A Tutorial Introduction to STAPLE

By Alireza Akhondi-Asl and Simon K. Warfield

Computational Radiology Laboratory, Children's Hospital Boston, Harvard Medical School, Boston MA 02115. Please address enquiries to: simon.warfield@childrens.harvard.edu

Introduction

STAPLE is an expectation-maximization algorithm for Simultaneous Truth And Performance Level Estimation. The algorithm considers a collection of segmentations and computes a probabilistic estimate of the hidden, implicit, true segmentation and a measure of the performance level achieved by each segmentation. The source of each segmentation in the collection may be an appropriately trained human rater or raters, or automated segmentation algorithms. The probabilistic estimate of the true segmentation is formed by estimating an optimal combination of the segmentations, weighting each segmentation depending upon the estimated performance level, together with a prior model that can account for the spatial distribution of structures and spatial homogeneity constraints. STAPLE is straightforward to apply to clinical imaging data, it readily enables assessment of the performance of an automated image segmentation algorithm, and enables direct comparison of human rater and algorithm performance. The probabilistic segmentation that is estimated is a type of consensus segmentation, which is often used as a reference standard segmentation for assessing performance of different algorithms and when an improved segmentation is needed.

Segmentation Evaluation

Measurement tools are often characterized by assessment of their accuracy and precision. The accuracy of a human or an algorithm in creating a segmentation is the degree to which the segmentation corresponds to the true segmentation, and so the assessment of accuracy of a segmentation requires a reference standard, representing the true segmentation, against which it may be compared. Precision is determined by the reproducibility of the segmentations obtained repeatedly from the same image. The precision may be assessed without comparison to a reference standard. High accuracy and high precision are both desirable properties. An ideal reference standard for image segmentation would be known to high accuracy and would reflect the characteristics of segmentation problems encountered in practice. There is a tradeoff between the accuracy and realism of the reference standard. As the accuracy of the reference standard segmentation increases, the degree to which the reference standard reflects segmentation problems encountered in practice tends to decrease.

The accuracy of segmentations of medical images has been difficult to quantify in the absence of an accepted reference standard for clinical imaging data. Synthetic images can be constructed with known true segmentation (high accuracy) but typically lack characteristics encountered in practice. Phantoms can be built and imaged, which increases the realism of the model by incorporating the imaging system characteristics, but also reduces the fidelity with which the true segmentation is known. Although such physical and digital phantoms have an important role to play in quantifying algorithm performance, such phantoms do not fully reflect clinical images due to the difficulty of constructing phantoms that reproduce the full range of imaging characteristics (such as partial volume artifact, intensity inhomogeneity artifact and noise). In addition, such phantoms typically cannot reproduce both the normal and pathological anatomical variability observed in clinical data. Therefore, alternative approaches have an important role, as it is not necessarily possible to generalize performance measurements on a phantom to the results expected in practice. Patient data provides the most realistic model for a given application task, but is the most difficult for which to identify a reference standard.

A common alternative to phantom studies has been to carry out behavioral comparisons: an automated algorithm is compared to the segmentations generated by a group of experts, and if the algorithm generates segmentations sufficiently similar to the experts, it is considered to be an acceptable substitute for the experts. In addition to acceptable accuracy, good automated segmentation algorithms also typically require less time to apply and have better precision than interactive segmentation by an expert. The most appropriate way to carry out the comparison of a segmentation to a group of expert segmentations has been unclear. A number of metrics have been proposed to compare segmentations. Simply measuring the volume of segmented structures has often been used [1], [2].

Two segmentation methods may be compared by assessing the limits of agreement [3] of volume estimates derived from the segmentations. However, volume estimates may be quite similar when the segmented structures are located differently, have different shapes or have different boundaries [4] and so alternative measures have been sought [5]. Furthermore, measures inspired by information theory have been applied [6-7]. Agreement measures for comparing different experts, such as the kappa statistic, have also been explored [8]. Next, we describe commonly used evaluation methods in the literature for the segmentation evaluation.

Overlap Based methods

For *N* class (label) segmentation, confusion matrixes *C* of size *N*×*N* can be defined where C_{ij} is the number of voxels segmented with label *i* while the true label is *j*. Based on this information, there are many measures that can be defined for the quality assessment. For example, in [9] two different multi-class type I and II error classifications were defined. Where, $M_I^{(k)}$ and $M_{II}^{(k)}$ are equal to zero for the ideal segmentation.

$$\begin{gathered} M_I^{(k)} = \left[\sum_{\substack{i=1 \\ i\neq k}}^{N} C_{ik} \right] \Bigg/ \left[\sum_{i=1}^{N} C_{ik} \right] \\ M_{II}^{(k)} = \left[\sum_{\substack{i=1 \\ i\neq k}}^{N} C_{ik} \right] \Bigg/ \left[\sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq k}}^{N} C_{ij} \right] \end{gathered} \tag{1}$$

For any label k, we can define true positive (TP), true negative (TN), false positive (FP), and false negative (FN) based on the confusion matrix as follow:

$$\begin{aligned} TP &= C_{kk} & TN &= \sum_{\substack{i=1 \\ i\neq k}}^{N} \sum_{\substack{j=1 \\ j\neq k}}^{N} C_{ij} \\ FN &= \sum_{\substack{i=1 \\ i\neq k}}^{N} C_{ik} & FP &= \sum_{\substack{j=1 \\ j\neq k}}^{N} C_{kj} \end{aligned} \tag{2}$$

Based on these definitions, we can describe important and commonly used measures such as specificity (SP), sensitivity (SE), negative predictive value (NPV), and positive predictive value (PPV).

$$SP = TP/(TP+FN) \qquad SE = TN/(TN+FP) \tag{3}$$

$$NPV = TN/(TN+FN) \qquad PPV = TP/(TP+FP) \tag{4}$$

In addition, we can define two important spatial overlap measures [11], [12]:

$$\begin{gathered} Dice = 2\times TP/(2\times TP+FP+FN) \\ Jaccard = TP/(TP+FP+FN) \end{gathered} \tag{6}$$

It should be mentioned that there is a nonlinear relation between Jaccard and Dice.

$$Dice = \frac{2\times Similarity}{1+Similarity} \tag{7}$$

In addition, it can be easily shown that always $Dice \geq Similarity$. Both of these measures take values between 0 and 1 where 1 indicates perfect segmentation.

It is not easy to say that which one these metrics should be used in the practice. In the other words, it is possible to find cases that some of them fail to grade segmentations precisely. However, between these metrics PPV has some specific properties that make it desirable for many applications. First of all it does not depend on the size of the ground truth. This property is very important when you have multi category segmentation problem with variety of structure sizes. In addition, PPV gives the probability of the truth given the decision, which makes it very desirable metric for many applications.

Distance Based methods

All of the above measures are based on the misclassified and correctly classified voxels. In the other words, if two different voxels are misclassified they have the same effect on the quality of the segmentation regardless to their distance to the ideal segmentation. To do this, another class of evaluation methods has been introduced which take to account distance information. To define these measures, we define two vectors. For each voxel on the boundary of the true label, we find the distance to the closest voxel on the boundary of the segmentation volume. We put the values in the vector and name it d_r. We use the same strategy vice versa to find the vector d_s. Based on these two vectors we can define following measures.

$$\begin{gathered} ND = \frac{\sqrt{\sum_{i=1}^{N_s} d_s{}^2(i)}}{N_s} \\ HD = \max\left(\max(d_s), \max(d_r)\right) \\ MD = \frac{\sum_{i=1}^{N_s} d_s(i)}{N_s} \\ RMSD = \frac{\sqrt{\sum_{i=1}^{N_s} d_s^2(i) + \sum_{i=1}^{N_r} d_r^2(i)}}{N_s+N_r} \end{gathered} \tag{7}$$

$$ASSD = \frac{\sum_{i=1}^{N_s} d_s\left(i\right) + \sum_{i=1}^{N_r} d_r\left(i\right)}{N_s + N_r}$$

These are Normalized Discrepancy (ND), Hausdorff Distance (HD), Mean Distance (MD), Root Mean Square Distance (RMSD), and Average Symmetric Surface Distance (ASSD). Between these measure HD and MD are usually used in the literature. There are many other evaluation methods in the literature which is beyond scope of this tutorial. For further information please refer to [10-11].

Voting Approach

A reference standard has sometimes been formed by taking a combination of expert segmentations. For example, a voting rule as often used in practice selects all voxels where some majority of experts agree the structure to be segmented is present [12-13]. This simple approach unfortunately does not provide guidance as to how many experts should agree before the structure is considered to be present. Furthermore vote counting strategies treat each voter equally without regard to potential variability in quality or performance amongst the voters, and do not allow for a priori information regarding the structure being segmented to be incorporated. In the case of binary segmentations, a majority vote is at least unique in the case of an odd number of voters.

However, in the case of multicategory segmentations, the category with the most votes may not be unique and may not reflect the overall preferred choice of the voters. Preferential voting strategies, operating on votes or class probabilities, have been examined in the context of classifier fusion [14]. Examples include the Borda count [15] for preferential vote ordering, class probability combining strategies (the Product Rule and the Sum Rule which can be used to express the Min Rule, the Max Rule, the Median Rule and the Majority Vote Rule as described by [16]), and strategies which assume each classifier has expertise in a subset of the decision domain [16-19]. Similar issues in combining decisions from multiple raters or studies arise in content- based collaborative filtering [20] and in meta-analysis of diagnostic tests [21-23]. Estimating performance in the presence of an imperfect or limited reference standard has also been explored [22, 24].

STAPLE takes a collection of segmentations of an image, and computes simultaneously a probabilistic estimate of the true segmentation and a measure of the performance level represented by each segmentation. The source of each segmentation in the collection may be an appropriately trained human rater or raters, or it may be an automated segmentation algorithm. The algorithm is formulated as an instance of the expectation-maximization (EM) algorithm [25-26] and builds upon the earlier work [27-28]. Next, we describe different extensions of the STAPLE.

Algorithm Description and its variations

Binary STAPLE [29-30]

In the binary STAPLE the goal is to segment a structure in an image by indicating the presence or absence of the structure at each voxel when multiple expert segmentations are available. The algorithm takes a collection of segmentations of an image, and computes simultaneously a probabilistic estimate of the true segmentation and a measure of the performance level represented by each segmentation. The source of each segmentation in the collection may be an appropriately trained human rater or raters, or it may be an automated segmentation algorithm. The algorithm is formulated as an instance of the expectation-maximization (EM) algorithm.

In the formulation of our algorithm described here, the expert segmentation decision at each voxel is directly observable, the hidden true segmentation is a binary variable for each voxel, and the performance level, or quality, achieved by each segmentation is represented by sensitivity and specificity parameters.

STAPLE [31]

Generalization of the algorithm to unordered multicategory labels, suitable for when the hidden true segmentation at each voxel is one of an unordered finite set of labels is the STAPLE algorithm. This algorithm utilizes a global prior, a spatially varying prior, and a spatial homogeneity prior encoded with a Markov Random Field, and an efficient solution strategy.

Continuous STAPLE

Continuous STAPLE is an extension of STAPLE algorithm to estimate the reference standard together with bias and variance parameters for a set of vector images (it can be expert segmentations). This method also has a parameter comparison framework in order to compute hypothesis tests to detect statistically significant differences in the parameters [32] [33].

STAPLE MAP [34]

The algorithm extends STAPLE by incorporating prior probabilities for the expert performance parameters. This is achieved through a Maximum A Posteriori formulation, where the prior probabilities for the performance parameters are modeled by a set of beta distributions. Fusion from data with some structures not segmented by each expert should be carried out in a manner that accounts for the missing information. STAPLE MAP enables improved fusion with partial delineation. It also avoids convergence to undesirable local optima in the presence of strongly inconsistent segmentations, which can occur in the STAPLE algorithm. The symptom of this problem is convergence to a local optimum at which one or more labels, which may occur frequently in the input segmentations, are regarded as being unreliably segmented, and the consensus segmentation replaces these labels with an alternative label. The MAP prior can be used to reflect additional information that such an local optimum is undesirable, and to ensure convergence to a different local optimum.

Download and Installation

You can download the binaries or the source code for the software both Mac OS X, Linux and Windows operating systems. The software is designed primarily to operate as command line tool, with a graphical user interface for visualization and simple applications of STAPLE.

Linux/Unix

First visit our web site: http://crl.med.harvard.edu/software/STAPLE/request_access.php and complete the form. You will receive an email providing a script. This script will download, build and install the latest stable version of the CRKit, including STAPLE. In order to run the script you must have installed g++, gcc, make, svn , curl , gunzip. The script downloads CMake, ITK, VTK, Qt and CRKit.

Windows

If you want to use Windows system to build the CRKit, you need CMake, ITK, VTK, and Qt. Please consider that when you want to build the VTK, you should set the VTK_USE_GUISUPPORT and VTK_USE_QVTK options to 'ON'. In the next step you are ready to install Qt. To build Qt using Microsoft Visual Studio (MSVS) 2005 do the following steps [35]:

1- Add the qt-all-opensource-src-4.5.3/bin directory to PATH environment variable.
2- Open a command prompt through Start -> All Programs -> Microsoft Visual Studio 2005 -> Visual Studio Tools -> Visual Studio 2005 Command Prompt.
   - Navigate to the qt-all-opensource-src-4.5.3 directory and type configure -platform win32-msvc2005 -no-webkit -no-dbus -no-phonon -debug-and-release -opensource –shared
3- Agree to the license terms by typing 'y' and pressing return.
4- Qt will take a while to configure.
5- Type 'nmake' when the configuration is done and wait a long time for the compilation to happen.
6- That's it.

After building ITK, VTK, and QT, you can use cmake to build CRKit.

If you are using MSVS 2008 or MSVS 2010 you should add "-platform win32-msvc2008" as the option in the configuration step [36].

Command-line Utilities

There are useful command-line utilities in CRKit. Here, we introduce some of them by an example. Suppose that we want to segment a structure (say left caudate) in an intensity image using a multiple-atlas-based segmentation strategy. STAPLE as a fusion algorithm is an excellent candidate for this purpose. In addition, suppose that we have four atlases with corresponding manual segmentation of multiple structures including left caudate. Here are the steps to do the segmentation.

- Take a look at image information and orientation
- Rigid registration of atlases to the intensity image
- Affine registration of atlases to the intensity image using output transform of rigid registration as the initialization transform.
- Non-rigid registration of atlases to the intensity image using output transform of affine registration as the initialization transform.
- Applying output transformation of non-rigid registrations to the manual segmentation
- Extraction of left caudate from the registered manual segmentations
- Apply STAPLE for the fusion
- Evaluation of the results.
- Visualization

Image Information

For image registration, it is important that the image metadata providing orientation and voxel spacing information be present and correct. *crlImageInfo* provides information about the image intensity statistics and geometry. For example, using the example image provided in the apps/staple/data/prostate/mri directory, *crlImageStats prostatemri.mhd* displays the following information:

```
        $crlImageStats prostatemri.mhd
        ============ prostatemri.mhd ============
        Intensity properties:
        Minimum 0.0 Maximum 4482.0 Mean 234.0 Variance 90142.95 Sum 15341058.
Geometric properties:
        Size: [256, 256, 1]
        StartIndex: [0, 0, 0]
        Spacing: [0.468750000000000, 0.468750000000000, 1.000000000000000]
        Origin: [0.000000000000000, 0.000000000000000, 0.000000000000000]
        Direction:
        1.000000000000000 0.000000000000000 0.000000000000000
        0.000000000000000 1.000000000000000 0.000000000000000
        0.000000000000000 0.000000000000000 1.000000000000000
```

It can be helpful to ensure images have the same orientation. An image can be oriented using *crlOrientImage* as follows:

```
crlOrientImage InputImageName OutputImageName axial
```

If the metadata about the image is recorded incorrectly, *crlChangeImageInformation* can be used to alter the metadata. Usually, it is not a good idea to change the image information but it is sometimes unavoidable. To use this command a reference image should be set which is used to set the new information. For example, if the origin and direction of an image are not correct, the following form of the command can be used:

```
crlChangeImageInformation -i InputImageName -r RefImageName -x -d  -o OutputImageName
```

In addition, there is another useful command that can be used to convert between different file formats.

```
crlConvertBetweenFileFormats -in inputfilename -out outputfilename [-opct] outputPixelComponentType
```

where outputPixelComponentType is one of unsigned_char, char, unsigned_short, short, unsigned_int, int, unsigned_long, long, float, double.

Registration

As an example, suppose we have four images and we want to register them to the intensity image. We name them MovingImage1,… MovingImage4. Also, we name their corresponding segmentations as MovingSegmentation1, … MovingSegmentation4. In addition, suppose that we name the intensity image that we want to segment as TargetImage. First we apply rigid registration. For this reason, crlRigidRegistration may be used as follows (where *i* varies between 1 and 4):

```
crlRigidRegistration TargetImage MovingImage${i} RigidImage${i} RigidTransform${i}
```

Next, we use crlAffineRegistration in the following form:

```
crlAffineRegistration TargetImage MovingImage${i} AffineImage${i} AffineTransform${i}
RigidTransform${i}
```

Finally, we use crlNonrigidBSplinePyramid

```
crlNonrigidBSplinePyramid TargetImage  MovingImage${i}  OutputImage${i} 30 1 .0005 OutputTransform 3 32
1000000 500 OutputFieldImage${i} AffineTransform${i}
```

Next, we apply the output field image to the manual segmentations using crlDeformScalarImage to extract labelmaps.

```
crlDeformScalarImage MovingSegmenation${i} OutputFieldImage${i} OutputSegmenation${i}  nearest
```

In the next step, we can use STAPLE for the fusion. However, we are just interested in left caudate. Thus, we first create binary images from the registered manual segmentations. To this end, we use crlBinaryThreshold as follows:

```
crlBinaryThreshold OutputSegmenation${i}  ${i}.nrrd  lowthreshold hithreshold 1 0
```

In this case values equal to and between lowthreshold and hithreshold are set to 1 and values outside the thresholds are set to 0.

STAPLE

Now, everything is ready to use STAPLE. crlSTAPLE is the command that is used for this goal. *crlSTAPLE* has many parameters that may be set, but the default parameters are often good choices. It will create a probabilistic output soft segmentation file called Weights.nrrd, when used as follows:

```
crlSTAPLE  -o Weights.nrrd 1.nrrd 2.nrrd 3.nrrd 4.nrrd
```

The output is 4D image soft segmentation where fourth dimension indicates each label. For our example, there are two labels: 1 which shows the left caudate, and 0 which indicates the background.

A maximum likelihood segmentation can be estimated for each voxel, by selecting the label with the highest probability at each voxel. To do this, the command *crlIndexOfMaxComponent*, can be used in the following form:

```
crlIndexOfMaxComponent Weights.nrrd Seg.nrrd
```

Now, we can compare the results with manual segmentation. Here just DICE is used for comparison. crlOverlapstats3d is the command that we use for this purpose.

```
crlOverlapstats3d 1.nrrd REF.nrrd 1
intersection 1928 union 2651 jaccard 0.727273 dice 0.842105 im1background 8386259 im1foreground 2349
im2background 8386378 im2foreground 2230
```

```
crlOverlapstats3d 2.nrrd REF.nrrd 1
intersection 1889 union 2780 jaccard 0.679496 dice 0.809167 im1background 8386169 im1foreground 2439 im2background 8386378 im2foreground 2230
crlOverlapstats3d 3.nrrd REF.nrrd 1
intersection 1603 union 2480 jaccard 0.646371 dice 0.785207 im1background 8386755 im1foreground 1853 im2background 8386378 im2foreground 2230
crlOverlapstats3d 4.nrrd REF.nrrd 1
intersection 1886 union 2548 jaccard 0.740188 dice 0.850699 im1background 8386404 im1foreground 2204 im2background 8386378 im2foreground 2230
crlOverlapstats3d Seg.nrrd REF.nrrd 1
intersection 2054 union 2623 jaccard 0.783073 dice 0.878341 im1background 8386161 im1foreground 2447 im2background 8386378 im2foreground 2230
```

Based on the results it can be seen that the quality of the segmentation is better than each one of the experts (automatic registrations). If you want to segment structures with huge number of voxels the performance of the all of the raters are very close. To have better performance measurement sometimes it is better to do not consider the voxels that all of the raters have the same decision. To do this you can use the option –assignConsensusVoxels in the following format:

```
crlSTAPLE --assignConsensusVoxels  -o OutputImage ListofInputImages
```

In this way you simply assign consensus voxels directly.

Continuous STAPLE

The same procedure can be done using Continuous STAPLE. However, first the binary images should be converted to distance function. crlSignedDanielssonDistanceMap3d is the command that can be used to find the distance function from the binary image.

```
crlSignedDanielssonDistanceMap3d ${i}.nrrd d${i}.nrrd outputVorono${i} outputVectorMap${i}
```

Now a file containing a list of images d${i}.nrrd is created (imagelist.txt). Finally, crlContinuousSTAPLE can be used in the following form:

```
crlContinuousSTAPLE  -p number-of-threads –o OutputFileName  -l imagelist.txt
```

In addition, following command lines are useful:

```
crlImageAlgebra image1 operator image2 outimage
```

It computes outimage = image1 operator image2 where valid operators are add, subtract, multiply, divide, difference, and squareddifference Also, crlImageAddMultiplyAdd which can be employed in the following form:

```
crlImageAddMultiplyAdd inmage add1 multiply add2 outimage
```

It calculates voxelwise outimage = (inimage + add1)*mult1 + add2. To create triangle models crlCreateTriangleModel can be utilized using the following format.

```
crlCreateTriangleModel segmentedImageFile threshold triangleFileName.vtk
```

For the above example the following command can be used:

```
crlCreateTriangleModel ${i} 1 triangle${i}.vtk
```

Visualization

crlViz is an easy to use graphical user interface (GUI) that can be utilized to view and compare images, segmenations, probabilities, shapes, …. In addition, there is an implementation for the STAPLE which enables to run it using default options from the GUI. The general view of the crlViz is shown in the Figure 1. You can load intensity image, probability map, segmented volume, fMRI activation map, tensor volume, RGB tensor volume, and triangle model. Suppose that we want to apply STAPLE in the same manner described before. For this purpose, first, in the file menu we choose Add Segmented Volume (Figure 2).

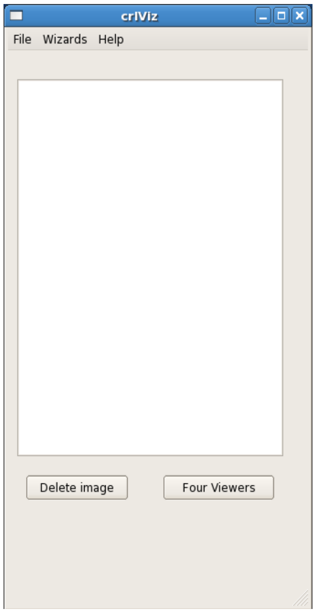


Figure 1. General view of crlViz.

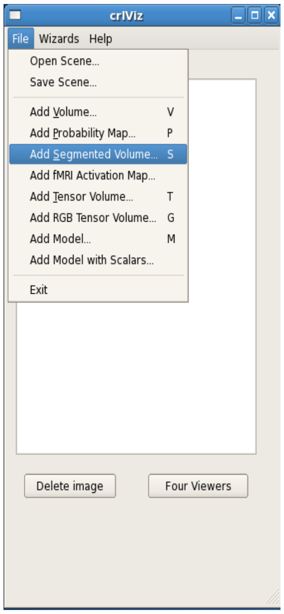
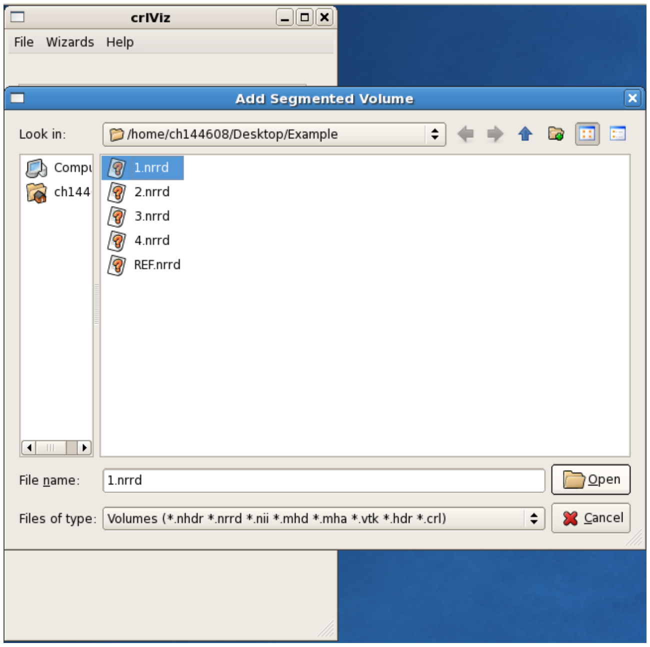


Figure 2. To add a segmentation volume, from the file menu we choose Add Segmented Volume.

Then, we find the segmentation images on the disk and we load them to the crlViz (Figure 3). In the next step, we choose STAPLE from Wizards menu (Figure 4). This is the wizard that can be used to apply STAPLE based on its default setting (Figure 5). In the next step, we drag input files from the crlViz main window and drop them to STAPLE Wizard window (Figure 6).



Look in: /home/ch144608/Desktop/Example

File name: 1.nrrd

Files of type: Volumes (*.nhdr *.nrrd *.nii *.mhd *.mha *.vtk *.hdr *.crl)

Figure 3. Locate the segmentation volumes in the disk and load them to the crlViz.

There is couple of outputs in the STAPLE algorithm. In this step, we set their names and location on disk. They are weights and maximum component file. The weights file is a 4D image which indicates soft segmentation and maximum component is the hard segmentation (Figure 7-Figure 9).

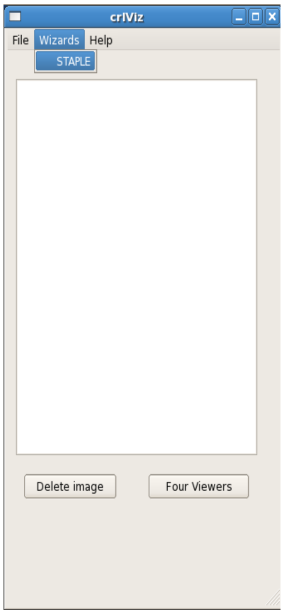
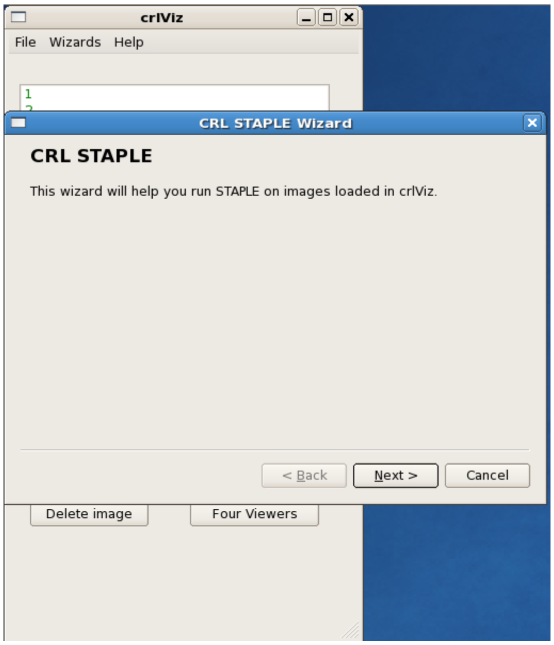


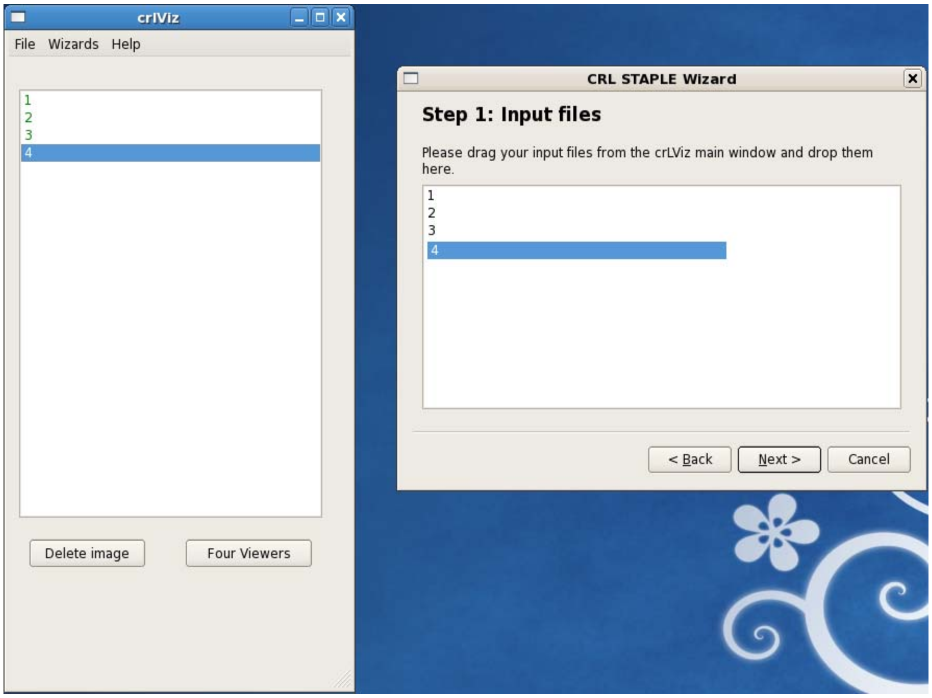
Figure 4. To use STAPLE algorithm, Choose STAPLE from Wizards menu.



CRL STAPLE

This wizard will help you run STAPLE on images loaded in crlViz.

Figure 5. Wizard to apply STAPLE using its default setting.

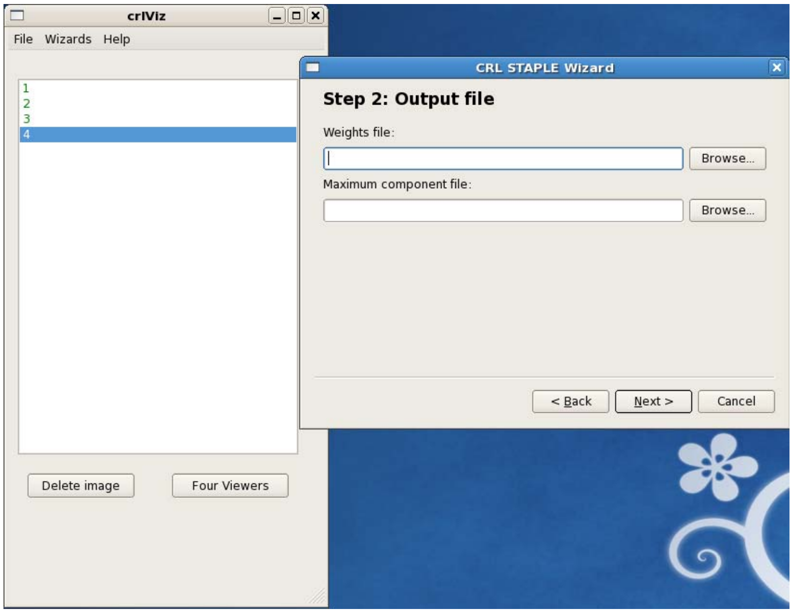


Step 1: Input files

Please drag your input files from the crLViz main window and drop them here.

1
2
3
4

Figure 6. Drag input files from the crlViz main window and drop them to STAPLE Wizard window.

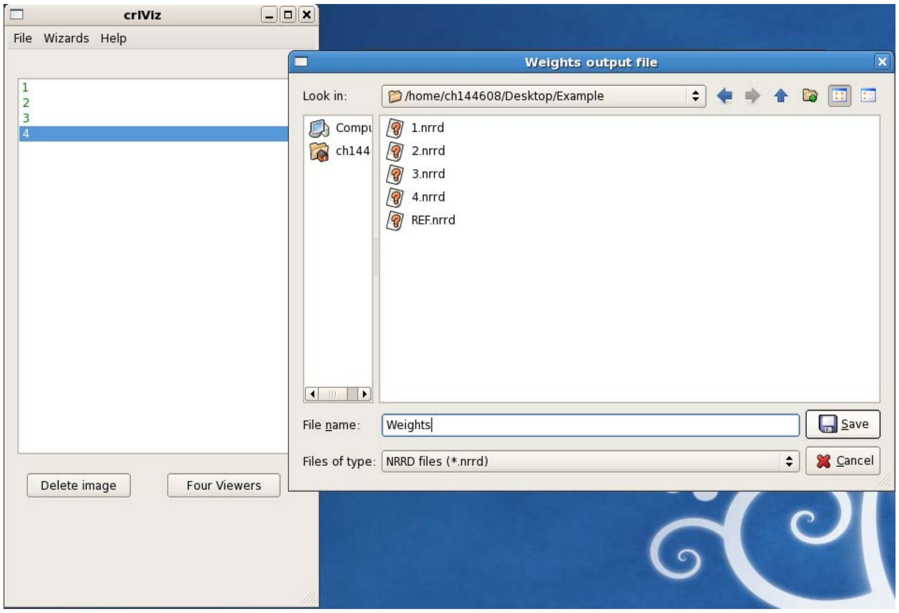


Step 2: Output file

Weights file:

Maximum component file:

Figure 7. Choose names for the Weights output files.

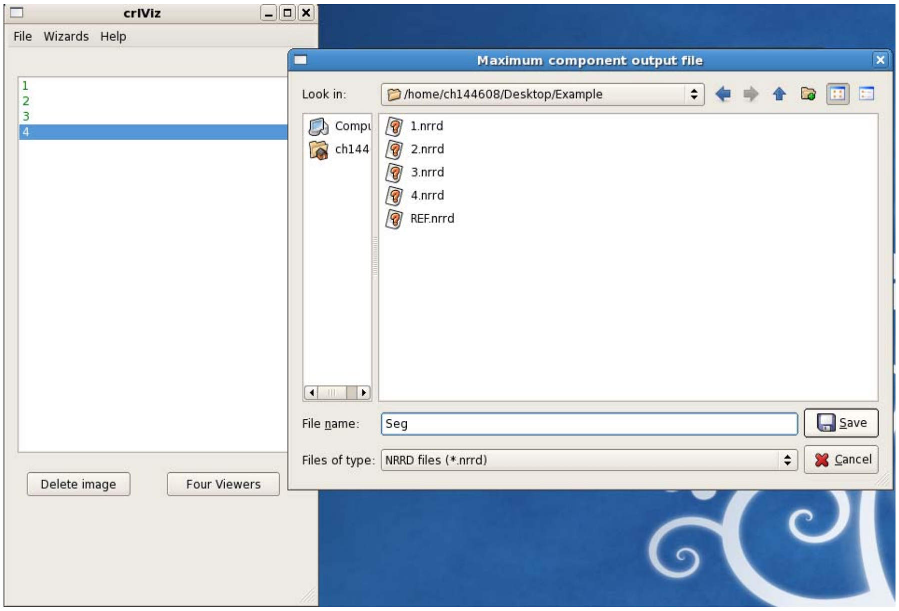


Look in: /home/ch144608/Desktop/Example

File name: Weights

Files of type: NRRD files (*.nrrd)

Figure 8. Choose names for the Weights.



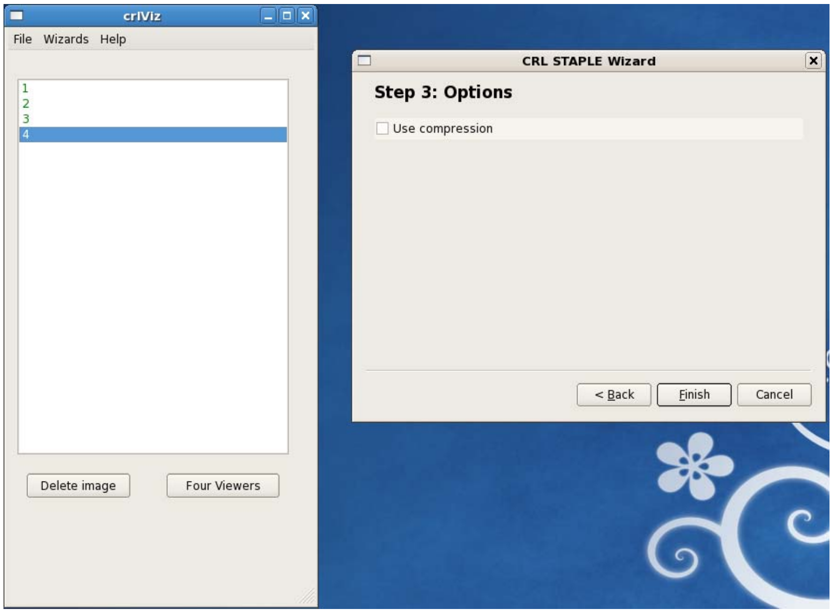
Look in: /home/ch144608/Desktop/Example

File name: Seg

Files of type: NRRD files (*.nrrd)

Figure 9. Choose names for the Maximum Component files.

Usually, the size of the weights image is too large and if you have disk space limitation, using compression option is a smart choice (Figure 10). Next, we just run the algorithm to complete the process.



Step 3: Options

□ Use compression

Figure 10. If there is a disk space limitation, Use compression is applicable.

Usually, we want to see the segmentation result on the intensity image. To this end, we first load the intensity image using Add volume option in the file menu (Figure 11). After loading the intensity image, you can view the image in the new window by clicking on it (Figure 12). In addition, you can manage the viewer using options in view menu (Figure 13). When the cursor is active, you can choose the points by clicking on the image. In addition, you can save, load, and remove the points (Figure 14). Also, you can change interactor from 2D to 3D (Figure 15) which is sometimes useful to have a sense about the location of structure of interest. Again you can use view menu to show/hide Marker, Bounding Box, Orientation Cube, and Small cube (Figure 16).

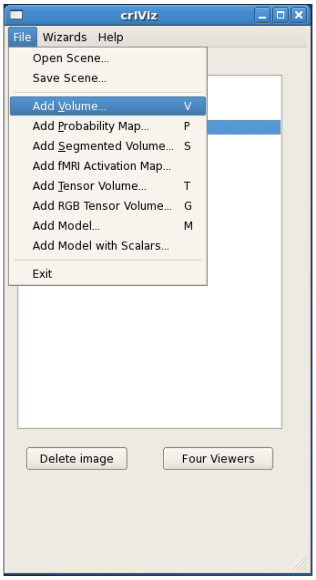
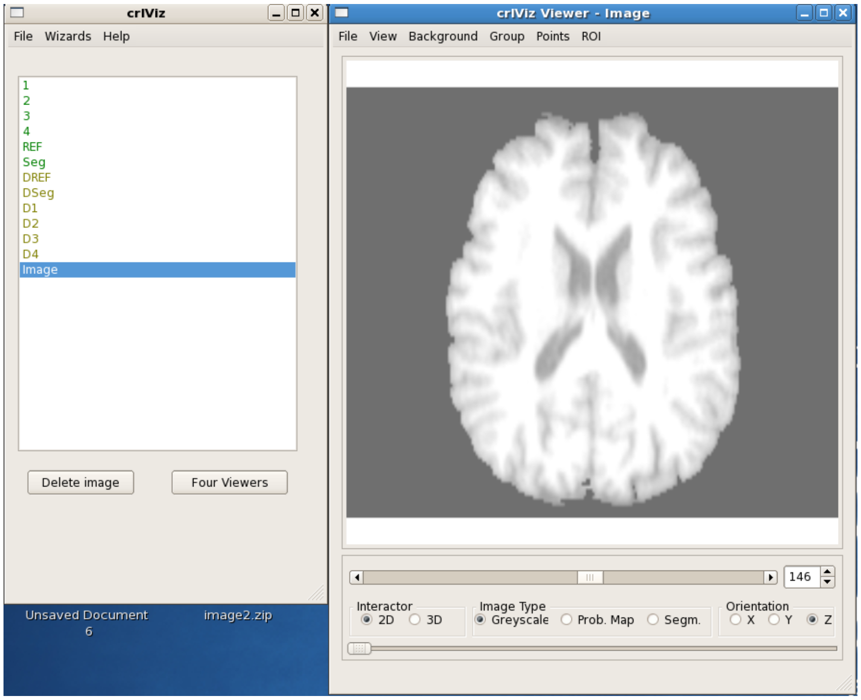
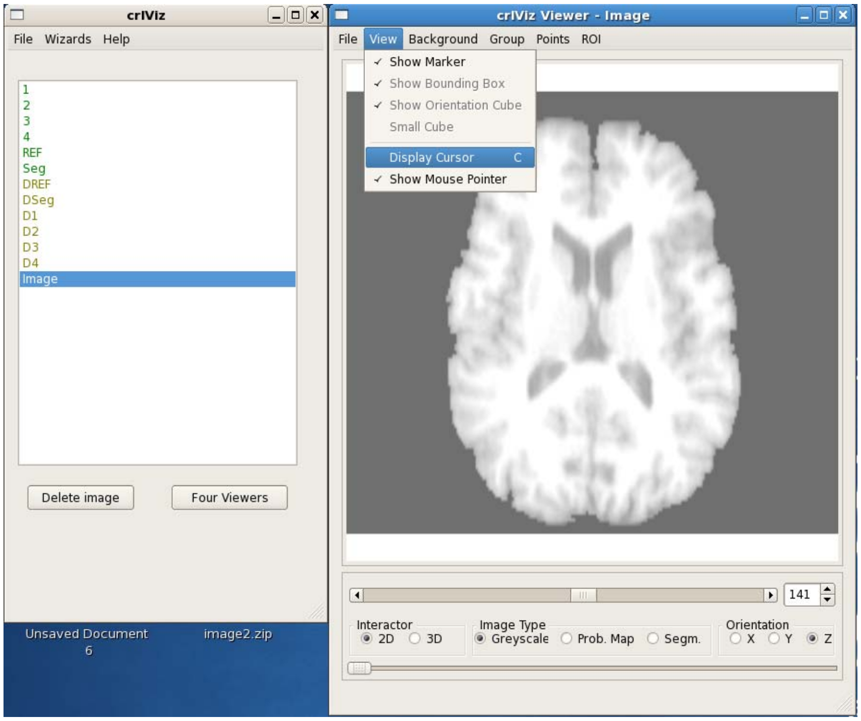


Figure 11. Load the intensity image using Add Volume option in File menu.



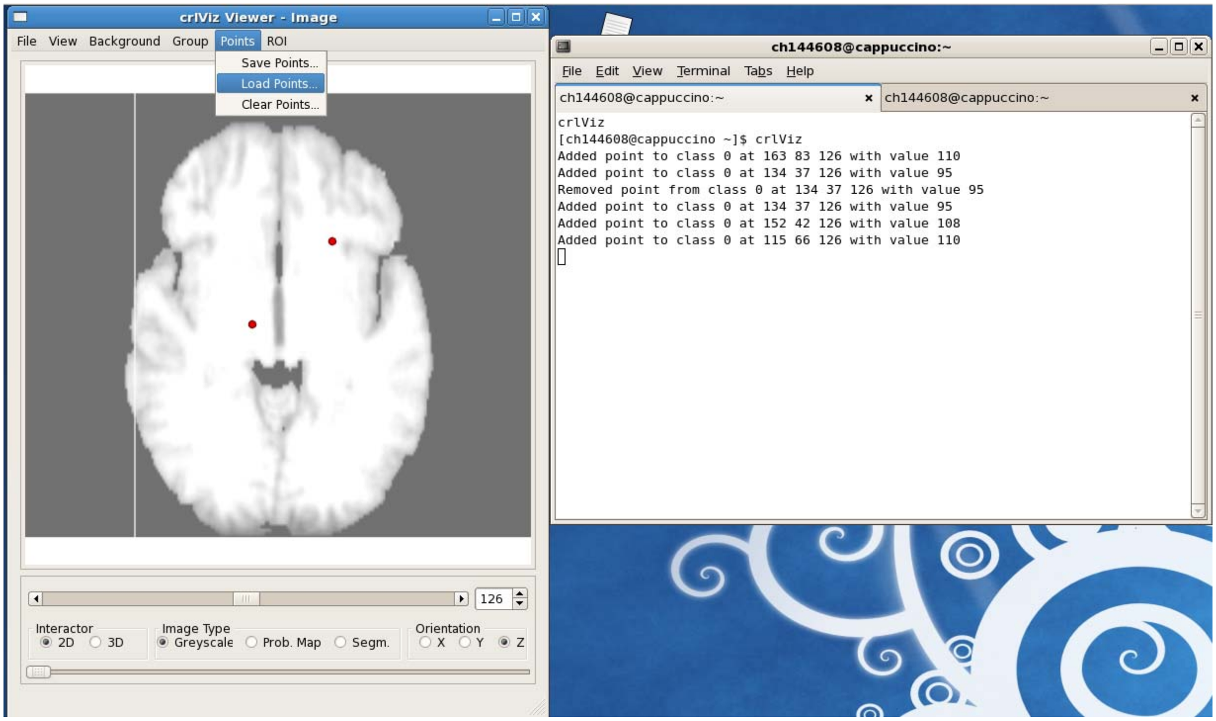
Interactor ● 2D ○ 3D
Image Type ● Greyscale ○ Prob. Map ○ Segm.
Orientation ○ X ○ Y ● Z

Figure 12. Intensity image can be loaded in the new window by clicking on it in the main window.



Interactor ● 2D ○ 3D
Image Type ● Greyscale ○ Prob. Map ○ Segm.
Orientation ○ X ○ Y ● Z

Figure 13. Options for the viewer window in the View menu.



```
crlViz
[ch144608@cappuccino ~]$ crlViz
Added point to class 0 at 163 83 126 with value 110
Added point to class 0 at 134 37 126 with value 95
Removed point from class 0 at 134 37 126 with value 95
Added point to class 0 at 134 37 126 with value 95
Added point to class 0 at 152 42 126 with value 108
Added point to class 0 at 115 66 126 with value 110
```

Figure 14. Adding points to the image when cursor is active. The points can be saved, loaded, and cleared.

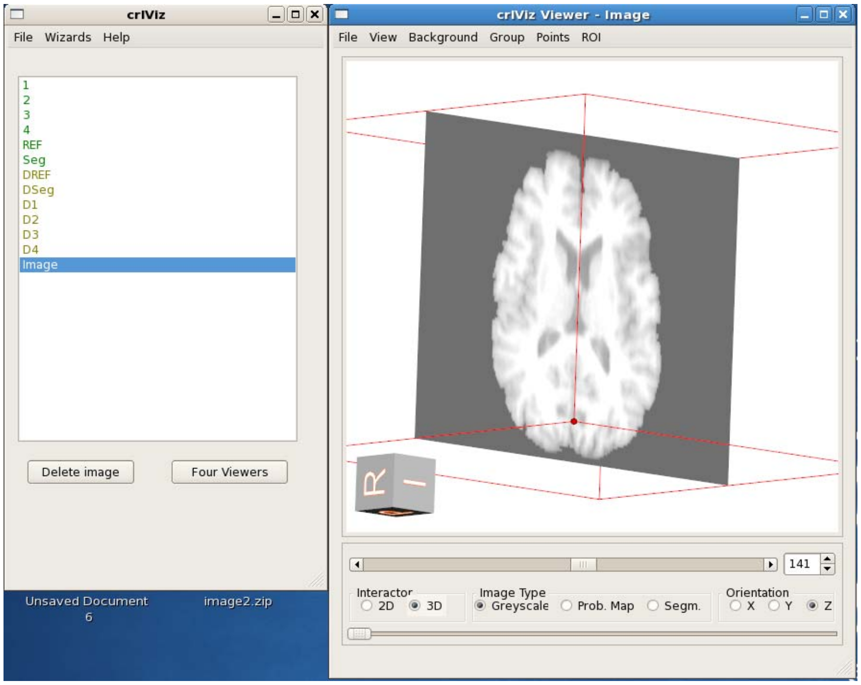
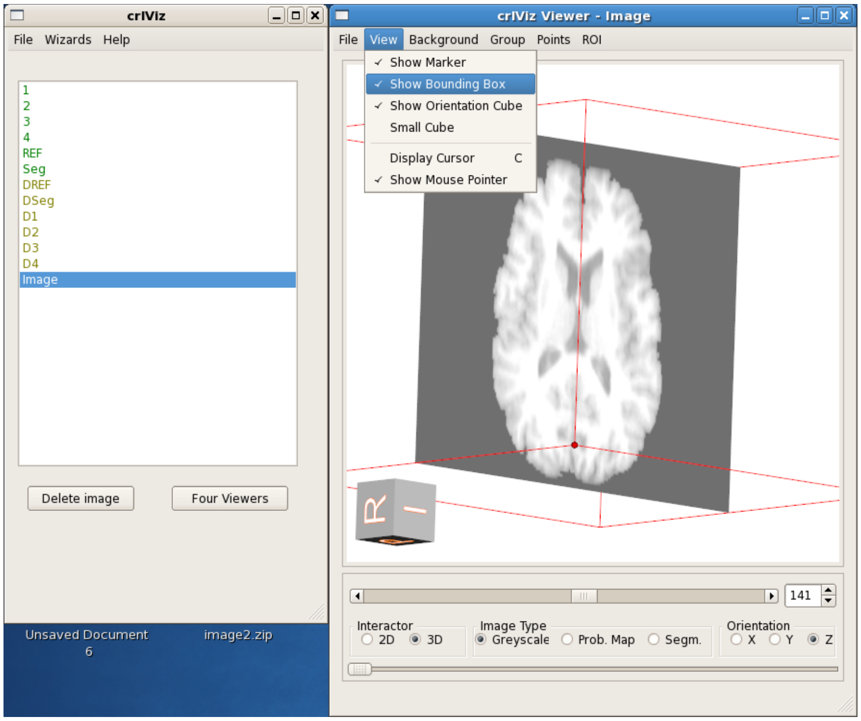


Figure 15. 3D view of the intensity image by changing the interactor from 2D to 3D.



141

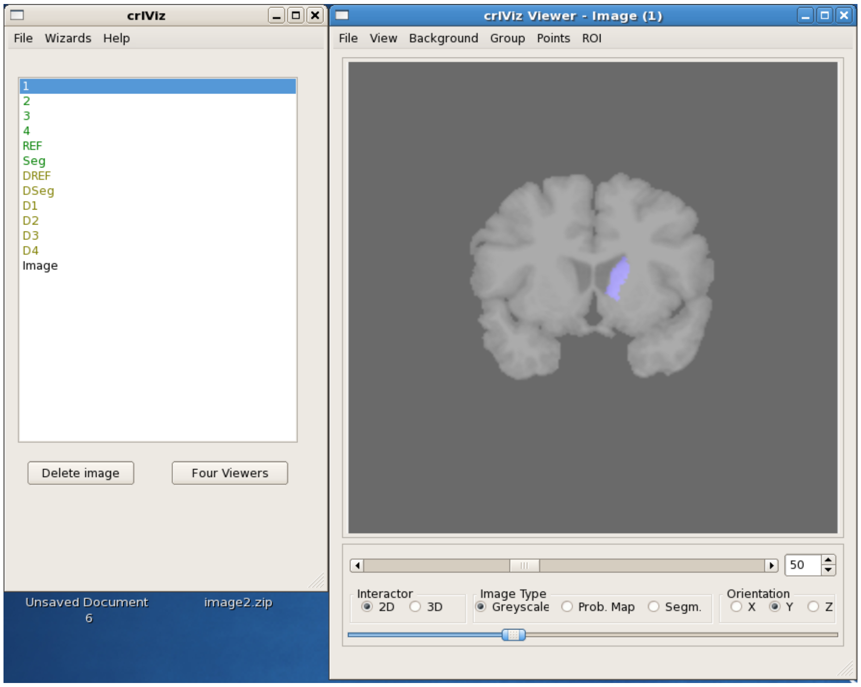
Interactor ○ 2D ● 3D

Image Type ● Greyscale ○ Prob. Map ○ Segm.

Orientation ○ X ○ Y ● Z

Figure 16. Show/hide Marker, Bounding Box, Orientation Cube, and Small cube.

To overlay images, you can open one image in a viewer window and drag the second image to the window (Figure 17). Also, you can use cursor to view intensity of the two overlaid images (Figure 18). In addition, it is possible to group multiple images using group menu (Figure 19).



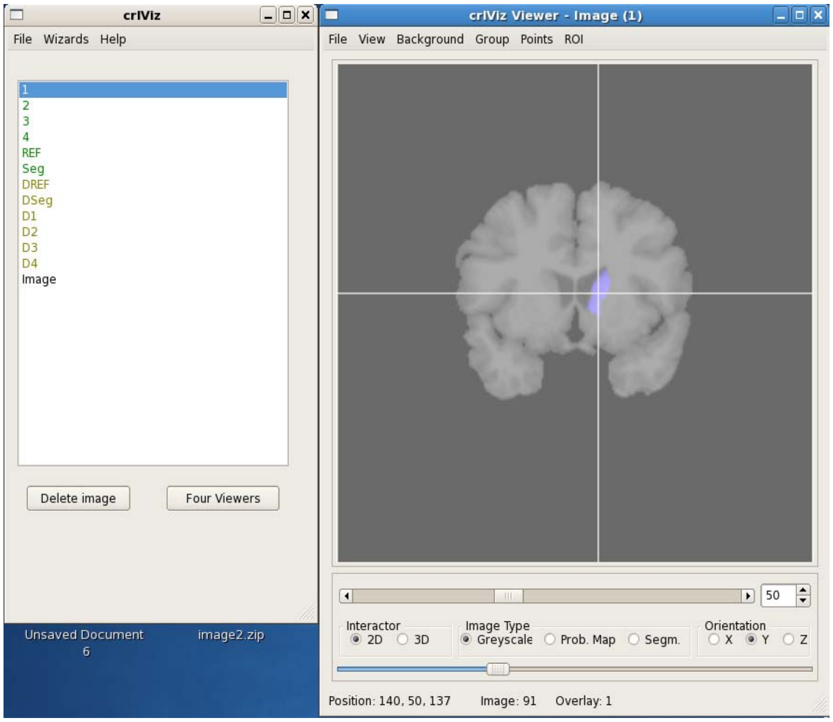
50

Interactor ● 2D ○ 3D

Image Type ● Greyscale ○ Prob. Map ○ Segm.

Orientation ○ X ● Y ○ Z

Figure 17. It is possible to overlay the images by dragging them to the viewer window and to change their opacity using the bar designed for this purpose.



50

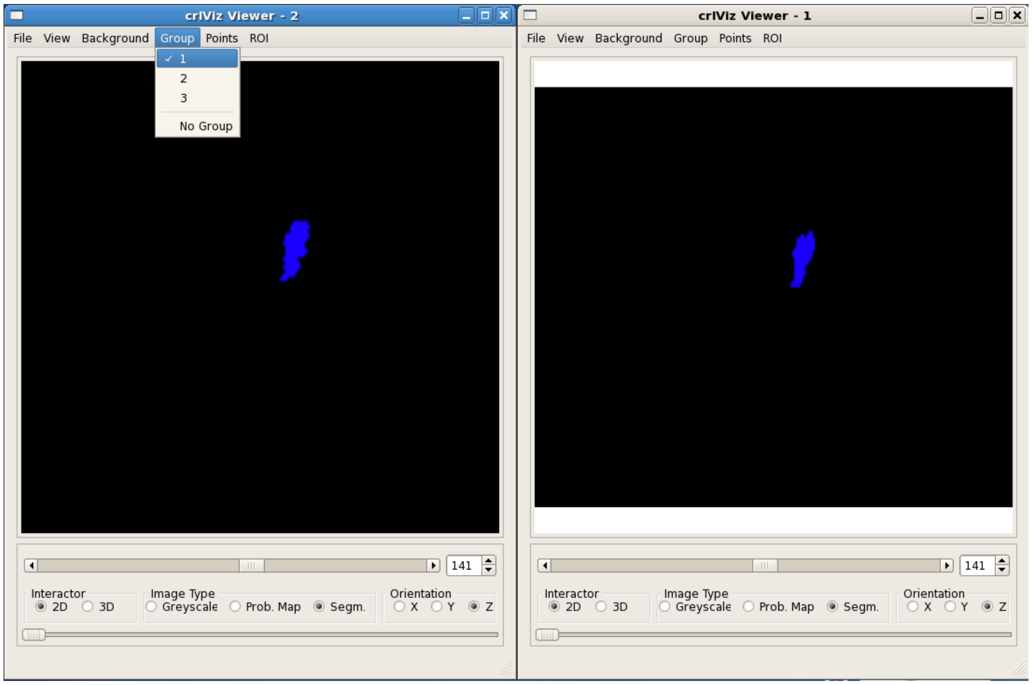
Interactor ● 2D ○ 3D

Image Type ● Greyscale ○ Prob. Map ○ Segm.

Orientation ○ X ● Y ○ Z

Position: 140, 50, 137 Image: 91 Overlay: 1

Figure 18. It is possible to use cursor to view intensity of the two overlaid images.


crlViz Viewer - 2
File View Background Group Points ROI
✓ 1
2
3
No Group
crlViz Viewer - 1
File View Background Group Points ROI

crlViz Viewer - 2:

141

Interactor ● 2D ○ 3D

Image Type ○ Greyscale ○ Prob. Map ● Segm.

Orientation ○ X ○ Y ● Z

crlViz Viewer - 1:

141

Interactor ● 2D ○ 3D

Image Type ○ Greyscale ○ Prob. Map ● Segm.

Orientation ○ X ○ Y ● Z

Figure 19. Synchronizing multiple images using group menu.

To view the triangle models, we can use Add Model option in the File menu (Figure 20). To view the 3D model in different angles interactor should be changed to the 3D (Figure 21).

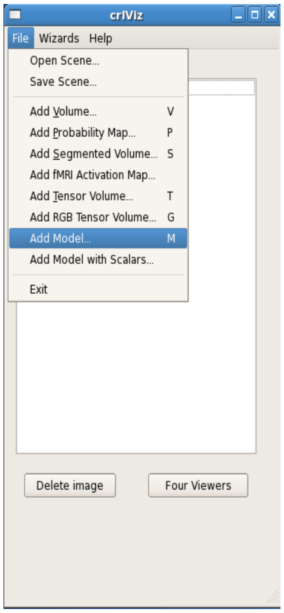
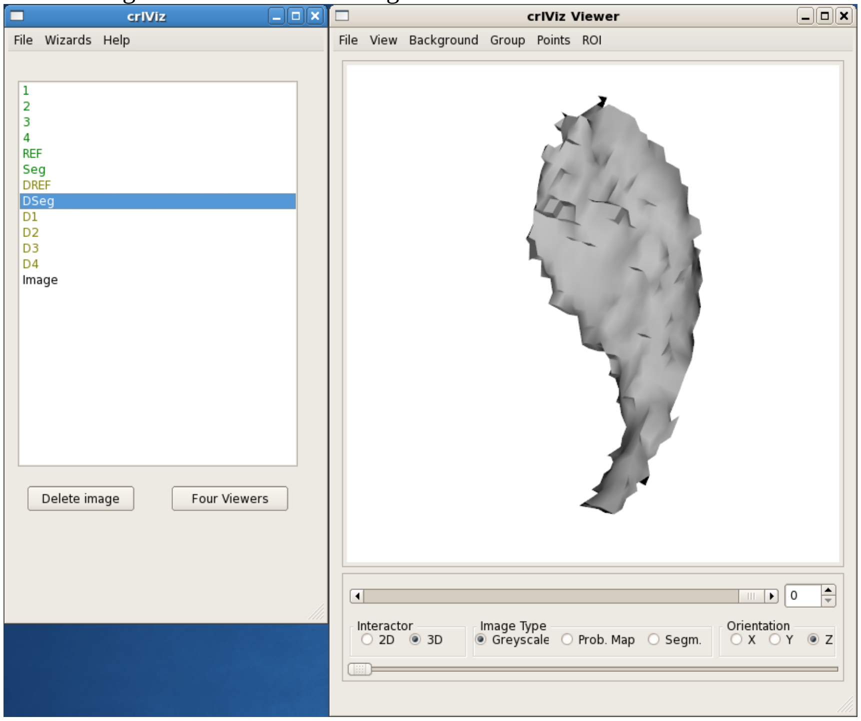


Figure 20. To add the triangle model Add Model can be used.



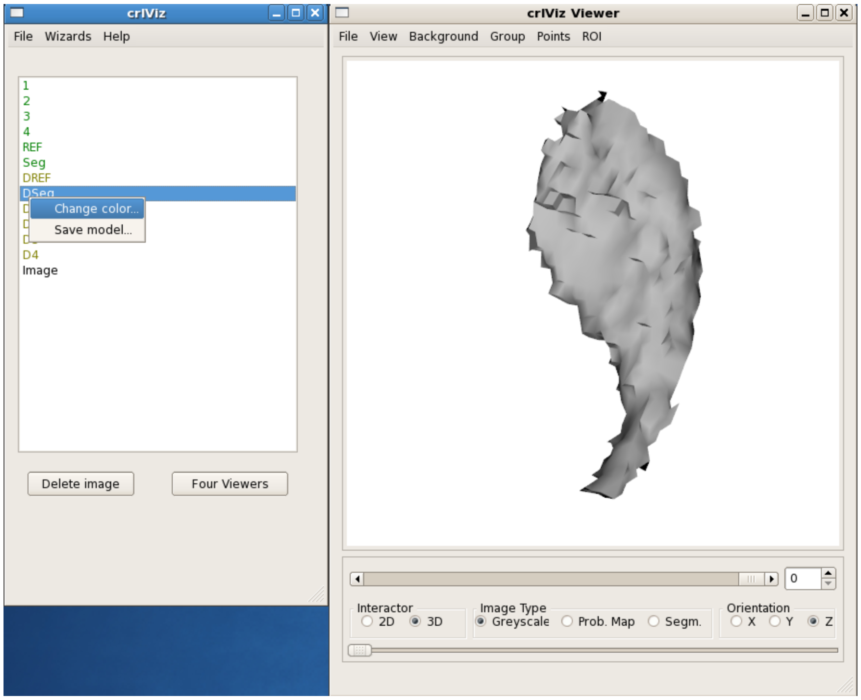
Interactor ○ 2D ● 3D

Image Type ● Greyscale ○ Prob. Map ○ Segm.

Orientation ○ X ○ Y ● Z

Figure 21. To view the 3D model in different views the intreactor should be changed to 3D.

By right clicking on the model in the main menu (Figure 22) we can change the color (Figure 23). In addition, to change the opacity of the model you can do shift+right click on the model in the viewer menu (Figure 24-Figure 25). In addition, you can drag and drop another model to the viewer window to compare to 3d models (Figure 26).

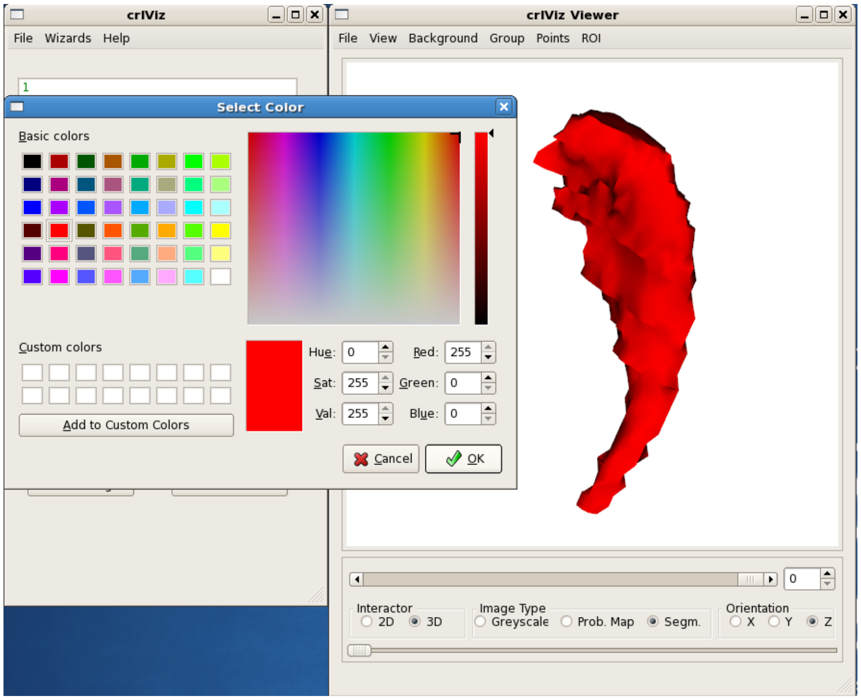


Interactor: ○ 2D ● 3D

Image Type: ● Greyscale ○ Prob. Map ○ Segm.

Orientation: ○ X ○ Y ● Z

Figure 22. Right clicking on the model in the main menu gives options for the changing the color and saving the model.



Hue: 0

Sat: 255

Val: 255

Red: 255

Green: 0

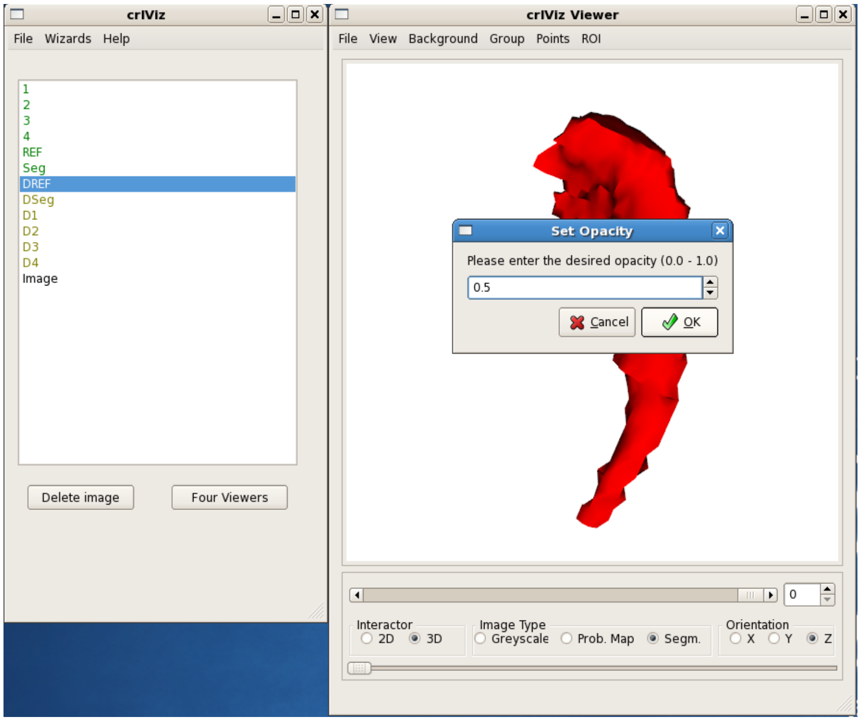
Blue: 0

Interactor: ○ 2D ● 3D

Image Type: ○ Greyscale ○ Prob. Map ● Segm.

Orientation: ○ X ○ Y ● Z

Figure 23. Change of the color of the model.



Please enter the desired opacity (0.0 - 1.0)

0.5

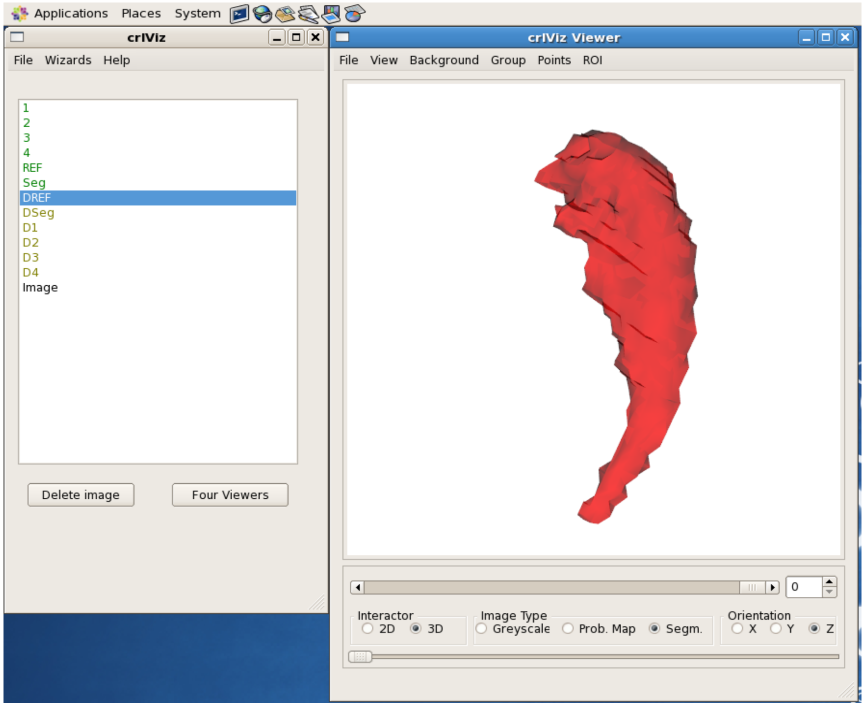
0

Interactor ○ 2D ● 3D

Image Type ○ Greyscale ○ Prob. Map ● Segm.

Orientation ○ X ○ Y ● Z

Figure 24. By shift+Right Clicking on the model in the viewer window the opacity of the model can be changed.



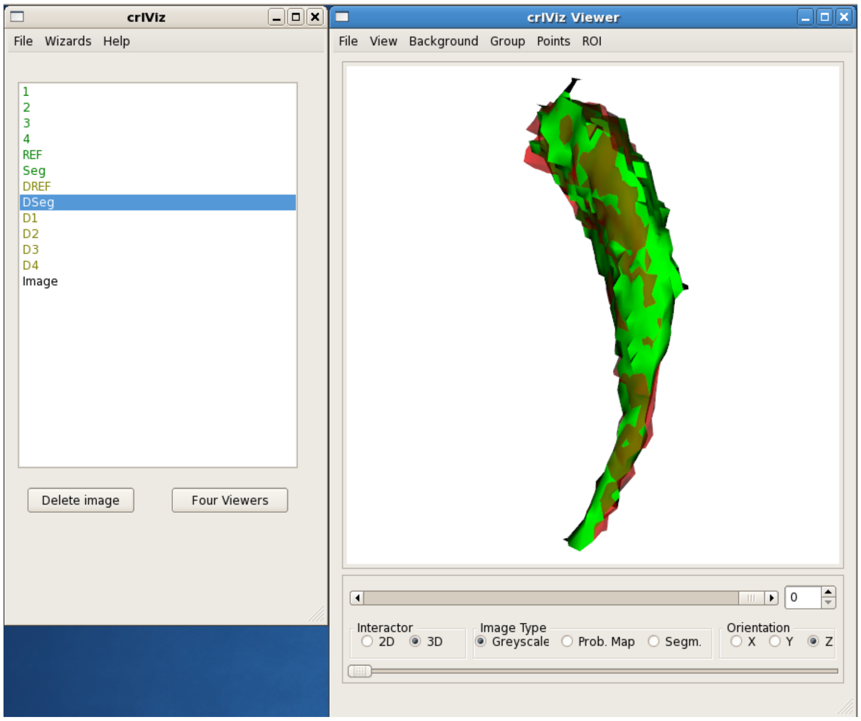
0

Interactor ○ 2D ● 3D

Image Type ○ Greyscale ○ Prob. Map ● Segm.

Orientation ○ X ○ Y ● Z

Figure 25. Opacity of the model changed to the 0.5.

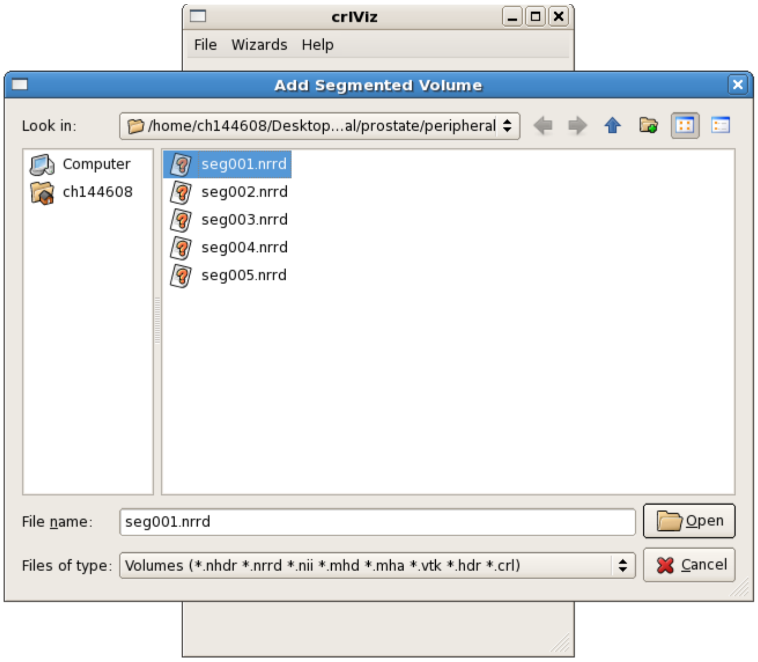


Interactor ○ 2D ● 3D
Image Type ● Greyscale ○ Prob. Map ○ Segm.
Orientation ○ X ○ Y ● Z

Figure 26. Overlay of two 3d models by dragging the second model to the viewer window.

An Example from [31]

Finally, we use the wizard to regenerate the result of the figure 7 in [31]. To this purpose first, we load the five images in apps/staple/data/prostate/peripheral to the crlViz (Figure 27).



Look in: /home/ch144608/Desktop...al/prostate/peripheral

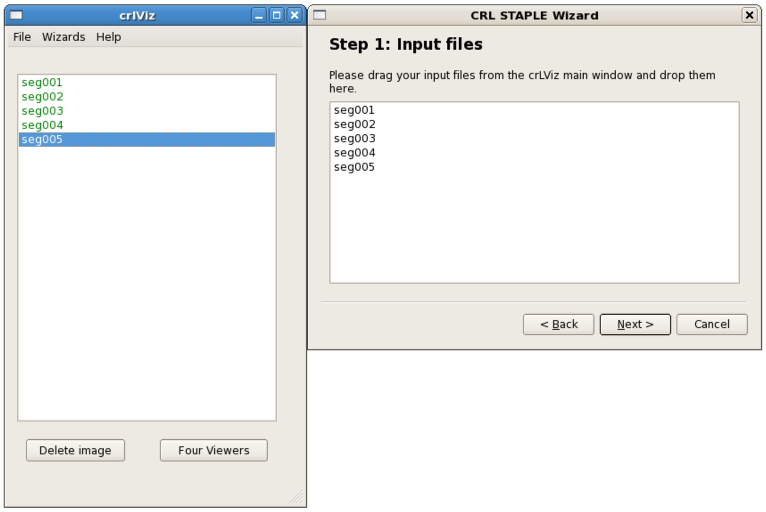
seg001.nrrd
seg002.nrrd
seg003.nrrd
seg004.nrrd
seg005.nrrd

File name: seg001.nrrd

Files of type: Volumes (*.nhdr *.nrrd *.nii *.mhd *.mha *.vtk *.hdr *.crl)

Figure 27. Load the images from apps/staple/data/prostate/peripheral.

Next, choose STAPLE from the Wizards, and drag and drop the images from the crlViz main window (Figure 28).



seg001
seg002
seg003
seg004
seg005

Step 1: Input files

Please drag your input files from the crlViz main window and drop them here.

seg001
seg002
seg003
seg004
seg005

Figure 28. Load drag and drop the images from the crlViz main window.

Then set the weights and maximum component files and click Next. Finally, if you need to compress the result check the compress and click Finish. The maximum component image is the STAPLE truth estimate (figure 7.d of [31]). Figure 29 shows the probability of the prostate peripheral zone overlaid on the MRI. In addition, Table 1 shows the positive predictive value for each input segmentation computed from the converged result.

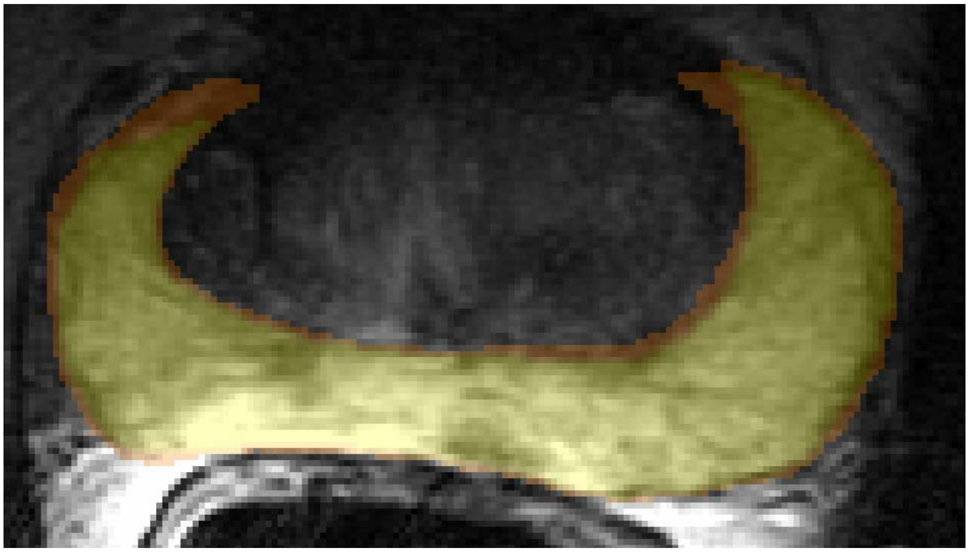


Figure 29. STAPLE truth estimate based on the example III.C of [31].

Table 1. Raters performances based on STAPLE estimated ground truth using PPV.

Rater	1	2	3	4	5
PPV	0.987	0.927	0.991	0.996	0.991

Acknowledgements

This work was supported in part by NIH grants R03 EB008680, R01 EB008015, R01 RR021885, and R01 LM010033.

References

[1] D. V. Iosifescu, M. E. Shenton, S. K. Warfield, R. Kikinis, J. Dengler, F. A. Jolesz, and R. W. McCarley, "An Automated Registration Algorithm for Measuring MRI Subcortical Brain Structures," *NeuroImage,* vol. 6, pp. 12-25, 1997.

[2] S. K. Warfield, R. V. Mulkern, C. S. Winalski, F. A. Jolesz, and R. Kikinis, " An Image Processing Strategy for the Quantification and Visualization of Exercise Induced Muscle MRI Signal Enhancement " *J Magn Reson Imaging,* vol. 11, pp. 525-531, May 2000.

[3] J. M. Bland and D. G. Altman, "Applying the right statistics: analyses of measurement studies," *Ultrasound Obstet Gynecol,* vol. 22, pp. 85-93, 2003.

[4] D. Levin, X. Hu, K. Tan, and S. Galhotra, "Surface of the brain: three-dimensional MR images created with volume rendering," *Radiology,* vol. 171, pp. 277-280, 1989.

[5] M. Everingham, H. Muller, and B. Thomas, "Evaluating Image Segmentation Algorithms Using the Pareto Front," in *ECCV* vol. LNCS 2353, ed, 2002, pp. 34-48.

[6] K. H. Zou, S. K. Warfield, J. R. Fielding, C. M. C. Tempany, W. M. Wells, M. R. Kaus, F. A. Jolesz, and R. Kikinis, "Statistical Validation Based on Parametric Receiver Operating Characteristic Analysis of Continuous Classification Data," *Acad Radiol,* vol. 10, pp. 1359-1368, 2003.

[7] F. Bello and A. C. F. Colchester, "Measuring Global and Local Spatial Correspondence Using Information Theory," in *MICCAI 98: First International Conference on Medical Image Computing and Computer-Assisted Intervention,* ed. Boston, 1998, pp. 964-973.

[8] A. P. Zijdenbos, B. M. Dawant, R. A. Margolin, and A. C. Palmer, "Morphometric Analysis of White Matter Lesions in MR Images: Method and Validation," *IEEE Transactions on Medical Imaging,* vol. 13, pp. 716-724, December 1994.

[9] A. Ortiz and G. Oliver, "On the use of the overlapping area matrix for image segmentation evaluation: A survey and new performance measures," *Pattern Recognition Letters,* vol. 27, pp. 1916-1926, Dec 2006.

[10] H. Zhang, J. E. Fritts, and S. A. Goldman, "Image segmentation evaluation: A survey of unsupervised methods," *Computer Vision and Image Understanding,* vol. 110, pp. 260-280, 2008.

[11] R. Cardenes, R. de Luis-Garcia, and M. Bach-Cuadra, "A multidimensional segmentation evaluation for medical image data," *Computer Methods and Programs in Biomedicine,* vol. 96, pp. 108-124, 2009.

[12] R. Kikinis, M. E. Shenton, G. Gerig, J. Martin, M. Anderson, D. Metcalf, C. R. G. Guttmann, R. W. McCarley, W. Lorensen, H. Cline, and F. A. Jolesz, "Routine Quantitative-Analysis of Brain and Cerebrospinal-Fluid Spaces With MR Imaging," *Jmri-Journal of Magnetic Resonance Imaging,* vol. 2, pp. 619-629, Nov-Dec 1992.

[13] S. Warfield, J. Dengler, J. Zaers, C. R. G. Guttmann, W. M. Wells III, G. J. Ettinger, J. Hiller, and R. Kikinis, "Automatic Identification of Grey Matter Structures from MRI to Improve the Segmentation of White Matter Lesions," *J Image Guid Surg,* vol. 1, pp. 326-338, 1995.

[14] L. P. Cordella, P. Foggia, C. Sansone, F. Tortorella, and M. Vento, "Reliability Parameters to Improve Combination Strategies in Multi-Expert Systems," *Pattern Analysis and Applications,* vol. 2, pp. 205-214, 1999.

[15] T. K. Ho, J. H. Hull, and S. N. Srihari, "Decision Combination in Multiple Classifier Systems," *IEEE Trans Pattern Analysis Machine Intelligence,* vol. 16, pp. 66-75, 1994.

[16] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On Combining Classifiers," *IEEE Trans Pattern Analysis Machine Intelligence,* vol. 20, pp. 226-239, 1998.

[17] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive Mixtures of Local Experts," *Neural Computation,* vol. 3, pp. 79-87, 1991.

[18] M. I. Jordan and R. A. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm," 1993.

[19] D. Windridge and J. Kittler, "A Morphologically Optimal Strategy for Classifier Combination: Multiple Expert Fusion as a Tomographic Process," *IEEE Trans Pattern Analysis Machine Intelligence*, vol. 25, pp. 343-353, 2003.

[20] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-Boosted Collaborative Filtering for Improved Recommendations," in *Proceedings of the Eighteenth National Conference on Artificial Intelligence* vol. AAAI-2002, ed, 2002, pp. 187-192.

[21] L. Irwig, P. Macaskill, P. Glasziou, and M. Fahey, "Meta-analytic methods for diagnostic test accuracy," *Journal of Clinical Epidemiology*, vol. 48, pp. 119-130, 1995.

[22] T. A. Alonzo and M. S. Pepe, "Using a Combination of Reference Tests to Assess the Accuracy of a new Diagnostic Test," *Statistics in Medicine*, vol. 18, pp. 2987-3003, 1999.

[23] S.-L. Normand, "Tutorial in Biostatistics Meta-Analysis: Formulating, Evaluating, Combining, and Reporting," *Statistics in Medicine*, vol. 18, pp. 321-359, 1999.

[24] S. V. B. SV, G. Campbell, K. L. Meier, and R. F. Wagner, "On the problem of ROC analysis without truth : the EM algorithm and the information matrix," in *Proc. SPIE 3981*, ed, 2000, pp. 126-134.

[25] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood From Incomplete Data Via EM Algorithm," *Journal of the Royal Statistical Society Series B-Methodological*, vol. 39, pp. 1-38, 1977.

[26] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York, New York: Wiley-Interscience, 1996.

[27] S. K. Warfield, K. H. Zou, M. R. Kaus, and W. M. Wells, "Simultaneous Validation of Image Segmentation and Assessment of Expert Quality," in *International Symposium on Biomedical Imaging: Macro to Nano; 2002 Jul 7-10; Washington D.C., USA*, ed. New York, USA: IEEE, 2002, pp. 1494:1-4.

[28] S. K. Warfield, K. H. Zou, and W. M. Wells, "Validation of Image Segmentation and Expert Quality with an Expectation-Maximization Algorithm," in *MICCAI 2002: Fifth International Conference on Medical Image Computing and Computer-Assisted Intervention; 2002 Sep 25-28; Tokyo, Japan*, ed. Heidelberg, Germany: Springer-Verlag, 2002, pp. 298-306.

[29] S. K. Warfield, K. H. Zou, M. R. Kaus, and W. M. Wells, "Simultaneous Validation of Image Segmentation and Assessment of Expert Quality," *International Symposium on Biomedical Imaging: Macro to Nano; 2002 Jul 7-10; Washington D.C., USA,* pp. 1494:1-4, 2002.

[30] S. K. Warfield, K. H. Zou, and W. M. Wells, "Validation of Image Segmentation and Expert Quality with an Expectation-Maximization Algorithm," *MICCAI 2002: Fifth International Conference on Medical Image Computing and Computer-Assisted Intervention; 2002 Sep 25-28; Tokyo, Japan,* pp. 298-306, 2002.

[31] S. K. Warfield, K. H. Zou, and W. M. Wells, "Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation," *IEEE Trans Med Imaging*, vol. 23, pp. 903-21, Jul 2004.

[32] O. Commowick and S. K. Warfield, "A Continuous STAPLE for Scalar, Vector, and Tensor Images: An Application to DTI Analysis," *IEEE Transactions On Medical Imaging*, vol. 28, pp. 838-846, Jun 2009.

[33] O. Commowick, N. I. Weisenfeld, H. Als, G. B. McAnulty, S. Butler, L. Lightbody, R. M. Robertson, and S. K. Warfield, "Evaluation of White Matter in Preterm Infants With

Fetal Growth Restriction," in *Proceedings of the Workshop on Image Analysis for the Developing Brain (held in conjunction with MICCAI'09)*, 2009.

[34] O. Commowick and S. K. Warfield, "Incorporating priors on expert performance parameters for segmentation validation and label fusion: a maximum a posteriori STAPLE," *Med Image Comput Comput Assist Interv*, vol. 13, pp. 25-32.

[35] *http://wwwx.cs.unc.edu/~cquammen/wp/2010/04/13/how-to-build-qt-4-5-3-from-source-on-windows-xp-32-bit-suitable-for-vtk/*.

[36] B. Q. w. V. C. 2010, "http://stackoverflow.com/questions/1644172/building-qt-4-5-with-visual-c-2010."