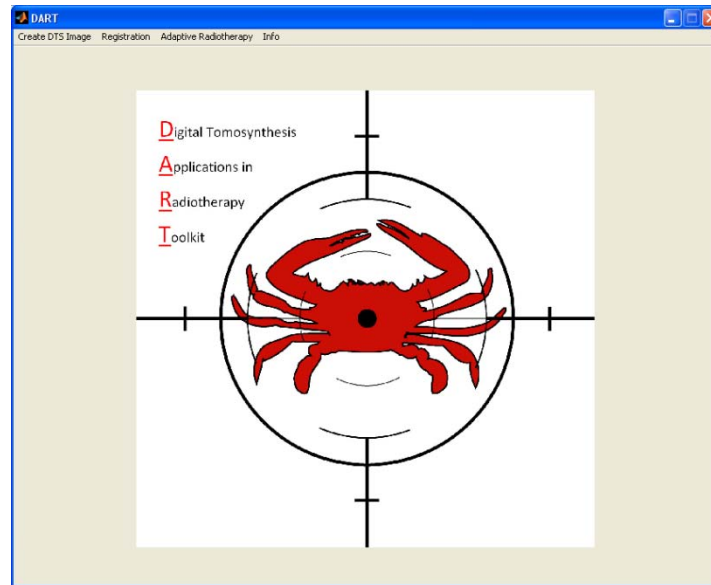


## DART User's Guide

This guide was written for use with the set of demo data provided. Since all the data is already available in the 'Demo data' folder, you will be asked to replace some of the files when you create them. You can either do this or name them differently.

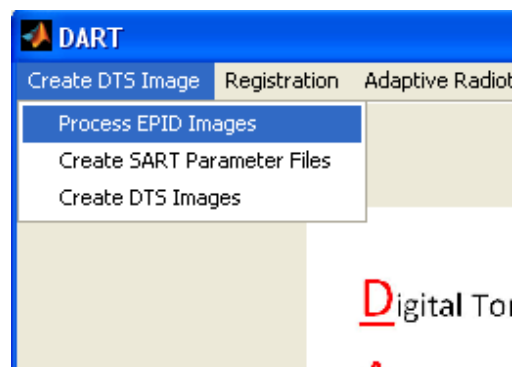
To start the program, set the path to the folder that contains all the code (typically the 'DART' folder) and type DART at the command line. This should launch the main window shown below.



### Section 1 – Generating DTS Images

#### **Part 1 – Creating Source Images**

This function is accessed through 'Create DTS Image' → 'Process EPID Images'.



This brings up the tool for processing raw images and stacking them. Images are expected to be in numerical order, with no extension for the EPID images and a \*.dcm extension for Pinnacle DRRs.

As input,

1. Choose the first file
2. Choose the last file
3. Give the increment
4. Tell the GUI whether these are EPID images or Pinnacle DRRs
5. Choose the filename to save

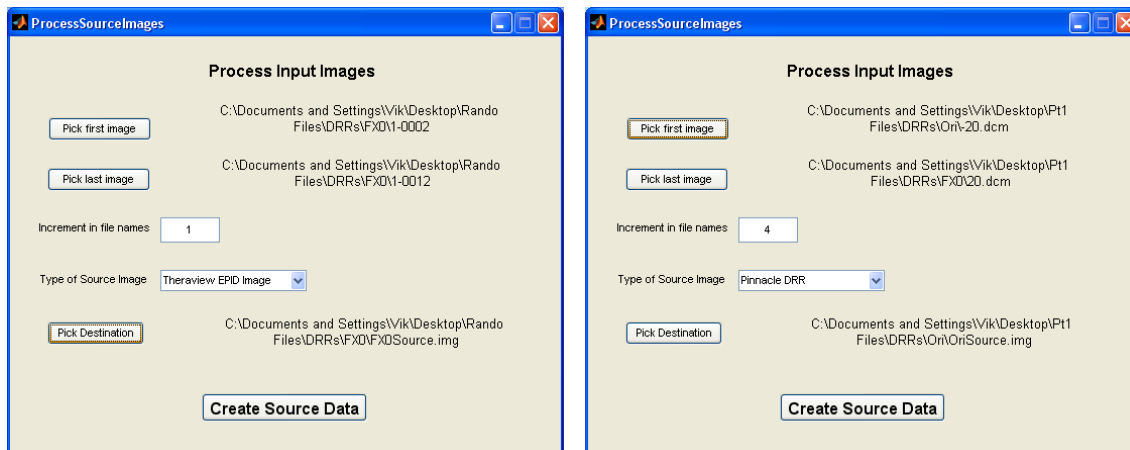
As an example (screenshots follow the instructions),

A. For EPID Images

1. Pick the file `..\Demo Data\Raw Images\EPID Images\1-0002` as the first file
2. Pick the file `..\Demo Data\Raw Images\EPID Images\1-0012` as the last file
3. Set the increment to 1
4. Select 'Theraview EPID Image' as the source type
5. Pick the filename to save (there is an example EPIDSource.img in the Demo Data folder)
6. Click 'Generate Source Data' and wait until you get the message to say the data was generated
7. You can see what was done by opening the file as a raw data set in Image J. The file contains 11 slices, each with 512 x 512 pixels.

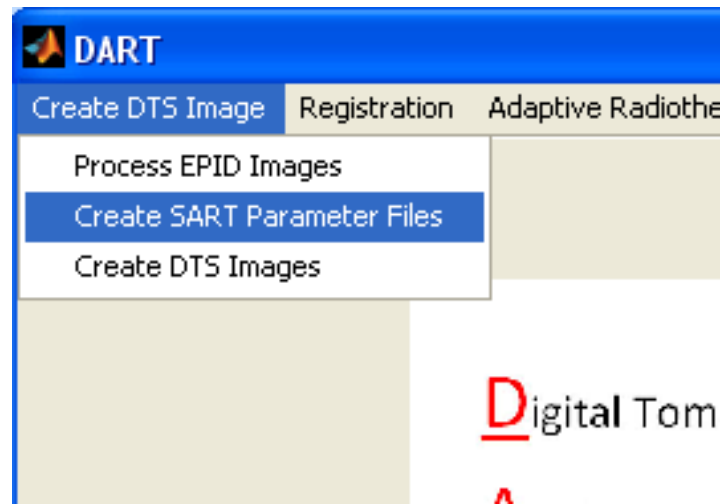
B. For Pinnacle DRRs

1. Pick the file `..\Demo Data\Raw Images\DRR\20.dcm` as the first file
2. Pick the file `..\Demo Data\Raw Images\DRR\20.dcm` as the last file
3. Set the increment to 4
4. Select 'Pinnacle DRR' as the source type
5. Pick the file name to save (there is an example DRRSource.img in the Demo Data folder)
6. Click 'Generate Source Data' and wait until you get the message to say the data was generated
7. You can see what was done by opening the file as a raw dataset in ImageJ. The file contains 11 slices, each with 462 x 462 pixels.



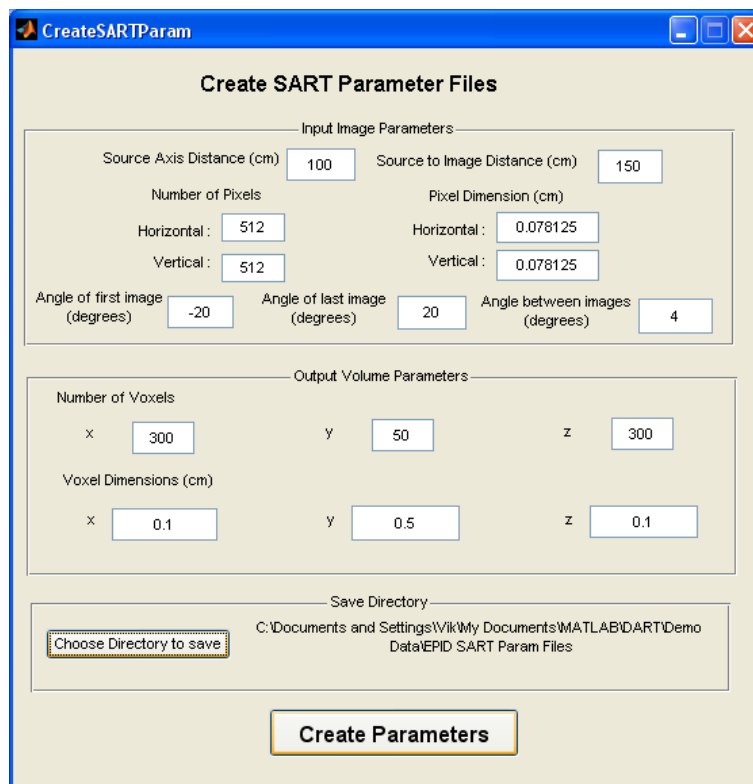
## Part 2 – Creating SART Parameters

This function is accessed through 'Create DTS Image' → 'Create SART Parameter Files'.



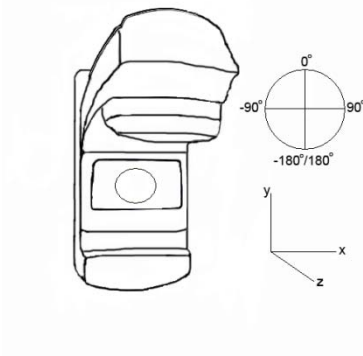
This brings up the GUI for the function used to create the files required during SART. Since the process takes a while and the files would be the same if the same geometry is used, this GUI allows for the files to be created once and used as many times as required.

The GUI has values pre-filled for the EPID. For a DRR, the number of pixels is unchanged but the size is 0.09756 cm.

The image shows a screenshot of the 'CreateSARTParam' GUI. The window has a blue title bar with the text 'CreateSARTParam'. The main area is beige and contains several sections of input fields. The first section is titled 'Input Image Parameters' and includes fields for 'Source Axis Distance (cm)' (100), 'Source to Image Distance (cm)' (150), 'Number of Pixels' (Horizontal: 512, Vertical: 512), 'Pixel Dimension (cm)' (Horizontal: 0.078125, Vertical: 0.078125), 'Angle of first image (degrees)' (-20), 'Angle of last image (degrees)' (20), and 'Angle between images (degrees)' (4). The second section is titled 'Output Volume Parameters' and includes fields for 'Number of Voxels' (x: 300, y: 50, z: 300) and 'Voxel Dimensions (cm)' (x: 0.1, y: 0.5, z: 0.1). The third section is titled 'Save Directory' and includes a 'Choose Directory to save' button and a text field showing the path 'C:\Documents and Settings\WIK\My Documents\MATLAB\Demo Data\EPID SART Param Files'. At the bottom, there is a large 'Create Parameters' button.

As input,

1. Give the SAD and SID (both in cm)
2. Give the number and size of pixels (in cm) in both direction
3. Give the angle corresponding to the first and last image, as well as the angle between the images. For the purposes of this software, the coordinate system shown below is used.



4. Choose the number and size of voxels (in cm) in each direction.
5. Choose the directory to save the results in.
6. Press the 'Create Parameters' button. An information window will appear to provide feedback while the files are being created. Two files are created for each angle – one with the .SARTPARAM extension and a second with the .WEIGHTS extension.

### **Part 3 – Generating DTS Images**

This function is accessed through 'Create DTS Image' → 'Create DTS Images'.

The GUI shown and inputs required vary depending on the reconstruction algorithm chosen. Some of the required inputs are pre-filled according to the values for an EPID reconstruction.

#### **A. FDK Algorithm**

The inputs are:

1. SAD and SID (in cm)
2. The input image file (the one created in part 1)
3. The number and size (in cm) of pixels making the source images
4. The angle of the first and last image as well as the increment between one angle and the next
5. The name to use to save the output
6. The number and size (in cm) of voxels in each direction.

An example reconstruction is found in the 'Demo Data' folder as FDKReconstruction.raw . The file was created using the default values of the GUI.

## B. Shift and Add Algorithm

**Generate DTS Images**

Reconstruction Algorithm:

Source to Axis Distance (cm)  Source to Image Distance (cm)

Input Parameters

C:\Documents and Settings\Viik\My Documents\MATLAB\Demo Data\EPIDSource.img

Number of Pixels

Horizontal  Pixel Dimensions (cm)

Vertical

Angle of first Image (degrees)  Angle of last Image (degrees)  Angle between Images (degrees)

Output Parameters

C:\Documents and Settings\Viik\My Documents\MATLAB\Demo Data\SAARReconstruction.raw

Number of Slices to Rebuild

Slice Separation (cm)

The inputs are:

1. SAD and SID (in cm)
2. The input image file (the one created in part 1)
3. The number and size (in cm) of pixels making the source images. For this algorithm, the pixels are assumed to be square in shape.
4. The angle of the first and last image as well as the increment between one angle and the next
5. The name to use to save the output
6. The number of slices to reconstruct.
7. The slice separation in cm.

An example reconstruction is found in the 'Demo Data' folder as SAARReconstruction.raw . The file was created using the default values of the GUI.

### C. SART

CreatedTSImages

**Generate DTS Images**

Reconstruction Algorithm: SART

Input Parameters

Get Input Images C:\Documents and Settings\Viik\My Documents\MATLAB\BART\Demo\Data\EPIDSource.img

Number of Pixels

Horizontal 512

Vertical 512

Get Directory with Parameter Files C:\Documents and Settings\Viik\My Documents\MATLAB\BART\Demo\Data\EPID SART Param Files

Angle of first Image (degrees) -20 Angle of last Image (degrees) 20 Angle between Images (degrees) 4

Output Parameters

Save As C:\Documents and Settings\Viik\My Documents\MATLAB\BART\Demo\Data\SARTReconstruction.raw

Number of Voxels

x 300 y 50 z 300

Voxel Dimensions (cm)

x 0.1 y 0.5 z 0.1

Number of iterations 2 Relaxation Parameter 0.2

Create DTS Image Set

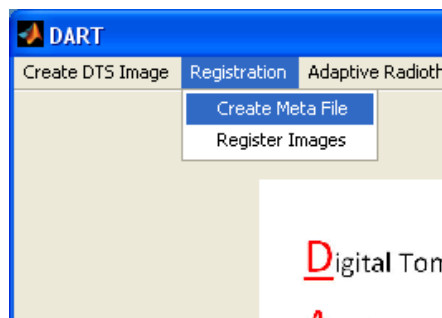
The inputs are:

1. The input image file (the one created in part 1)
2. The directory with the SART parameter files created in part 2 (if you have not done this step, this data does not exist and you need to create it first)
3. The number of pixels making the source images
4. The angle of the first and last image as well as the increment between one angle and the next
5. The name to use to save the output
6. The number and size (in cm) of voxels in each direction
7. The number of iterations for which to run the algorithm
8. The relaxation parameter

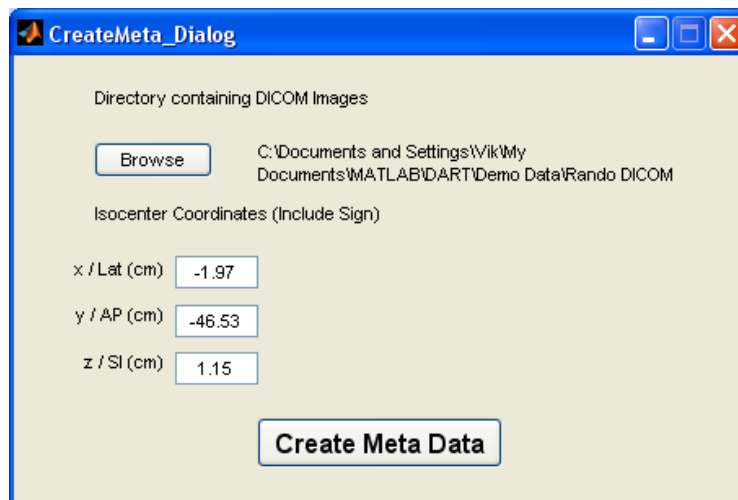
An example reconstruction is found in the 'Demo Data' folder as SARTReconstruction.raw . The file was created using the default values of the GUI.

### **Part 4 – Creating Metafiles for images**

This function is accessed through 'Registration' → 'Create Meta File'.



This function reads in all the DICOM files from the CT and stacks them into a single \*.raw file. It also creates two header files, one with the original coordinate system and one with an updated coordinate system where the plan isocenter is the new origin.



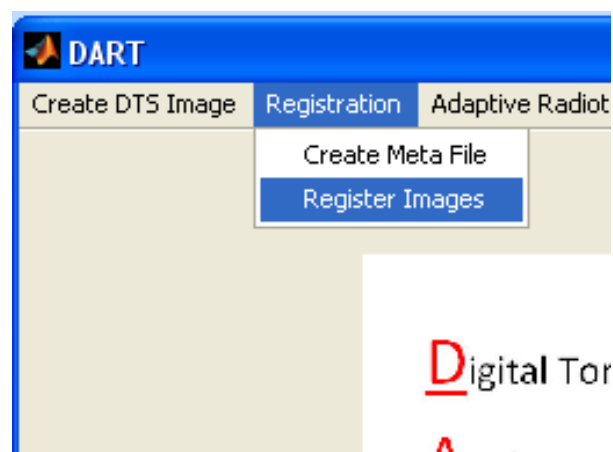
The inputs are:

1. The directory with the DICOM CT files
2. The coordinates of the treatment isocenter (in cm and including the sign). This information is obtained directly from Pinnacle

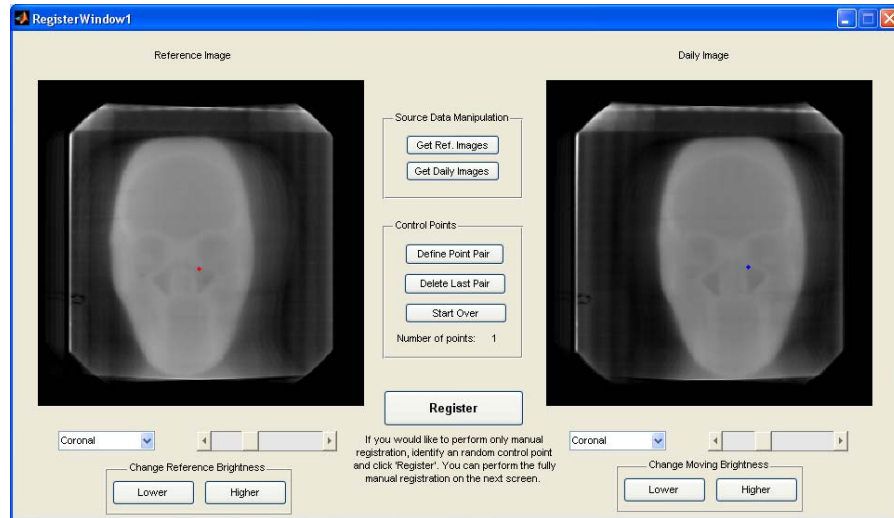
As an example, the directory '..\Demo Data\Rando DICOM\' . The isocenter location for this case was  $x = -1.97$  cm,  $y = -46.53$  cm, and  $z = 1.15$  cm

### **Part 5 – Registering Conebeam Datasets**

This function is accessed through 'Registration' → 'Register Images' to bring up the first registration window.



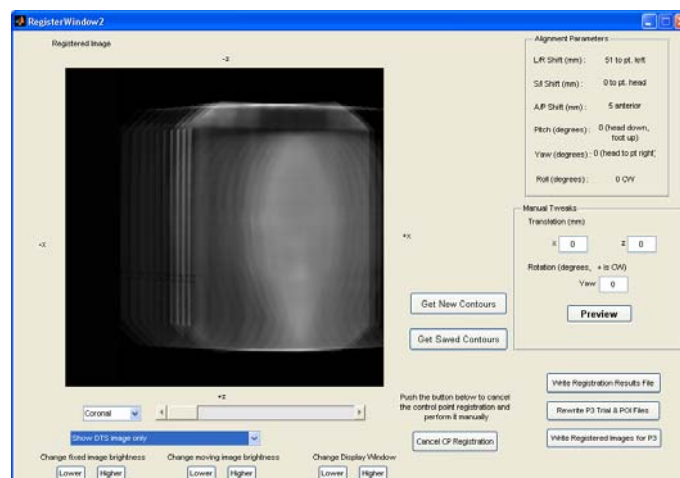
As an example, you can choose ‘../Demo Data/SARTreconstruction’ as the reference images and ‘../Demo Data/DailyCBCT’ as the daily images.



Since the images are coronal along the primary direction, this plane should be used. The first screen is used to define control points. Points need to be defined on the reference set first and then on the daily set. The points are displayed on the slices where they are selected but these are not kept in memory and moving from one slice to another and back will not cause them to be displayed any more.

Once at least one point pair is defined, the Register button appears. Pressing it will cause the daily dataset to be resampled to the same size and origin as the reference set. Once this is done, the second register window is shown.

Two options are now available to further tweak the registration. The fixed image set can be displayed in checkerboard fashion with the daily image set. Alternatively, the user can use contours imported from Pinnacle to help with the registration. Note: If the registration results are obviously wrong, the original control-point-based registration can be completely cancelled and a manual process can be done. If this is chosen, the option goes away and the original daily dataset is reloaded.

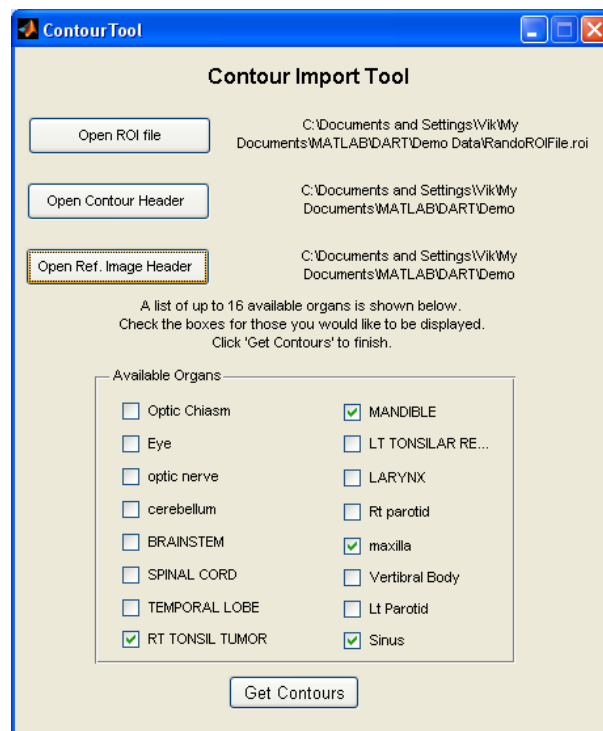




Small labels also tell the user in what direction they need to make translations (in the above figure, if the daily images are supposed to be moved up, then it requires a negative z value). Any manual tweaks are previewed first, where only the current slice selected is moved, and then applied to the whole dataset. The 'Alignment Parameters' panel is not updated until the changes have been applied to the whole dataset.

### Using contours

Special contour files are either created (the first time this needs to be done) or they can be imported from previously processed files. To create new contours, the 'Get New Contours' button is pressed. This brings up the contour processing tool shown below.



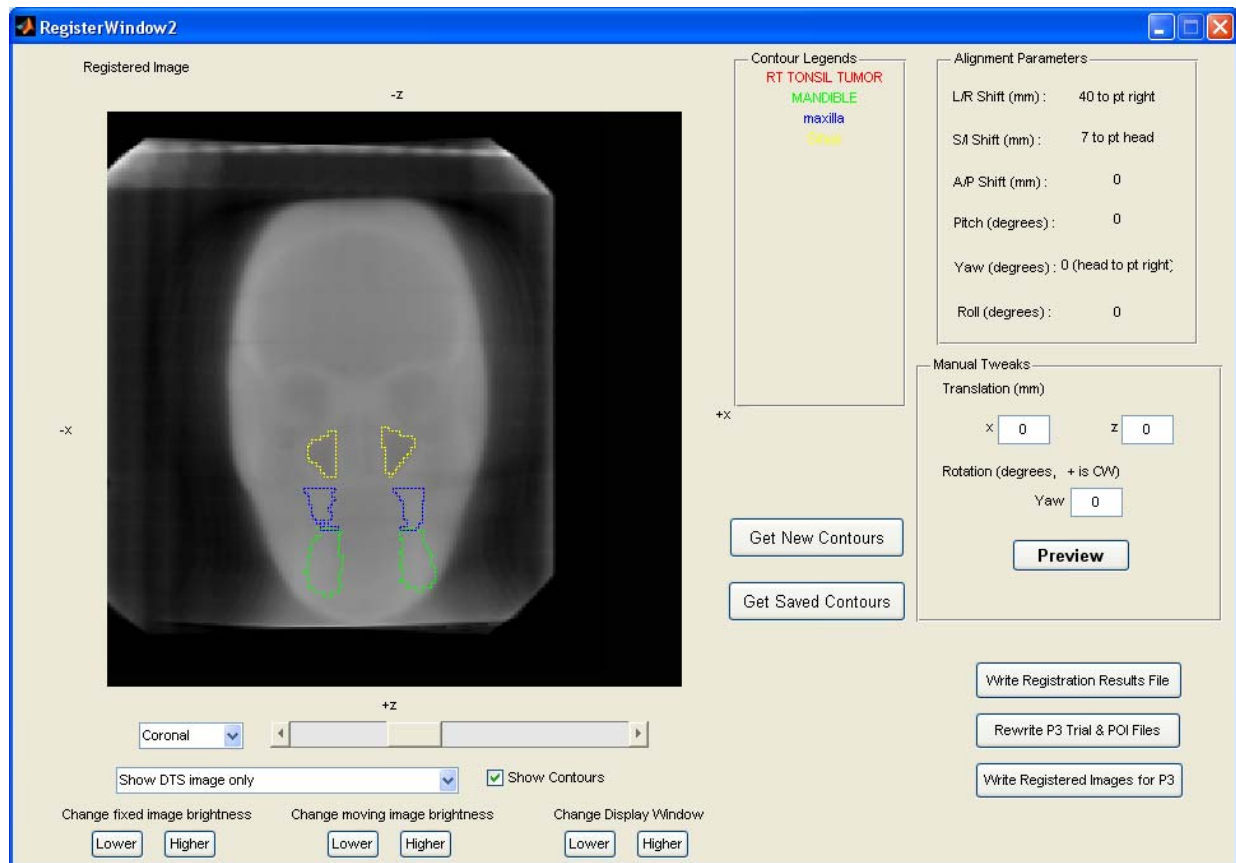
As inputs, you need to provide

1. An ROI file (for instance you can choose the '..\Demo Data\RandoROIFile.roi' file)  
Once this file is opened, you are given the option to process any or all of the first 16 ROIs in the file. In the case shown in the figure, we have chosen only four of the 16 ROIs.
2. A \*contour.mhd file (for example the '..\Demo Data\OriginalCTcontour.mhd' file)
3. The .mhd file for the reference dataset (In this case, the '..\Demo Data\SARTReconstruction.mhd' file)

Clicking on the 'Get Contours' button then asks for the name to save the files as and executes the function to create the files needed. The user is given feedback while the code is running and, once it is done, the GUI disappears. The 'Show contours' option is not chosen by default and needs to be chosen for contours to be shown.

Contours can also be opened from previously saved files. This looks for a \*.contour file and then processes the data and displays the contours on the correct slices. As an example, the '..\Demo Data\contourexample.contour' set of files. The \*.contour file contains only the names of the regions of interest. Accompanying files with extension contour.n, where n is an integer between 1 and 16, contain the actual information about the contour location.

With contours, it is typically easier to use only the DTS (daily) images only to register. For this particular case, the shifts should be of the order of 40 mm to left and 7 mm up to give the best registration parameters for matching the daily imageset with the contours (shown below).

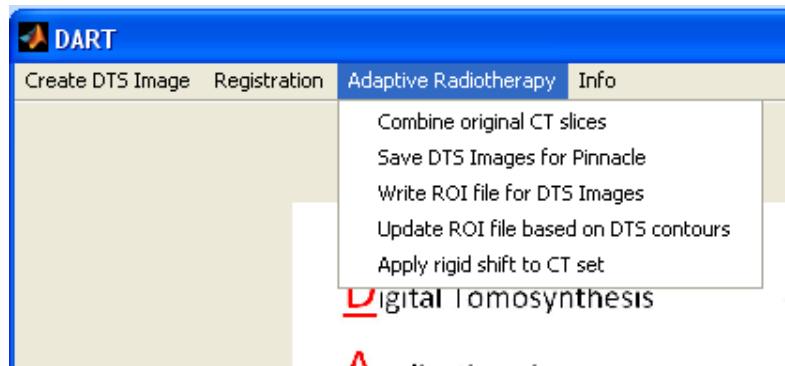


Since the reference images were not created from EPID images taken with the phantom positioned at its intended position. Thus, if you choose to do a registration with the reference dataset without the use of contours, the results will be different.

Once the user is satisfied with the registration, the results can be written to file using the 'Write Registration Results File' button or the Pinnacle \*.POINTS and \*.TRIAL files can be re-written for use in dose recalculations (the user can try this using the '..\Demo Data\OriginalTrialFile.trial' and the '..\Demo Data\OriginalPOIFile.points' files included in the demo data folder). Along with the original .trial and .points files, the user needs to provide the coordinates of the plan isocenter. Finally, the registered images can be written for the purposes of importing into Pinnacle.

## **Part 6 – Adaptive Radiotherapy**

There are several functions for adaptive radiotherapy as shown below.



1. The 'Combine original CT slices' function is used to combine the DICOM slices into a single file with an accompanying header file. This differs from the function described in part 4 of this guide in that the CT slices are not sampled down to twice the pixel width during the reading process.
2. The 'Save DTS Images for Pinnacle' function saves the files in a format expected by Pinnacle and uses the Pinnacle coordinate system in the header file.
3. The 'Write ROI file for DTS Images' function resamples the original ROI file into one to be used for the CBDT images saved in the previously described function.
4. The 'Update ROI file based on DTS contours' takes in an ROI file that was created using Pinnacle and CBDT datasets, resamples the contours to the original CT dataset and saves as a replacement for the original CT set.
5. The 'Apply rigid shift to CT set' function resamples the original CT datasets to take into account shifts saved in a registration result file.

These functions have not been fully tested and results may not be what is expected.