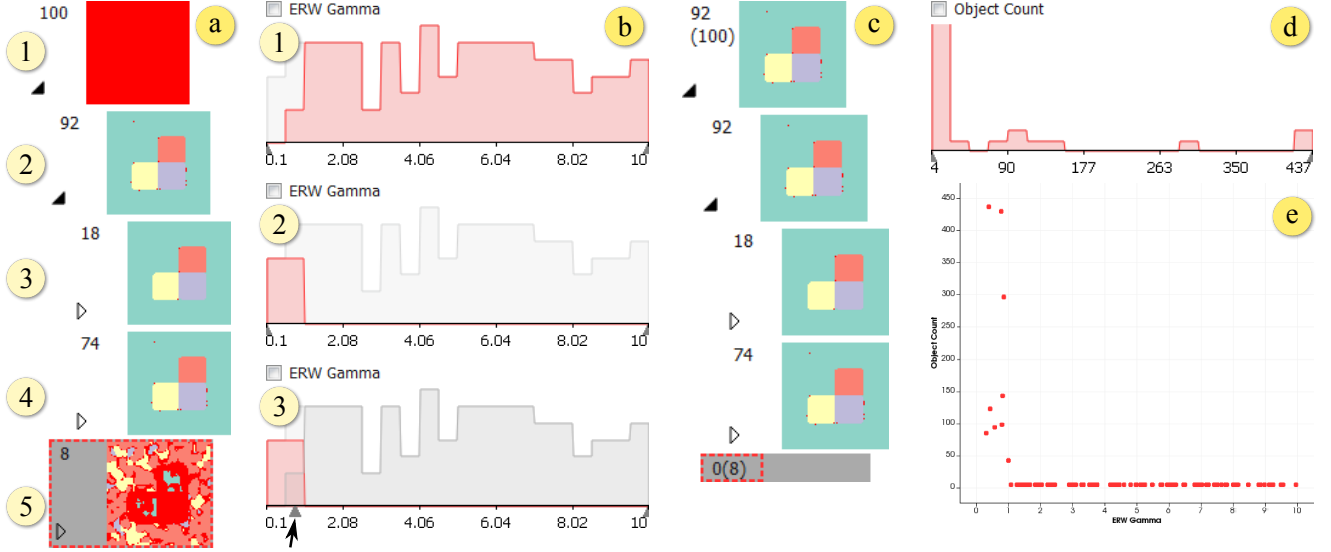


Figure 5: Cluster view (a) of the synthetic K-Edge dataset. Histogram b1 shows the γ_{erw} distribution of node a2, b2 that of a5, b3 demonstrates the filtered histogram, (c) the resulting cluster tree view. Histogram (d) shows the object count for node a1, the scatterplot (e) correlates γ_{erw} (x-axis) to the object count (y-axis).

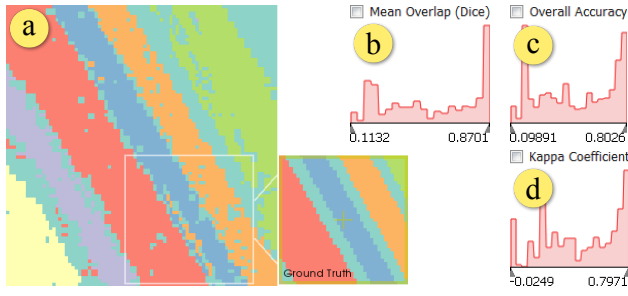


Figure 6: (a) label image with best kappa value, the ground truth is shown in the magic lens for comparison. (b-d) objective measure histograms.

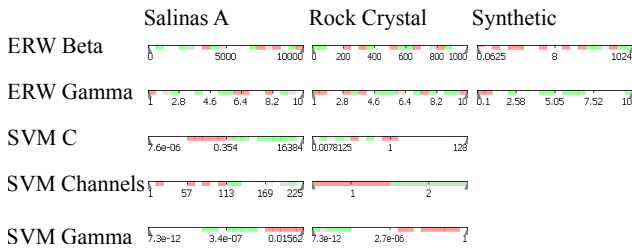


Figure 7: Comparison of ranked parameter ranges for all case studies.

insight into the cluster variation, there is a need for researching further methods of representing a cluster, for example through a scale encoding the number of differences for each voxel. Linked Histograms and filtering have proven to be successful for analyzing the correlations between input parameters and output, as shown in Section 6.2.1 and Section 6.2.2. Finding stable parameter regions, at least for the most influential parameters, was shown to be successful through our rating, axis coloring and filtering interactions. As can be seen in Figure 7 different modalities require different parameter settings. Hence, there is a need for GEMSe in helping practitioners find these settings without the help of segmentation experts. Future work could look further into the multi-channel aspect, for example through integrating image fusion for comparison visualizations, or through evaluating multiple different segmentation algorithms at once. The segmentation pipeline we employed can produce a probabilistic result, another promising area for future work is incorporating information on the segmentation uncertainty, which these probabilistic results give access to. Regarding clustering, one could investigate ways to combine clustering by result similarity as we have applied here, with clustering or splitting by parameter values, or perform dimensionality reduction instead of clustering.

Acknowledgements

We thank our anonymous reviewers for their very helpful comments. The research leading to these results has received funding from the K-Project ZPT+ of the Austrian Research Promotion Agency (FFG).

References

- [AD10] ARENS S., DOMIK G.: A survey of transfer functions suitable for volume rendering. In *IEEE/ EG Symposium on Volume Graphics*

- (2010), Westermann R., Kindlmann G., (Eds.), The Eurographics Association. doi:10.2312/VG/VG10/077-083. 2
- [BM98] BEDERSON B., MEYER J.: Implementing a zooming user interface: Experience building pad++. *Software-Practice & Experience* 28, 10 (Aug. 1998), 1101–1135. doi:10.1002/(SICI)1097-024X(199808)28:10<1101::AID-SPE190>3.0.CO;2-V. 3
- [CCK*14] CHIFFRE L. D., CARMIGNATO S., KRUTH J.-P., SCHMITT R., WECKENMANN A.: Industrial applications of computed tomography. *CIRP Annals - Manufacturing Technology* 63, 2 (2014), 655–677. doi:10.1016/j.cirp.2014.05.011. 1
- [CG08] CONGALTON R. G., GREEN K.: *Assessing the accuracy of remotely sensed data: principles and practices*, 2nd ed. CRC press, 2008. doi:10.1111/j.1477-9730.2010.00574_2.x. 8
- [CL11] CHANG C.-C., LIN C.-J.: LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 3 (May 2011), 27:1–27:27. doi:10.1145/1961189.1961199. 5
- [Dic45] DICE L. R.: Measures of the amount of ecologic association between species. *Ecology* 26, 3 (1945), 297–302. 3
- [FLPZ51] FLOREK K., LUKASZEWICZ J., PERKAL J., ZUBRZYCKI S.: Sur la liaison et la division des points d'un ensemble fini. *Colloquium Mathematicae* 2 (1951). 3
- [FNMA06] FIRSCHING M., NIEDERLOHNER D., MICHEL T., ANTON G.: Quantitative material reconstruction in CT with spectroscopic X-ray pixel detectors – a simulation study. In *Nuclear Science Symposium Conference Record, 2006. IEEE* (Oct. 2006), vol. 4, pp. 2257–2259. doi:10.1109/NSSMIC.2006.354363. 7
- [GFL04] GRADY L., FUNKA-LEA G.: Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. In *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis* (Prague, Czech Republic, May 2004), ECCV 2004 Workshops CVAMIA and MMBIA, Springer, pp. 230–245. doi:10.1007/978-3-540-27816-0_20. 1, 3
- [GJ*10] GUENNEBAUD G., JACOB B., ET AL.: Eigen v3. <http://eigen.tuxfamily.org>, 2010. Online. Retrieved March 10, 2016. 5
- [Gra05] GRADY L.: Multilabel random walker image segmentation using prior models. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (June 2005), vol. 1, pp. 763–770. doi:10.1109/CVPR.2005.239. 2, 3
- [HB03] HARROWER M., BREWER C. A.: ColorBrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal* 40, 1 (2003), 27–37. doi:10.1179/000870403235002042. 3
- [HKRs*06] HADWIGER M., KNISS J. M., REZK-SALAMA C., WEISKOPF D., ENGEL K.: *Real-time Volume Graphics*. A. K. Peters, Ltd., Natick, MA, USA, 2006. 2
- [KAM99] KRISHNAMACHARI S., ABDEL-MOTTALEB M.: Image browsing using hierarchical clustering. In *Proceedings of the IEEE International Symposium on Computers and Communications* (1999), pp. 301–307. doi:10.1109/ISCC.1999.780837. 4
- [KLF*15] KANG X., LI S., FANG L., LI M., BENEDIKTSSON J. A.: Extended random walker-based classification of hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing* 53, 1 (Jan. 2015), 144–153. doi:10.1109/TGRS.2014.2319373. 1, 3, 8
- [MHG10] MALIK M. M., HEINZL C., GRÖLLER E.: Comparative visualization for parameter studies of dataset series. *IEEE Transactions on Visualization and Computer Graphics* 16, 5 (Sept. 2010), 829–840. doi:10.1109/TVCG.2010.20. 4
- [Nie00] NIELSEN J.: Why you only need to test with 5 users. <http://goo.gl/uLD01>, Mar. 2000. 5
- [PBCR11] PRETORIUS A. J., BRAY M.-A., CARPENTER A. E., RUDDE R. A.: Visualization of parameter space for image analysis. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec. 2011), 2402–2411. doi:10.1109/TVCG.2011.253. 2, 8
- [PRH10] PRASSNI J.-S., ROPINSKI T., HINRICHS K.: Uncertainty-aware guided volume segmentation. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 1358–1365. doi:http://doi.ieeecomputersociety.org/10.1109/TVCG.2010.208. 2
- [SHB*14] SEDLMAIR M., HEINZL C., BRUCKNER S., PIRINGER H., MÖLLER T.: Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (Dec. 2014), 2161–2170. doi:10.1109/TVCG.2014.2346321. 2, 3
- [SHM10] SAAD A., HAMARNEH G., MÖLLER T.: Exploration and visualization of segmentation uncertainty using shape and appearance prior information. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Vis 2010)* 16, 6 (10 2010), 1365–1374. doi:10.1109/TVCG.2010.152. 2
- [SMH10] SAAD A., MÖLLER T., HAMARNEH G.: ProbExplorer: Uncertainty-guided exploration and editing of probabilistic medical image segmentation. *Computer Graphics Forum (Proceedings EuroVis 2010)* 29, 3 (6 2010), 1113–1122. doi:10.1111/j.1467-8659.2009.01691.x. 2
- [SML06] SCHROEDER W., MARTIN K., LORENSEN B.: *The Visualization Toolkit*, 4th ed. Kitware, 2006. 5
- [THA10] TOP A., HAMARNEH G., ABUGHARBIEH R.: Spotlight: Automated confidence-based user guidance for increasing efficiency in interactive 3d image segmentation. In *Medical Image Computing and Computer-Assisted Intervention Workshop on Medical Computer Vision (MICCAI MCV)* (2010), pp. 204–213. doi:10.1007/978-3-642-18421-5_20. 2
- [TWSM*11] TORSNEY-WEIR T., SAAD A., MÖLLER T., HEGE H.-C., WEBER B., VERBAVATZ J.-M.: Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 1892–1901. doi:10.1109/TVCG.2011.248. 2
- [VMV] VEGANZONES M. Á., MARQUÉS I., VILCHES B. A.: Hyperspectral remote sensing scenes. http://www.ehu.es/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes. Online. Retrieved March 26, 2015. 8
- [WHA07] WILLETT W., HEER J., AGRAWALA M.: Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov. 2007), 1129–1136. doi:10.1109/TVCG.2007.70589. 4
- [YAL*02] YOO T. S., ACKERMAN M. J., LORENSEN W. E., SCHROEDER W., CHALANA V., AYLWARD S., METAXAS D., WHITAKER R.: Engineering and algorithm design for an image processing api: a technical report on itk - the insight toolkit. *Studies in health technology and informatics* (2002), 586–592. 5