Interactive visualization of multi-field medical data using linked physical and feature-space views

Jorik Blaas, Charl P. Botha and Frits H. Post

Data Visualization Group, Delft University of Technology

Abstract

Multi-field datasets contain multiple parameters defined over the same spatio-temporal domain. In medicine, such multi-field data is being used more often every day, and there is an urgent need for exploratory visualization approaches that are able to deal effectively with the data-analysis. In this paper, we present a highly interactive, coordinated view-based visualization approach that has been developed especially for dealing with multi-field medical data. It can show any number of views of the physical domain and also of the abstract high-dimensional feature space. The approach has been optimized for interactive use with very large datasets. It is based on intuitive interaction techniques, and integrates analysis techniques from pattern classification to guide the exploration process. We will give some details about the implementation, and we demonstrate the utility of our approach with two real medical use cases.

1. Introduction

Multi-field datasets contain multiple variables defined over the same physical domain. As this type of data is being used more often every day in clinical research and practice, it is becoming increasingly important to have techniques with which multi-field data can be effectively explored and visualized. Some examples of medical multi-field data are:

Multi-modality data PET-CT is a multi-modality type of dataset that combines CT (computerized tomography) with PET (Positron Emission Tomography) data. The CT data supplies anatomical information as context, while the PET yields functional data, for example on the precise location of tumors and their metastases.

Registered data Different datasets can be registered in order to extract more information from the combination. This can be done for different modality datasets of the same patient, datasets acquired at different time points or datasets of different patients.

DTI Diffusion Tensor Data is inherently multi-field data in that it contains at least one symmetric second-order tensor per data point. In the basic single-tensor model, there are six unique tensor elements per data point.

Data with derived features This is an important class of multi-field medical data where extra features have been calculated for each data point. For example, based on a CT dataset, the gradient or curvature volumes can be calculated. The purpose of these extra features is often to be used in subsequent segmentation, quantification or visualization steps.

Visualization and image processing techniques could be used by clinicians to make effective use of multi-field data for diagnosis, therapy planning and surgical guidance. However, before this stage can be reached with a particular type of multi-field data and application, visualization and image processing scientists have to develop and refine the necessary techniques for routine clinical use. This is a challenging task, as the necessary exploration of multi-field medical data is complex and there is very little supporting infrastructure. In fact, there is currently no unified visualization approach that satisfactorily addresses this problem.

The approach that we present in this paper was developed for these scientists and has the main goal of facilitating the crucial exploratory stage of the clinical algorithm development pipeline. Our approach facilitates numerous often-occurring tasks during the exploration of multi-field datasets, for example:

- Investigating combinations of features for finding spatial separations of point sets in physical space (segmentation)
- Classification of feature point sets (pattern classification)
- Finding clusters, patterns, and relations in the data

- Selective visualization based on feature value ranges
- Investigation of possible features and their suitability for segmentation or as pattern classifiers

Our approach is based on the use of interactive dynamic coordinated views. This is based on related work concerning multiple coupled views and multi-field (non-medical) data. Our solution is set apart by three major factors. Firstly, it has been optimized for highly interactive use with specifically multi-field medical datasets. In contrast to problems in information visualization, multi-field medical datasets might have a smaller number of features or variables per data point, but can be very large. The inherent physical component of medical data, i.e. the fact that it is continuously defined over a spatial domain, requires a different approach. Secondly, our approach supports arbitrary coordinated projections of the high-dimensional multi-field feature spaces. Existing solutions support selecting discrete axes or features from feature space to act as the spanning vectors of two-dimensional scatter plots. Through an intuitive interaction technique, our framework supports *arbitrary* projections of feature space. In other words, each axis of the resultant coordinated scatterplot can represent a user-defined combination of features. Finally, we have integrated a number of analysis techniques that are commonly used in pattern recognition, such as clustering and transformation algorithms, linear discriminant analysis, and principal component analysis. These techniques can be applied interactively to result in a more effective exploration.

The rest of this article is structured as follows. We discuss related work in Section 2, and in Section 3 our approach is explained in detail. Section 4 gives design considerations for implementation, and some performance figures. In Section 5, two real medical applications are described, demonstrating the exploration process. We present conclusions and future directions in Section 6.

2. Related work

XmdvTool [War94] combines a number of techniques for the visualization of multi-variate data. These techniques include scatter plots, glyphs, parallel coordinates, hierarchical techniques, and importantly, linked N-dimensional brushing. Linked brushing refers to the functionality that selections made in any one view will be reflected in all other views on the same data.

The Grand Tour [Asi85] is a method for automatically selecting a continuous sequence of 2D projections of N-D data that are asymptotically arbitrarily close to the set of all possible projections. Practically, this allows one to visualize a high-dimensional dataset with a single animation consisting of a sequence of 2D projections [BA86].

XGobi [SCB98], as well as its more modern successor GGobi [SLBC03], also supports linking of views, various types of scatterplots, 3D rotation views, parallel coordi-nate views and linked brushing. It distinguishes itself from XmdvTool in that it supports a number of extensions of the Grand Tour method described above.

Where XmdvTool and the Gobi systems are applications, Orca is a pipeline-based programming framework for building visualization applications for multi-variate data [SRL*00]. The framework supports multiple linked views, linked brushing and Grand Tour like automated generation of sequences of 2D views. The focus was more on the architecture and design of the programming framework, and not so much on the application of the methods themselves.

WEAVE [GRW*00] is to our knowledge the first published work explicitly integrating linked 3D scientific visualizations with multidimensional statistical representations. All views are coordinated and the statistical representations support linked brushing. The 3D visualizations do not support brushing or selection.

The SimVis software system [DGH03] builds on the ideas presented in WEAVE in that it combines scientific visualization views with information visualization views. It extends this work by presenting a feature definition language (FDL) that formalizes the subsets of data that can be selected by for instance linked brushing.

WEAVE and SimVis depart from the multi-variate information visualization tradition by integrating scientific visualization views, in the form of volume visualizations. However, neither of these systems support linked brushing or selection in the scientific visualization view. Also, whilst Grand Tour-like techniques generate sequences of arbitrary 2D projections of N-D data, all other work discussed in this section that support the interactive definition of 2D projections of N-D data do so only by the selection of two discrete variables from the multi-variate data.

Our work continues in the tradition of WEAVE and SimVis by combining scientific visualization (physical space) and information visualization views, but makes the following new contributions: it supports the interactive visual definition of arbitrary 2D projections of N-D datasets, it also allows linked brushing and selection in the scientific visualization view and, importantly, it allows features selected in information visualization to be directly visualized in the physical view. In addition, we have integrated techniques from pattern analysis that can be interactively applied to enhance the data exploration process.

3. Methods

Our approach employs dynamic multiple-linked-view techniques in an environment where tight coupling enhances the users perception and exploration of the data.

There are four main architectural components in our approach. Firstly the data-model presents a clear and straightforward manner of representing structured volumetric multi-scalar data. Secondly the view component contains the

visual components used for user interaction. Thirdly, a coupling component describes the bindings between the views. Finally, a processing component contains the data-processing methods that provide high-level control.

3.1. The data model

While the medical data we are focusing on are mostly defined on regular grids, our data model supports arbitrarily positioned measurements. The basic element in our model is a data point, which is the set of measurements that share a location.

Each data point contains the physical coordinates at which it was defined, and a feature-space part which represent the measured or derived values. Each data point can be part of a single selection set, which is identified by a numerical identifier.

Two feature types: measured and derived

There are two types of features in our data model. Firstly, the values directly sampled on the physical domain are called the measured features. Since the exploration process later on uses each data point as a separate entity, there is no locality involved that relates for example the value of a point to the value of its neighbors. However, in segmentation this type of local information can be very helpful (see also Section 5.1.2).

For this reason our data model adds derived features. The values for these derived features are computed using locally available measured features. Examples of these computations are gradient computation, denoising and morphological operations.

The main advantage of introducing these derived features alongside the original data is that any combination of derived and measured features can be studied during exploration.

3.2. Views

The components in this part of the architecture present the user with a direct visualization of the data, with which the user interacts to explore the data. The user is presented multiple linked but independent views of the same data. Each of the views shows either a physical-space or a feature-space representation of the dataset.

The physical-space view is based on the actual physical locations of the data points, so that neighboring points in these views represent physical proximity. Usually this mapping is very straightforward, such as a slice view or another form of projection.

The feature-space view complements this by showing only the feature values, unrelated to their physical location. Points close to each other in these spaces share the same characteristics, but can be positioned anywhere in the physical volume.

3.2.1. High-dimensional projection of feature-space points

Since feature spaces are often high-dimensional, a reduction to lower dimensions is often needed for display. Any two-dimensional display reduces the dimensionality of the feature points. We have chosen to let our feature space view show a projection of the feature values. The high dimensional points are projected onto a plane that can be arbitrarily positioned in feature space.

These extra degrees of freedom over feature-selection techniques allow the user to study any linear combination of features as a separate feature. Figure 1 shows how an anti-diagonally positioned plane is used to study the difference between two feature values.
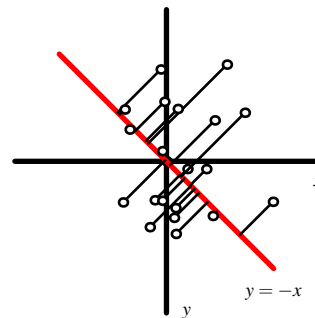
Figure 1: *A projection plane that maps (x, y) to $x - y$, which can be used to study the difference between two feature values directly.*

Compared to the feature-selection techniques that are most often used, this provides a number of advantages:

1. The relation between several features can be studied at the same time
2. Interactive manipulation of the projection plane provides extra insight into the higher dimensional structure of the feature space
3. Differences between features can be used in the same way as the original features, merely by repositioning the hyper-plane

Usually the interest lies in finding clusters, outliers or separation planes. For these measures a projection to one or two dimensions often suffices. Once the data has been reduced to two dimensions, we can employ a scatter plot, or we can use histograms for one dimensional projections.

3.2.2. Main view components

While feature-space views provide information about the similarity of points, it is of prime importance to be able to locate these points in the physical space. Our approach currently includes two physical-space view components and two feature-space view components (see Figure 2).

The basic physical-space view is the slice-viewer. The slicer shows 2D orthogonal planes in physical space, on which the feature-space values are represented as colors or
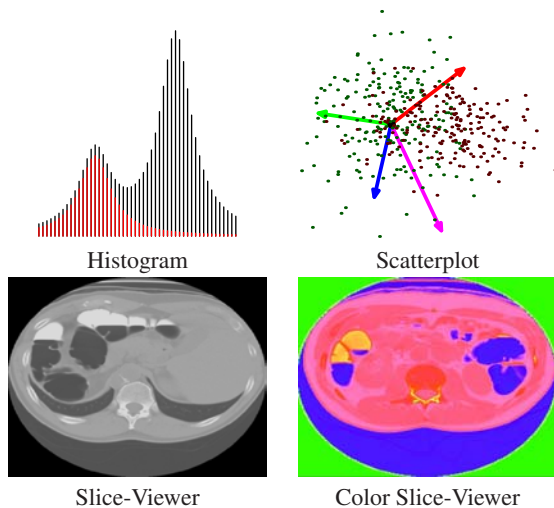
Figure 2: *The four view components: histogram, scatterplot, slice-viewer and pseudo-color slice-viewer.*

Figure 3: *Two clusters of points separated by a single feature. The separation can be easily found by moving the axis (red) corresponding to the separating feature. After moving the axis (right), the separating clusters can be distinguished*

grayscales. To map feature-space values to a color, a projection of the feature values is needed. This projection can be as simple as selecting a single feature and using that as a grey value, but it can also involve an arbitrary projection that maps a set of features onto RGB colors.

The scatterplot component has some improvements over the scatterplot normally used, in order to deal with the large number of points visible at the same time, and the high dimensionality of the data. Each data-point is rendered as a single point. The scatterplot uses additive rendering to give a good impression of the density where points overlap. Where often in high-dimensional data-analysis the scatterplot is used to visualize only two selected features, we allow arbitrary combinations of features to be visualized.

3.2.3. Dynamic axis manipulation

Since high-dimensional projections are difficult to control, we have chosen to use a straightforward direct control mechanism. We describe the projection-plane as being spanned by two (high dimensional) vectors in feature space. Instead of modifying these vectors directly, we transform the two n-dimensional vectors into n two-dimensional vectors, so that each can be displayed and manipulated in the view's coordinate system.

This provides the user with a single two-dimensional vector for each feature (see Figure 3). These vectors are represented as visual axes in the scatterplot, and can be manipulated dynamically through pointing and dragging.

Since each feature has its own manipulable axis, there is a direct link between the movement of the axis and the values of the points for that feature: The larger the value the more the points will move around when the axis is moved. In this way, it is easy to pick up a feature and see how the currently visible points are distributed along this axis.
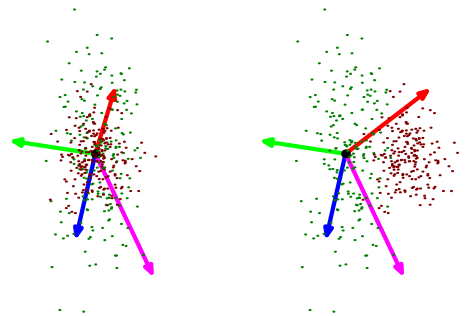
3.2.4. Selection and filtering

While a view of all data points may give a good overview, filtering is required to be able to focus on a specific subset. For example, the removal of air in a medical data set may be necessary to see all the details in the rest of the data.

To support making these selections, we currently use multiple rectangular brush areas, which can be used in physical space as well as in feature space. In this way making a selection of a region of interest works in exactly the same way as marking all points with the same properties (such as air).

To filter out uninteresting points, any selection can be hidden and consequently removed from further processing steps. While a single selection set is often useful enough, more advanced segmentation/exploration methods may require multiple selection sets to be active simultaneously.

For this purpose, we provide the possibility to mark and store selections with colors. Consequently points in the dataset can be divided into groups, each visible with a unique color. Each point belongs to a single group. With our selection system the visibility of selected sets of points can also be toggled, so that only the region of interest is visible and clutter is reduced.

3.3. Coupling

While each of the view components presented in the previous section can by itself be used to explore the dataset, the combination of multiple views is much more powerful. For that reason, we have chosen to use multiple simultaneous viewing components, between which a strong coupling is kept. Selections made in one viewport are immediately propagated to other viewports, and also projections can be linked between views. This section explains in more detail how the views are coupled and how this enhances the data exploration process.

3.3.1. Tightly coupled views

Our approach maximizes the link between the visible views in a highly interactive setting. The dynamic propagation of

selection sets and filters to the other views gives direct feedback of the links between the visible points. The use of color coded, additively rendered, points adds to this experience, since the different selection sets can be distinguished in any view.

Since the brushing feedback is interactive, we can quickly investigate whether a group of points visible in one projection is also a group in a different projection, by simply making a selection in one view and by watching the resulting changes in the other views.

Each view by itself presents a complete representation of the active dataset. The filters introduced earlier are always applied to all views at the same time, and all views share the same colored selections. This helps to maintain a sense of uniformity, especially when many viewing components are active simultaneously.

3.3.2. Linking of projections

While the selection sets are always shared between all views, some other properties can be explicitly linked between a pair of views. For example, the histogram view can be directly linked to the axes of a scatterplot view, so that the histogram will use the projected values. To examine how well two classes can be linearly separated by a one-dimensional separator, we can set up a scatterplot showing both groups of feature-points that we wish to separate and then link the x-axis of the scatterplot to our histogram view. When linked, the histogram view will show a 1D histogram of the projected points, keeping selection colors intact (see Figure 4).

This type of linking of projections is also applicable to the color slice viewer. Since each physical point in the slice viewer has a feature value, the same projection can be used to produce color values. For a single axis link, the color slice viewer will show the projected one dimensional features using a pre-set colormap. To use a higher dimensional projection, each axis can be linked to a color channel of the image.
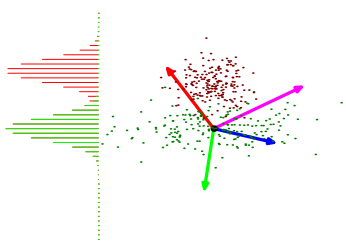
Figure 4: *A histogram linked to a scatterplot. The histogram uses the Y coordinates of the projected points in the scatterplot as a source.*

Changes in projection are, just as changes in the selections, directly propagated into the linked views. Hence, feedback from the other views can directly be used in the interaction loop for finding a suitable projection that separates the points in the desired manner.

3.4. Processing

While the direct manipulation of the scatterplot's projections is an effective to gain insight in high-dimensional spaces, it is often a lot easier to use statistical analysis techniques in order to find proper projections. This component contains all methods that use calculation on the (selected) data to influence the way it is displayed by the visual components.

3.4.1. Using selections to create new projections

Exploring high dimensional feature spaces can be very difficult, even when direct links between feature and physical space are used to put the data into context.

An often recurring task is finding a projection that best shows a selected point set, or that best shows the differentiating features between two sets of points. In the next sections we discuss both situations, and how the processing methods available in our system can assist in these tasks.

In the first case, the high dimensional feature space often shows quite a bit of correlation, which can get in the way of finding intricate details in our area of interest. In these cases, manually finding the projection that keeps most of the important information intact may be tedious and difficult. Therefore we provide the user with an automated way of finding these projections, by the use of principal component analysis (PCA).

In the second case, where the interest lies in separating two sets of points (air and tissue for example), we are often able to make two selection sets typical for each 'material' by brushing in the slice viewer. Once a selection has been made for each of the two materials, linear discriminant analysis (LDA) is applied to find the optimal separating plane between the two sets. This plane will provide us with a single axis for which the spread between the two groups of projected points is maximized.

In all cases the axes we find can be directly applied to any view, and to keep a sense of coordination, we change the current projection to the new projection in a smoothly animated fashion.

For some situations brushing can be very tedious, especially when the clusters in feature space have non-linearly separable shapes. Therefore a third processing method is available in the form of k-nearest-neighbor (kNN) classification. This processing method aims to expand the currently painted selections to points that are similar, or nearby in feature space. It takes as input a partially marked set of points, then for each unmarked point it finds the closest point in feature space and copies its selection ID.

This type of processing method is extremely valuable in a paint-by-example setting. An initial set of selections is made for a number of important tissue types, after which the kNN process expands the selection to the whole dataset, resulting in a segmented volume.

4. System/Implementation

We have implemented the above methods in a prototype system, while keeping in mind the following goals:

1. Clear visual presentation of all components
2. Interactive display and manipulation at high frame rates
3. Good scalability up to datasets that contain tens of millions of points

The prototype system has been implemented in C++ using OpenGL for rendering, for maximum flexibility and speed.

4.1. Arbitrary number of views and components

The implemented system allows an arbitrary number of view components to be active at the same time. Figure 5 shows the layout of the screen during an exploration session. To simplify the user interface we have chosen a regular gridded layout. Each cell in the grid can contain a single viewing component, such as slice-viewers or scatterplots.
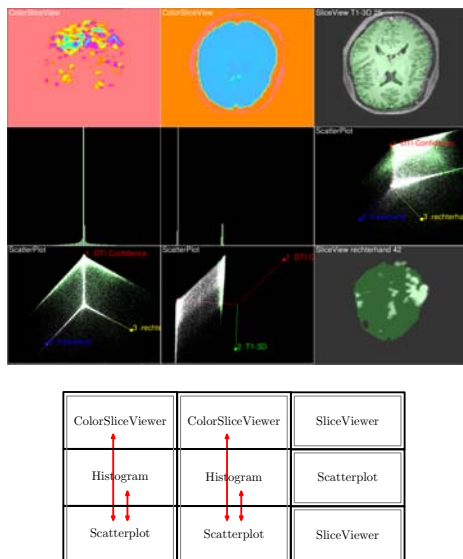
Figure 5: *Top: The interface during a user session. Bottom: The component grid, where red arrows indicate the coupling between components.*

The configuration of these components can be stored and loaded for task-specific exploration scenarios. The coupling between views can be modified at run-time, but a pre-set combination of couplings is initially present. Most parts of the user interface are stateless, so there is no hidden internal state hindering the user experience.

4.2. GPU-efficient data structures

To achieve optimal interactivity, even when processing datasets containing millions of points, efficient data-structures are needed throughout the processing pipeline. To exploit modern graphics hardware, our system uses GPU-friendly data-structures. The feature points are stored on the

graphics card in vertex-buffers as well as in main memory. Since the high-dimensional projections used throughout the view components are all written as GPU shaders, almost no communication is needed between GPU and CPU, which usually forms a bottleneck in these kinds of applications. This allows us to transform and draw over 80 million points per second for 4D datasets on a 2GHz AMD64 system with an ATI x1900 videoboard.

Number of features	MPoints/sec.	Frames/sec.
4	80.2	61.2
8	44.2	33.7
12	41.7	31.8

Table 1: *Rendering performance for the scatterplot with 1.3 million feature-points for a varying number of features.*

The state of the selection groups has to be kept synchronized between the two storages, for which a simple but effective update mechanism was implemented. All changes are propagated from the main memory to the vertex buffers in a batch-wise fashion, updating as few points as possible for a given optimal batch size.

5. Applications

To show the applicability of the exploration techniques we have used the presented methods to perform a number of often required explorations tasks on medical data.

5.1. Material classification in CT virtual colonoscopy

In virtual colonoscopy, an important processing step is to find the surface of the colon wall, so that polyps can be detected. To gain more contrast in differentiating the wall, scans are made with a contrast liquid and extra CO_2 gas is added inside the colon. Even with these measures, segmenting the colon surface can be tricky because of partial volume effects that occur on the boundary between tissue and air, or tissue and contrast medium.

The following sections describe how our exploration methods can be applied to this type of problem. The data exploration in this case involves two basic steps. Firstly, we describe the exploration process for finding a linear separation by brushing. Secondly we show how the additional derived features can be used to find a better segmentation.

We will start with a simple scalar CT dataset, which we load into a 2×2 gridded user interface as shown in figure 6.

5.1.1. Separating tissue, air and contrast medium

Since the main interest lies in the colon wall, we build an initial segmentation that includes the three main materials. Using the brushing interface of the slice-viewer, we mark parts of air, contrast medium and tissue, as seen in Figure 7(a).

Once the selection has been made, we want to expand this selection to similar points. The histogram at this point shows
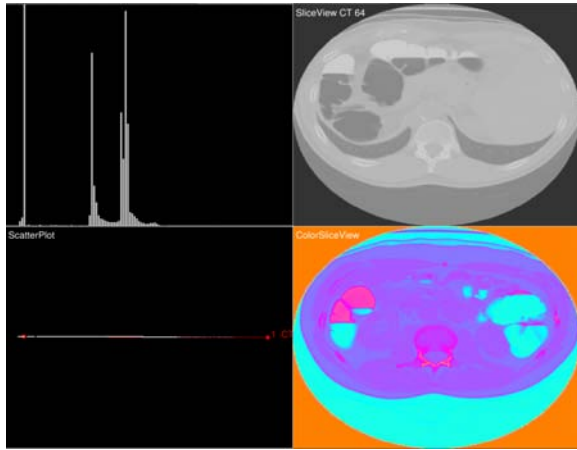
Figure 6: *The gridded* 2 × 2 *layout of the user interface with the CT dataset loaded. From top left to bottom right: histogram, slice view, scatter plot, color slice view*

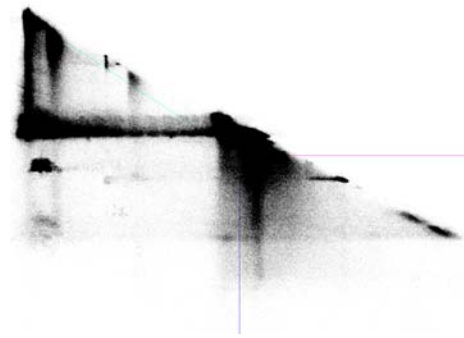(a) Initial Selection   (b) Expanded Selection
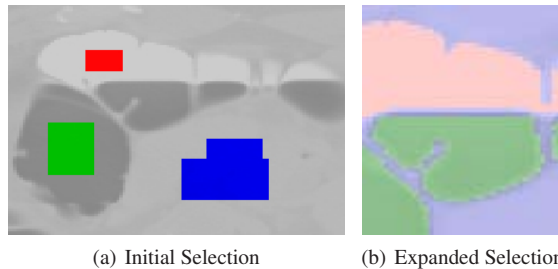
Figure 7: *A shows the initial selection created through brushing in the slice-viewer. Red represents contrast liquid, green represents gas and blue marks tissue. B shows a detail of the expanded selection through application of the kNN classification process, note that the boundary is not correctly classified.*

fairly well separated peaks, suggesting that a simple classification/separation can be found. By selecting the kNN classification process (see Section 3.4.1), each unmarked point is assigned the selection ID of the closest marked point in feature-space. The result of this processing step is a fully marked dataset, as seen in Figure 7(b).

This initial segmentation however has misclassified the boundary. The boundary between air and contrast medium was wrongly classified as tissue. This is caused by the partial volume effect, whereby the voxels on the boundary have the same scalar value as surrounding tissue.

5.1.2. Introduction of derived features

Since direct classification of the voxels at the air-contrast boundary is hampered by the partial volume effect, further steps are taken to introduce derived features (see Section 3.1). Derived features are computed based on physical-space neighborhoods, so a local gradient or minimum/maximum can at this point be added to the dataset.

Figure 8: *Scatterplot showing the distribution of points in LH-space. Horizontal axis is L, vertical is H.*
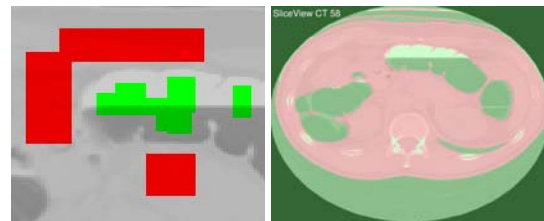
Figure 9: *Expanded selection through application of the kNN classification process on the feature space that was expanded with the derived LH feature.*

We choose to add LH features [SVSG06]. The LH values represent the local minima (L) and maxima (H) by following the local gradient direction. In this way, the boundary between air and contrast medium will be identified by an L value corresponding to air, and an H value corresponding to contrast medium.

By examining the scatterplot for the newly added features (see Figure 8), we already see a number of clusters emerging. Exploring these interactively by brushing in the scatterplot and watching the slice-views change, we note that (as expected) transitions between materials are presented by clusters in LH-space.

To test how well our LH space helps us to separate the materials, we start by brushing a new selection set which explicitly includes the boundary surface (Figure 9). We expand the selection through kNN and note that the two extra dimensions (L and H) in our feature space help to find a proper segmentation, as shown in Figure 9. Note how the materials are correctly classified and the colon surface is correctly segmented; the boundary no longer poses a problem.

5.2. Functional MRI combined with structural MRI

A number of different scanning techniques are currently employed in brain imaging through MRI. The different techniques provide different types of 3D volumes, and are often used together. We will focus on neurosurgical planning for the resection of brain tumors, where it is of prime importance to prevent damage to the important functional areas. For this purpose, multiple scans are made; First a structural MRI to

provide a whole-brain scan as a reference image. Secondly, multiple functional MRI scans map the activation areas for the main motor tasks, such as hand and foot movement

For each of the motor tasks, the functional MRI scan is preprocessed to provide a volume containing the Z values, which describe the correlation of each voxel to the specific task. In total, we have five measured features: one structural MRI and four functional MRIs (for hand and foot motion on each side).

5.2.1. Mapping the motor tasks

To find out the positioning of the location of the major activation areas, we used histogram brushing on the fMRI features to threshold the Z values. In this way, each motor task area is marked with a unique color. The sliceviewer can then be used to inspect these areas as they are overlaid on the structural MRI.

5.2.2. Adding a new derived feature: tumor distance

In planning the surgery, it is often important to use a safety margin around the tumor and see if any functional activation areas are within this range. For this, a semi-manual segmentation of the tumor is made beforehand, which we can use to introduce a new derived feature. We apply a distance field processing step, generating a distance map for the tumor, in which each voxel stores the minimum distance to the tumor. This volume can consequently be added as a new derived feature.

Using brushing and filtering on this distance feature, we can now focus on the area inside the safety margin around the tumor. Since filtering does not interfere with the selections assigned previously, the activation areas inside this area can be directly studied.

6. Conclusions and future work

With this work, we have made the following contributions:

- We presented a dynamic coordinated view approach for the interactive visualization of (medical) multi-field data.
- We showed how scatter plots can be created with arbitrary projections of high-dimensional feature space.
- We demonstrated how techniques from pattern analysis can be interactively applied to facilitate a visualization-based multi-field exploration process.
- Using two case studies, we demonstrated the application of our approach to real medical data.

The techniques we have described can be used for a wide variety of tasks. Especially a quick evaluation of the relevance of certain features for a given problem is extremely useful in practice. Manual exploration of 2D feature-space projections is augmented by the PCA, LDA, and kNN analysis, and in this way the user is guided quickly toward important views of a high-dimensional feature space. Throughout, two facilities are essential: the dynamic feedback displayed in

multiple views, and the direct axis manipulation. Of course the system as it is now is only intended for users with a background in visual data analysis. For clinical use, a task- oriented user interface layer is required, which will require a major design effort.

There are many possible directions for future research. The current system can be extended with more components for analysis and viewing, such as volume rendering and parallel axis views. Also, more support for time-varying data can be added, such as adding local change metrics as new derived features.

For better support of the interaction and exploration process, hierarchical selection techniques could be added. This could be combined with a logging mechanism to document the analysis process and facilitate backtracking.

References

[Asi85]   ASIMOV D.: The grand tour: a tool for viewing multidimensional data. *SIAM J. Sci. Stat. Comput. 6*, 1 (1985), 128–143.

[BA86]   BUJA A., ASIMOV D.: Grand tour methods: an outline. In *Proc. 17th Symp. on the Interface of Computer Sciences and Statistics* (1986), Elsevier North-Holland, Inc., pp. 63–67.

[DGH03]   DOLEISCH H., GASSER M., HAUSER H.: Interactive feature specification for focus+context visualization of complex simulation data. In *VISSYM '03: Proc. Symposium on Data visualisation* (2003), Eurographics Association, pp. 239–248.

[GRW*00]   GRESH D., ROGOWITZ B., WINSLOW R., SCOLLAN D., YUNG C.: Weave: a system for visually linking 3-d and statistical visualizations applied to cardiac simulation and measurement data. In *Proc. IEEE Visualization* (2000), pp. 489–492.

[SCB98]   SWAYNE D. F., COOK D., BUJA A.: XGobi: Interactive dynamic data visualization in the x window system. *J. Comp. Graph. Stat. 7*, 1 (March 1998), 113–130.

[SLBC03]   SWAYNE D. F., LANG D. T., BUJA A., COOK D.: GGobi: evolving from XGobi into an extensible framework for interactive data visualization. *Comput. Stat. Data Anal. 43*, 4 (2003), 423–444.

[SRL*00]   SUTHERLAND P., ROSSINI A., LUMLEY T., LEWIN-KOH N., DICKERSON J., COX Z., COOK D.: Orca: A visualization toolkit for high-dimensional data. *J. Comp. Graph. Stat. 9*, 3 (September 2000), 509–529.

[SVSG06]   SEREDA P., VILANOVA A., SERLIE I. W. O., GERRITSEN F. A.: Visualization of boundaries in volumetric data sets using lh histograms. *IEEE Trans. on Visualization and Computer Graphics 12*, 2 (2006), 208–218.

[War94]   WARD M. O.: Xmdvtool: integrating multiple methods for visualizing multivariate data. In *Proc. IEEE Visualization* (1994), IEEE Computer Society Press, pp. 326–333.