# Extending DeVIDE – Fourfold

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

J.B. van Velzen[1] and R. Nieuwenhuizen[1]

[1]TU Delft, The Netherlands

**Abstract**

*In DeVIDE, it used to be a tedious job to create segmentations from 3D DICOM data. This article presents an explanation of the newly created module multiDirectionalSlicedViewSegmentation3dVieWeR, which is created to visualize 3D contours based on one or more easy to select seed points. The module does not introduce new algorithms, it rather combines existing VTK algorithms into a single framework and presents them in an intuitive and easy to use way.*

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 therefore decided to create an intuitive approach of making selections in 3D DICOM data, and representing this in a user friendly way.

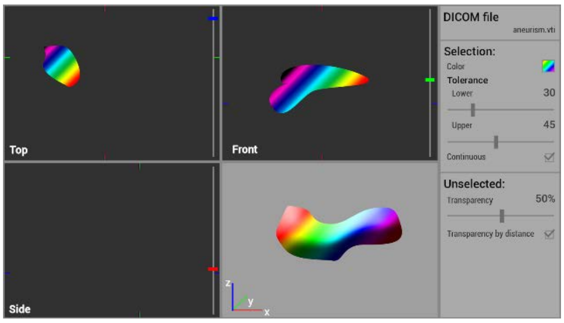
## 2. Related work

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] and Slicer[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.

## 3. 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.



DICOM file
aneurism.vti

Selection:
Color
Tolerance
Lower 30
Upper 45
Continuous ☑

Unselected:
Transparency 50%
Transparency by distance ☑

**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] http://www.mitk.org
[2] http://www.slicer.org

### 3.1. Interface specifications

- The new DeVIDE module will have the working title *‘multiDirectionalSlicedViewSegmentation3dVieWeR’*.
- The module will scale up and down when the window is resized, making sure the important content gets more priority.
- 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, brighter, 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

### 3.2. Usability specifications

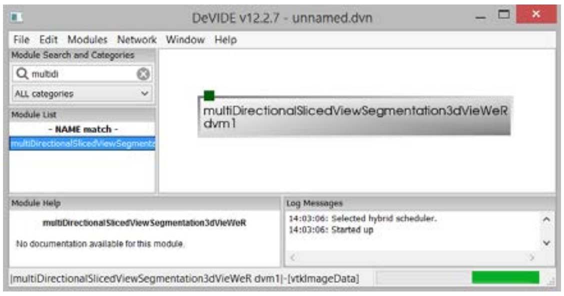
- VTKImageData can be loaded into the module through DeVIDE’s network and its contents will be displayed in the previously mentioned views.
- VTKImageData can also be loaded by pressing a button in the top part of the control panel and then selecting a .vti-file in the file explorer.
- When a user clicks in a 2d view, the selected point will be added to the list of seed points 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.

### 3.3. 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.

## 4. Results

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 find the *multiDirectionalSlicedViewSegmentation3dVieWeR* in the module list. This module has one inputPoint where (optionally) vtkImageData can be supplied. The easy way to supply vtkImageData from a .vti-file is with a vtiRDR module. This can be seen in Figure 2 respectively Figure 3.


DeVIDE v12.2.7 - unnamed.dvn
File
Edit
Modules
Network
Window
Help
Module Search and Categories
multidi
ALL categories
Module List
- NAME match -
multiDirectionalSlicedViewSegment
multiDirectionalSlicedViewSegmentation3dVieWeR
dvm1
Module Help
Log Messages

multiDirectionalSlicedViewSegmentation3dVieWeR

No documentation available for this module.

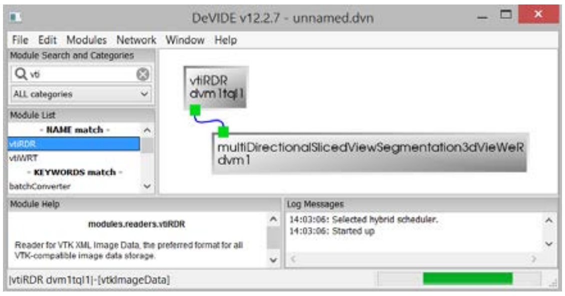
```
14:03:06: Selected hybrid scheduler.
14:03:06: Started up
```

|multiDirectionalSlicedViewSegmentation3dVieWeR dvm1|-[vtkImageData]

**Figure 2:** *The minimal DeVIDE Network.*

When the user loads its data through the DeVIDE network, a different .vti-file can still be selected through the control panel.

Comparing the original design with the final product, as seen in Figure 2, we can see they there are very similar. The default window that the module opens on every time can be seen in Figure 4.


DeVIDE v12.2.7 - unnamed.dvn
File Edit Modules Network Window Help
Module Search and Categories
vti
ALL categories
Module List
- NAME match -
vtiRDR
vtiWRT
- KEYWORDS match -
batchConverter
vtiRDR
dvm1tql1
multiDirectionalSlicedViewSegmentation3dVieWeR
dvm1
Module Help
Log Messages

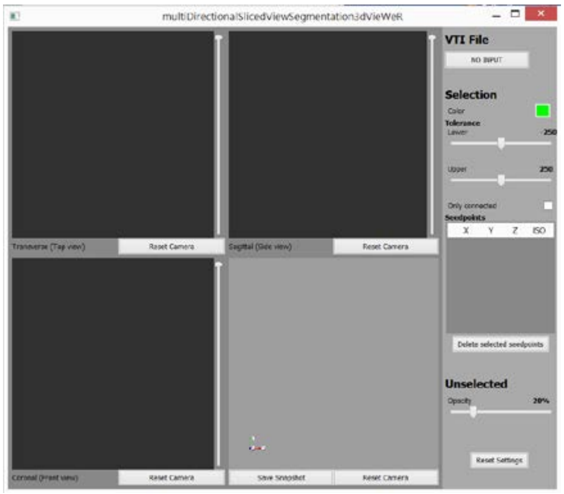
modules.readers.vtiRDR

Reader for VTK XML Image Data, the preferred format for all VTK-compatible image data storage.

```
14:03:06: Selected hybrid scheduler.
14:03:06: Started up
```

|vtiRDR dvm1tql1|-[vtkImageData]

**Figure 3:** *Loading vtkImageData through from the DeVIDE network.*

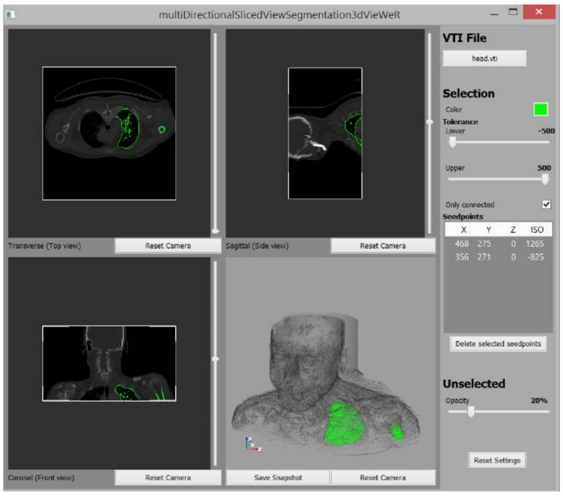


VTI File: NO INPUT
Lower: -250
Upper: 250
Only connected: □
Opacity: 20%

**Figure 4:** *The default multiDirectionalSlicedViewSegmatation3dVieWeR view.*

To illustrate the results this application can produce, we have taken three different datasets and tried to produce sensible results from them.

The first is from the file *lungs.vti.* We selected 2 points, one in the left lung, the other in the left humerus (upper arm bone). We selected an 'only-connected' result, and set the thresholds to their maxes. The result can be seen in Figure 5.



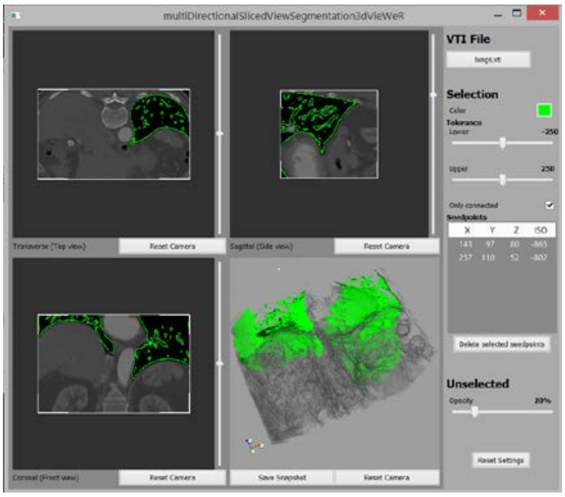
VTI File: head.vti
Lower: -500
Upper: 500
Only connected: ☑
Seedpoints

| X | Y | Z | ISO |
|---|---|---|---|
| 468 | 275 | 0 | 1265 |
| 356 | 271 | 0 | -825 |

Opacity: 20%

**Figure 5:** *An active multiDirectionalSlicedViewSegmatation3dVieWeR view on head.vti.*

Our second dataset is from the file lungs.vti. Again two points are selected (one in each lung), the thresholds are left untouched and 'only-connected' is enabled. As suspected this yields a similar result as with the head.vti dataset and can be seen in Figure 6.



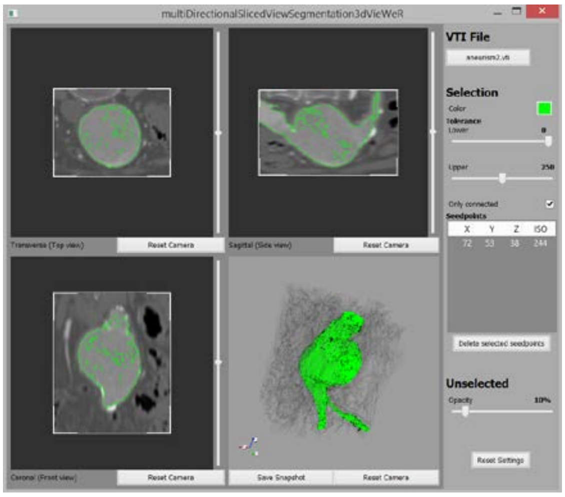
VTI File: lungs.vti
Lower: -250
Upper: 250
Only connected: ☑
Seedpoints

| X | Y | Z | ISO |
|---|---|---|---|
| 143 | 97 | 80 | -865 |
| 237 | 110 | 52 | -802 |

Opacity: 20%

**Figure 6:** *An active multiDirectionalSlicedViewSegmatation3dVieWeR view on lungs.vti.*

Lastly, in order to test on a dataset that had different data values than the two datasets near the lungs, we took the aneurism.vti dataset. With a single seed point, 'only-connected' enabled and the lower threshold margin set to 0, we were able to get a very good segmentation, as is shown in Figure 7. We made this even clearer by setting the 'unselected-transparency' to 10%.



VTI File: aneurism2.vti
Lower: 0
Upper: 250
Only connected: ☑
Seedpoints

| X | Y | Z | ISO |
|---|---|---|---|
| 72 | 53 | 38 | 244 |

Opacity: 10%

**Figure 7:** *An active multiDirectionalSlicedViewSegmatation3dVieWeR view on aneurism.vti.*

## 5. Conclusion

As presented in the introduction, DeVIDE was in need of a module that combined several VTK techniques into a single and easy to use module. Looking at the results section, this is exactly what we have achieved. There were several changes compared to the original specifications, but the essence really stayed the same.

Achieving this result was a hard job at first. The lack of documentation on many DeVIDE modules and the small amount of VTK support available on the web did not make it easy to find a place to begin. Eventually, using other viewers and VTK source code as a reference, we got to understand how things worked.

We can conclude by saying this project was successful. The module does yield very good results, with minimal effort for the user.

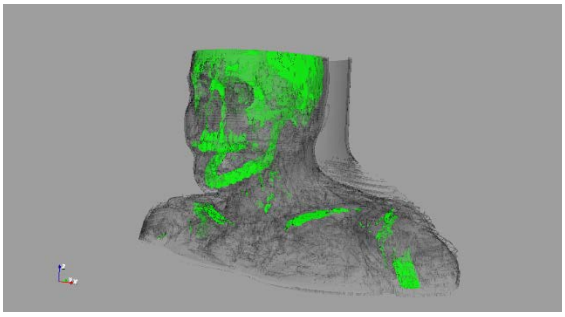


**Figure 8:** *An example snapshot*

### 6. Future work

This module has some improvements that can be made, but were left out or that we came up with after the product was in the final stages.

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 colored indicators could be added. It might be difficult to create them as an overlay in the VTK framework, but it would be helpful to see what slices the other 2d views are currently showing.

Another feature that we originally envisioned but never got in the final product was the ability to change the transparency of unselected data by the distance that it is from the selected data.

We found that, when selecting very specific seed points, it is sometimes hard to get the selection right, because of the resolution of the 2d views, even when maximizing the entire window. A solution for this, besides using a really big monitor, would be to be able to (temporarily) make a 2d view take up the entire screen, so a precise selection can be made.

It might also be a good idea, efficiency wise, to optimize the selection in the end, because currently the results for each of the seed points are just combined, without checking for selections that are selected through another seed point.

Lastly, it might be nice to have some shading on the final selection, since now it can be a little hard to distinct shapes when looking at a selection head-on. The semi-transparent unselected data helps here to give an indication of the shape, but this does not work in all cases.

Delft University of Technology