Extending DeViDe – Fourfold

Final Assignment for the TU Delft course *IN4307 Medical Visualization*

J.B. van Velzen1 and R. Nieuwenhuizen1

1TU Delft, The Netherlands

**Abstract***The ABSTRACT is to be in fully-justified italicized text, between two horizontal lines, in one-column format, below**the author and affiliation information. Use the word “Abstract” as the title, in 9-point Times, boldface type, leftaligned to the text, initially capitalized. The abstract is to be in 9-point, single-spaced type. The abstract may be up to 3 inches (7.62 cm) long.*

*Leave one blank line after the abstract, then add the subject categories according to the ACM Classification Index (see* [*http://www.acm.org/class/1998/*](http://www.acm.org/class/1998/)*).*

Categories and Subject Descriptors (according to ACM CCS): I.4.10 [Computer Graphics]: Image Representation—Volumetric

1. **Introduction**

At the beginning of the course IN4307 Medical Visualization at the TU Delft, we were introduced to DeViDe (the Delft Visualization and Image processing Development Environment). This software package was also used during the practical assignments of this course, where we got an actual taste of what was possible with DeViDe. Because of the experiences we had in the practical exercises, we decided to use DeViDe for the final project.

During the practical exercises we obtained some experience with segmentation from DICOM data, but we always felt that the experience we had obtaining a specific segmented result from a DICOM dataset could use some improvement. We therefor decided to create a intuitive approach of making selections in 3D DICOM data, and representing this in a user friendly way.

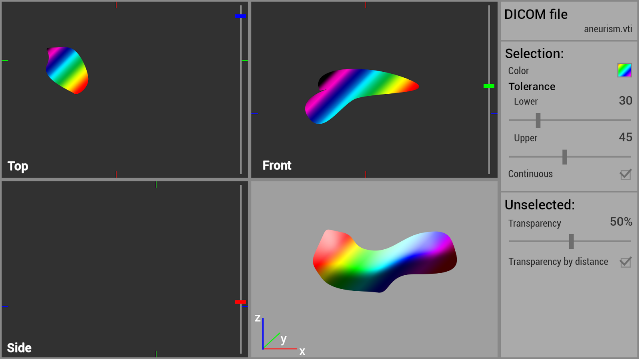
1. **Related work**

In DeViDe, of course there were some comparable viewers. One example is the *slice3dVWR* we used a lot in the practical. Other viewers that we were pointed to by the assistants -for which we thank you- are the EmphysemaViewer and the SkeletonAUIViewer. These provided a nice skeleton for us to start with.

Furthermore, we looked at other software that displays data in the same way we envisioned our final result would. Two of those being MITK[[1]](#footnote-1) and Slicer[[2]](#footnote-2). These inspired us to implementing some of the features described in the next paragraph, on top of the features we already put in our project proposal.

1. **Proposed method**

The purpose is to create a DeViDe module that can create a Graphical User Interface as roughly designed in Figure 1. This module will contain of a frame with three two-dimensional views displaying the CT or MRI slices of the DICOM data, each from a different side (Top/Transverse, Side/Sagittal and Front/Coronal). The fourth view will be tree-dimensional, displaying a 3d object contour of the DICOM data. On the right of these four views is a control panel with which the user has some freedom in creating/extracting a selection from the start points that are selected in the 3 two-dimensional views. These are, for example, selection thresholds and the way the unselected part of the data will be presented. The exact specifications are listed after.



**Figure 1:** *A rough design for the desired user interface and functionality.*

NOTE that these specifications do not 100% correspond with the original rough design, since some features are added after, and some were left out in the final product.

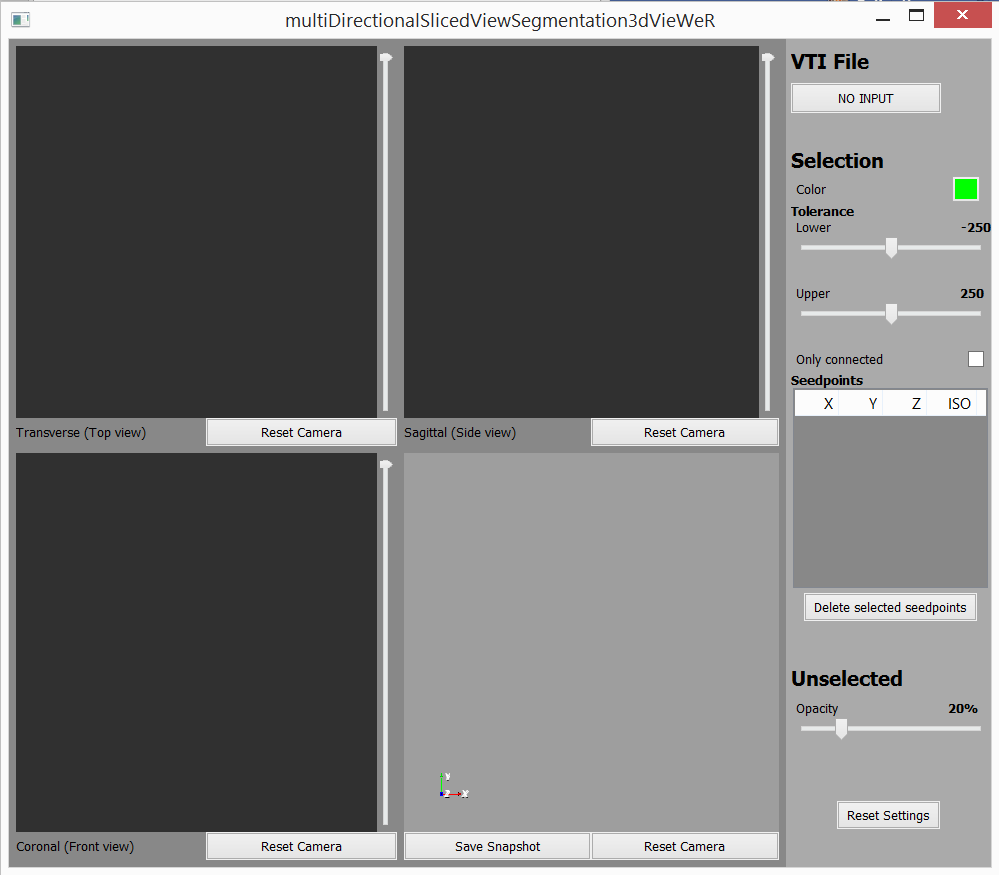
* 1. **Interface specifications**
* The new DeViDe module will have the working title *‘multiDirectionalSlicedViewSegmentation3dVieWeR’*.
* The module should load a frame with 4 views and a control panel on the right side for the user to influence the selection settings.
* The three darker view are 2D views that display a slice of the DICOM data, each from a different side (transverse/axial, sagittal and coronal)
  + The content of each of these views can be moved by clicking and holding a certain mouse button outside the slice data and then dragging the mouse.
    - Rotating: Left mouse button
    - Translating: Middle mouse button
    - Scaling: Right mouse button
  + Sliders on the right of all three 2D views can be used to navigate through the individual slices in the data.
* The fourth, lighter, view displays in 3D the unselected data in a transparent way, and the selected part in an opaque color.
  + The content of the 3D view can be moved by clicking and holding a certain mouse button and then dragging the mouse.
    - Rotating: Left mouse button
    - Translating: Middle mouse button
    - Scaling: Right mouse button
* All four views have a button on the bottom allowing the user to reset the camera.
* The fourth, 3d, view also has a button allowing the user to take a snapshot of the current shown 3d render in that view.
* The control panel settings include:
  + A file-selection-button
  + Settings for obtaining a selection:
    - The selection color,
    - The selection tolerance (upper and lower),
    - The connectivity of the selection,
    - A list showing all currently selected points and
    - A button to delete selected points from the previously mentioned list.
  + Settings for the unselected data:
    - The transparency.
  + A button to reset all settings
  1. **Usability specifications**
* VTKImageData can be loaded into the module through DeViDe’s network and its contents will be displayed in the previously mentioned views.
* VTKImageData can also be loaded by pressing a button in the top part of the control panel and then selecting a .vti-file in the file explorer.
* When a user clicks in a 2d view, the selected point will be added to the list of seedpoints and a new selection will be calculated.
  + If an ‘only-connected’ selection is required, the selection will be calculated by taking all the seed points and their values and use them in a region-growing algorithm, using the threshold bounds to get the final result.
  + If an ‘only-connected’ result is not required, the selection will be calculated by simply thresholding the DICOM data for each seed point and adding results to get the final result.
* After a new selection is calculated, the 2d views will show the current selection in the current slice. The 3d view will show a 3d selection, in combination with the unselected data. The unselected data has a transparency that is defined in the control panel.
  1. **Implementation**

Since the base of this project is using existing algorithms and combining them into user-friendly interface, we did not implement any of the actual thresholding and contouring algorithms that we used to create our final product. The entire implementation we did was on the part of the Graphical User Interface and stringing together already available VTK algorithms.

1. **Results**

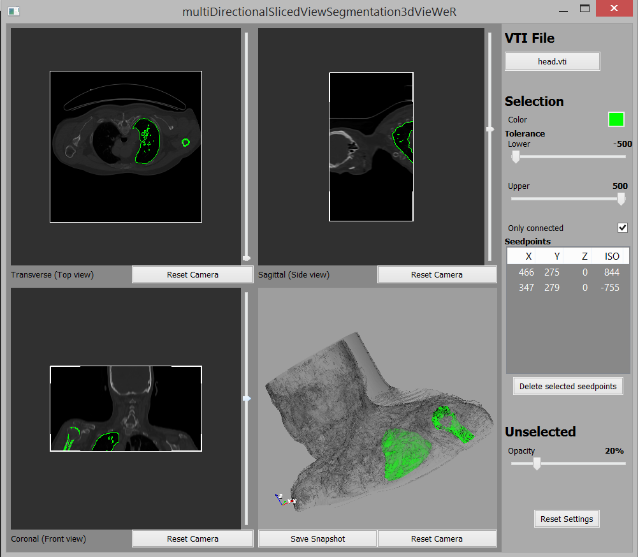
Comparing the original design with the final product, as seen in Figure 2, we can see they there are very similar.

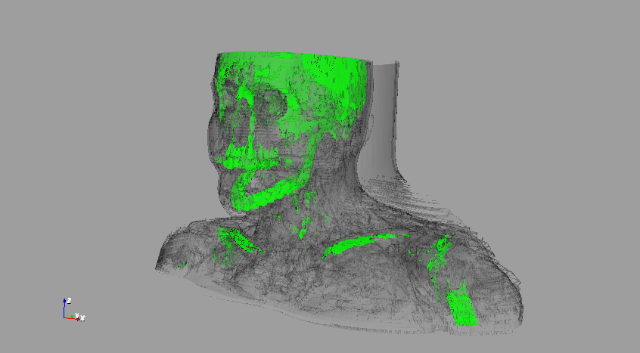
When starting up DeVIDE with the new module included. (using %PATH\_TO\_DeVIDE-RE%/dre.cmd devide –extra-module-paths %PATH\_TO\_MODULE%), you will be able to locate the *multiDirectionalSlicedViewSegmentation3dVieWeR* in the module list. Load one into your network, and you will see the frame like in figure 2.



**Figure 2:** *The default multiDirectionalSlicedViewSegmatation3dVieWeR view.*

At this point, you can choose either to load up a vtiRDR in your network as well, and then execute the network, or you can directly choose a file from your file system.

 **Figure 3:** *An active multiDirectionalSlicedViewSegmatation3dVieWeR view.*



**Figure 4:** *An example snapshot*

1. **Conclusion**

DeViDe was in need of help and we came to the rescue ?

WAS KUT, want geen documentatie

//Did not end up in here:

* Indicators
* On

1. **Future work**

This module has some improvements that can be made, but were left out. One of those improvements is to make it possible to select a different color for each seed point, this would greatly improve the segmentation options, and the visibility of the results.

For further improvements the colored indicators from the original specifications could be added. It might be difficult to create them as an overlay in the vtk framework, but it would be nice to see how the other views are zoomed, compared to a specific 2d view.

Transparency by distance

2d views fullscreen.

1. <http://www.mitk.org> [↑](#footnote-ref-1)
2. <http://www.slicer.org> [↑](#footnote-ref-2)