Eurographics Conference on Visualization (EuroVis) 2014
H. Carr, P. Rheingans, and H. Schumann
(Guest Editors)

Volume 33 (2014), Number 3

GuideME: Slice-guided Semiautomatic Multivariate Exploration of Volumes

L. Zhou and C. Hansen

Scientific Computing and Imaging Institute, University of Utah, United States

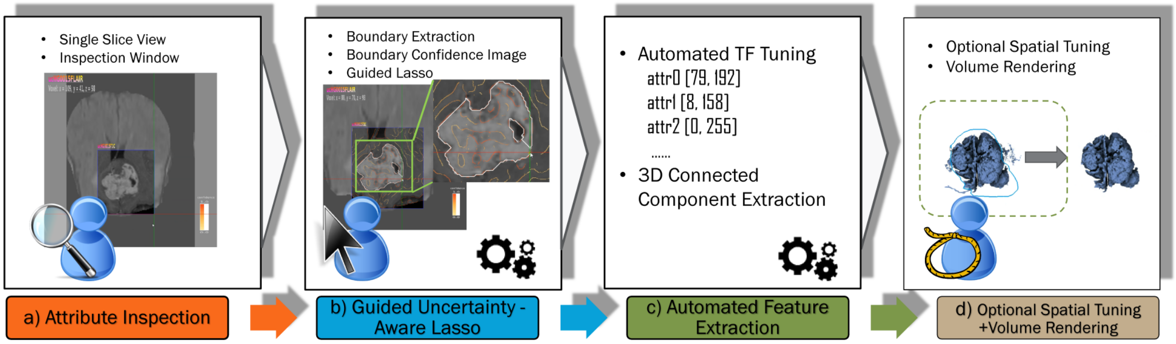

Figure 1: *The workflow of our proposed method which comprises four stages: attribute inspection, guided uncertainty-aware lasso for defining features, feature extraction through automated transfer function tuning and finally spatial fine tuning and visualization. Shown in this figure is the example of extracting the tumor core in a multimodal MR brain scan data.*

Abstract

Multivariate volume visualization is important for many applications including petroleum exploration and medicine. State-of-the-art tools allow users to interactively explore volumes with multiple linked parameter-space views. However, interactions in the parameter space using trial-and-error may be unintuitive and time consuming. Furthermore, switching between different views may be distracting. In this paper, we propose GuideME: a novel slice-guided semiautomatic multivariate volume exploration approach. Specifically, the approach comprises four stages: attribute inspection, guided uncertainty-aware lasso creation, automated feature extraction and optional spatial fine tuning and visualization. Throughout the exploration process, the user does not need to interact with the parameter views at all and examples of complex real-world data demonstrate the usefulness, efficiency and ease-of-use of our method.

Categories and Subject Descriptors (according to ACM CCS): I.4.10 [Image Processing and Computer Vision]: Image Representation—Volumetric I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

The exploration and visualization of multivariate volume has been an active research area for a decade. The state-of-the-art methods for exploring multivariate volumes is user interaction with multiple linked view systems. These methods require the user to explore the volume using parameter views, e.g. parallel coordinate plots (PCP) or histograms, in a trial-and-error manner [AM07, BBP08, GXY11]. Although they have shown successes in simulation datasets where the user understands the "recipe" of the parameter space, i.e. knows what combinations of value ranges of attributes may result in interesting features, it is difficult for the user to explore complex measured datasets, e.g. seismic datasets. To this end, research efforts have addressed the exploration of complex datasets such as seismic data [ZH13, HBG*11, HFG*12]. In [ZH13], the approach allows domain users to apply their expertise into the finding of features by directly select a region of interest in a multi-panel slice view. However, these methods either work only on univariate seismic data for a certain type of features [HBG*11, HFG*12] or still requires transfer function tuning with a PCP based or a histogram based editor [ZH13] which can be unintuitive and time consuming for domain users. Moreover, switching between multiple views may be somewhat distracting.

Seismic datasets are an important tool to the petroleum industry and is the driving application of our method. Geophysicists interpret features which indicate potential oil and gas reservoirs, including channels and salt domes, on 3D seismic data slices. To interpret the seismic data, they first identify and locate geological features on slices from the 3D seismic data through examination and selection. With the advancement of multivariate 3D seismic [CM07] interpretation, attributes derived from the seismic amplitude are used to aid in the extraction of relevant features. Interpretation is done mainly by free-hand drawing on slices and refinement of the features through multi-attribute transfer functions [ZH13, TLM03]. Users typically perform the following tasks during interpretation: selecting features by drawing on 2D slices, refine those features through transfer function manipulation, examination of results in 3D renderings. Other domains use similar tasks and we demonstrate the generality of our approach with multi-modal MR brain scans.

As such, we propose GuideME: a novel method for multivariate volume exploration which strives to provide the user with a very simple and intuitive exploration process for highly complex datasets. Instead of multiple linked views, our method has only one slice view coupled with a focus overlay and a volume rendering view, and the tedious trial-and-error interactions are largely replaced by a guided uncertainty-aware lasso and automated feature extraction. The user starts the exploration by browsing through the slices and detects a feature of interest using his/her domain knowledge. A focus window which allows the user to inspect other attributes can be placed over the feature of interest. Through the inspection, the user determines attributes that best describe the boundaries of the feature. A boundary confidence image is then blended with the slices, and the user can easily select the region with a guided uncertainty-aware lasso which automatically snaps to the detected feature boundaries. The selected region is then used as input for the automated feature extraction. Eventually, the feature is volume rendered and may be optionally edited directly in the 3D view. Using a highly complex real-world seismic dataset and multi-modal MR brain scans, we show our approach is efficient, and is able to create comparable results that given by previous method and ground-truth segmentations.

In this paper, we make the following contributions:

- A novel slice-guided semiautomatic multivariate volume exploration workflow. The user is freed from unfamiliar parameter space views and tedious trial-and-error transfer function tunings.
- A guided uncertainty-aware lasso for region selection, based on edge detection and Dijkstra shortest path algorithms.
- A technique to automatically fine-tune a multivariate transfer function given the lasso, based on the optimization of a response function.

2. Related Work

Multivariate volumes are often visualized using transfer functions in a linked view system. In this section, we give an overview of the relevant techniques.

The most frequently used transfer functions (TFs) are a 1D TF that uses scalar values of the volume or a 2D TF that has the gradient magnitude of the volume as a second property for better classification [KD98]. The TFs can be interactively defined by 1D or 2D parameter space widgets proposed by Kniss *et al.* [KKH02]. However, to design a good TF, the user has to manipulate the TF widgets in a trial-and-error fashion which is laborious and time consuming. To address this issue, researchers have proposed methods to automate the TF design process. Maciejewski *et al.* [MWCE09] utilize kernel density estimation to structure the data parameter space to generate initial TFs. Wang *et al.* [WCZ*11] initialize TFs by modeling the data value space using Gaussian mixture model and render the extracted volume with pre-integrated volume rendering. Alternatively, the system proposed by Guo *et al.* [GMY11] allows the user to directly manipulate on the volume rendering view with intuitive screen space stroking tools similar to 2D photo editing applications. The above methods work well on volumes with one or two attributes, however, extending the ideas to more attributes or higher dimensional TFs is not straight forward.

In recent years, a large amount of work on multivariate volume exploration and visualization has been proposed. The SimVis system [Dol07, PKH04] allows the user to interact with several 2D scatter plot views using linked brushes to select features of interest in particle simulations rendered as polygons and particles. Akiba and Ma [AM07] propose a tri-space exploration technique involving PCPs together with time histograms to help the design of high-dimensional TFs for time-varying multivariate volume datasets. Blaas *et al.* [BBP08] extend parallel coordinates for interactive exploration of large multi-time point datasets rendered as isosurfaces. Zhao and Kaufman [ZK10] combine multidimensional reduction and TF design using parallel coordinates. Guo *et al.* [GXY11] propose an interactive high-dimensional TF design framework using both continuous PCPs and multidimensional scaling technique accelerated by employing an octree structure. Unfortunately, the techniques described above require the user to design high-dimensional TFs in the parameter space which may be unintuitive for domain users and impractical for complex real-world datasets. Zhou and Hansen [ZH13] proposed a hybrid method that allows the user to specify initial TFs from region selected in the object space, fine-tune the TFs in the parameter space and optionally edit the classified volume directly in the object space. However, their work still requires trial-and-error tuning of the TF in the parameter space.

Our work introduces a novel workflow that replaces the manual TF tuning procedure with an automated optimization method, simplifies the user interface to a single slice view and introduces a guided uncertainty-aware lasso for region selection. We strive to make multivariate volume exploration an intuitive experience without burdening the domain user with unfamiliar spaces and views. As a result, we achieve a very simple system which produces comparative results to previous methods. Via a real-world multivariate 3D seismic dataset and multi-modal MR brain scans, we will show the usefulness, efficiency and ease-of-use of our proposed technique.

3. Method Overview

Our approach utilizes automated methods to replace a laborious user workflow. A guided uncertainty aware lasso which snaps to feature boundaries is proposed to assist region selection, automated transfer function tuning is applied to avoid trial-and-error transfer function design, and finally a 3D connected component is automatically extracted. The result of the method is a 3D connected component which best represents the intention of the user. As shown in Figure 1, our approach comprises four conceptual stages: attribute inspection, uncertainty aware lasso drawing, feature extraction based on automated transfer function tuning and volume visualization with optional spatial fine tuning. In the following, we explain attribute inspection in this section, detail the uncertainty aware lasso in Section 4 and the automated feature extraction in Section 5, and briefly describe the volume rendering and spatial fine tuning in Section 6.

During attribute inspection, the user inspects one attribute at a time through a focus window on slices. The focus window serves as a "magic lens" [BSP*93] to overlay the chosen attribute with the contextual background. Then, the user selects one or more attributes that can properly represent the feature boundaries. The selected attribute(s) are then used to generate uncertainty information in terms of boundary confidence as shown in Figure 2 as the color coded curves . With the uncertainty information, the user is able to draw guided uncertainty-aware lassos which snap to feature boundaries via a few mouse clicks as the white curves seen in Figure 2. Next, the feature is extracted using an automated feature extraction approach which minimizes false positives outside the lasso region while preserving true positives inside, and finds the dominant 3D connected component within the lasso region. Finally, further spatial fine tuning can be conducted in the 3D view.

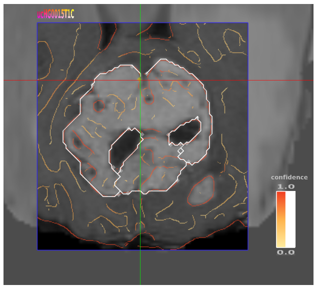

Figure 2: *On an MR brain scan dataset, the inspection window with attribute, T1C, is shown over a tumor region with the FLAIR attribute as background. The boundary confidence derived from T1C, is rendered overlaying the data slice with a color map shown to the right.*

4. Guided Uncertainty-Aware Lasso

In this stage, we first extract feature boundaries using Canny edge detection on an anisotropic diffused image of the data slice. A boundary confidence image describing the uncertainty can then be derived from the feature boundaries of the selected attributes. Next, the system calculates an optimal path between user clicks based on the uncertainty information to create a guided uncertainty-aware lasso. The details of each component will be described in the next subsections.

4.1. Boundary Extraction

The feature boundaries are extracted via edge detection on an anisotropic diffused image of current slice I_a of attribute a. We apply Perona and Malik [PM90] anisotropic diffusion to I_a to remove noise while preserving edges. The flow function $g(\nabla I_a)$ shown below is used, where K is a constant which is empirically set to 30 which gives a good diffusion stopping effect, and the partial differential equation is numerically solved with a small number of iterations.

$$g(\nabla I_a) = e^{-(\frac{||\nabla I_a||}{K})^2}$$

Then, the edges in the filtered image are extracted by Canny edge detection [Can86] which is simple and has good accuracy. The gradient field is first derived, and we then compute the direction of the gradient and classify it into four cases: horizontal, vertical and two diagonals. We remove pixels that are not maximal in the pixel's classified direction in the non-maximal suppression step. Finally, we conduct the hysteresis step via recursive edge tracing. To avoid user involvement in the setup of the lower and upper thresholds, we compute the histogram of the gradient magnitude and accumulate histogram bins until the sum is equal or greater than a certain portion T_{gm} of the count of voxels on the given slice and take the gradient magnitude value of that bin as the upper threshold t_{up} [Can86]. The lower threshold t_{lo} is then computed by multiplying the upper threshold with a constant k_l. We adopt the settings of $T_{gm} = 0.7$ and $k_l = 0.4$ from Matlab and find they work well on all datasets we use.

4.2. Boundary Confidence Image

A boundary confidence image can be derived from the extracted boundary images of user chosen attributes $\bar{A_s}$ from the pop-up menu in the inspection window to indicate the uncertainty. As an uncertainty measurement, the boundary confidence should be in the range $[0,1]$ which is defined by Equation 1.

$$I_b = \begin{cases} 1, & \text{if } ||\nabla I_a|| > t_{up} \\ \max_{\bar{A_s}} \frac{||\nabla I_a|| - t_{lo}}{t_{up} - t_{lo}}, & \text{if } t_{lo} \leq ||\nabla I_a|| < t_{up} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The boundary confidence of each attribute is computed by normalizing the gradient magnitude of the extracted boundaries. The normalization uses the upper and lower thresholds defined in the edge extraction process, and values greater than the upper threshold are mapped to 1. Pixels that are not detected as edges are simply mapped to 0. Next, the boundary confidence value for all selected attributes is calculated by blending individual boundary confidence using the *MAX* operator which keeps the blended value inside the range $[0,1]$. An equal weight is assigned to each attribute so as to avoid having the boundary confidence of one attribute reduces the importance of others, and to remove the requirement of user involvement. A sequential color map scheme suggested by Color Brewer [HB03] is used for the rendering of the boundary confidence image as seen in Figure 2. The color map range and opacity can be interactively modified to remove or highlight certain confidence value ranges.

4.3. Guided Uncertainty-aware Lasso

Given the boundary confidence image I_b with its pixels P, and two user defined end points u and v, an uncertainty-aware lasso which snaps to feature boundaries can be thought of as finding an optimal path that minimizes the transition energy between each pixel as shown in Equation 2.

$$E(I) = \sum_{p \in P} E_s(p) \quad (2)$$

$$E_s(p) = 1 - ||I_b(p)|| \text{ where } p \in P$$

The energy in Equation 2 can be efficiently optimized using the Dijkstra's algorithm [Dij59] from end point u to v.

To compute the optimal path using Dijkstra's algorithm, we convert image I_b into a bidirected graph where each pixel p is assigned a node and the edge from p to its neighboring pixel p_n has energy $E_s(p_n)$ as weight. Thanks to the efficiency of Dijkstra's algorithm, the user is able to interactively set the end points u, v by clicking on the boundary confidence image inside the inspection window to setup, and edit the end points by a click and drag interaction.

5. Automated Feature Extraction

In this stage, we extract the feature based on the lasso region via an automated feature extraction procedure. An initial transfer function is generated and tuned using our novel automated transfer function tuning method. The resulting transfer function gives minimum false positives outside the lasso region while preserving true positives inside the lasso. Then, the dominant 3D connected component in the classified volume is extracted.

5.1. Automated Transfer Function Tuning

The core of our feature extraction approach lies in automated transfer function tuning. By watching the domain experts manually fine-tuning the transfer functions using existing tools, we observed that they focus only on the lassoed region and tried to minimize false positives outside the lasso while preserving true positives inside the lasso. Therefore, we mimic this procedure by formulating an optimization problem. For a multivariate volume of M attributes: $\bar{A} = (A_1, A_2, \ldots, A_M)$, we model the M-dimensional transfer function space by conducting `AND` operation between the M 1D spaces. This avoids erroneous classification caused by a separable M-D transfer function composed of M 1D transfer functions multiplied together as shown on page 258 in [HKRs*06]. We use only binary values 0 and 1 to indicate the selection of attribute values, and denote such a binary transfer function as f and its i-th 1D subspace f^i. An initial transfer function can be setup, and then optimized by maximizing a response function.

5.1.1. Initial Transfer Function Setup

Given the lasso region, an initial transfer function, f_0, can be created by querying the attribute values of the pixels inside the lasso. We conduct the query by a simple traversal over the slice and test if the current pixel on the slice falls inside the lasso. If true, we record the pixel's M-queried results into the corresponding locations of the histogram array H. Where H is a 1D histogram array of M-layers, and each layer H_i is a 1D histogram associated with an attribute A_i. Then, an initial binary transfer function f_0 is generated by setting non-zero histogram locations to one.

5.1.2. Response Function Formulation

We formulate a response function $R(I_s, I_c)$ of two binary images: the user lasso image I_s and a connected component image I_c of the transfer function classified image I_f. Since we focus on the lasso region only, we take the dominant connected component of the classified image inside the

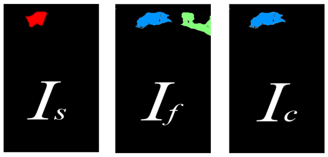

Figure 3: *Left, shows the lasso image I_s where the red indicates the lasso region. The transfer function classified image I_f is seen in the center where the blue and green regions are classified by the transfer function. Right, sees the connected component image I_c where blue is the dominant connected component.*

lasso. Specifically, we extract all connected components in the classified image and create a histogram of tag values inside the lasso. Then we keep only the connected component with the most frequent tag in this histogram and discard other connected components. As shown in Figure 3, the relationship between I_s, I_c and I_f is clearly demonstrated. For a multivariate transfer function f, a slice of M-attribute volume with pixels p, I_f can be denoted as:

$$I_f = \{p | f(\vec{v_p}) > 0\} \tag{3}$$

where $\vec{v_p} = (v_p^1, v_p^2, \dots v_p^M)$ is the multivariate value of pixel p. In practice, whenever the transfer function changes, we update I_f and extract the dominant connected component.

The response function $R(I_s, I_c)$ can then be written as:

$$\begin{aligned} R(I_s, I_c) &= w_r \cdot r(I_s, I_c) + (1 - w_r) \cdot s(I_s, I_c) \\ \text{subject to } & N_c \geq N_{min} \\ \text{where } & w_r \in [0, 1] \end{aligned} \tag{4}$$

where $r(I_s, I_c)$ is the cross-correlation coefficient of images I_s and I_c, $s(I_s, I_c)$ is a smoothness term, w_r is a tunable weight which is empirically set to 0.7 by default, and the non-zero pixel count N_c of image I_c has to be greater or equal to N_{min} which we empirically set to be 90% of the non-zero pixel count of the lasso image I_s. The cross-correlation coefficient is computed by treating the images as arrays of binary pixels as seen in Equation 5.

$$r(I_s, I_c) = \frac{\sigma_{I_s I_c}}{\sigma_{I_s} \sigma_{I_c}} = \frac{\sum_{i=1}^{N} (I_{s_i} - \bar{I}_s)(I_{c_i} - \bar{I}_c)}{\sqrt{\sum_{i=1}^{N} (I_{s_i} - \bar{I}_s)^2} \sqrt{\sum_{i=1}^{N} (I_{c_i} - \bar{I}_c)^2}} \tag{5}$$

The smoothness term $s(I_s, I_c)$ measures the normalized differences of non-zero pixels p_c and the neighborhood p_{cn} of I_c inside the lasso, in our case eight neighbors n, in the classified region inside the lasso:

$$s(I_s, I_c) = -\frac{\sum_{P_s} \sum_n (p_c - p_{cn})}{s_{max}} \tag{6}$$

where P_s are non-boundary pixels of I_s and s_{max} denotes the maximum possible differences inside I_s. Specifically, we derive s_{max} by considering the extreme case that all non-zero pixels are surrounded by zero pixels which gives $s_{max} = 8 \times \frac{1}{1+(4\times 0.25+4\times 0.5)} \times |P_s| = 2 \times |P_s|$ as the corner pixels are shared by four neighboring stencils while the middle pixels on each side are shared by two stencils. Maximizing the response function R encourages higher correlation between the classified region and the lasso, while penalizing the elimination of true positives inside the lasso. As a result, maximizing Equation 4 minimizes false positives outside the lasso while preserving true positives inside the lasso.

5.1.3. Transfer Function Optimization

Since there is no direct link between the transfer function and R, Equation 4 is hard to optimize using methods like gradient descent or conjugate gradient. As such, we propose the following greedy algorithm to approximately maximize $R(I_s, I_c)$. As seen in Algorithm 1, we first determine the order for optimizing the 1D subspace of individual attributes of the transfer function. This step is necessary because this ordering affects the final result. We assume that an attribute which has higher R than others is likely to require fewer changes for optimization than others, and this is confirmed by experiments on the datasets we used. Therefore, we use a conservative heuristic which optimizes the 1D subspaces from more contributing ones (higher R) to less contributing ones (lower R) for the feature of interest. The reason is twofold: first, this heuristic may lead to minimal iterations of optimization. Second, if we start with less contributing attributes, it is likely to overly eliminate true positives inside the lasso and other attributes may never have the chance to remedy such an error. We first get the binary images classified by the initial transfer function of individual attribute f_0^i for all attributes $\bar{A}$. The response function value R is evaluated for each binary image, and then we sort the attributes by R. Next, we select the attribute that has the highest response function value, and maximize the response function R by optimizing the transfer function's j-th subspace.

Algorithm 1 TF_Opt(I_s, f_0)

```
for Attribute $A_i$ in all $M$ attributes do
    Generate $I_c^i$ with $f_0^i$
end for
Sort all attributes $\bar{A}$ with descending order of $R(I_s, I_c^i)$
$f = f_0$
for Attributes $A_j$ in sorted $\bar{A}$ do
    ModifyTF($f^j, H_j$)
end for
```

To optimize the individual subspace of the transfer function, we propose a simple yet efficient transfer function bin dropping approach as seen in Algorithm 2. In Figure 4, the steps of the bin dropping method are clearly illustrated on the top. The method starts with the computation of the mean value μ of the associated queried histogram of the given attribute. Then, the farther end of the attribute to μ is chosen as

Algorithm 2 ModifyTF(f^i, H_i)

```
Compute $\mu$ of $H_i$
Select direction $d$ whose bin is farther to $\mu$
for true do
    if $R^{n+1}(I_s, I_c) > R^n(I_s, I_c)$ then
        Drop bin from $f^i$ in direction $d$
    else
        if Both directions have been tried then
            break
        else
            Switch direction $d$
        end if
    end if
end for
```

direction d as it is likely to contain more false positives. The algorithm finds the optimal point that maximizes R by dropping bins from the transfer function in the direction d, and then performs the same operations on the other direction until converges. The effect of the automated transfer function tuning process is shown in the bottom of Figure 4.

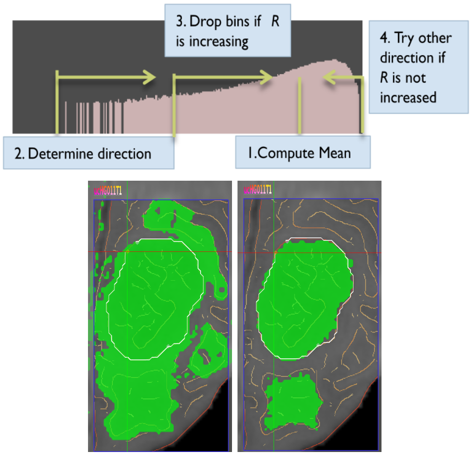
3. Drop bins if R is increasing
4. Try other direction if R is not increased
2. Determine direction
1.Compute Mean

Figure 4: *On the top, shows the steps involved in transfer function modification by bin dropping. In the bottom left is the initial transfer function classified result (green) using the queried values from the lasso (white) on the MR brain scan HG11. To the right, the optimized transfer function classification result.*

5.2. 3D Connected Component Extraction

Transfer function does not contain any spatial information, and as such, even with an optimized transfer function may contains false positive in 3D. Therefore, the last step of feature extraction is to apply connected component finding to extract the intended feature in the transfer function classified volume. We first extract all connected components in the classified volume, and then, query tag values inside the user lasso on the slice. The connected component whose tag value is most frequent is then selected. Next, the selected connected component is given a color and opacity and other components are discarded.

6. Volume Rendering and Spatial Fine Tuning

Once the 3D connected component has been extracted from the automated feature extraction stage, the classified result is stored as a tag volume and visualized using volume rendering. We adopt the directional occlusion shading method from Schott *et al.* [SPH*09] which provides better depth cues than local shading models as demonstrated by [LR11] and has shown to provide more insights into seismic datasets [PBVG10]. To provide smooth tag volume rendering, we utilize a simplified version of [HBH03] which can be efficiently computed in the GPU shader.

In some cases, even the user intended connected component contains false positives. As such, we provide the user a simple yet flexible means of spatial fine tuning: volume rendered image space lasso. This image space lasso allows the user to select voxels inside the back projected volume of the volume rendered image. Two modes are provided: the keep mode keeps voxels inside the lasso while removing others, and the remove mode does just the opposite which is similar to [WOCH12].

7. Implementation

Our proposed method has been implemented in C++, with OpenGL and CUDA for rendering and computation. The user interface has been implemented using Qt. Most image processing procedures and value querying tasks, slice rendering and volume rendering have been implemented on the GPU using GLSL shaders with the `GL_EXT_shader_image_load_store` extension. The rendering of lassos and texts are accelerated with NVidia's `NV_path_rendering` SDK [NVI]. Correlation coefficient computation is implemented with thrust CUDA library [HB]. Graph creation from the slice and Dijkstra's algorithm are implemented on the CPU. Efficient connected component extraction is realized with CONNEXE library [Mal].

8. Examples

To demonstrate the usefulness and efficiency of our method, we apply it to complex multivariate datasets in two different disciplines: a multivariate seismic data in the petroleum industry and multi-modal brain scans from the 2013 Medical Image Computing and Computer Assisted Intervention (MICCAI) Conference challenge. To validate our method, we compare the method against previously extracted features by domain experts for the seismic example and hand-segmented ground truths for the MR brain example.

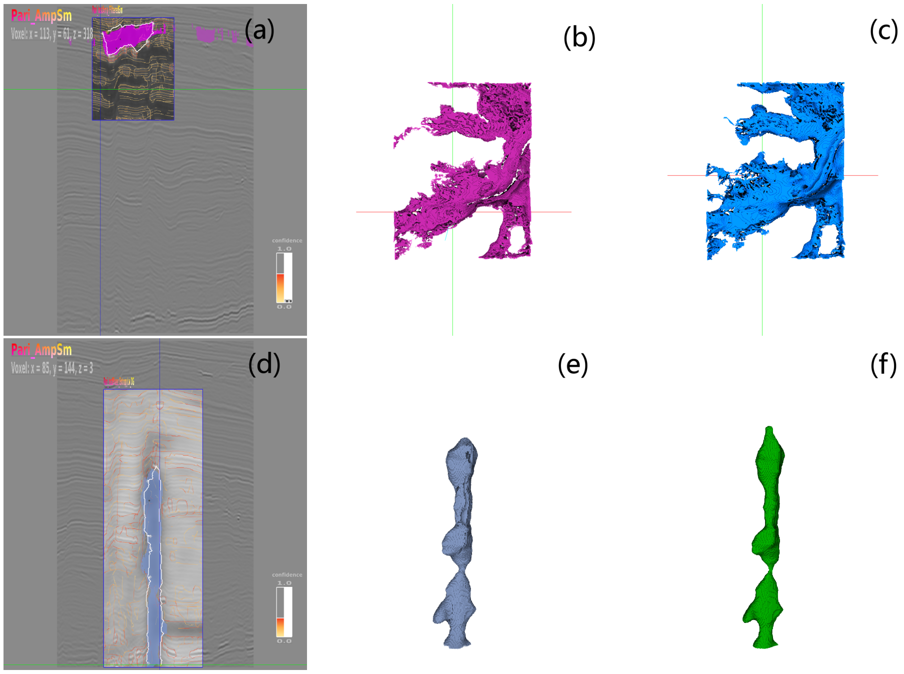

Figure 5: *The first row shows the upper channel in the New Zealand seismic data. Shown in subfigure (a) is the lasso which extracts the feature, (b) is the result using GuideME, and (c) is the result generated by a domain expert using [ZH13]. The salt dome is shown to the bottom. In subfigure (d) shows the lasso region drawn for feature extraction, in (e), shows the result using GuideME, and in (f), the result extracted by the domain expert.*

8.1. Seismic Dataset

The seismic dataset we used is a part of the public New Zealand seismic data. Six attributes have been computed from the original seismic amplitude: namely instantaneous amplitude `InstAmp`, instantaneous phase `InstPhase`, entropy of instantaneous phase `InstPhase_Entropy`, horizon layers `Layer_Seg`, semblance `Semb` and thickness of semblance `Semb_Thick`. The user starts the exploration on slices in 'inline' direction which in our case is the slices on the `'YZ'` plane. A potential channel structure draws the user's attention, and the user zooms in and places the inspection window over this feature of interest. As seen in Figure 5(a), after inspecting different attributes, it is decided that `InstAmp` best represents the boundary of this channel structure. Next, the user creates an uncertainty-aware lasso by placing several anchor points on the feature's boundary with relatively high boundary confidence and fine-tunes it by dragging the anchor points. The feature is then extracted as shown in Figure 5(b). Compared to Figure 5(c) where the feature is extracted by our collaborating geophysicists using [ZH13], our automated method provides similar result which captures the connected body and its meander details. However, the proposed method greatly reduces the time to achieve such a result. With the same feature lassoed, our method takes around a second as shown in Section 8.3 to extract the feature while pure interactive tuning takes minutes. Moreover, the lasso drawing process is guided which may also be faster than free hand drawing. Next, we would like to extract the salt dome structure found near the center of the volume. Again, the user utilizes the inspection window to examine attributes that emphasis this feature, and finds that in addition to the `InstAmp` attribute, the `InstPhase_Entropy` attribute best illustrates the boundary. Then, a lasso is drawn with boundary confidence information calculated from these two attributes as shown in Figure 5(d). The result as seen in Figure 5(e) is comparable to the one from domain expert interactions as seen in Figure 5(f).

To make a quantitative comparison, we compute the dice score, i.e. twice the number of overlapping voxels from two data sets divided by the sum of all voxels from the two data, for our proposed approach against the results conducted by the domain expert. The dice score for the upper channel is 0.84 while the score for the salt dome is also both 0.84 as shown in Table 1. Both cases demonstrate that our method is able to extract features that are similar to interactively extracted and fine-tuned features generated by domain experts, but is faster and easier. Furthermore, the entire user interaction in our method happen on slices and the 3D view which may be more familiar and intuitive to domain users.

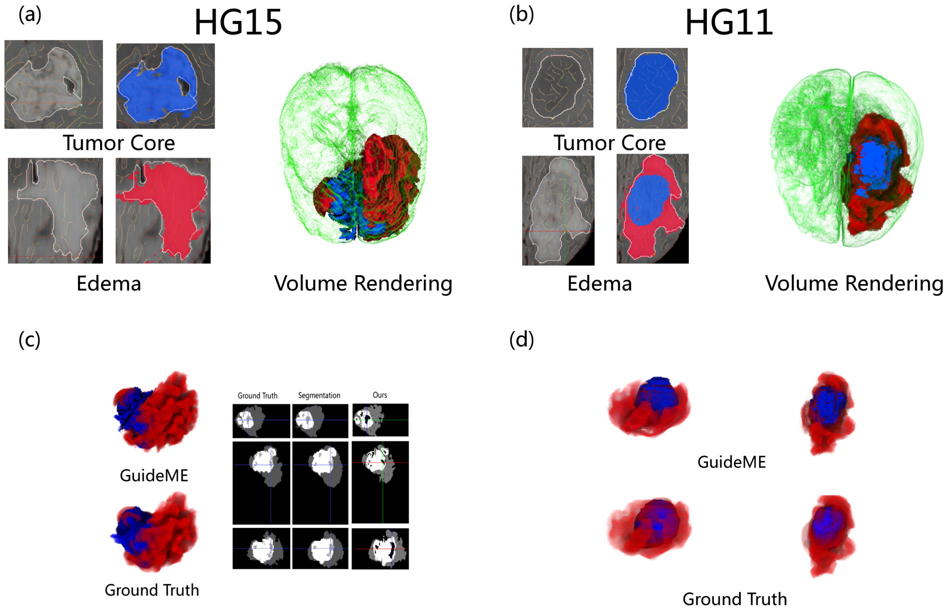

Figure 6: *GuideME is applied to multi-modal brain scans of four modals from the BRATS 2013 challenge. Subfigures (a) and (b) show the visualization result and the lassos on the slice for HG15 and HG11. The extracted active tumors are rendered in blue, the edemas in red in the context of brain tissue in green. In (c) and (d), we compare the volume rendering of the tumor and the edema of the GuideME result against the ground truth. In (c) we include a slice comparison against the ground truth as well as the method proposed in [CMDA13].*

Dataset	Size	#Attr	Feature	Dice	Timing
	x 274	7	Upper Channel	0.84	1.231s (741+490ms)
Seismic	y 426		Salt Dome	0.84	1.373s (888+485ms)
	z 245				
	x 161	4	Tumor Core	0.85	0.833s (411+422ms)
MR HG15	y 216		Edema	0.81	0.817s (402+415ms)
	z 177		Both	0.87	—
	x 161	4	Tumor Core	0.84	0.730s (350+380ms)
MR HG11	y 216		Edema	0.76	0.815s (405+410ms)
	z 177		Both	0.87	—

Table 1: *Quantitative comparisons and timing results for the features extracted in the example datasets. For the timings, the first numbers in the parenthesis are the automated transfer function tuning time while the second numbers are the 3D connected component finding time.*

8.2. Brain Scan

To demonstrate the generality of GuideME, brain tumor image data from the NCI-MICCAI 2013 Challenge on Multimodal Brain Tumor Segmentation [FRM*] (BRaTS) were used. The data consists of multi-contrast MR scans of 30 glioma patients with expert annotations for the tumor core and the edema as ground truths. The datasets in the challenge all contain four channels: FLAIR, T1, post-Gadolinium T1 (T1C) and T2. We chose high-grade subject HG15 and HG11 which the methods in the proceedings of the Challenge gave good agreement with the ground truths.

We describe detailed operations to extract the tumor in HG15. Setting the FLAIR as the context attribute, and browsing the slices on XY direction, a large tumor region is observed. First, we extract the tumor core. Visualizing different attributes inside the inspection window, it is apparent that the T1C attribute is the best candidate for boundary confidence for the tumor core. A lasso is then drawn around the tumor to extract it, and a volume rendered image space lasso is used to fine tune it. The extracted tumor core is seen in blue in Figure 6(a). Next, we extract the edema. Checking with different attributes inside the inspection window, the FLAIR attribute best describes the edema. Clicking along the edema boundary, the feature is then extracted and fine-tuned using the volume rendered image space lasso. The final classification of the edema is shown in red as seen in Figure 6(a). To validate the result, we compare our classification against the ground truth segmentation using the dice score. The dice score for the tumor and the edema together is: 0.87, and respectively 0.85, and 0.81 as seen in Table 1. The result is also compared to a method [CMDA13] proposed in the BraTS challenge as seen in Figure 6(c), in which the method gives slightly above 0.91 dice score for the whole tumor region, and around 0.90 for the tumor while around 0.86 for the edema.

Similarly, we extract the tumor core and the edema for HG11 as seen in Figure 6(b), and a comparison can be seen in Figure 6(d). The resulting dice score for the core and the edema is: 0.87, while the core has dice score 0.84 and the edema has 0.76 as shown in Table 1. In comparison, the scores for the method in [CMDA13] are: just above 0.90 for the whole tumor, around 0.82 for the core and 0.70 for the edema.

8.3. Performance

All the performance timings are conducted on a workstation with a single Intel Core i5 3.30GHz CPU, 16GB of main memory and an Nvidia GeForce GTX 480 with 1.5GB memory running 64-bit Windows 7 system. The creation of the boundary confidence image is around 200ms, and the update of guided uncertainty-aware lasso typically takes below 20 milliseconds. The timing results of features in both examples can be found in Table 1. For the seismic dataset, the classification of the upper channel takes 714ms for 43 response function iterations, while it takes 55 iterations (888ms) for the salt dome. For the brain scan datasets, both subjects have the same size and thus show similar timings. Specifically for the subject HG15, the time for the automated transfer function tuning for the tumor takes 411ms with a total number of 53 iterations. While for the edema region, the timing is 402ms for 49 iterations.

8.4. Discussion

Although we have not conducted a formal user study, our collaborating experts from the petroleum industry found GuideME an improvement over previous tools and provided informal comments. As in their traditional workflow, the datasets are examined and analyzed using slices. Since they have the expertise to identify a certain feature on slices, interactively selecting feature boundaries is not an imposition to them. As they are familiar with free-hand drawing on seismic slices, selecting an appropriate slice and view angle is naturally part of their workflow. While selecting features from multi-attribute slices is interactive and thus done through trial and error, they have the expertise to identify a feature on slices. A previous method [ZH13] required the user to use a free-hand lasso tool to select features of interest on slices. The free-hand lasso was cumbersome to use. GuideME guides the user through the uncertainty-aware lasso interactions where boundaries can be more rapidly and concisely defined by the drawing interaction. The domain experts commented that the GuideME system is faster and easier to use than previous tools. The experts also complained about the trial-and-error transfer function tuning in [ZH13]. Having the automated transfer function tuning freed the experts from this tedious step. Given that the extracted features are comparable to the previous method as shown in 8.1, the domain experts found that with our proposed method, they can be more focused on their geological interpretation tasks.

Our approach is not without limitations. Our proposed method is not an automated feature extraction method for multivariate volume datasets. It requires the user to identify features of interest on slices and select them using the uncertainty-aware lasso. The selection of feature boundaries is an interactive process which requires the user's expertise and understanding of the data. The lasso region chosen for features is critical to the final visualization result. Furthermore, our method extracts features that are connected in the 3D data. As such, our method would not work on datasets in where features are not distinguishable on 2D attribute slices or where features are not connected in the 3D space. While the user has to browse through the slices to detect features of interest and the lasso region drawn for features is critical to the final visualization result, we argue that this is where the expertise of the user applies, and is the main focus of our method.

9. Conclusion and Future Work

We have introduced GuideME: a novel slice-guided semiautomatic multivariate volume exploration method. The user explores the volume on slices, inspects different attributes via an inspection window and draws guided uncertainty-aware lassos on feature of interest, and then the features are extracted through an automated feature extraction approach. More specifically, a boundary confidence measurement which is derived from edge detection provides the user with hints and the uncertainty of feature boundaries. A guided uncertainty-aware lasso that snaps to the feature boundary facilitates region selection. An automated feature extraction method minimizes false positives outside the lasso while preserving true positives inside the lasso. Our experiments have shown that GuideME gives comparable results to those generated by previous methods and expert segmentations, but is more efficient and easier in terms of interaction. We believe that the presented method is a first step towards a visual analytic environment in which the domain experts can focus on spaces and views of their expertise without worrying about unfamiliar and tedious interactions.

In the future, we would like to investigate how to provide the user more guidance towards a feature using advanced image processing and machine learning techniques. Also, we plan to provide the user quantitative visual analysis tools to better understand the features in the dataset.

Acknowledgments

This research was sponsored by the DOE SciDAC Institute of Scalable Data Management Analysis and Visualization DOE DE-SC0007446, NSF ACI-1339881, NSF IIS-1162013.

References

[AM07] AKIBA H., MA K.-L.: A tri-space visualization interface for analyzing time-varying multivariate volume data. In *Proceedings of Eurographics/IEEE VGTC Symposium on Visualization* (May 2007), pp. 115–122. 1, 2

[BBP08] BLAAS J., BOTHA C., POST F.: Extensions of parallel coordinates for interactive exploration of large multi-timepoint data sets. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (Nov.-Dec. 2008), 1436 –1451. 1, 2

[BSP*93] BIER E. A., STONE M. C., PIER K., BUXTON W., DEROSE T. D.: Toolglass and magic lenses: The see-through interface. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (1993), SIGGRAPH '93, pp. 73–80. 3

[Can86] CANNY J.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 8*, 6 (Nov 1986), 679–698. 3, 4

[CM07] CHOPRA S., MARFURT K. J.: *Seismic Attributes for Prospect ID and Reservoir Characterization (Geophysical Developments No. 11)*. Society Of Exploration Geophysicists, 2007. 2

[CMDA13] CORDIER N., MENZE B., DELINGETTE H., AYACHE N.: Patch-based segmentation of brain tissues. In *Proceedings of NCI-MICCAI BRATS 2013* (2013), pp. 6–17. 8, 9

[Dij59] DIJKSTRA E. W.: A note on two problems in connexion with graphs. *Numerische Mathematik 1* (1959), 269–271. 4

[Dol07] DOLEISCH H.: Simvis: interactive visual analysis of large and time-dependent 3d simulation data. In *Proceedings of the 39th conference on Winter simulation* (2007), WSC '07, pp. 712–720. 2

[FRM*] FARAHANI F., REYES M., MENZE B., GERSTNER E., KIRBY J., KALPATHY-CRAMER J.: NCI-MICCAI 2013 Challenge on Multimodal Brain Tumor Segmentation (BRaTS). The challenge database contains fully anonymized images from the following institutions: ETH Zurich, University of Bern, University of Debrecen, and University of Utah and publicly available images from the Cancer Imaging Archive (TCIA). 8

[GMY11] GUO H., MAO N., YUAN X.: Wysiwyg (what you see is what you get) volume visualization. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Dec. 2011), 2106–2114. 2

[GXY11] GUO H., XIAO H., YUAN X.: Multi-dimensional transfer function design based on flexible dimension projection embedded in parallel coordinates. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE* (2011), pp. 19–26. 1, 2

[HB] HOBEROCK J., BELL N.: Thrust: a c++ template library of parallel algorithms. 6

[HB03] HARROWER M., BREWER C. A.: Colorbrewer.org: An online tool for selecting colour schemes for maps. *Cartographic Journal 40*, 1 (Jun 2003), 27–37. 4

[HBG*11] HÖLLT T., BEYER J., GSCHWANTNER F., MUIGG P., DOLEISCH H., HEINEMANN G., HADWIGER M.: Interactive seismic interpretation with piecewise global energy minimization. In *Pacific Visualization Symposium (PacificVis), IEEE 2011* (March 2011), pp. 59–66. 2

[HBH03] HADWIGER M., BERGER C., HAUSER H.: High-quality two-level volume rendering of segmented data sets on consumer graphics hardware. In *Visualization, 2003.IEEE* (Oct. 2003), pp. 301 –308. 6

[HFG*12] HÖLLT T., FREILER W., GSCHWANTNER F., DOLEISCH H., HEINEMANN G., HADWIGER M.: SeiVis: An interactive visual subsurface modeling application. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (Dec. 2012), 2226–2235. 2

[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. 4

[KD98] KINDLMANN G., DURKIN J.: Semi-automatic generation of transfer functions for direct volume rendering. In *IEEE Symposium on Volume Visualization* (1998), pp. 79–86. 2

[KKH02] KNISS J., KINDLMANN G., HANSEN C.: Multidimensional transfer functions for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics 8*, 3 (2002), 270–285. 2

[LR11] LINDEMANN F., ROPINSKI T.: About the influence of illumination models on image comprehension in direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Dec. 2011), 1922 –1931. 6

[Mal] MALANDAIN G.: Connexe: connected component extraction. 6

[MWCE09] MACIEJEWSKI R., WOO I., CHEN W., EBERT D.: Structuring feature space: A non-parametric method for volumetric transfer function generation. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (Nov. 2009), 1473–1480. 2

[NVI] NVIDIA C.: Nv path rendering. 6

[PBVG10] PATEL D., BRUCKNER S., VIOLA I., GROLLER E.: Seismic volume visualization for horizon extraction. In *Pacific Visualization Symposium (PacificVis), 2010 IEEE* (March 2010), pp. 73 –80. 6

[PKH04] PIRINGER H., KOSARA R., HAUSER H.: Interactive focus+context visualization with linked 2d/3d scatterplots. In *Coordinated and Multiple Views in Exploratory Visualization, 2004. Proceedings. Second International Conference on* (July 2004), pp. 49 – 60. 2

[PM90] PERONA P., MALIK J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 7 (1990), 1364–1371. 3

[SPH*09] SCHOTT M., PEGORARO V., HANSEN C. D., BOULANGER K., BOUATOUCH K.: A directional occlusion shading model for interactive direct volume rendering. *Computer Graphics Forum 28*, 3 (2009), 855–862. 6

[TLM03] TZENG F.-Y., LUM E., MA K.-L.: A novel interface for higher-dimensional classification of volume data. In *Visualization, 2003. IEEE* (Oct 2003), pp. 505–512. 2

[WCZ*11] WANG Y., CHEN W., ZHANG J., DONG T., SHAN G., CHI X.: Efficient volume exploration using the gaussian mixture model. *IEEE Transactions on Visualization and Computer Graphics 17*, 11 (Nov. 2011), 1560–1573. 2

[WOCH12] WAN Y., OTSUNA H., CHIEN C.-B., HANSEN C.: Interactive extraction of neural structures with user-guided morphological diffusion. In *Proceedings of the IEEE Symposium on Biological Data Visualization* (2012), pp. 1–8. 6

[ZH13] ZHOU L., HANSEN C.: Transfer function design based on user selected samples for intuitive multivariate volume exploration. In *Pacific Visualization Symposium (PacificVis), 2013 IEEE* (March 2013), pp. 73–80. 2, 3, 7, 9

[ZK10] ZHAO X., KAUFMAN A.: Multi-dimensional reduction and transfer function design using parallel coordinates. In *Volume Graphics 2010, IEEE/EG International Symposium on* (May. 2010), pp. 69–76. 2