

DIRART (Deformable Image Registration and Adaptive Radiotherapy) Software Suite (Version 1.0a)

User Instruction Manual

Version 0.1

Deshan Yang, PhD
Issam El Naqa, PhD

Department of Radiation Oncology
School of Medicine
Washington University in Saint Louis

Copyright by Deshan Yang, PhD

Mar, 2009

Last updated: Mar. 11, 09

Table of Content

DIRART (Deformable Image Registration and Adaptive Radiotherapy) Software Suite.....	1
(Version 1.0a)	1
User Instruction Manual	1
Version 0.1	1
Deshan Yang, PhD	1
Issam El Naqa, PhD	1
Department of Radiation Oncology	1
School of Medicine	1
Washington University in Saint Louis.....	1
1 Introduction.....	5
2 More introduction	6
3 Introduction of functions and features	7
3.1 Image processing	7
3.2 Rigid image registration / alignment.....	7
3.3 Deformable image registration (DIR).....	8
3.4 Adaptive radiotherapy.....	8
3.5 Graphic user interface and visualization.....	9
3.6 Convenient features	10
3.7 Performance	10
3.8 What does DIRART not do?.....	11
4 Installation.....	12
5 A regular workflow	13
6 Starting the program	13
7 Loading images.....	15
7.1 Loading images from CERR projects	15
7.2 Loading images (2D or 3D) from MATLAB .mat files.....	16
7.3 Load DICOM image sets	17
7.4 Loading 2D images	17
7.5 Problems of image loading	17
8 After loading images.....	18
8.1 Aligning two images	18
8.2 Cropping images	18
8.3 Noise reduction for MVCT images.....	19
8.4 Resampling images	20
9 Image preprocessing	20
10 Image alignment.....	22
11 Deformable registration	22
11.1 Introduction.....	22
11.2 Definition of the deformation vector field (very important).....	22
11.3 Requirements for the images	23
11.4 Two deformable registration frameworks.....	24
11.4.1 Asymmetric registration framework.....	24
11.4.2 Symmetric / inverse consistency registration framework.....	24
11.5 Iterations and multi-resolution.....	24
11.5.1 Multigrid approach.....	24

11.5.2 Multiple pass approach	24
11.6 Parameter for registration iteration control	25
11.7 Algorithms and variants	26
11.8 Other options for registration computation	29
11.9 Start registration	29
11.10 Results	30
12 Validation and post-registration analysis	30
12.1 Smoothing the DVF	31
12.2 Deform the moving image or the fixed image	31
12.3 Invert the DVFs	31
12.4 Remove the DVFs (from DIRART)	31
12.5 Validation	32
12.5.1 Landmark based validation	32
12.5.2 Ground truth based validation	32
12.6 Inverse consistency analysis	32
12.7 DVF Jacobian analysis	34
12.8 Image similarity measurement	34
13 GUI and Visualization	35
13.1 General features	35
13.2 3D viewing directions	35
13.3 Viewing the images	36
13.4 Viewing DVF	38
13.5 Viewing structures	41
13.6 Viewing doses	42
13.7 GUI layouts	44
13.8 Locking and Unlocking display panels	45
13.9 Window level setting	46
13.10 Zooming and the geometry limits of display	47
13.11 Shortcut key combinations	48
13.12 Control GUI functions with mouse	49
13.13 Other GUI options	50
14 File and project management	54
14.1 Saving and loading projects	54
14.2 Saving and loading results	55
14.3 Reset results	55
14.4 Generate and export images	55
14.5 Event logging	56
15 Adaptive radiotherapy (ART)	57
15.1 Introduction	57
15.2 General working flow	58
15.3 Structures	58
15.3.1 Loading structures from CERR plans	58
15.3.2 Mesh-based structure presentation in DIRART	59
15.3.3 Structure manager	59
15.3.4 Smoothing a structure	59
15.3.5 Contract and expand a structure	60

15.3.6	Copy a structure to the other image	60
15.3.7	Deform structures.....	61
15.3.8	Export structures to CERR plan.....	62
15.3.9	Compare two structures	63
15.4	Doses.....	63
15.4.1	Load doses from CERR plans.....	63
15.4.2	The dose manager	63
15.4.3	Dose deformation.....	64
15.4.4	Export processed dose to CERR plan	65
15.5	Other ART tasks	65
16	Data inside DIRART.....	65
16.1	Access to DIRART internal data	65
16.2	The DIRART data structure.....	65
16.2.1	Images	66
16.2.2	DVFs	66
16.2.3	Structures	66
16.2.4	Doses.....	66
17	Advanced topics.....	67
17.1	Using the registration software without GUI.....	67
17.2	Important utility MATLAB functions	67
18	Contacts.....	67
19	References.....	67

1 Introduction

Recent years have witnessed tremendous progress in IGRT technology and potential possibilities for adapting treatment planning on a daily basis. However, there is a lack in having software tools toward this goal. Therefore, we have implemented the DIRART software suite for deformable image registration (DIR) plus adaptive radiotherapy (ART) research.

DIRART is a large set of programs developed using MATLAB. It contains DIR algorithms, common ART functions, integrated graphics user interfaces, visualization features, and dose metrics analysis functions. In addition, it complementarily works together with CERR (Computational Environment for Radiotherapy Research) to offer more functions. DIRART is designed around the concepts of the interactive RT objects, including images, structures, doses and deformation vector fields (DVF) etc.

By exchanging RT objects with TPS via DICOM-RT files, DIRART provides a full featured working environment for ART related research tasks. It is capable of transforming dose distributions and structures between the planning CT and the daily images according to the computed DVF so that such RT information could be viewed and evaluated on either image sets. It can also rescale, subtract and sum up the transformed doses, and convert isodose lines to structures.

For DIR applications, DIRART is a toolbox which provides 20+ DIR algorithms, including the newer inverse consistency algorithms to provide consistency DVF in both directions with better accuracy.

As a DIR research environment, DIRART dose not only provide a good set of image preprocessing algorithms and post-processing functions to facilitate the development and testing of DIR algorithms, but also offers a good amount of options for results visualization, evaluation and validation.

DIRART is designed in a data-oriented style with focus on usability, user-friendliness, performance, accuracy, flexibility, features, configurability and stability. It has great potential for the ART and DIR research.

2 More introduction

DIRART is a relative large and ambitious project. This project started in 2006 as a deformable image registration MATLAB program with simple graphics. After continuous improvements and major redesigns, it has become a complete software environment for deformable image registration and adaptive radiotherapy research. The scale of DIRART is fairly large. Up to now, it contains 450+ MATLAB program files with 40000+ program lines, plus many C/C++ programs to interface with ITK (Insight Segmentation and Registration Toolkit) and other software packages.

From users' standpoint, DIRART is: 1) a collection of DIR (deformable image registration) algorithms under two common frameworks, plus visualization and validation features, 2) an ART (adaptive radiotherapy) toolkit which is able to perform dose and structure remapping, dose accumulation and analysis using the DIR results, 3) a treatment plan viewer with many unique visualization options, 4) a complimentary package to CERR (Computational Environment for Radiotherapy Research)¹ to provide additional DIR and ART functions to CERR. Moreover, by exchanging DICOM-RT data, it could be used an external interface to commercial TPS (treatment planning system).

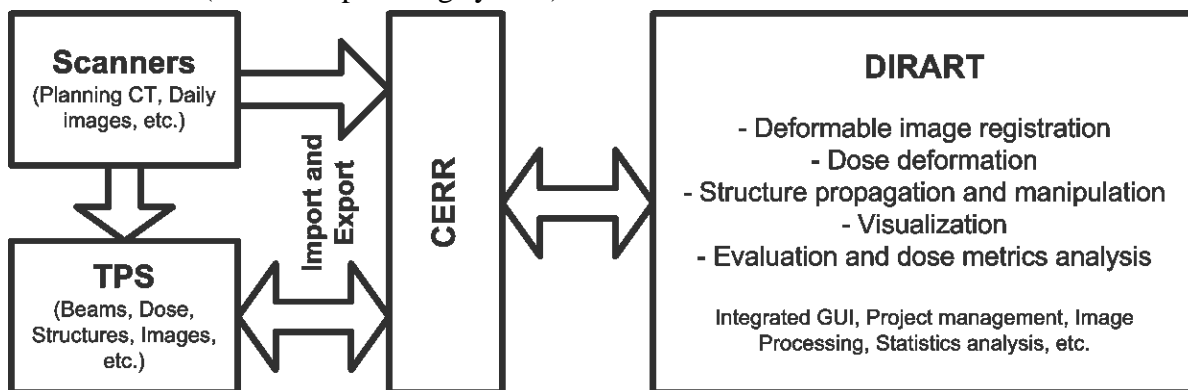


Figure 1: The general work flow. DIRART imports RT data from scanners and TPS by using CERR. While mostly carrying out its functions alone, DIRART complementarily works with CERR by sending results for DICOM exporting, additional data analysis.

DIRART uses many CERR MATLAB functions, especially functions for DICOM-RT import and export, dose metrics computation, etc. Therefore, DIRART requires installation of CERR to be fully functional. Figure 1 outlines the major functions of DIRART and its interactive relationships with CERR, TPS (treatment planning system) and imaging devices.

As illustrated in Figure 2, DIRART is designed around the concepts of the individual and interactive RT objects, including images, structures, doses and deformed vector fields (DVF) etc. All RT objects are processed independently based on their physical representations (position coordinates, voxel sizes, origin, orientation, etc). For example, a structure object contains a collection of contour points that are defined in the same coordinate system as the associated CT image. The structure is independent from the CT image because its contour points are not necessarily on the CT slices. A dose volume shares the same physical coordinate system as the CT scans and structures, but could have a completely independent volumetric coverage and voxel sizes. By enforcing such independencies, DIRART has the flexibility to transform RT objects among different imaging coordinate systems. For example, image registration could be performed on the cropped image volumes with user configurable spatial resolutions in order to

improve the computation efficiency. The resulting DVF can then be applied on the original CT scans (before cropping and with different spatial resolution) or the structures for various ART tasks.

Besides its DIR and ART functions, DIRART provides an integrated graphics user interface which offers accesses to all major functions, visualization features and configurable options. In the other side, DIRART functions could also be invoked from MATLAB scripts so that DIR and ART tasks could be streamlined without user interactions.

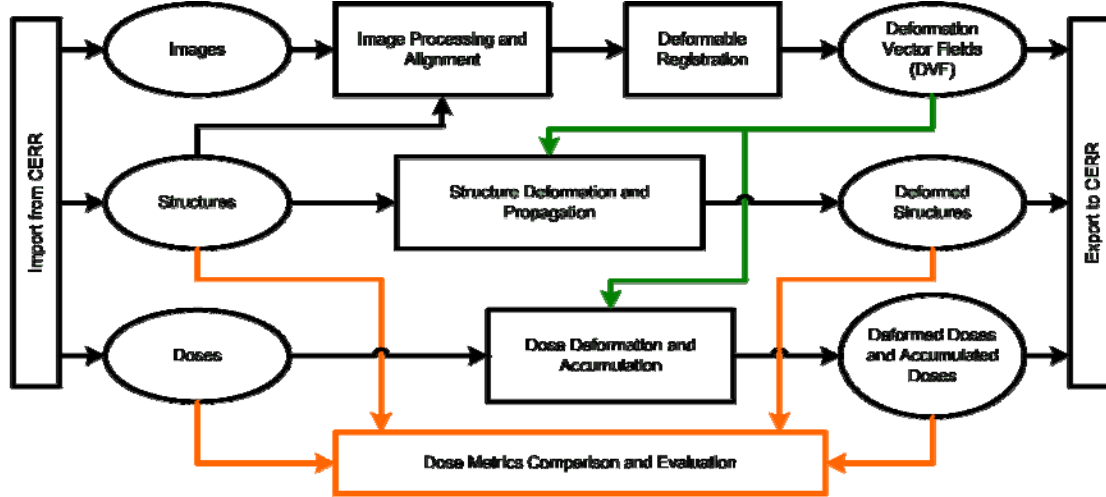


Figure 2: DIRART is designed around the concepts of RT objects, including images, structures, doses and deformation vector fields (DVF). These individual RT objects interact in the DIRART system via different DIR and ART tasks.

3 Introduction of functions and features

Figure 3 demonstrates the general workflow of image processing and registration (rigid and deformable) with DIRART.

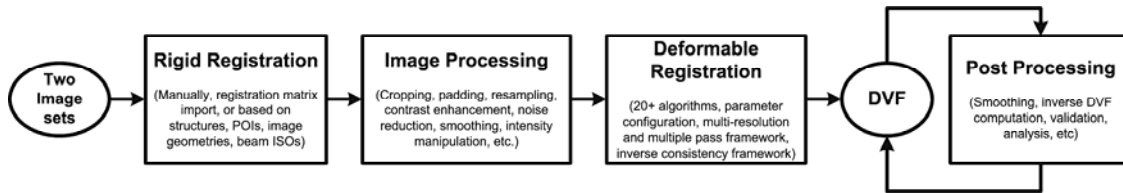


Figure 3: The general workflow of image processing and registration in DIRART.

3.1 Image processing

We implemented the following image processing procedures: 1) Smoothing (Gaussian low pass, 10 different edge preserving smoothing filters). 2) Histogram equalization (regular, contrast limited). 3) Window level transformation. 4) KVCT to MVCT intensity remapping. 5) Detect and paint bowel gas pockets². 6) Re-sampling (with various resampling filters). 7) Cropping. 8) Padding. 9) Image intensity manipulation by structures. 10) Mathematic operations on image intensity values. These image processing techniques help DIR algorithms to achieve better accuracy.

3.2 Rigid image registration / alignment

DIRART does not have many rigid registration algorithms. Instead, it imports rigid image registration results (as transformation matrix) from CERR, or from DICOM-RT files. DIRART

however has a good image alignment method, which allows two images to be aligned by POIs (points of interest), and geometry or mass centers of structures, etc.

3.3 Deformable image registration (DIR)

We have implemented two common DIR frameworks in DIRART, the asymmetric registration framework and the inverse consistency registration framework. The inverse consistency framework³ is able to generate DVF and inverse DVF simultaneously without explicitly inverting the regular DVF in a post-registration operation. Both frameworks support multi-resolution approach and multiple pass approach. They also provide necessary image resampling, DVF resampling, concatenation and optional smoothing operations after passes and multi-resolution stages. The GUI is constantly updated during the DIR computation. Therefore users could view the progresses of the computation.

Actual DIR algorithms are implemented under the frameworks. We have implemented many DIR algorithms and variants, and the interface to ITK so that DIRART can execute the DIR algorithms implemented in ITK. The available DIR algorithms are: 1) 12 optical flow algorithms⁴, 2) 6 demons algorithm⁵ and variants, 3) 4 level-set algorithms, 4) 5 DIR algorithms from ITK (including demons algorithms and B-spline algorithm).

DIRART offers the following post-registration operations on DVF after a DVF is computed: 1) smoothing, 2) computing inverse DVF for asymmetric DIR algorithms, 3) accuracy validation (with a ground truth DVF, or landmark pairs), 4) analysis (Jacobian, magnitude histogram, etc).

3.4 Adaptive radiotherapy

ART is a very broad topic. In DIRART, we have implemented the functions to transform information (image data, structures and dose) between the two image sets so that information originally defined on one image could be processed (visualized, evaluated, etc) on the other image domain. Transformation could be a coordinate value shifting (without deformation) and a DVF based coordinate remapping (with deformation).

Remapping (with or without deformation) of doses and images are resampling of these volumetric data and it is technically easier after both DVF and inverse DVF are computed. Remapping information defined on the moving image onto the fixed image domain needs to use (regular) DVF. Remapping information in the other direction needs to use the inverse DVF.

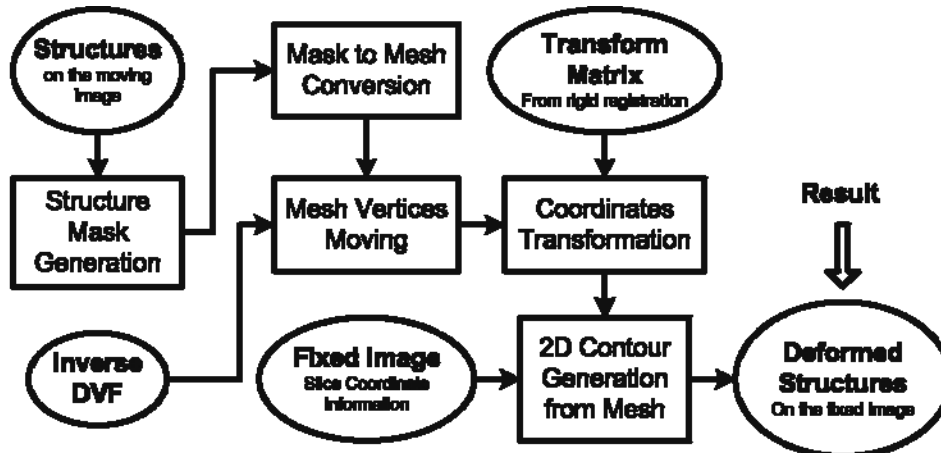


Figure 4: DIRART structure propagation workflow. This procedure deforms the moving image structures onto the fixed image by using the inverse DVF.

Remapping structures is less straightforward and worth further explanation. Figure 4 shows the DIRART work flow to remapping the structures defined on the moving image onto the fixed image. A structure imported from DICOM-RT is as a collection contour points defined on the 2D slices of the moving image. After remapping these contour points, we need to reformat the remapped structure data into a same style (as contour points defined on the slices of the fixed image) before the generated new structure data could be sent to CERR and exported to DICOM-RT files. This is the reason why the relative complicated workflow is employed.

The procedure to remap structures from the fixed image to the moving image is similar to Figure 4 but the regular DVF is used instead of the inverse DVF. Structure remapping could also be done by deforming the structure mask and then converting the deformed mask volume into 2D contour points. This alternative workflow is also implemented in DIRART.

DIRART has a dose manager and a structure manager. The dose manager can load, delete, rescale, add/subtract and rename doses. Dose subtraction and summation are important to evaluate the dose difference between fractions and to compute the dose accumulation. Both subtraction and summation are performed in conjunction with resampling if doses are in different voxel positions and/or resolutions. The structure manager has functions to rename, cleanup, shrink, expand, smooth, transfer and delete structures, as well as to turn structure display on/off, and change structure display colors. DIRART can also convert user configured isodose contours into structures, and further process (transfer, smooth, export, etc.) them as regular structures.

3.5 Graphic user interface and visualization

Visualization is a very important part of DIRART. DIRART is able to display all RT objects together in multiple individually configurable display panels. In fact, DIRART could be used as a treatment plan viewer. Comparing to CERR, it has many different visualization options, especially for structures and doses. Figure 5 shows a screen capture of the DIRART GUI. See Table 1 and Table 2 for very detailed explanation for the visualization features and options.

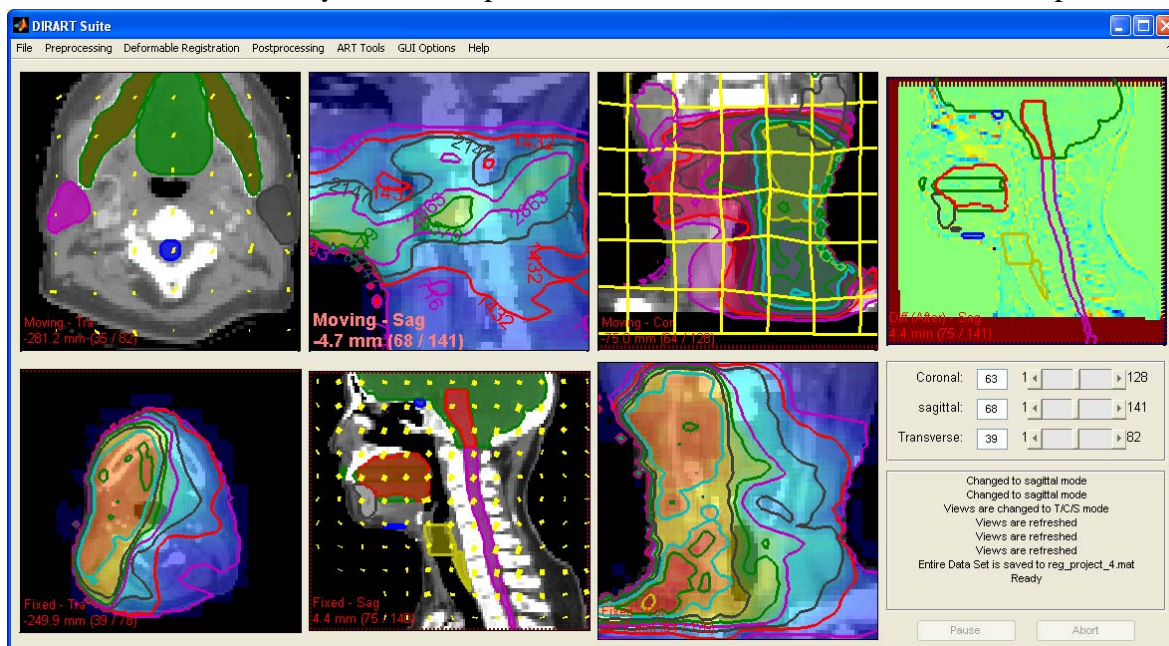


Figure 5: Screen shot of the DIRART main GUI. This is a H/N cancer patient. This example demonstrates some of the DIRART visualization options. There are 7 display panels, each is individually configured to display different or combined data, in different 3D views, zooming settings, color setting, etc. There are two

doses, one for the moving image and another for the fixed image. Doses could be displayed in isodose lines (with or without labels) and/or in colorwashes. Structures could be displayed with only contour lines, or with color filled. DVF could be displayed as vectors or as deformed grid lines. Line colors and thickness are configurable. The difference images could be shown in either colors or in gray scales.

Table 1: Visualization options for RT objects

Objects	Visualization Options
Image	1) The original images, 2) the difference images (before or after registration), 3) the deformed images, 4) images in composite color channels, 5) checkerboard images (before and after registration), in gray or in color, with and without grid lines, with configurable grid sizes
Dose	1) Isodose lines (absolute or percent doses), with labels on or off, with or without color filled between lines, 2) dose colorwash, 3) both
Structures	Image association: 1) only with associated image, 2) on any image (automatic coordinate transformation) Color: user can select color for every individual structure. Display mode: 1) in contours, 2) color filled, 3) both
DVF	DVF to display: 1) backward DVF, 2) forward DVF, 3) per-iteration delta DVF, 4) per-pass delta DVF, 5) per-stage (multi-resolution stage) delta DVF. Vector display mode: 1) In arrows, 2) As deformation grids DVF scalar data display in color wash: 1) projection in L-R, 2) in A-P, 3) in S-I, 4) magnitude, 5) Jacobian

Table 2: General GUI options

Options	Possible choices
Colormap	User can select the different colormaps for color wash displays options, for dose, image, structures and DVF
Window levels	Images can be displayed in user selected window levels, or one of the 6 predefined window level settings
Display panels	Up to 7 individual display panels are supported. Each panel can be individually configured. 5 predefined display layouts are available: 7 small panels, 1 larger + 3 small, and so on. Image position (to be displayed) can be locked or unlocked between different display panels
Other display options	1) Draw image boundary box, 2) Draw NaN value boundaries, 3) Aspect ratio, 4) display landmarks for registration validation purpose. 5) Configurable font sizes, line thickness, transparency.
Mouse actions	1) Windows level adjustment, 2) zoom in/out, 3) panning, 4) slice changing, 5) reading image intensity values, 6) active display panel selection, 7) position localization cross display panels, etc.
Key shortcuts / combination	1) Slice changing, 2) image alignment adjustment, 3) select window level presets, 4) turn DVF display on/off, 5) image display orientation selection, 6) mouse action selection, etc.

3.6 Convenient features

DIRART allows users to save the entire working data set into a disk file, and to load the entire project back in a later time to continue previous works. DIRART also allows its internal data to be exported to MATLAB environment so that the data could be further processed by user provided MATLAB programs or scripts.

3.7 Performance

Computation speed of DIR algorithms in DIRART is acceptable. DIRART allows user to crop the images before an actual DIR computation and allows users to configure voxel resolution for DIR computation. In this ways, the actual dimensions of the images in DIR computation are limited, and the DIR computation time and computer memory requirement are controllable. For example, we use a resolution $2 \times 2 \times 3 \text{ mm}^3$ for head-neck patients. For image sizes of $130 \times 140 \times 82$, DIR computation finishes in 80 seconds on a Dell Optiplex 755 desktop PC with 2.66 GHz CPU and 2GB RAM. The actual computation speed is of course depended on the DIR voxel resolution and the size of image cropping region. In general, DIR computation time has

become a less important factor in the total time of an ART task, instead of the major factor as it used to be.

3.8 What does *DIRART* not do?

DIRART does not have a dose computation engine to compute daily doses from daily images and treatment plans. However, this could be compensated by using the Monte Carlo dose computation engine with CERR (will be also available for DIRART). DIRART also does not do online or offline adaptive treatment planning re-optimization based on daily images.

4 Installation

No need of special installation procedures. Just copy the software package into a certain folder and add the folder to program searching path of MATLAB. CERR must also be installed and within the MATLAB searching path for DIRART to fully functional because DIRART uses many CERR MATLAB functions. Please refer to CERR information at

<http://radium.wustl.edu/CERR/about.php>.

DIRART requires MATLAB version 2007 or later.

5 A regular workflow

1. Start the main GUI
2. Loading images from CERR plans
3. Align, crop and resample the images
4. Apply image preprocessing filters
5. Set registration parameters and start deformable registration computation
6. View and analyze the registration results, if necessary, redo step 4 and 5 with certain parameter changes
7. Loading doses and structures from the same CERR plans
8. Perform doses and structures deformations
9. Analyze the deformed doses and structures
10. Export the deformed doses and structures back to the CERR plans
11. Export deformed doses and structures to DICOM-RT files from CERR. These files could be loaded into TPS systems later.

6 Starting the program

Run “dirart” from MATLAB command line, you will get the welcome message window. Click OK button to continue and you will get the main GUI with blank panels Figure 7.

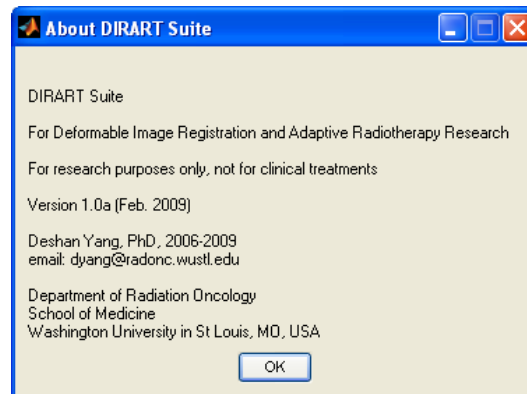


Figure 6: The welcome message

It is possible to start the program with different command line parameters, for example, to pass the two images to be registered. Please see later sections for command line options.

It is possible to choose different GUI layout, for example, to use 7 image display panels instead of 4 panels. Please see later sections for GUI options.

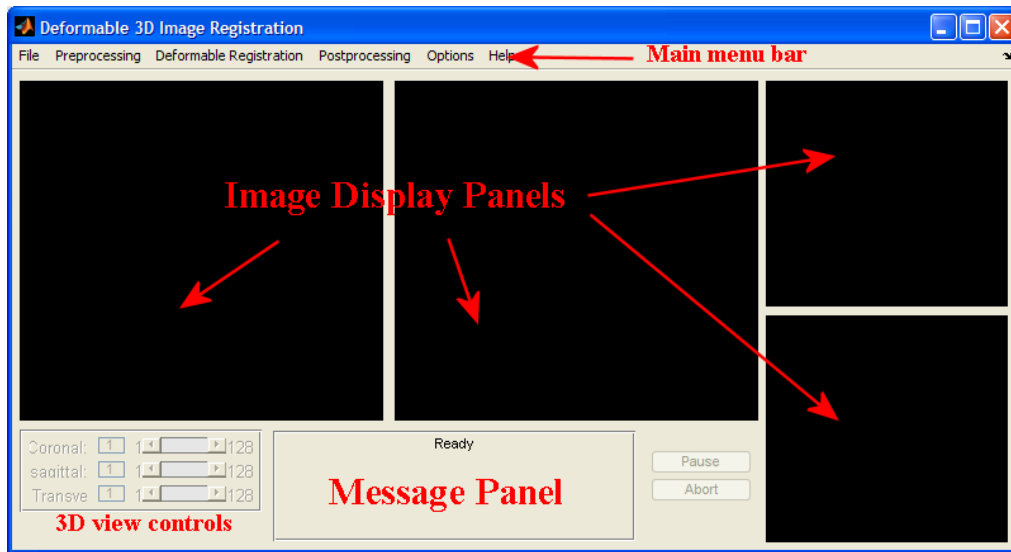


Figure 7: Blank GUI screen shot

After images are loaded, the 3D view controls will be initialized according to the image dimension. Images will be displayed in the display panels. There are many different ways to view the images and results. Figure 8 shows the screen shot after image is loaded. Please see the later sections for options of visualization.

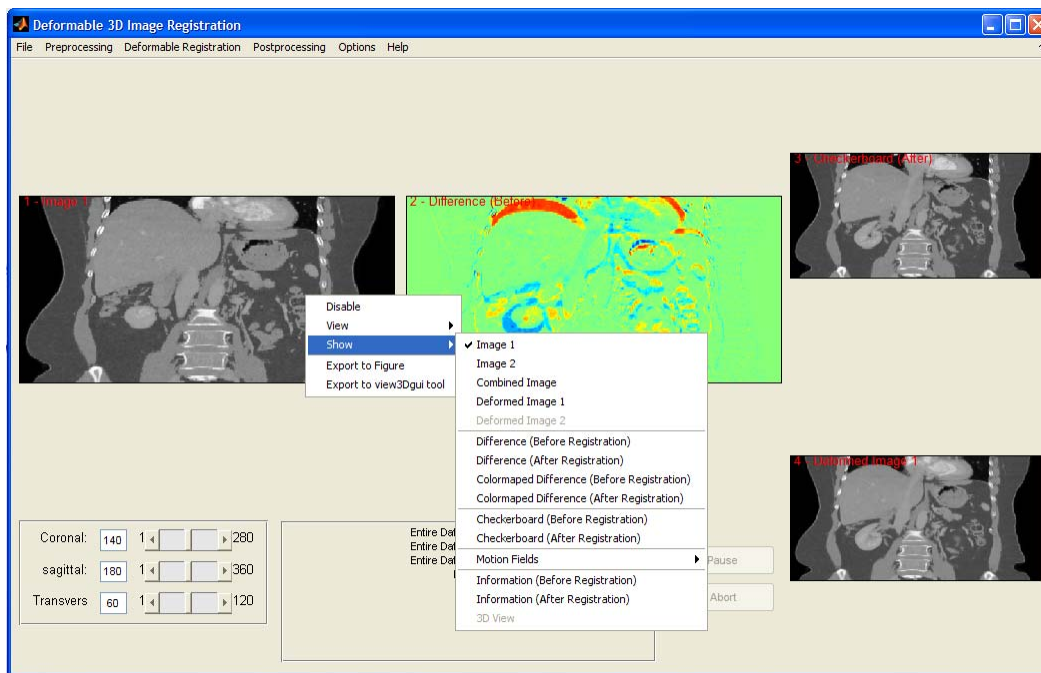


Figure 8: GUI screen shot after images are loaded.

All GUI features and options are accessible through the main menus and the mouse right click popup menu. Many important features are associated with short-cut key combinations.

7 Loading images

7.1 Loading images from CERR projects

CERR¹ (Computational Environment for Radiotherapy Research) is a software platform for developing and sharing research results in radiation therapy treatment planning (<http://radium.wustl.edu/CERR/about.php>). It is programmed in MATLAB, and is able to import and display treatment plans through DICOM-RT files.

DIRART is able to load image data from CERR files. In fact, we recommend all users to load images via CERR by creating the whole dataset as CERR plans, including images, structures, doses, etc, so that other information besides images could be loaded from the same CERR files in later steps.

- Load the whole RT data set (images, doses, etc) into CERR, then save the CERR plan file. Each image set should be saved as one CERR file. (CERR archive files are MATLAB *.mat files)
- In DIRART, use “File → Load Images → Load images from CERR plans” to load two image sets from two CERR plans. As shown in Figure 9, DIRART will ask the user to select a CERR plan file to load the image from.
- If there are multiple images (scans) in one CERR plan, user will be asked for which scan to load (Figure 10).
- If there are structures in the CERR plan, user will be given a chance to load structures after loading an image set (Figure 11).

In the current version, DIRART will list all structures in the CERR plan. However, only the structures associated with the selected scan should allow being loaded. DIRART will fix this problem in the later version. At this moment, users should know which structures are associated with the selected CT scan, and should only select these structures.

User may want to only load the structures that will be used for image alignment purpose (See the next section about image initial alignment) at this point. For example, only load the POIs (point of interest) that indicate the beam isocenters. Users can always load the rest structures later into DIRART.

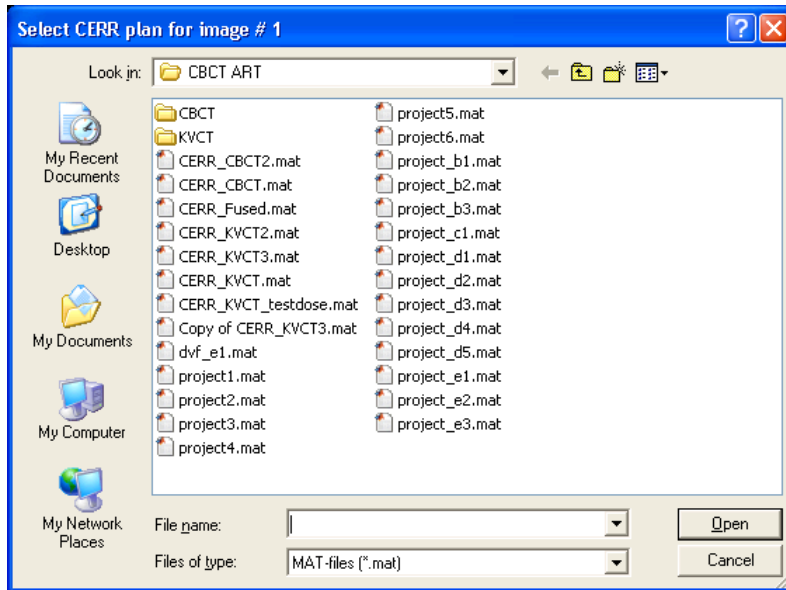


Figure 9: Screen shot of loading an image from a CERR plan

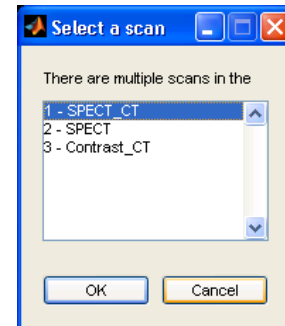


Figure 10: Screen shot of selecting a scan from a multiple scan CERR plan

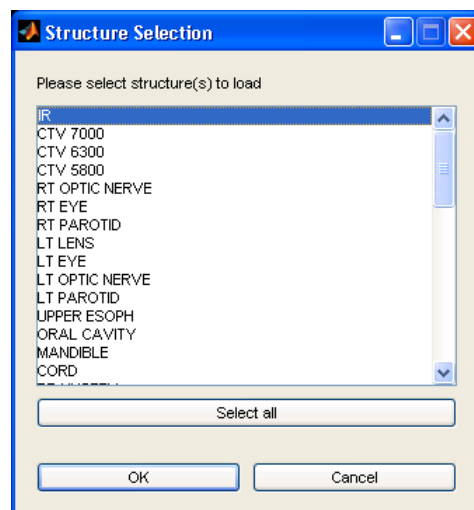


Figure 11: After DIRART loads an image from the CERR plan, it will ask user to select structures in the CERR plan to load.

7.2 Loading images (2D or 3D) from MATLAB .mat files

CERR can only load images from DICOM files or from a RTOG plan. If the images cannot be loaded in CERR, then they can be loaded into MATLAB and saved into MATLAB .mat files, and loaded into DIRART from the saved .mat files.

To load images from .mat files, use “File → Load images → Load images in MATLAB format”. Two images will be loaded together. Users will be asked to select the .mat files for the images one by one.

Images in MATLAB are just 2D or 3D binary array variables. Please note that array variable memory allocation in MATLAB is different from the regular image memory storage. In MATLAB, for a 2D array with x and y coordinate indices, the y index is the first index, and the x

index is the second. The array data stored in memory is continuous in y , not x . For a 3D array with x , y and z coordinate indices, the y is the first index, x is the second and z is the third. Array data is continuous in y , then x , then z .

For 3D images, after both images are loaded. Users will be asked to manually enter the voxel sizes in mm because image voxel sizes are not stored together with the image data in the .mat files. In later DIRART versions, users will also be asked for the image origin, which is the physical coordinates of the very first voxel, and the voxel coordinate increments (could be negative).

Images, as MATLAB 2D or 3D array variables, can be stored as MATLAB .mat file. One MATLAB .mat file should only store one image (a 2D or 3D MATLAB variable, with its own variable name). By default, the .mat filename should be the same as the variable name. For example, an image as a MATLAB binary array variable with name “lung_img_1”, it should be saved in a .mat file named as “lung_img_1.mat”.

3D images in other format must be loaded into MATLAB as 3D binary variables outside the deformable registration tool, and then saved into MATLAB .mat files before they can be loaded into DIRART. The helper functions listed in Table 3 could be used for such a task.

Table 3: Utility functions (already included in the software package) to load 3D images into MATLAB

load_3d_image.m	to interactively load a series of 2D images one by one
load_3d_image_dicom.m	By giving a filename filter, to load all DICOM files selected by the filter at once
load_3d_image_raw.m	To load binary 3D image files

7.3 Load DICOM image sets

DIRART could directly load a DICOM image sets. In fact, DIRART uses CERR DICOM loading functions to load such DICOM image sets without saving them into a CERR plan. However, we still suggest users to use CERR to create CERR plan files for such DICOM image sets, because CERR plans also support other DICOM data besides images.

Please note that CERR converts the image coordinates in the DICOM files to RTOG coordinates, therefore image coordinates loaded into DIRART (from CERR plans, or from a DICOM image sets by using CERR DICOM functions) are different from the original image coordinates in the DICOM files.

7.4 Loading 2D images

More image file formats are supported for 2D images than 3D images. In fact, most image file formats, include DICOM, PNG, TIFF, JPG, GIF etc., are supported. To load 2D images, use “File→Load images → Load 2D images”. If the file format is not supported, errors will be reported in the MATLAB command line. Then these image files must be loaded into MATLAB outside the registration software and saved into MATLAB .mat files before they can be used for registration computation.

Pixel spacing in x and y directions are assumed to be the same for 2D images. In later versions, DIRART will ask for the pixel resolution for 2D images. If the resolutions of two 2D images are not the same, DIRART will perform resampling operations.

7.5 Problems of image loading

Possible problems of image loading:

- Out of memory
This is a general MATLAB problem. A general solution will be discussed in the later sections. However, a few important features in DIRART, especially image cropping, resampling, and dissociation between different object, have designed to ease such MATLAB memory problems.
- Not all image file types are supported. DIRART will support them as far as the images could be loaded into MATLAB and saved as .MAT files.
- Don't know the image voxel physics positions.

8 After loading images

DIRART forces users to perform cropping, aligning, smoothing and resampling on the two images just loaded.

8.1 Aligning two images

It is very important to initially align the two loaded images. Alignment means translation only registration, without rotation, scaling or deformation.

As shown in Figure 12, a dialog window will pop up after two images are loaded for the user to align the two images. DIRART allows aligning images using the POIs, or the centers-of-mass of the objects associated with each image.

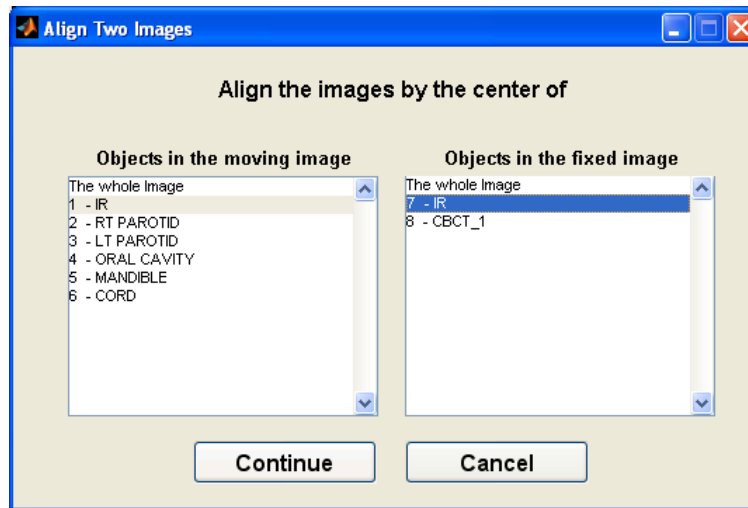


Figure 12: Image alignment window screen shot. The objects “IR” for each image are the POIs defined for the beam isocenters for the associated image.

Users can realign two images later. Please see later sections for this topic.

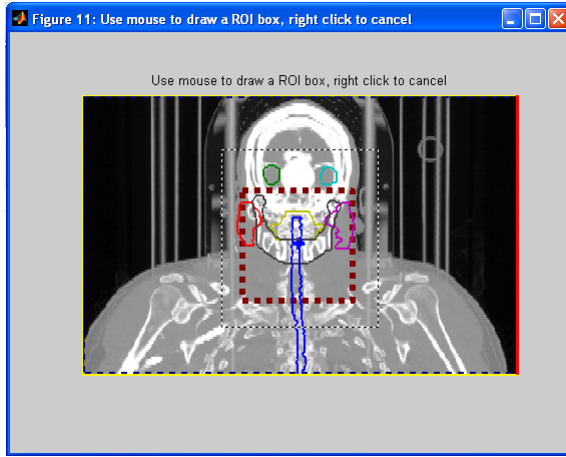
8.2 Cropping images

Image cropping reduces the image dimension. The goal is to reduce total memory require so as to reduce the chances of OUT OF MEMORY errors, and to reduce the overall time of later image preprocessing and deformable image registration computations.

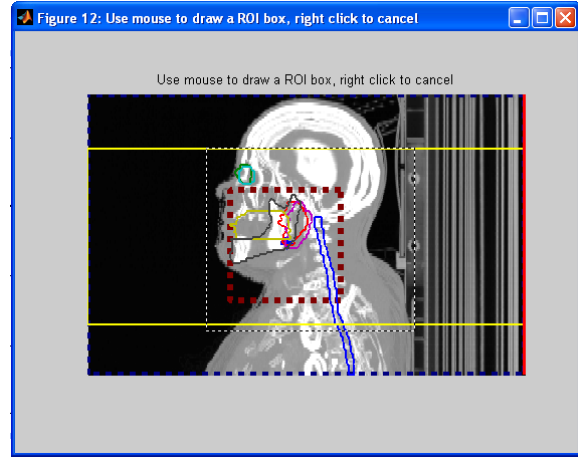
See the demonstration in Figure 13. DIRART asks user to sequentially define the cropping boundary box in the three 3D viewing directions: coronal, sagittal and transverse (Figure 13 a to c). Users will use mouse to draw the cropping boxes on top of the MIP (max-intensity-projection) images overlaid with other structures and boundary boxes. Displaying the MIP images and structures helps users to clearly see through the entire images in order to

determine the cropping boundaries. At the end, users will be asked to confirm the defined cropping boundaries (Figure 13d). If not satisfied, DIRART will ask users to redo it.

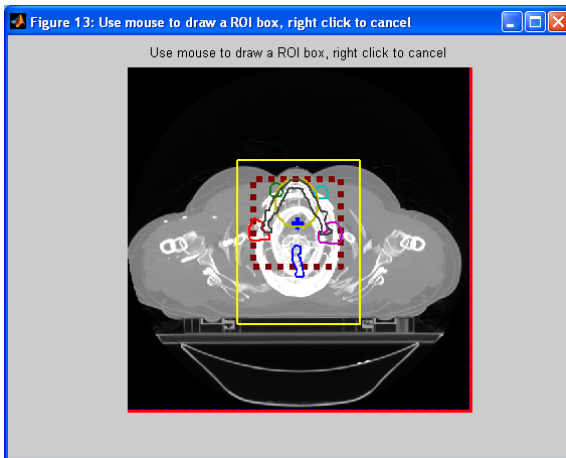
At each viewing direction, users can right click the mouse to skip this step, which means to use the current boundary box without drawing a new one.



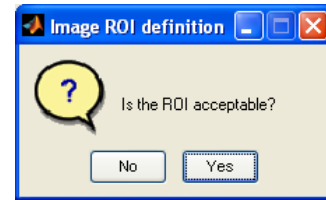
(a) Step 1: coronial view



(b) Step 2: sagittal view



(c) Step 3: transverse view



(d) User confirmation

Figure 13: Screen shots of image cropping. The MIP (max-intensity-projection) image is displayed. The yellow boxes are the current cropping box which is initially the whole image, and redefines itself after users sequentially define it using mouse in the three viewing directions. The thin gray dash lines are the boxes user is drawing using the mouse. The thick brown dashed boxes are the image boundary box of the other image overlay in the current image being cropped. The structures associated with the current image are also displayed in MIP fashion.

Users can redo the image cropping later. Please see the later section for this topic.

8.3 Noise reduction for MVCT images

If one or both images are MVCT scans, DIRART will ask if users want to perform noise reduction filtering on it before the images are resampled. There are some advantages to perform noise reduction for MVCT images before these images are resampled, and these are the reasons why noise reduction is performed at this point.

Please note that DIRART uses the image type and scanner manufacturer information from the original image DICOM header information to determine if the images are MVCT scans.

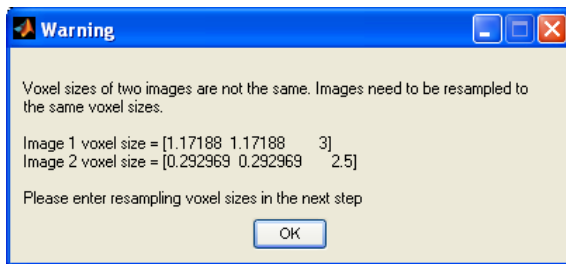
If DIRART cannot determine if the images are MVCT scans, it will not ask user to perform the optional noise reduction filter. Users can always perform such noise reduction filtering later.

See the later section for image preprocessing for details about the different filter types and parameter settings.

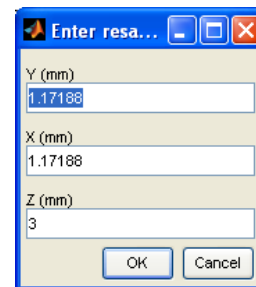
8.4 Resampling images

DIRART will prompt the user to perform image resampling after both images are cropped, as demonstrated in Figure 14. Both images have to be in the same spatial resolutions in order to proceed to the further registration computation. Therefore, resampling is mandatory if two images have different resolution. Resampling is optional if two images already have the same resolution.

DIRART will tell the current image resolutions (Figure 14a), and will ask user to enter the new voxel sizes for both images (Figure 14b). The default values suggested in Figure 14b are the max values of the original voxel sizes of both images. If the final voxel sizes are the same as the original image voxel sizes of either image, resampling will be skipped for that image.



(a)



(b)

Figure 14: Image resampling

User can resample the images again later. Please see the later section for this topic.

9 Image preprocessing

Image preprocessing tools are available at “main menu → Preprocessing → ...”. DIRART have included a selection of image processing tools that have been useful for our deformable image registration practices. However, the section is neither complete nor consistent. There are certainly other image processing filters that could be added in later version. We are also planning to integrate all these image preprocessing operations into a stand-alone GUI window, which will provide preview and allow user to confirm, undo, cancel and etc. Therefore, the ways that DIRART handles image processing could be changed constantly.

DIRART can undo the last image processing step, but for the last step only and not further back. DIRART actually saves the image data before an image preprocessing operation takes place, and reload these saved image data when undo is required. However, such undo information will be erased in order to reduce memory usage before a deformable registration computation. Therefore, users will be able to undo after a DIR computation.

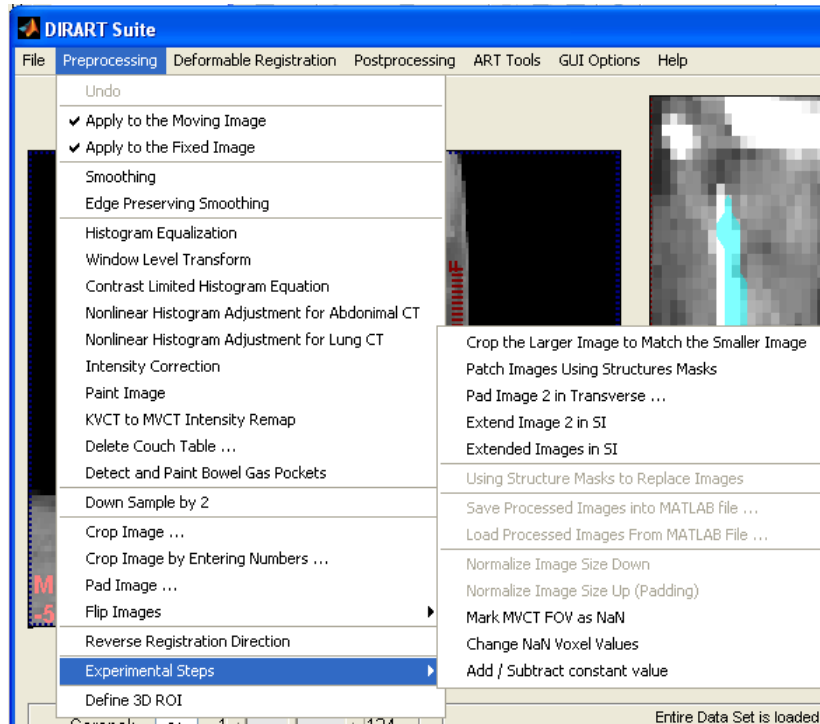


Figure 15: Image preprocessing menu screen shot

Users can control the target image of the preprocessing by using the menu item “Preprocessing → Apply to the Moving Image” and “Preprocessing → Apply to the Fixed Image”. Preprocessing will only be applied to the selected images.

Table 4: Image preprocessing tools

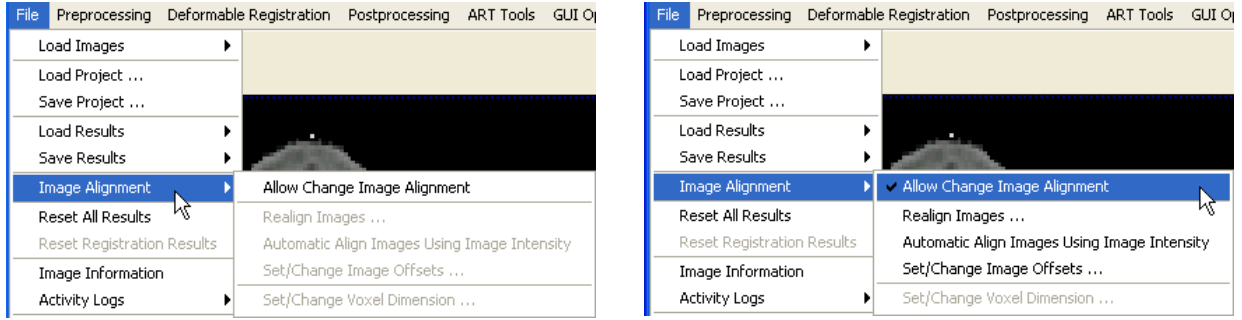
Preprocessing procedures	Explanation
Smoothing	Gaussian low pass filtering, sigma = 1 pixel
Edge preserving smoothing	
Histogram equalization	
Window Level Transform	This is for image contrast enhancement. Using the current window level setting to transform the image intensity. $I \leftarrow \min(\max(I, c+w/2), c-w/2) - (c-w/2)$ c: window center, w: window width, I: image intensity After the transformation, window level setting will be updated accordingly.
Nonlinear histogram adjustment for abdominal CT	
Nonlinear histogram adjustment for lung CT	
Intensity correction	For MR images, to correct the image intensity heterogeneity. Don't use this feature for other image types. Even for MR images, this feature does not usually work well.
KVCT to MVCT intensity remap	For KVCT-MVCT registrations, to be used on KVCT only
Detect and paint bowel gas	For abdominal CT with bowel gas

Down sampling by 2	Use GPReduce filter to down sample the image
Crop image	
Flip images	
Reverse registration directions	Swap image 1 and image 2
Normalize image size Up / Down	These are old features, no need anymore

10 Image alignment

DIRART always asks users to perform the initial image alignment right after two images are loaded (Figure 12). DIRART does not allow further re-alignment unless users explicitly enable it (Figure 16b), in which case all the computed DVFs and deformed images are invalidated.

DIRART offers 4 ways to align the images: 1) aligning using centers of the structures and the images (Figure 12), 2) automatic alignment based on image intensity, 3) manual alignment using shortcut key combinations and/or mouse, 4) manually enter the image voxel shiftings.



(a) Image alignment is not allowed yet

(b) Image alignment is allowed

Figure 16: Image alignment is only allowed after user explicitly enabled it from “File→Image Alignment→Allow Change Image Alignment”.

After image alignment is allowed, users can use key combinations to move two images relatively. Please see the later sections for the key combinations. After alignment is finished, users should lock the feature by disallowing image alignment (Figure 16a).

11 Deformable registration

11.1 Introduction

For two images I and J , while I is the moving image (or reference image) and J is the fixed image (or target image), deformable image registration is to compute the deformation vector field V in order to optimize the system energy equation:

$$E = \int_{\Omega} S(V(I), J) d\Omega + \alpha^2 \int_{\Omega} R(V) d\Omega \quad (1)$$

where V is the deformation vector field, also often called motion field, optical flow, etc, $V(I)$ is the deformed moving image I , S is the similarity function, R is the smoothness constraint function, Ω is the image domain, and α is a constant.

11.2 Definition of the deformation vector field (very important)

V is defined on the coordinates on J . It is the “pull-back” vector field, has the same array dimension as J . Each element of V is a 3D vector, associated with a voxel in J , and defines

“which voxel in I does the voxel in J comes from”. In another word, V defines the coordinates transformation (mapping) from J to I . V is defined on the grid of J , not on the image grid of I . The direction of the 3D vector is from the point in I to the matching point in J .

The way V is defined for deformable image registration is different from how motion is defined for rigid registration. Users usually think that a motion field defined how every voxel in the moving image moves. Such a motion field is often called as “*push-forward*” motion field.

In fact, the “*pull-back*” motion field is the “*push-forward*” motion field if the image registration direction is inverted (the two images are interchanged).

If the moving image and the fixed image have been determined for a registration computation, the most important difference between the two motion fields is that:

- The pull-back motion field is defined on the voxels in the fixed image
- The push-forward motion field is defined on the voxels in the moving image

Because the two motion fields are not defined for the same voxels (or on the same coordinate system), they are not directly negative to each other. Also because the coordinate mapping is not necessary one-to-one, given one motion field, the other one may not even “*well*” exist.

A “push-forward” motion field is often desirable for a certain application, but it cannot be used to compute the deformed moving image because a motion field defined on the target voxel is desirable to compute the deformed moving image, which is the “pull-back” motion field.

Giving one motion field, there are methods to directly compute the other one, if the given one is well behaved (smooth, positive Jacobian). Such procedures are called “*motion field inversion*”. In fact, a motion field inversion tool is included in the software package under “main menu → Postprocessing → Compute Reverse Motion Field”. This topic will be further discussed in the advanced topic sections.

The computed deformation vector field is in unit of pixel (or voxel for 3D), not in real physical unit like mm or cm. Users should have known the voxel size of the images

11.3 Requirements for the images

Requirements	Explanation
Image intensity matching	<ul style="list-style-type: none"> • Both images from the same image modality. • Similar image modalities, like kVCT to MVCT, or MR images obtained using different MR scan protocols, are not considered to be the same imaging modality since image intensities do not match between the two images • It is possible to perform certain kind of intensity mapping procedures, for example, to convert kVCT intensity to MVCT, then registration can be carried out.
Image Resolution Matching	<ul style="list-style-type: none"> • Voxel sizes of both images must be the same. If they are not the same, one or both images must be resampled so that the voxel sizes are the same.
Image Size	<ul style="list-style-type: none"> • The image #1 (the moving image) must completely cover the image #2 (the fixed image) for asymmetric registration • Both images must have the same dimensions for inverse consistent registration

- If this requirement cannot be met, image padding, cropping could be performed in order to meet the requirement. Please see the advanced topics about registration between mismatched images

11.4 Two deformable registration frameworks

We have implemented two deformable registration frameworks in DIRART, the regular asymmetric framework and the inverse consistency framework. But frameworks support multigrid, multiple passes and configurable smoothing between multigrid stages and between passes.

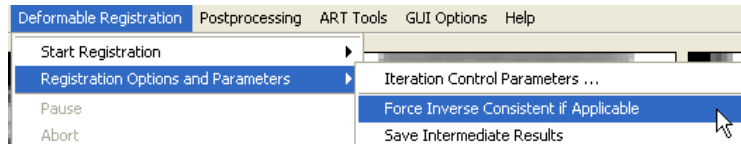


Figure 17: If the menu option “Force inverse consistency if applicable” is turned on, the inverse consistency frame work will be used instead of the asymmetric framework.

11.4.1 Asymmetric registration framework

In this framework, the image #1 is the moving image, and the image #2 is the fixed image.

11.4.2 Symmetric / inverse consistency registration framework

In this framework, both images are deformed towards each other.

11.5 Iterations and multi-resolution

11.5.1 Multigrid approach

Multigrid is an approach to down sample the images, and to perform image registration sequentially from low resolution to high resolution. After registration is finished in one image resolution stage, the result will be used as the initial condition for the next image resolution stage. Multigrid approach, up to 5 stages, is supported. Images are down sampled by factor of 2 for each stage.

11.5.2 Multiple pass approach

Multiple pass approach is similar to multigrid approach. The idea is to perform registration in multiple times on the same image resolution stage. Because the result of image registration is often not good enough in one computation, registration can be computed again in a new pass based on the result of the previous pass.

Multiple pass approach⁶ makes it possible to perform registration in multiple passes, and for each pass, the registration is computed with small number of iterations. Multiple passes with small number of iterations for each pass would generate better results than one pass with larger number of iterations.

Multiple pass approach is particularly useful for a few algorithms:

- Horn-Schunk (HS) optical flow algorithm⁷
- Lucas-Kanade (LK) optical flow algorithm⁸

It is not very useful for demons algorithms, because demons algorithm does iterative registration in a way similar to the multiple pass approach inside the algorithm itself.

11.6 Parameter for registration iteration control

To set the registration parameters, use “main menu → Deformable Registration → Option and Parameters → Iteration Control Parameter”. A dialog box, shown in Figure 18, will come up for user to enter the parameters.

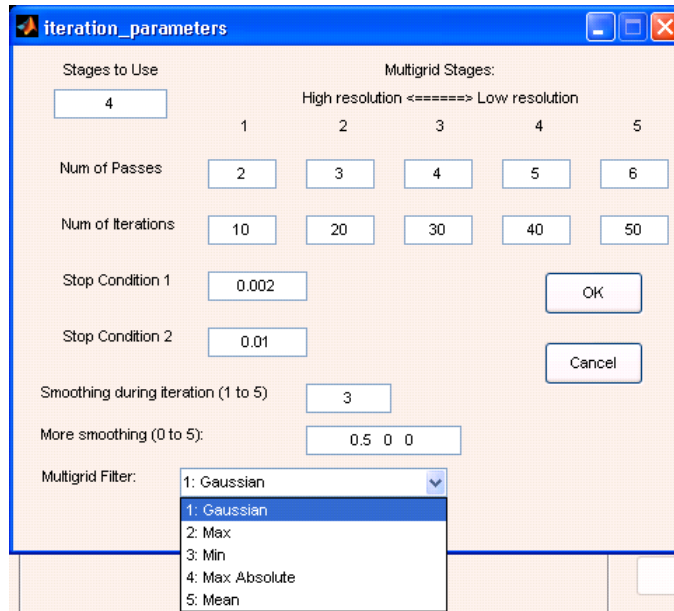


Figure 18: Deformable registration iteration control parameters dialog box

Most parameters are self-explanatory from the field names in the dialog box. A few parameters need to be further explained.

Table 5: Deformable registration iteration control parameters

Parameters	Explanation							
Stages to use Default = 4	Number of multigrid stages to use							
Num of passes Defaults = 2, 3, 4, 5, 6	Number of passes to use for each individual stages							
Num of iterations Defaults = 10, 20, 30, 40, 50	Number of iterations to compute for each pass, set for each individual stages							
Stop Condition 1 Default = 0.02	The iteration stop condition. Iteration will stop if the adjustment of motion field is less than this condition. The value is in unit of pixel for 2D or voxel in 3D.							
Stop Condition 2 Default = 0.001	The stop condition for the multiple passing. Pass will stop if the adjustment of motion field is less than this condition. The value is in unit of pixel for 2D or voxel in 3D.							
Smoothing during iterations Default = 3	<p>The values are designed to control the smoothing operation in the iteration.</p> <p>For HS algorithm⁷, the value is unit-less. It is remapped to the α^2 parameter in the optical flow equation.</p> <table><tr><td>Value</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>>5</td></tr></table>	Value	1	2	3	4	5	>5
Value	1	2	3	4	5	>5		

	α^2 0.05 0.1 0.2 0.3 0.4 0.5
	For demons algorithms ⁵ , the value is the Gaussian low-pass filter window size, in voxels
Smoothing after passes Defaults = [0.5 0 0] voxels	Value 1: after each pass, the deformation vector field computed by this pass could be smoothed by a Gaussian low pass filter. This value is the sigma of the Gaussian low pass window. If the value is 0, then no smoothing is performed. Value 2: smoothing of the composed motion field after each multigrid stage Value 3: smoothing of the composed motion field after the final stage
Multigrid filter	The filter to compute the image intensity during image down sampling for the multigrid approach. By fault, the Gaussian filter is used. However, Gaussian filter does not always give best results. For lung, the max filter is often a better choice.

11.7 Algorithms and variants

The software tool supports multiple deformable registration algorithms and variants. They are categorized into a few groups and listed in Table 6.

Table 6: Deformable image registration algorithms and variants

Groups	Variants	Note
Horn-Schunck (HS) optical flow algorithms ⁷		I have found that this algorithm working well for most images. It is also usually running faster than demons algorithm because it computes the deformed moving image in less number of times. Suggestion: try to use this algorithm for most images, also try to use different smoothness setting
	Integer	All computation is performed as integers. The deformation vector fields are also defined as integers. The algorithm is implemented in this way in order to save memory during computation for big images. Accuracy of the results is ok, but not very impressive. Suggestion: not to use this algorithm
	Memory saving	Similar to the original HS algorithm, with certain

		<p>extra steps in order to reduce the memory usage. Certain new features available in the HS algorithm are not implemented for this algorithm.</p> <p>Suggestion: Try it if you have out of memory problems. However, you will still get out of memory problem for most situations.</p>
	With Issam's non-linear smoothness	<p>This is the HS algorithm with different smoothness scheme during iterations.</p> <p>Suggestion: Try it, it may work better for you for certain images.</p>
	With divergence constraint	<p>Divergence constraint is for volume preservation. Therefore the algorithm is only for certain type of images that volume preservation may be true.</p> <p>Computation is much slower.</p> <p>Suggestion: It may work better for some images, but it is slower than the original HS algorithm.</p>
Lucas-Kanade (LK) optical flow algorithm ⁸		<p>Based on LMS principle.</p> <p>The original algorithm is really not working well for most images. It will probably work best for highest contrast images with feature points.</p> <p>Suggestion: It usually does not work well.</p>
	Improved	<p>The original LK algorithm with directional smoothing.</p> <p>It is working fairly well, comparable to HS algorithm. The computation speed is not impressive because of the extra computation of directional smoothing.</p> <p>It works better with multiple pass approach.</p> <p>Suggestion: It works fine. It is not very fast even though it is not iterative.</p> <p>It will work well if users are only interested to register isolated feature points instead of the whole image, but such a registration scheme is not implemented.</p>
Combined HS and LK algorithm		<p>It is not working as good as promised for most images. And the iteration usually converges much slower than the original HS algorithm.</p>

		Suggestion: Users should give it a try.
Level set algorithms		Suggestions: They are not bad algorithms, but may be not even worth to try because there are better algorithms. And the level set program files are lack of maintaining, could be buggy.
	Hierarchical ⁹	
	Affine approximation	
	With iterative smoothness	
Demons algorithms ⁵	Original ⁵	This is a very famous algorithm, working well for most images. Suggestion: It works, but not very fast, maybe too much smoothing on the deformation vector field, which can be changed by minor changes on the program code. The HS optical flow algorithm has been used more by the author.
	Modified	The algorithm is modified in the way that the gradient of the moving image is used instead of the gradient of the fixed image. Suggestion: Result should be comparable to the original demons algorithm.
	Symmetric force ¹⁰	Suggestion: Should work better than noisy images.
	Double force	Suggestion:
	Fast	
	Fast with elastic regularization	
Iterative SSD minimization		
Free deformation algorithm ¹¹		Slower, works ok, not better, comparing to HS and demons algorithm
Algorithms from ITK	<ul style="list-style-type: none"> • Demons • Symmetric demons • Level set • B-spline 	The deformable registration algorithms in ITK have been compiled into DLL files. The registration software calls the DLL, which is actually the ITK code, to perform computation. Suggestion: buggy, not always working. And the ITK demons algorithm, which should work fine, is usually working worse than the MATLAB demons implementation. And the ITK codes, that are written in C or C++, should run much faster,

		are actually not running faster than MATLAB code.
		B-spline algorithm should work for single modality or multimodality, if it works.

11.8 Other options for registration computation

There are other options under “Deformable Registration → Options and Parameters”. They are explained in Table 7.

Table 7: Other options for deformable image registration

Options	Explanation
Force inverse consistent if applicable	Enable the inverse consistency framework
Save intermediate results	The multigrid and multiple pass framework will save the intermediate results into MATLAB files
Log output	Log all MATLAB command output messages into a file, for debugging purpose.
Generate inverse consistent Motion Field	For inverse consistency algorithms, after all computation is finished, the deformable vector field in the inverse direction will be also computed (with minimal computation)
Total intensity preservation modulation for lung CT	When generating the deformed moving image during the computation, applying the total intensity preservation modulation. This is useful for lung CT. However, this feature is quite experimental. Don't use it for now.
Use both image gradient	If enabled, will always use the average image gradient of both images for everywhere image gradient is applied in the computation.
User previous results and continue	This option allows the previous registration to continue.
Region smoothing	Retired feature, not supported anymore
Don't deform the regions	Retired feature, not supported anymore

11.9 Start registration

After the images are loaded and preprocessed, user can select a registration algorithm under “Deformable Registration → Registration Algorithms”. To start registration computation, use “Deformable Registration → Start” (Figure 19).

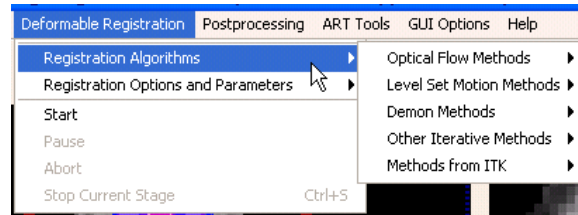


Figure 19: Screen shot of the deformable registration menus

It takes time to computation is fully finished. The displayed images will be updated during the computation, thus users can see the progresses. A lot of messages are reported on the MATLAB command line.

The “Pause” and “Abort” buttons will be enabled for use to pause or to abort the computation. The ‘Stop Current Stage’ menu will be enabled for use to stop the registration in the current multi-grid stage and continue to the next stage.

11.10 Results

The important results are the DVF and the deformed images. Users must remember that the deformation vector fields are in unit of voxel, not in mm or cm, and it is the “pull-back” motion field defined on the fixed image, not a “push-forward” motion field.

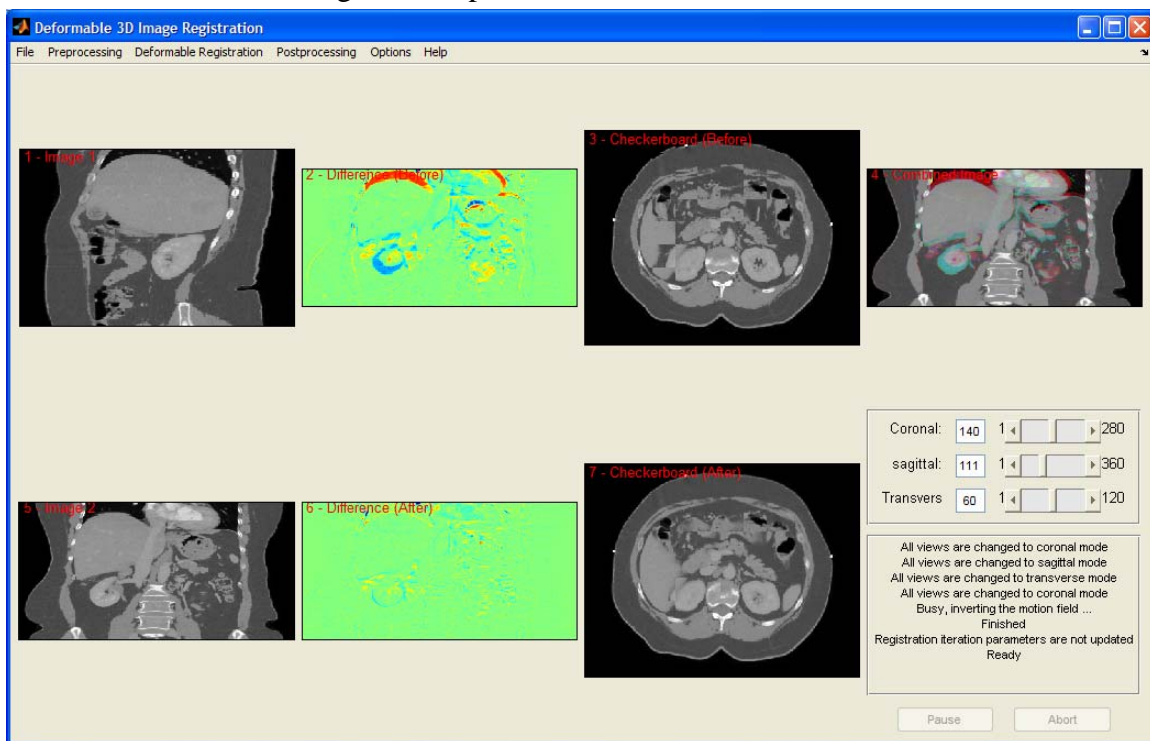


Figure 20: An example of liver CT-CT registration

12 Validation and post-registration analysis

After DVF is computed, the following validation and analysis procedures are available, as shown in Figure 21. Some options are only available after the DVF or the inverse DVF is available. Other options are only available after required procedures have finished.

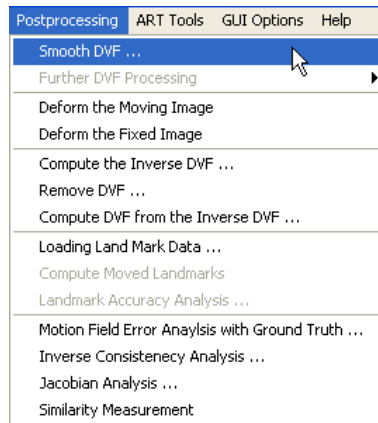


Figure 21: Screen shot of GUI menus for operations after deformable registration computation

12.1 Smoothing the DVF

DIRART allows DVF or the inverse DVF to be smoothed using Gaussian low-pass filter.

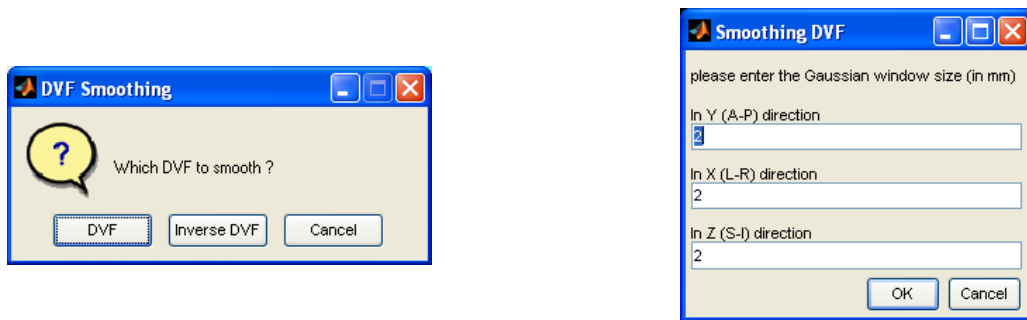


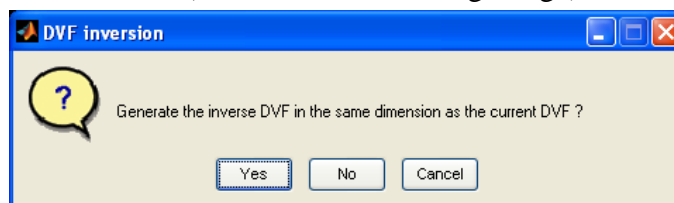
Figure 22: Smoothing DVF by entering the Gaussian smoothing window size

12.2 Deform the moving image or the fixed image

Users can use the DVF to deform the moving image, or use the inverse DVF to deform the fixed image.

12.3 Invert the DVFs

Users can compute the inverse DVF from the regular DVF (defined on the fixed image), or the regular DVF from the inverse DVF (defined on the moving image).



12.4 Remove the DVFs (from DIRART)

Users can delete the regular DVF or the inverse DVF from DIRART data memory.



12.5 Validation

12.5.1 Landmark based validation

* The format of the landmark data files is not standardized. In theory, landmark data should be in the format of point pairs with each point has its 3D coordinates. The current DIRART codes need to be fixed so to support such landmark data files. For now, users may need to modify the DIRART code to support their own landmark data files.

12.5.2 Ground truth based validation

* The format of the DVF data files is not standardized, either. For now, users may need to modify the DIRART code to support their own ground truth DVF data files.

12.6 Inverse consistency analysis

Inverse consistency analysis procedure is available if both the regular DVF and the inverse DVF are computed. The following reported will be printed at the MATLAB command line, as Table 8. Error histogram is plotted as Figure 24. The whole 3D volume of the inverse consistency error magnitude is also exported to view3dgui tool so that users can exam the errors visually (Figure 23). In addition, the inverse consistency errors are saved in the DIRART data memory. Users can then select to view the errors as color wash (Figure 25).

Table 8: Inverse consistency analysis report

```
=====
Reverse consistency analysis results:
=====
Mean = 1.092134e-001, std = 3.051496e-001
Median = 3.337924e-002, max = 4.548387e+000
WO boundary: Mean = 6.931237e-002, std = 1.685943e-001
WO boundary: Median = 3.265673e-002, max = 4.226503e+000
```

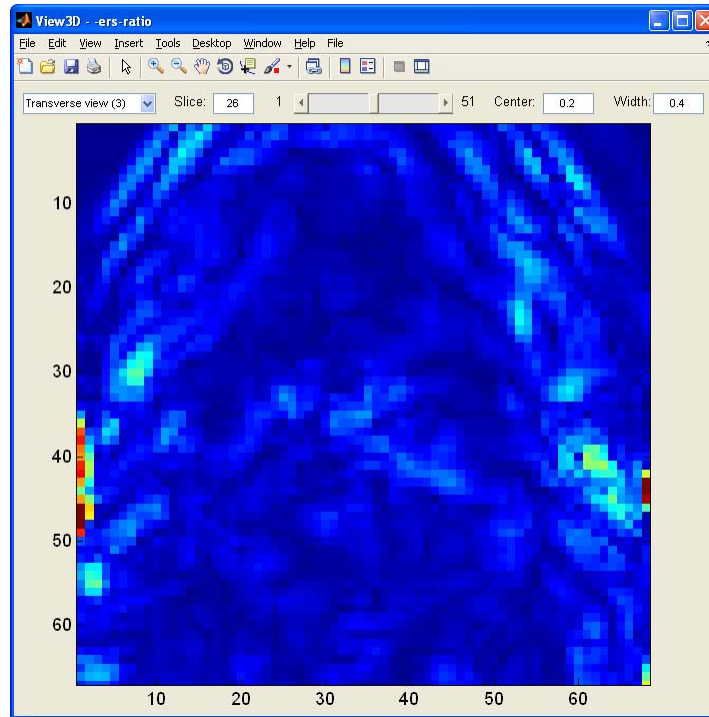



Figure 23: Viewing inverse consistency error 3D volume in the view3dgui tool.

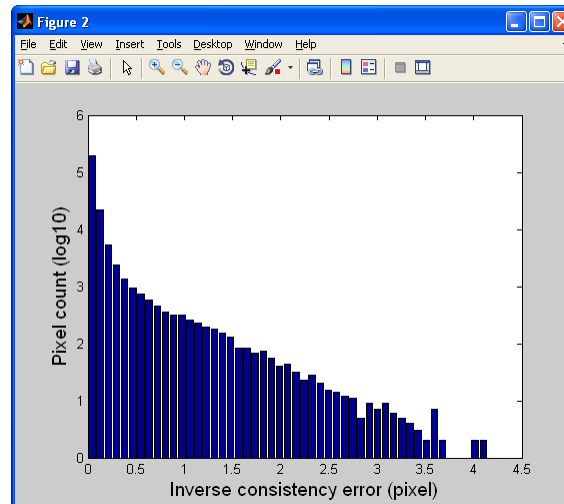


Figure 24: Inverse consistency error histogram plot

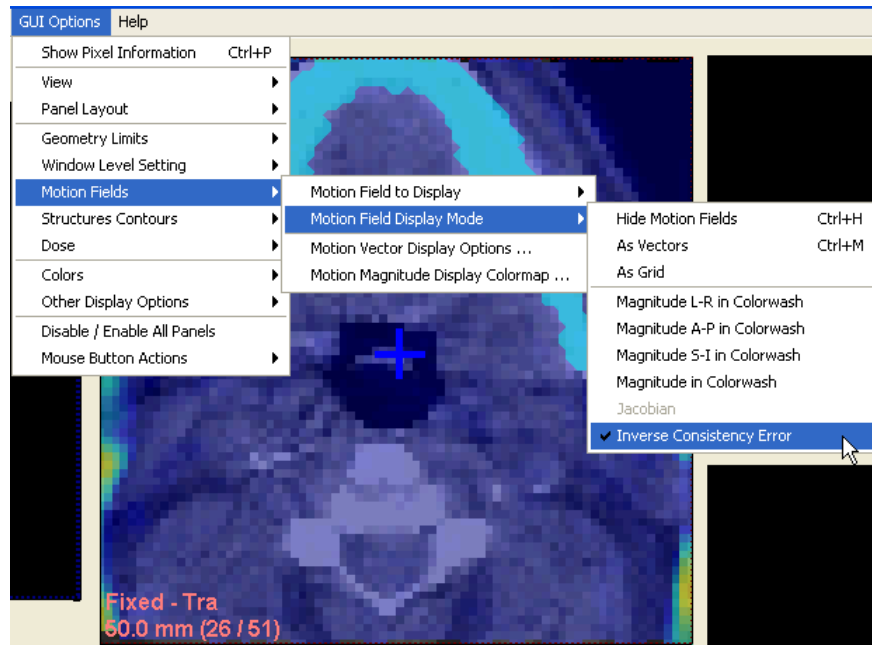
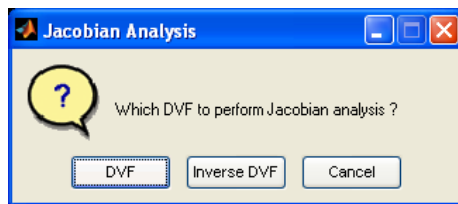


Figure 25: Viewing the inverse consistency error as color wash, overlaid on top of the images inside DIRART. The GUI options menus indicate that the such a DVF display mode is turned on.

12.7 DVF Jacobian analysis



(a)

```
=====
Jacobian analysis results:
=====
Mean = 1.005402e+000
STD = 4.638178e-001
Median = 9.845884e-001
Value <= 0: 1.4 percent
```

(b)

Figure 26: (a) Select a DVF to compute Jacobian. (b) Jacobian analysis report example

12.8 Image similarity measurement

Image similarity measurements before registration are computed between the moving image and the fixed image, and computed between the deformed moving image and the fixed image for after registration reports.

MI (mutual information), NMI (normalized MI), CC (Cross-correlation), MSE (mean square error), entropy are computed and reported, only for the image regions that both images overlap each other.

```
=====
Similarity measurement:
=====
Before registration:
MI = 2.25806
NMI = 1.23063
CC = 0.72850
MSE = 80160
Entropy = -14091

After registration:
MI = 3.03360
NMI = 1.30322
CC = 0.89664
```

MSE = 50515
Entropy = -12936

Figure 27: Image similarity measurement reports

13 GUI and Visualization

Visualization is one of the most important features in DIRART. In fact, DIRART has so many visualization options, many of which are fairly unique. This section will explain all these options. Figure 28 demonstrates some of the DIRART visualization possibilities.

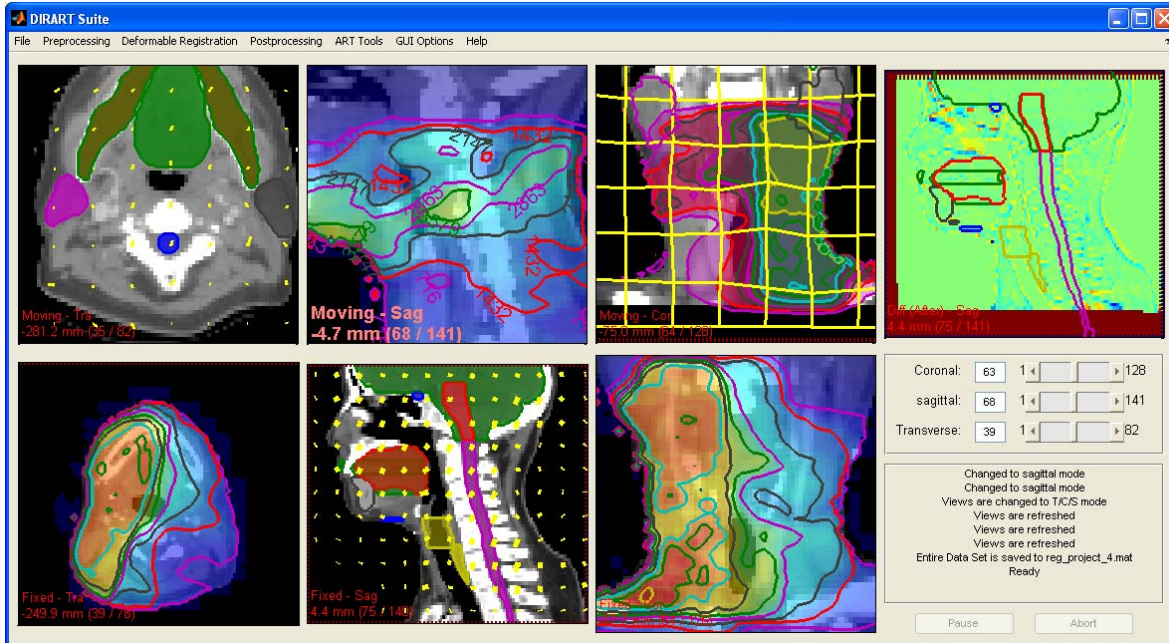


Figure 28: Screen shot of the DIRART main GUI.

13.1 General features

1. 3D Images and results can be viewed in 3D directions: coronal, sagittal and transverse (Figure 29).
2. 3D images and results can be viewed slice by slice. Users can use the slider controls in the 3D view control panel to change the current viewing slice. Users can also directly enter the slice number in the text input fields next to the slider controls (Figure 28).
3. Users can right click the mouse on the image viewing panels to bring up the context sensitive popup menu (Figure 30). Most visualization options are accessible through the popup menu.
4. The "GUI Options" menu on the main menu bar has more GUI options.
5. There are multiple display panels that can be individual configured to display different image in different options.
6. Besides the base images, other information, including doses, structures, DVFs, boundary boxes, etc, are displayed on top of the base images.

13.2 3D viewing directions

3D data can be viewed in one of the 3D directions: coronal, sagittal and transverse. The current viewing direction is labeled for each display panel, as demonstrated in Figure 29.

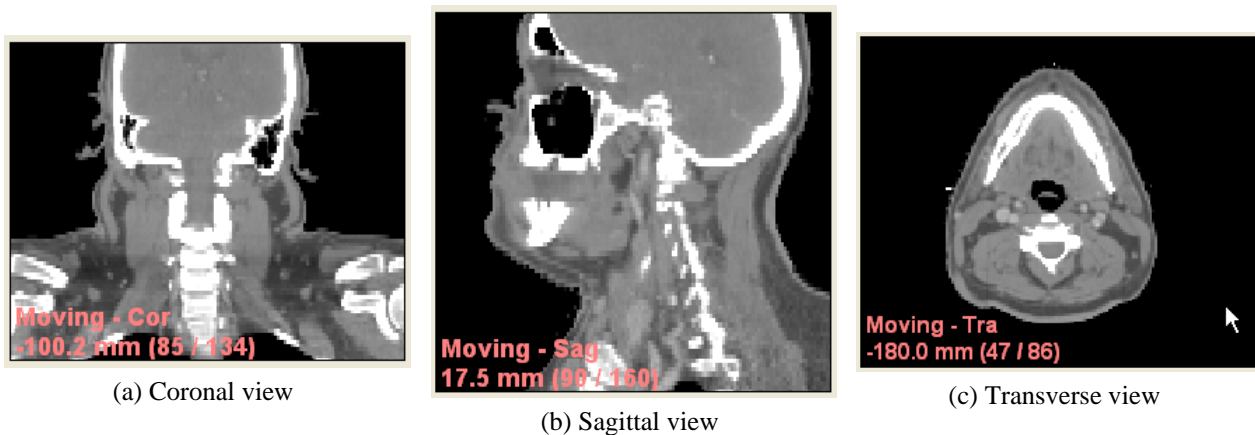


Figure 29: 3D viewing directions. The brown

Ways to select the viewing directions:

- 1) Right mouse click popup menu → view → ...
- 2) Main menu GUI Options → View → ...
- 3) Using shortcut keys: press 1 to set all display panels to coronal view, 2 to sagittal view and 3 to transverse view.

13.3 Viewing the images

Any of the image display panel can be set to view images in the different configuration. View modes can be selected from popup menu by clicking the display panel using the right mouse button (Figure 30).

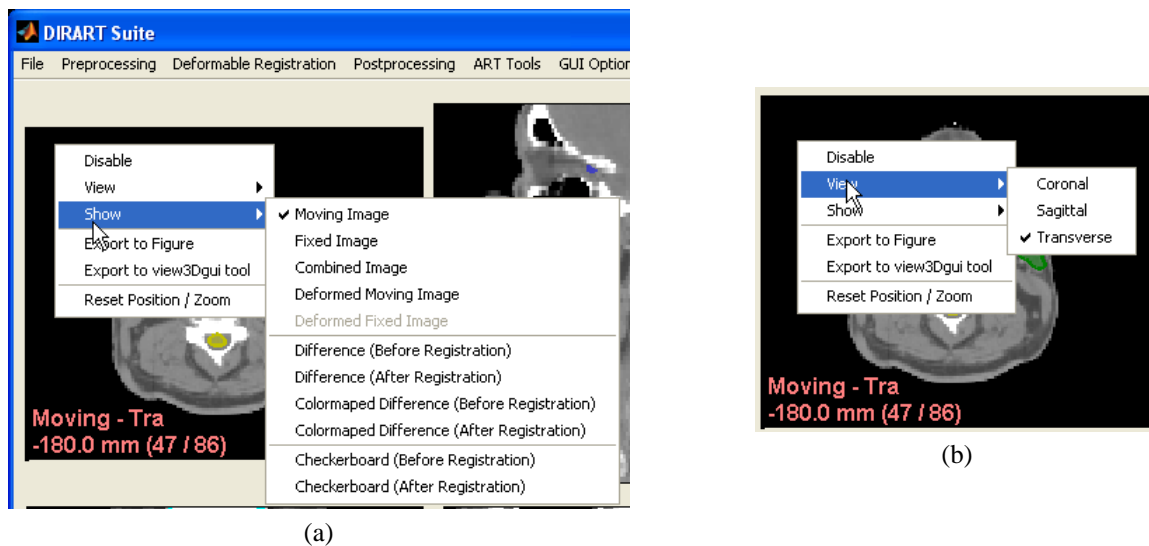

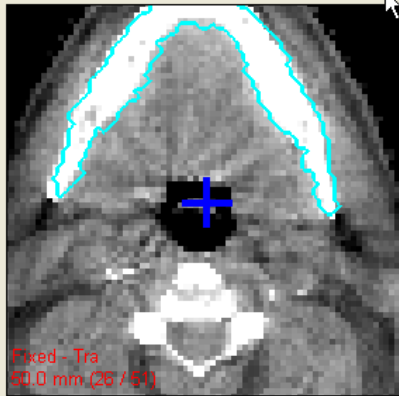



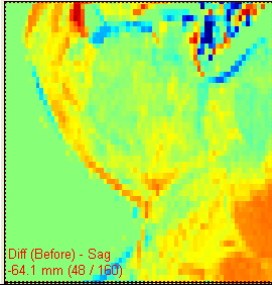
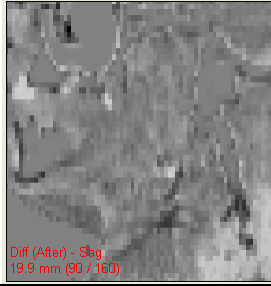
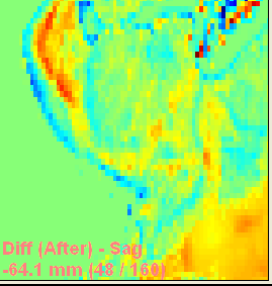
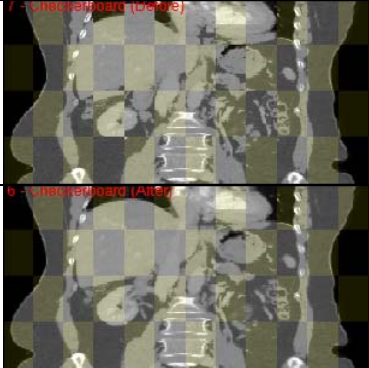



Figure 30: Screen shots of popup menu by placing the mouse cursor on the display images and clicking the right mouse button. The popup menu allows easy accesses to the base image display choices and view directions. Other information, like doses, structures, etc. is displayed by overlaying on top of the base images.

Table 9: Image viewing modes

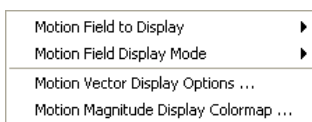
Display modes	Examples
---------------	----------

1	The moving image (image #1)	
2	The fixed image (image #2)	
3	The combined image (image 1 and image 2 are shown in different color channels and displayed together)	
4	The deformed moving image	
5	The deformed fixed image	Similar to mode 4 (only for inverse consistency algorithms)

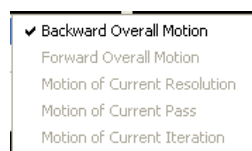
6	The difference image before registration (in gray scale image or in color)		
7	The difference image after registration (in gray scale image or in color)		
10	Checkerboard image before registration		
11	Checkerboard image after registration		

13.4 Viewing DVF

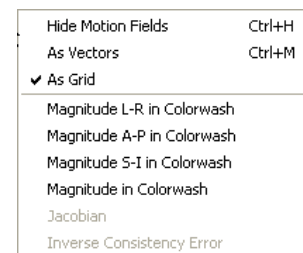
Deformation vector fields can be displayed by overlaying on top of the displayed images. For 3D images, DVFs can be projected into x , y and z directions. Projection on each of directions is in fact a scalar field. Such scalar fields can also be displayed as color wash on top of the displayed images. Furthermore, the magnitude, Jacobian of DVF, and inverse consistency errors can be computed and be displayed as scalar fields.



(a) DVF display options



(b) DVF to display

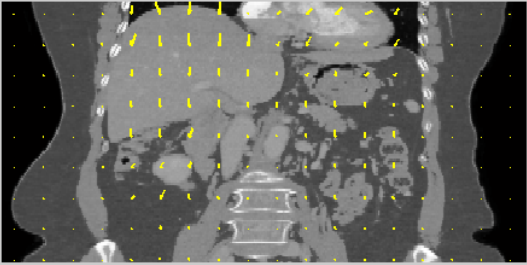
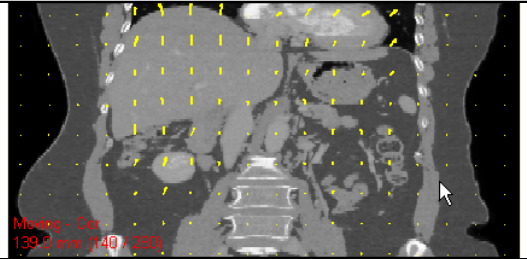
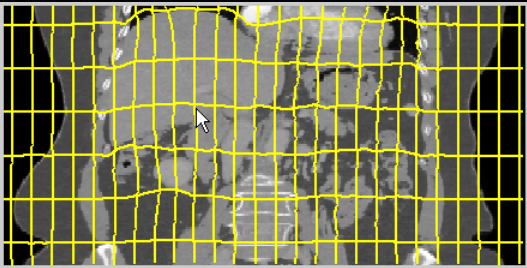
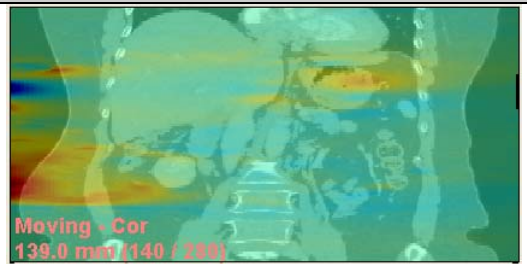
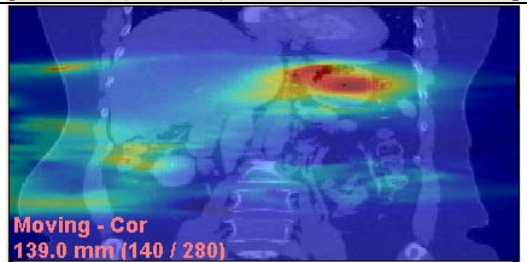
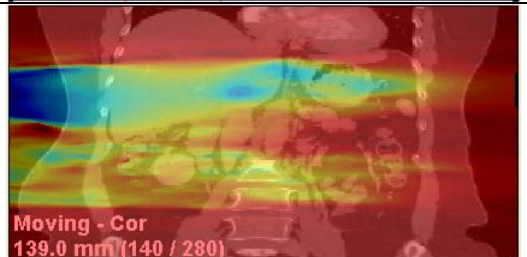


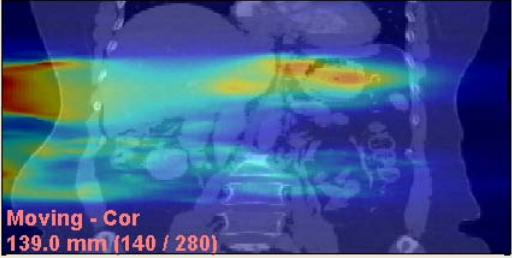

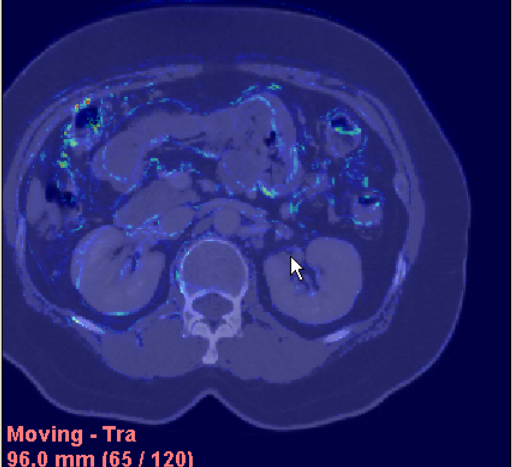
(c) DVF display modes

Figure 31: DVF display options GUI menus

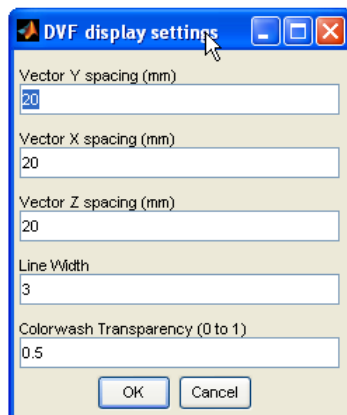
Figure 31 shows the screen shots of DVF display menus. Table 10 shows the example screen shots of DVF display.

Table 10: DVF display modes

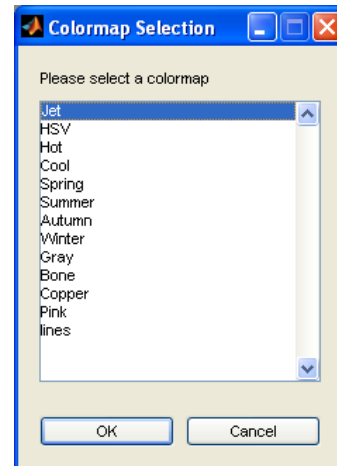
Explanation	Screen shots
<p>“Pull-back” DVF displayed as vector with arrows. (The ends of the arrows are aligned.)</p>	
<p>“Push-forward” DVF displayed as vector with arrows. (The beginnings of the arrows are aligned.)</p>	 <p>Moving - Cor 139.2 mm (140 / 280)</p>
<p>“Pull-back” motion as deforming grids</p>	
<p>DVF projected in L-R directions, displayed as color wash on top of the image</p>	 <p>Moving - Cor 139.0 mm (140 / 280)</p>
<p>DVF projected in A-P directions, displayed as color wash on top of the image</p>	 <p>Moving - Cor 139.0 mm (140 / 280)</p>
<p>DVF projected in S-I directions</p>	 <p>Moving - Cor 139.0 mm (140 / 280)</p>

DVF magnitude	
Jacobian of DVF	
Inverse consistency errors between the forward DVF and the inverse DVF	

There are a few extra GUI options that control more DVF display details, as Figure 32. These extra options control the DVF grid sizes, arrow vector line thickness, color wash transparency and color maps to be used for color wash displays.



(a) DVF display options



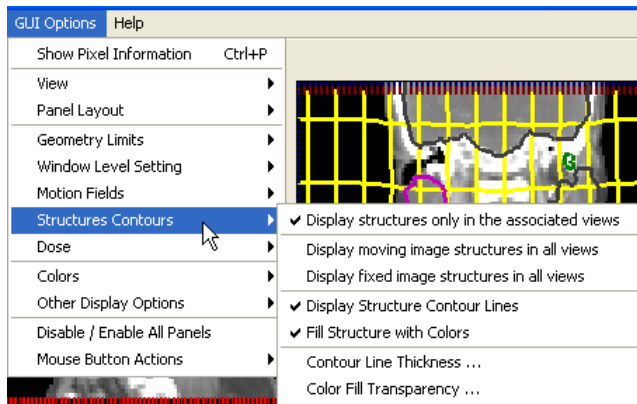
(b) Color map selections

Figure 32: Screen shot of the DVF display setting windows.

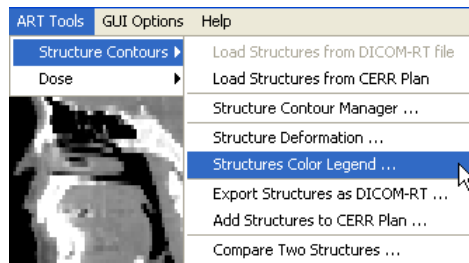
13.5 Viewing structures

In DIRART, structures are loaded from CERR plans and associated with either the moving image or the fixed image. DIRART offers a few options to control how the structures are displayed on top of the images. In general, structures can be displayed:

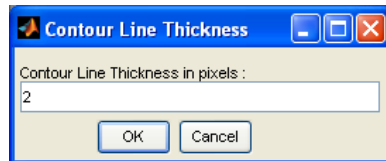
- On top of either image
- In any 3D viewing directions
- As contour lines with configurable line thickness
- With transparent color filling
- Individually turned on / off
- In user configurable colors



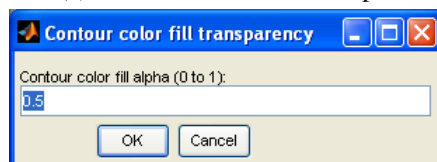
(a) Structure Display GUI options



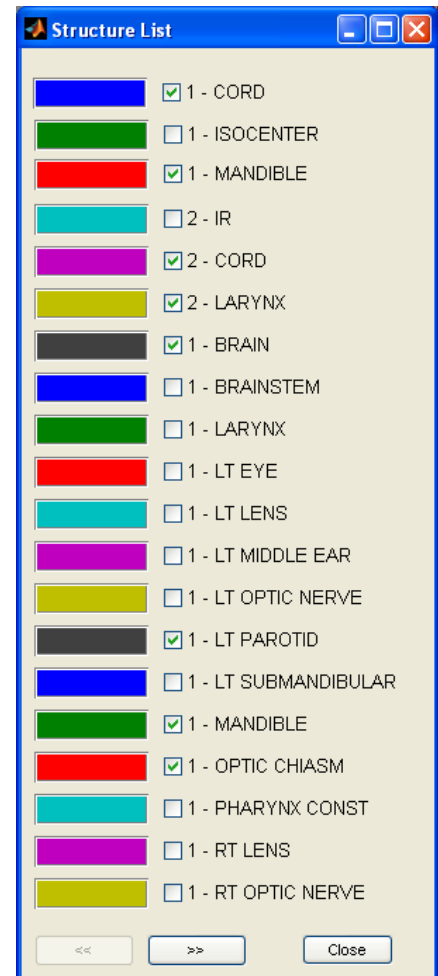
(b) Other structures options under the ART tool menus



(c) Contour line thickness input



(d) Contour color fill transparency input



(e) Structure list window in which users can turn structures on / off and assign colors

Figure 33: Screen shot of structure related GUI options

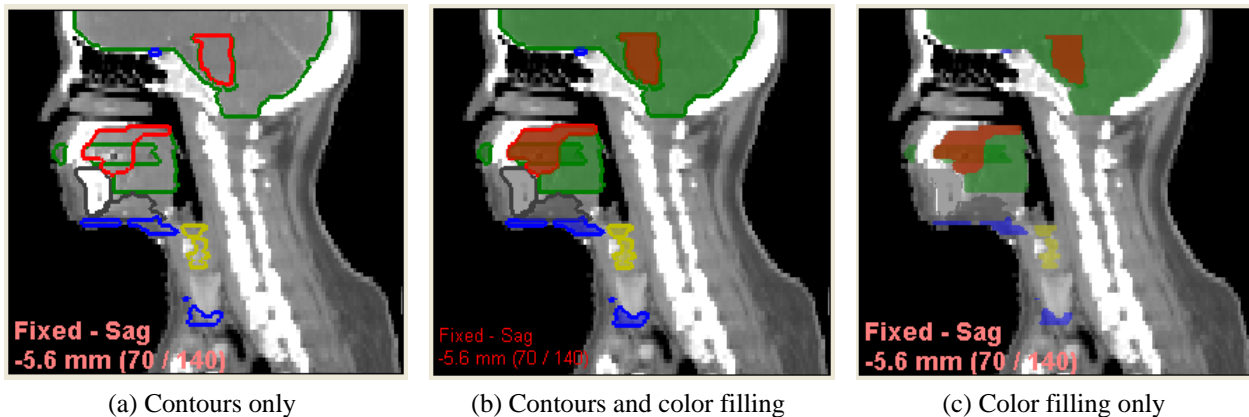


Figure 34: Examples of structures displayed with contours and color filling

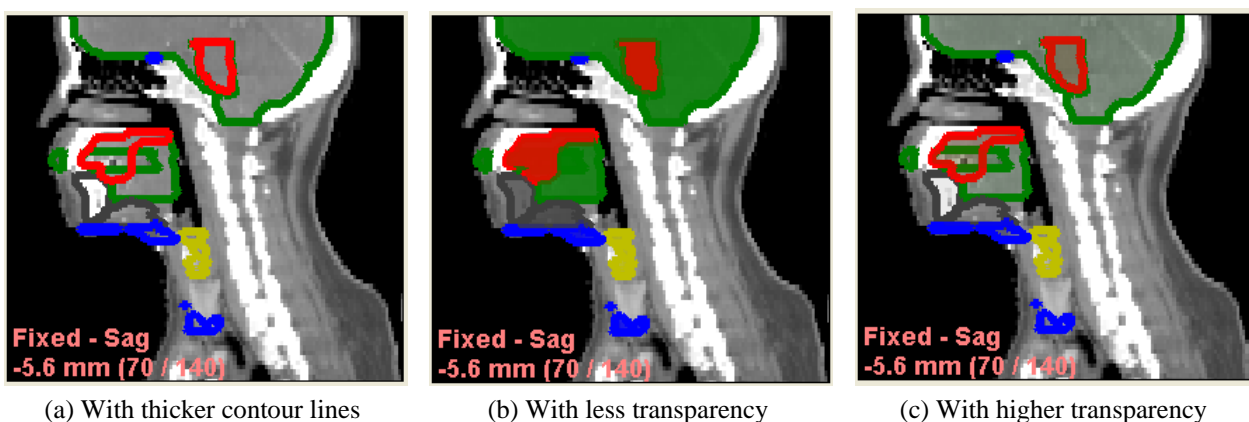


Figure 35: Examples of structures displayed with different line thickness and transparency settings

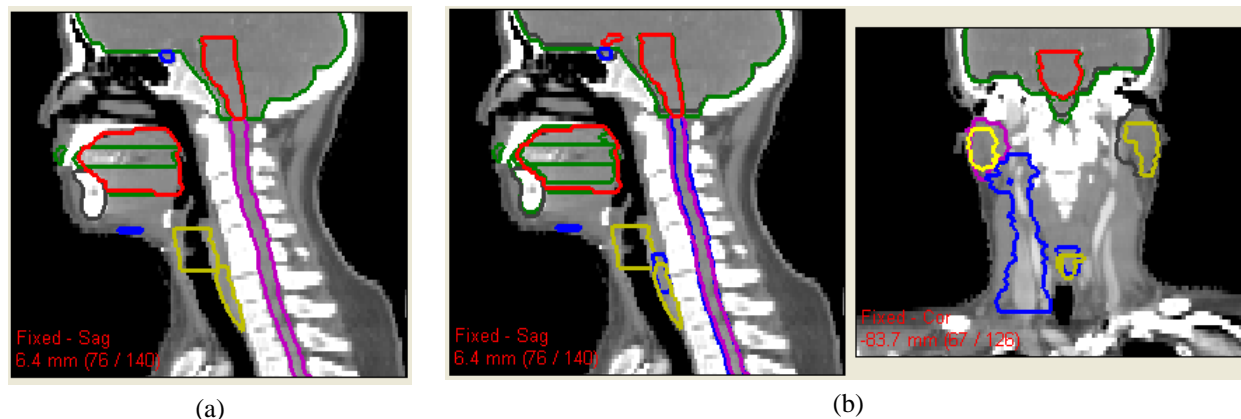


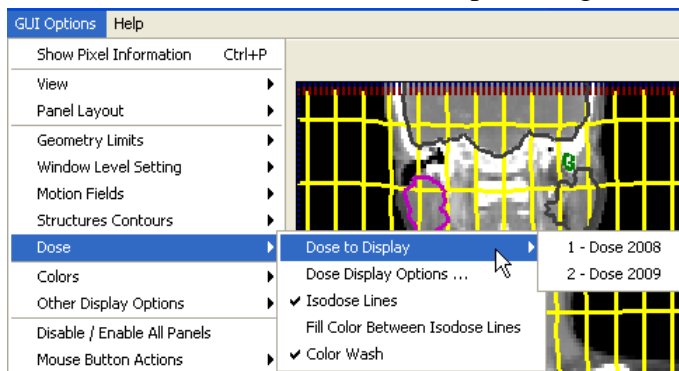
Figure 36: (a) Only fixed image contours are display. (b) Both moving image contours and fixed image contours are displayed on top of the fixed image

13.6 Viewing doses

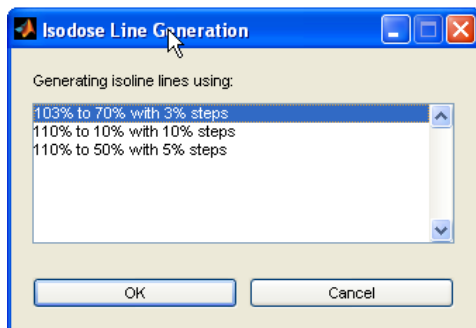
DIRART supports multiple doses that can be loaded from CERR plans and associated with either the moving image or the fixed image. DIRART has a few options to control how the doses are displayed on top of the images. As the options shown in Figure 37, doses can be displayed:

- Only one dose can be turned on for a display panel. Different display panels however can be configured to display different doses.

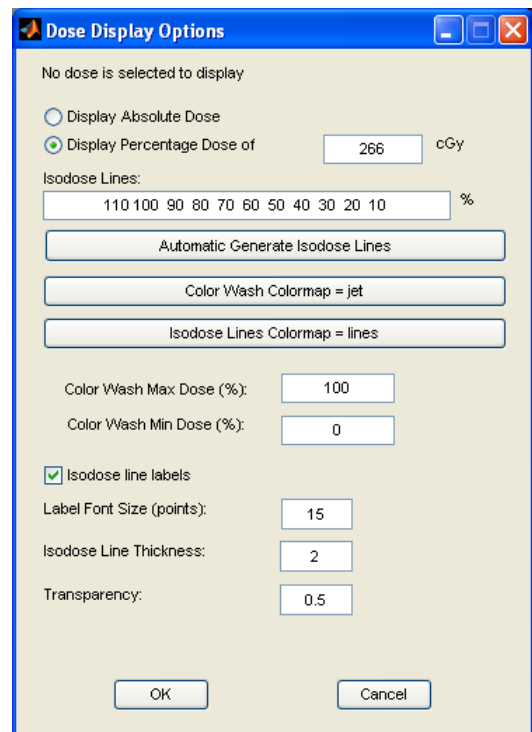
- On top of either image
- In any 3D viewing directions
- As isodose lines with configurable line thickness. Labels for isodose lines can be turned on or off, and with configurable font size. Color filling between isoline lines could be turned on / off.
- As color wash
- In user configurable colors
- As absolute doses or percentage doses



(a) Dose display GUI options



(c) Choices of isodose line automatic generation



(b) The dose display options window screen shot

Figure 37: Screen shot of dose display related GUI options

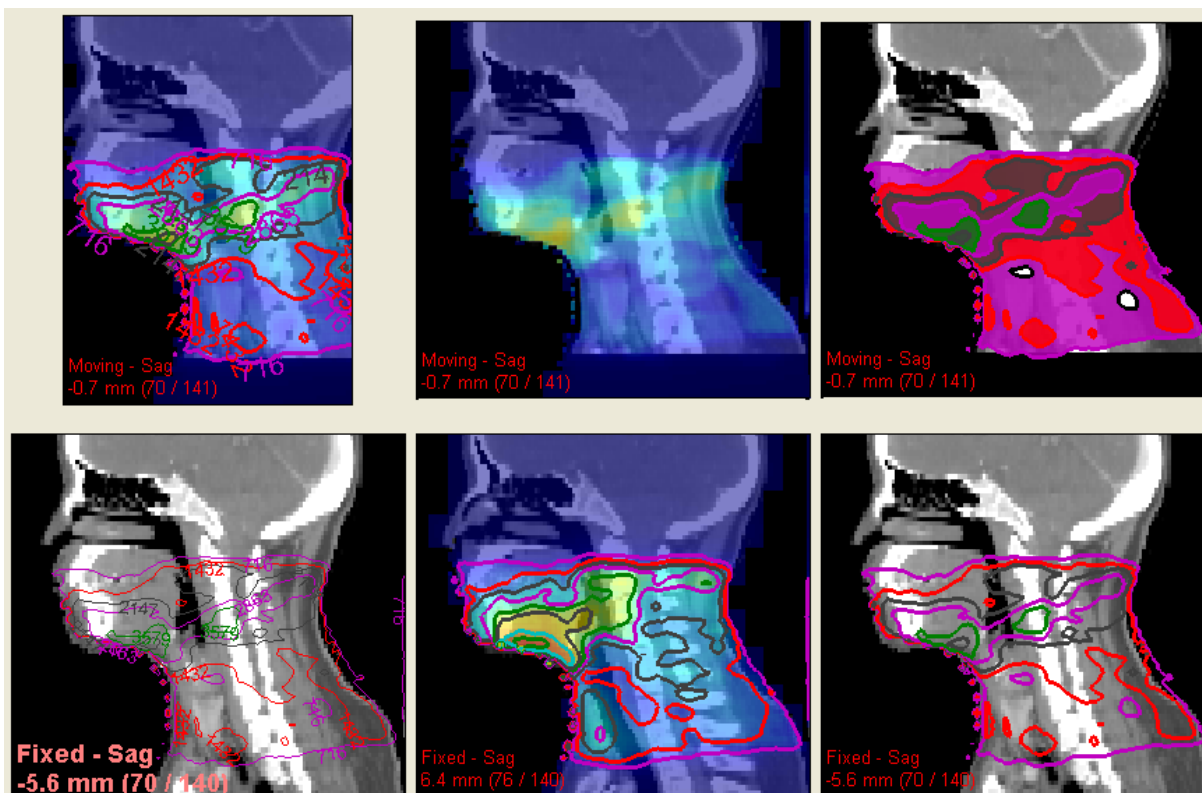
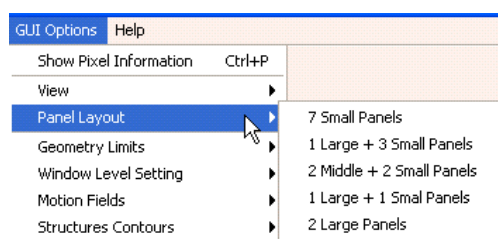


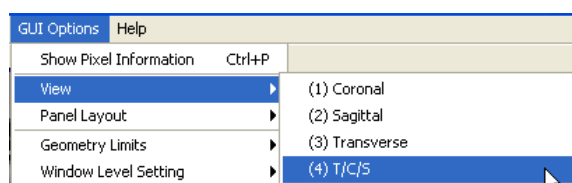
Figure 38: Demonstration of different dose display options, in colorwash, or as isodose lines, with and without color filling, with and without isodose line labels of different font sizes.

13.7 GUI layouts

GUI layouts can be selected from the menus shown in Figure 39. The following layout options are available.



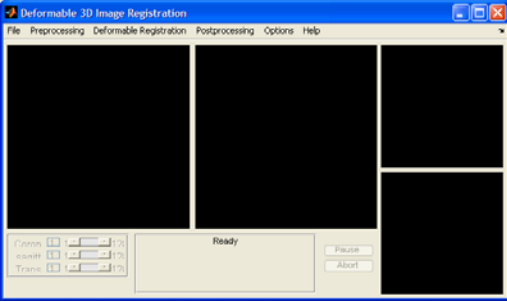
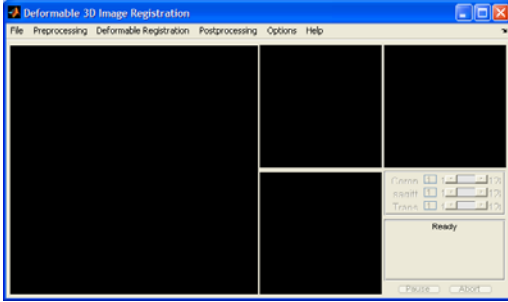
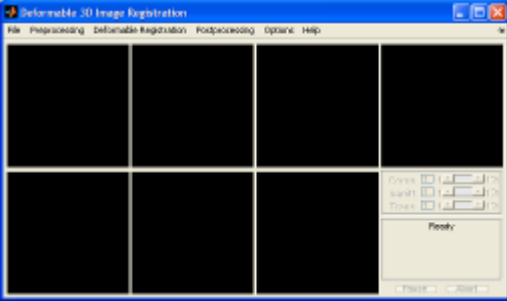
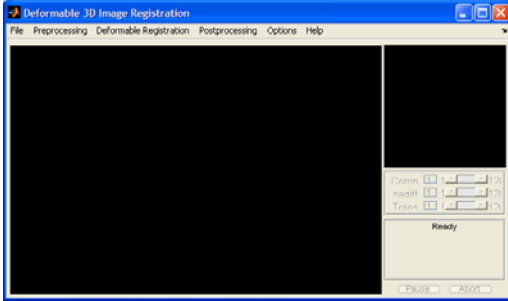
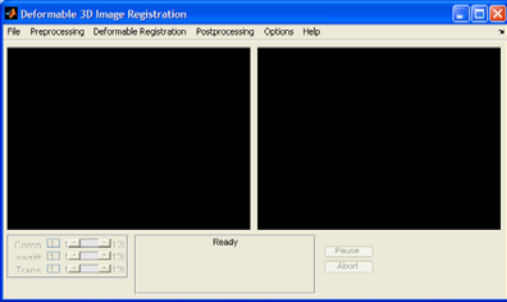
(a) Panel layout menus



(b) The T/C/S view will change the GUI layout to 7 small panels

Figure 39: The panel layout menu screen shot.

Table 11: GUI image display panel layouts

 <p>4 panels: 2 middle size + 2 small</p>	 <p>4 panels: 1 large + 3 small</p>
 <p>7 small panels</p>	 <p>2 panels: 1 large + 1 small</p>
 <p>2 middle size panels</p>	

13.8 Locking and Unlocking display panels

In DIRART, every display panel is individually configurable. For example, DVF display is turned on for one panel, and turned off for the next panel. One panel shows the 1st dose as color wash in the transverse view, and another panel shows the 2nd dose as isodose contours in the coronal view.

All the GUI controls and the active values of GUI options are associated with the active display panel (Figure 40). These GUI controls include the slice number sliders, shortcut and combination keys and mouse actions. User can use mouse to click on a display panel to make it the active panel. When an inactive panel becomes the active panel, all GUI controls and status are updated to reflect the current GUI options / configurations of the active panel.

However, users would like to configure the GUI options for all the display panels together. The menu “GUI Options → View → Lock between Display Panels (L)” can enable / disable such synchronization. If this menu item is checked, when users update a GUI option, this

GUI option will be updated for all the display panels. The “(L)” means this option could be turned on / off by pressing the “L” key from the keyboard.

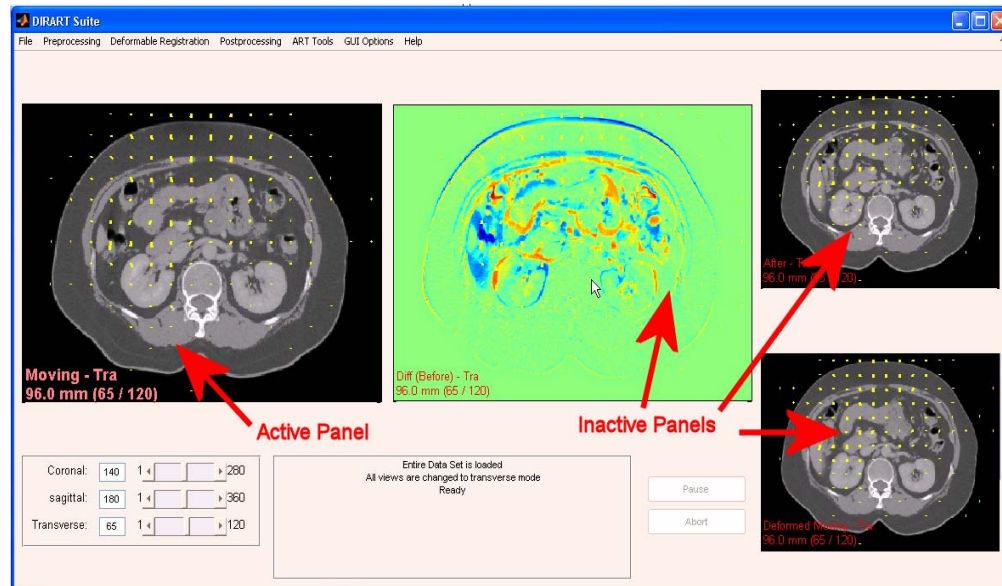


Figure 40: Active and inactive display panels. There is only one active panel, which has a text label of bigger font and lighter font color. All inactive panels have thinner and smaller text labels.

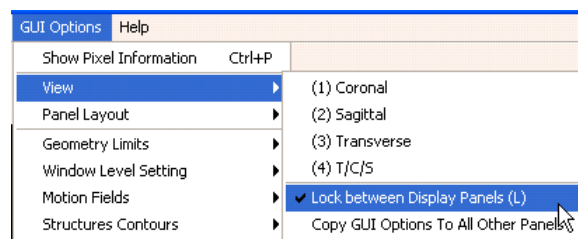
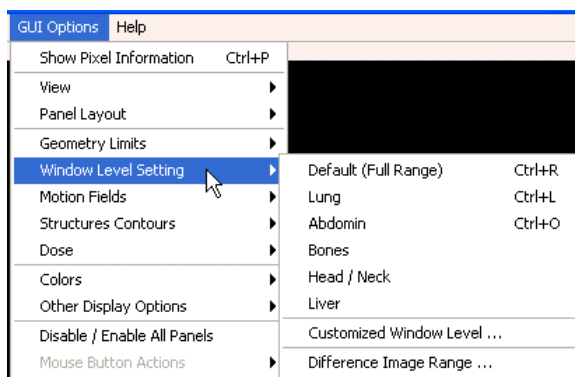


Figure 41: Screen shot of the menu option to lock/unlock between display panels.

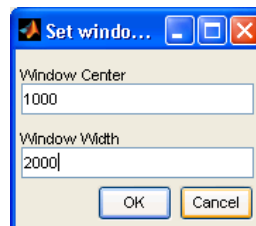
The menu “GUI Options → View → Copy GUI Options to All Other Panels” will copy the GUI configuration of the current active panel to all other display panels. When the display panel lock is turned from off to on, the current configuration of the active panel is automatically copied to all the other panels.

13.9 Window level setting

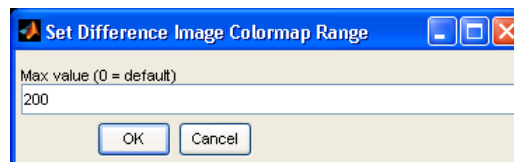
Users can select and set window level setting for image display purpose by accessing “main menu bar → Options → Window Level Setting → ...”.



(a) Window level setting menus



(b)



(c)

Figure 42: (a) Window level settings menus, (b) customized window level setting window screen shot, (c) difference image colormap scale range.

The “difference image colormap range” is for the different image display, equivalent to window width setting while the window center is 0. The current window level setting could also be used in the “Window Level Transformation” image preprocessing procedure. Please see the preprocessing section for more details.

13.10 Zooming and the geometry limits of display

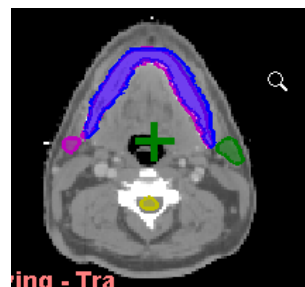
DIRART allows user to adjust the geometrical limits of image display by zooming in / out, resetting zooming and selecting predefined zooming settings. Figure 43 shows the related GUI options. Users can enable zooming feature from “GUI Options → Mouse Button Actions → Zoom (Z)” (Figure 43a) or by pressing the Z key from the keyboard. After zooming is enabled, the mouse cursor of the DIRART window becomes the magnify glass shape (Figure 43b), which means zooming is on.

Users can click the left button of the mouse to zoom in, or hold the shift key and left click to zoom out. The point where user clicks the mouse becomes the center of the display the zooming in or out. To reset the image display to the original position, user can right click on the image to bring up the context menu, and then click on the “Reset Position / Zoom” menu.

DIRART provides a few predefined image display geometry limit settings (Figure 43d). These options are self-explanatory. When the image zooming / position is reset, the selected predefined geometry limits will be used.



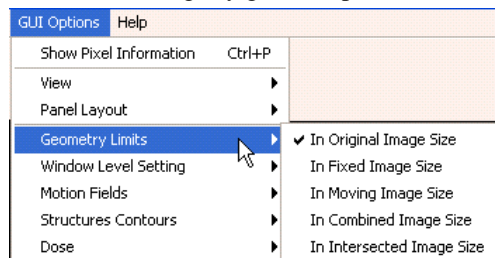
(a) Zooming can be turned on / off from this GUI option menu



(b) When zooming is on, the mouse cursor become the magnify glass shape



(c) Image zooming and display positions can be reset from the context menu



(d) The geometry limits menus are the other ways to set the image display positions / geometry limits

Figure 43: Screen shots of image display zooming and geometry setting related GUI options

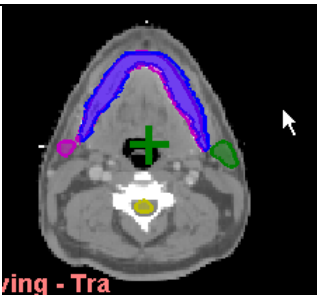
13.11 Shortcut key combinations

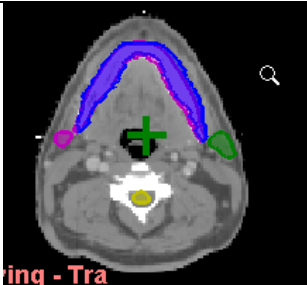
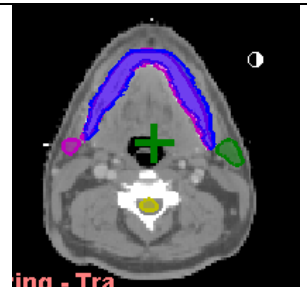
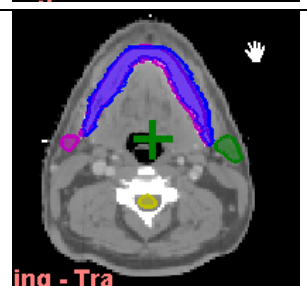
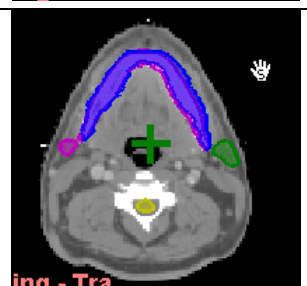
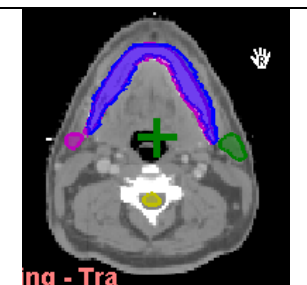
Table 12: List of shortcut key and key combinations

Key combination	Actions
1	Change all views to coronal
2	Change all views to sagittal
3	Change all views to transverse
4	T / C / S view in 7 display panels
C	Display images in color, on / off
Z	Enable / disable zooming with mouse click
S	Enable / disable slice changing with mouse
M	Enable / disable image panning with mouse
W	Enable / disable window level adjustment with mouse
L	Lock / unlock display panels
0	Reset zooming
9	Paint image at the current mouse clicking point
Left, up	Go to the last slice
Right, down	Go to the next slice
Page up	Go to the 5 slice backward
Page down	Go to the 5 slice forward
Home	Go to the first slice

End	Go to the last slice
Ctrl-S	Stop registration computation for the current multigrid stage
Ctrl-P	Show/hide pixel information
Ctrl-B	Show/hide colorbar
Ctrl-H	Hide deformation vector field overlay
Ctrl-M	Show overall deformation vector field in vectors
Ctrl-R	Reset window level setting
Ctrl-L	Using window level settings for lung
Ctrl-O	Using window level settings for abdomen
Ctrl-C	To break any MATLAB computation
Ctrl-left	If image alignment is allowed, moving the fixed image left by 1 pixel respecting to the moving image in the transverse plane. If shift key is also pressed, then moving the fixed image by 5 pixels instead of 1 pixel.
Ctrl-right	Moving the fixed image right by 1 (or 5 with shift key pressed) pixel respecting to the moving image in the transverse plane.
Ctrl-up	Moving the fixed image up by 1 (or 5 with shift key pressed) pixel respecting to the moving image in the transverse plane.
Ctrl-down	Moving the fixed image down by 1 (or 5 with shift key pressed) pixel respecting to the moving image in the transverse plane.
Ctrl-PageUp	Moving the fixed image by 1 (or 5 with shift key pressed) pixel respecting to the moving image in the superior direction.
Ctrl-PageDown	Moving the fixed image by 1 (or 5 with shift key pressed) pixel respecting to the moving image in the inferior direction.

13.12 Control GUI functions with mouse

Mouse Actions	Screen shot with different mouse cursors	Explanation
Left click	Any	Left clicking on a display panel makes this panel the active panel.
Right click	Any	Right clicking on a display panel brings out the context popup menu.
Regular		When no mouse action is enabled, the mouse cursor is an arrow.

Zooming		When zooming is enabled, the mouse cursor is a magnify glass. The zooming feature can be turned on / off by the “W” key, which means “window level”
Image window level / Contrast Adjustment		When window level / contrast adjustment feature is turned on, the mouse cursor is a half white and half black circle. The window level adjustment feature can be turned on / off by the “W” key, which means “window level”.
Panning		When the panning feature is turned on, the mouse cursor is a hand. The slice changing feature can be turned on / off by the “M” key, which means “move”.
Slice changing		When the slice changing feature is on, the mouse cursor is a hand marked with “S”, which stands for “slice”. The slice changing feature can be turned on / off by the “S” key.
Image alignment		When the image aligning feature is on, the mouse cursor is a hand marked with “R”, which stands for “registration”. There is no shortcut key for this feature on purpose.

13.13 Other GUI options

Table 13: The GUI options menu screen shots

GUI options menu screen shots	Explanation
-------------------------------	-------------

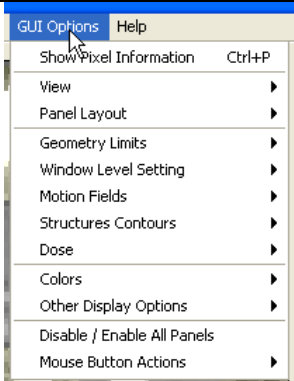
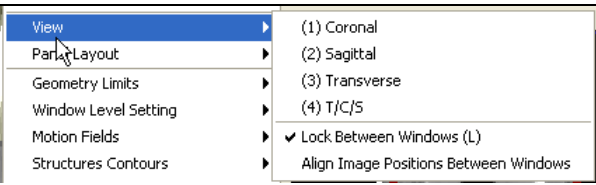
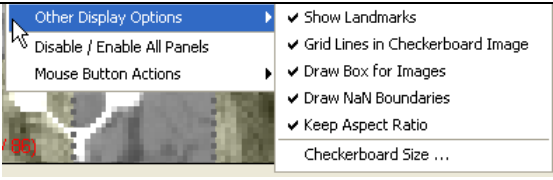

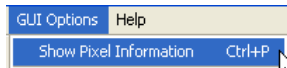
	The level 1 menu
	The view menu
	The other display options menu
	The mouse button actions menu

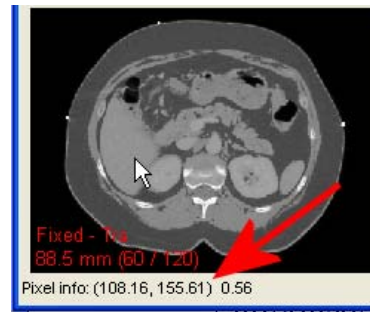
Table 14: Other GUI and visualization options. Options marked with * are advanced options, and will be further explained in the advanced topic sections.

Options	Explanation
<p>Disable a view</p> <p>1. Right mouse popup menu → Disable 2. GUI Options → View Mode → Disable/Enable All Views</p>	<p>If a view is disabled, it will show blank. This view will not be refreshed if image data or the results are changed. This feature helps to improve the GUI response speed.</p>

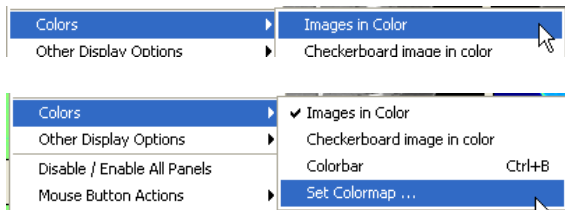
Show pixel information



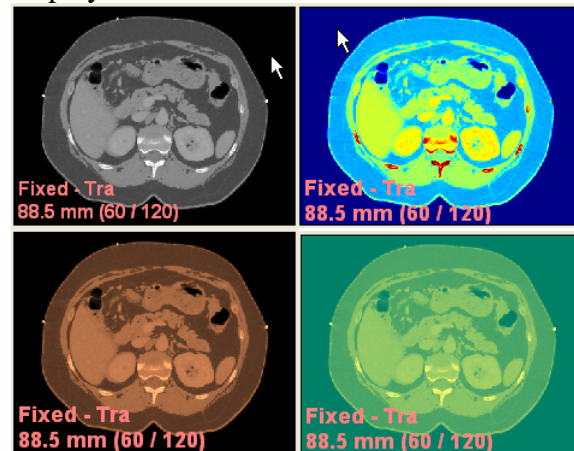
If this option is enabled, users can move the mouse pointer over the images to check the image intensity values which is displayed at the left-bottom corner of the DIRART window.



Display images in color

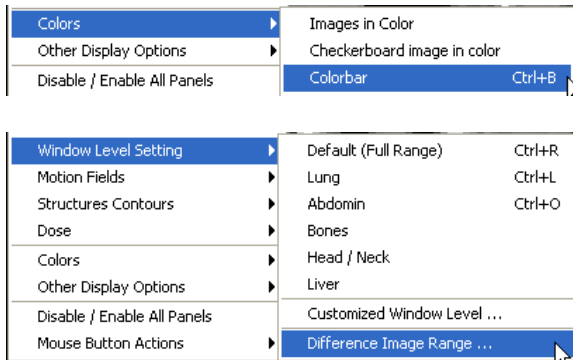


If this option is enabled, the gray images will be displayed in color.

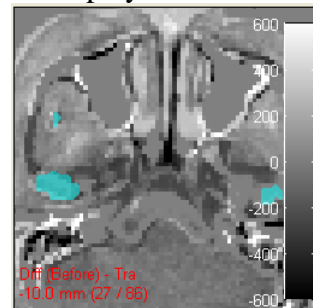


Examples of using different colormaps →

Colorbar



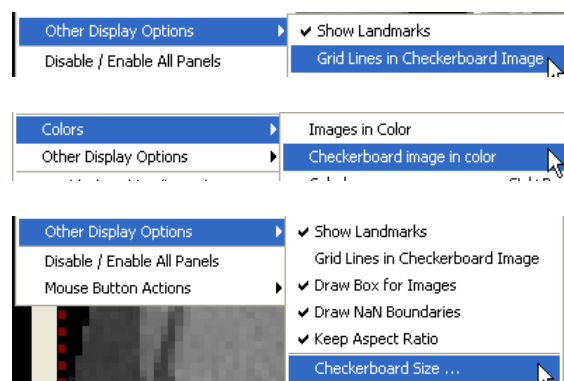
If this option is enabled, a color scale bar will be displayed next to the different images.



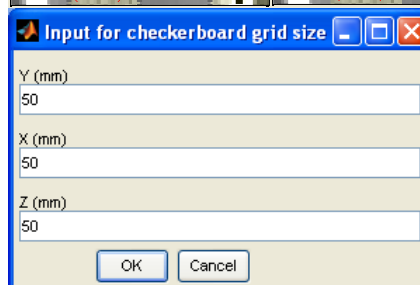
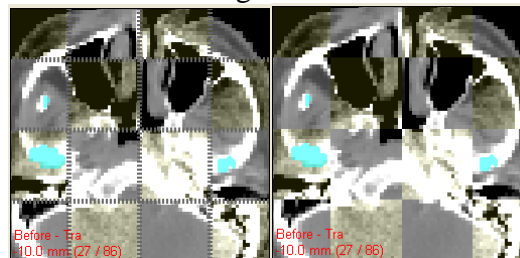
* Show landmarks
Options → Show Land Marks

To show / hide landmarks if landmark data are loaded

Grid line in checkerboard images



If enabled, will show grid lines in the checkerboard images

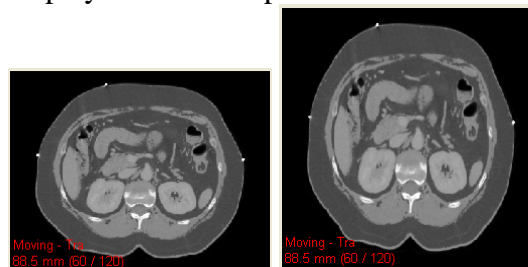


Keep aspect ratio

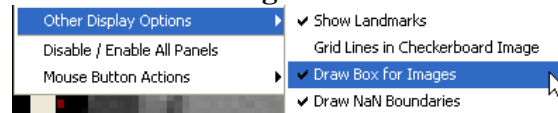


Examples of image displayed with this options enabled and disabled ➔

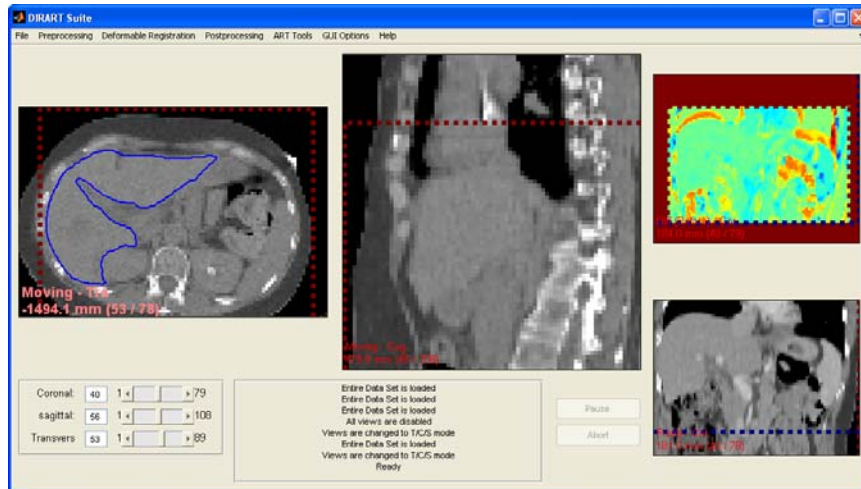
If enabled, images will be displayed with correct aspect ratio. Otherwise, images will be displayed to fill the panels.



Draw Box for Images

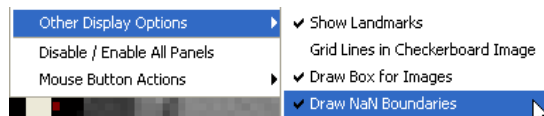


If enabled, the image boundary boxes will be drawn, as shown below.

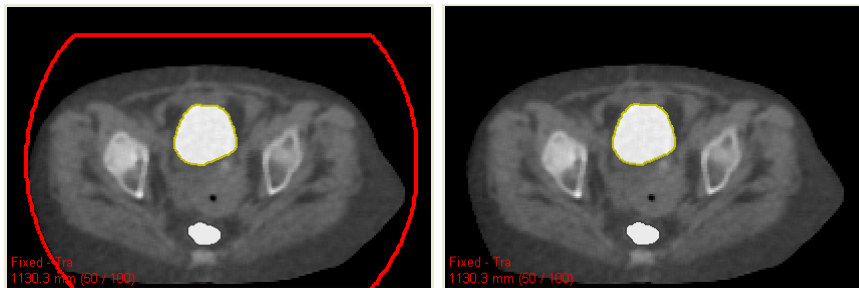


* Draw NaN Boundaries

Is enabled, will draw the NaN value boundaries. Please see the advanced topic about registration two images with mismatched content.



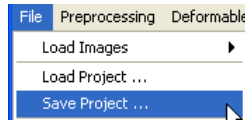
Examples of MVCT image displayed with this feature on versus off. The red contour is the NaN boundary.



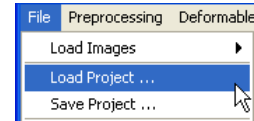
14 File and project management

14.1 Saving and loading projects

Users can save all current data, including all images, deformable vector fields, settings, GUI current display modes, image preprocessing results, etc., into project files by using:



(a) Save Project

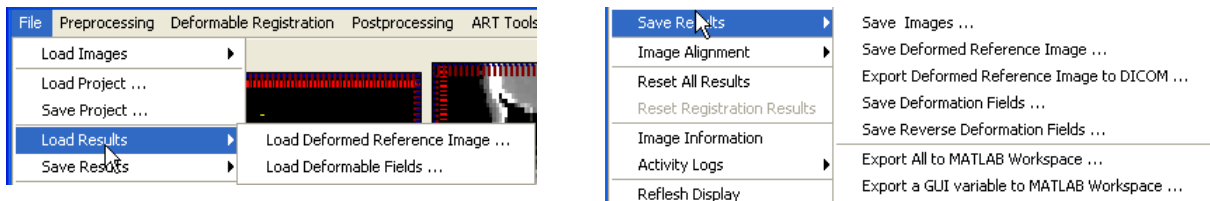


(b) Load Project

The entire data set will be saved into a MATLAB .mat file, which can be loaded back later. The features of project saving and loading let users to save the current working status, and resume it in a later time.

14.2 Saving and loading results

Saving and loading functions are available at “File → Save Results” and “File → Load Previous Results”.



The results here are referring to 1) preprocessed images, 2) computed deformation vector fields, 3) deformed moving image, 4) inversed deformation field if available.

It is also possible to export all data to MATLAB workspace, so that further actions can be taken under MATLAB command line outside the registration software. To do so, users should use “main menu → File → Save Results → Export All to MATLAB Workspace”.

14.3 Reset results

There are two ways to reset the registration results.

- “File → Reset All Results” will reset everything, and return the working status as if two images are just loaded and nothing has been done.
- “File → Reset Registration Results” will clear the computed deformation vector fields, but will not clear the image preprocessing results.

14.4 Generate and export images

At many times, for research or other purpose, users would like to save the displayed images or results, to an image file, or to a separated MATLAB figure. Users can use

- Context menu → Export to Figure
- Context menu → Export to view3Dgui tool

The “view3Dgui” tool is a separate MATLAB program (included in the software package) which is able to view 3D data. It has additional viewing functions, for example, zooming, higher resolution interpolation, etc. See Figure 44 for examples of 2D image resampling for display purpose.



(a) Image displayed in its native resolution

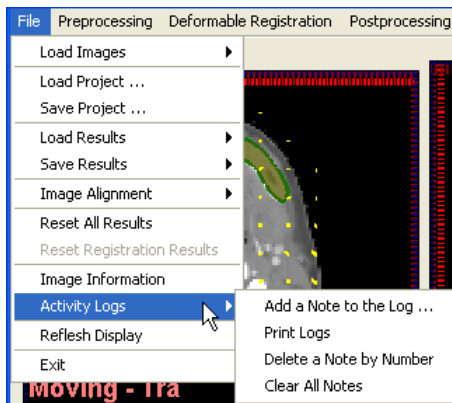


(b) Image displayed in high resolution

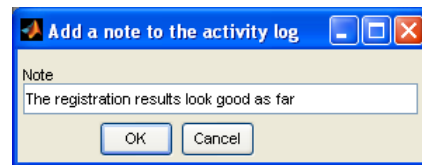
Figure 44: Screen shots of the view3dgui tool

14.5 Event logging

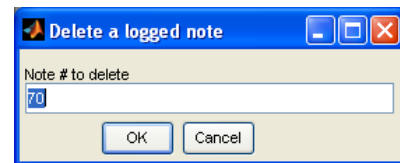
The tool will log any important steps to allow user to check what has been done step by step. The logged events can be printed to MATLAB command line. Event log function can be used from File → Activity Logs → *, as shown in the Figure 45a.



(a)



(b)



(c)

Figure 45: Menus and input windows related to event logging

Table 15: Sample event logs

=====	Event Logs	=====
1:	10-Mar-2009 10:25:39	
2:	Initialization ...	
3:	Image #1 is loaded from D:\Research\MATLAB\data\HN plans\HN_2008_CERR.mat	
4:	Image #1 voxel size = [0.976563 0.976563 3]	
5:	Image #2 is loaded from D:\Research\MATLAB\data\HN plans\HN_2009_CERR.mat	
6:	Image #2 voxel size = [0.976563 0.976563 3]	
7:	Image #2 is cropped using [140-398,110-422,31-108]	
	Image size after cropping = [259 313 78]	
8:	Image #1 is cropped using [130-383,115-403,27-99]	
	Image size after cropping = [254 289 73]	


```

9:      Resampling images 1 to voxel size = [2 2 3]
10:         Image size after resampling = [124 141 72]
11:      Resampling images 2 to voxel size = [2 2 3]
12:         Image size after resampling = [126 153 78]
13:         Image #1 size = [124 141 72]
14:         Image #2 size = [126 153 78]
15:      Using window setting for abdominal
16:      Image offsets is updated to -4 -1 1
-- -- -- --
36:      Image offsets is updated to -4 0 2
37:      Image 1 is pad with boundary = [0 0 10]
38:      Undo the last step
-- -- -- --
46:      Image #1 is cropped using [1-128,1-141,1-82]
      Image size after cropping = [128 141 82]
-- -- -- --
49:      Image #2 is cropped using [1-126,1-140,1-78]
      Image size after cropping = [126 140 78]
50:      Registration parameters are changed to:
           maxiters = [10 20 30 40 50]
           passes_in_stages = [2 3 4 5 6]
           Multigrid_Stages = [2]
           minimal_max_motion_per_iteration = [0.002]
           minimal_max_motion_per_pass = [0.01]
           smoothing_in_iteration = [3]
           smoothing_after_pass = [1 0 0]
           multigrid_filter_type = [1]
51:      Starting registration at 26-Feb-2009 12:34:24
52:      Method = [1]
53:      Inverse consistency = [1]
54:      Smoothing setting = [3 1 0 0]
55:      Num of stages = [2]
56:      Multigrid filter type = [1]
57:      Using inverse consistency framework
58:      Registration is finished at 26-Feb-2009 12:36:21
59:      Using window setting for abdominal
60:      Image #1 window level transformed using center = 1050, width = 600
61:      Image #2 window level transformed using center = 1050, width = 600
62:      Starting registration at 26-Feb-2009 12:37:23
63:      Method = [1]
64:      Inverse consistency = [1]
65:      Smoothing setting = [3 1 0 0]
66:      Num of stages = [2]
67:      Multigrid filter type = [1]
68:      Using inverse consistency framework
69:      Registration is finished at 26-Feb-2009 12:39:40
70:      Project is loaded from "D:\Research\ART\HN plans\reg_project_4.mat"
=====

```

15 Adaptive radiotherapy (ART)

15.1 Introduction

In ART, the treatments need to be adapted to patient anatomy changes. Treatment adaptation involves rescanning the patient (to obtain the daily CT images), evaluating previous delivered dose and performing daily re-planning (re-contouring, re-optimization). There will be a lot of extra works to do in order to achieve treatment adaptation.

Deformable image registration can provide many helps, especially on the following tasks:

1) automatic contour propagation between treatment planning CT and daily CT, 2) remap data between planning CT and daily CT either way for evaluation purpose, 3) register all daily dose (to planning dose grid), to accumulate them for evaluation.

15.2 General working flow

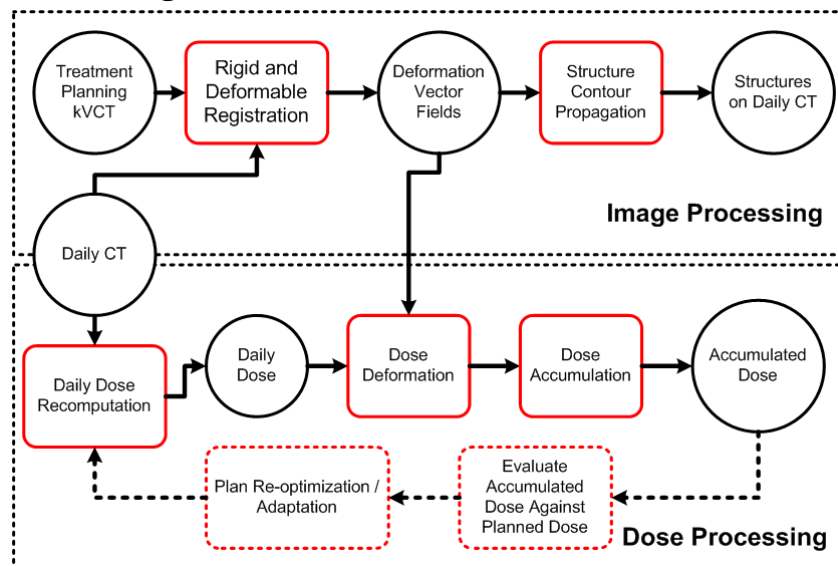


Figure 46: Offline ART workflow

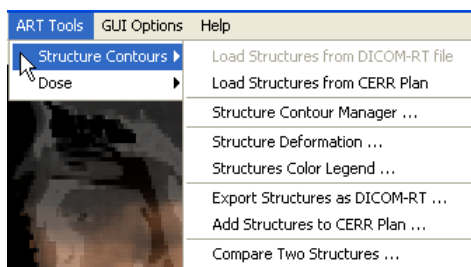
DIRART is able to perform the few tasks in the general working flow: 1) structure contour propagation, 2) dose deformation; 3) dose accumulation.

15.3 Structures

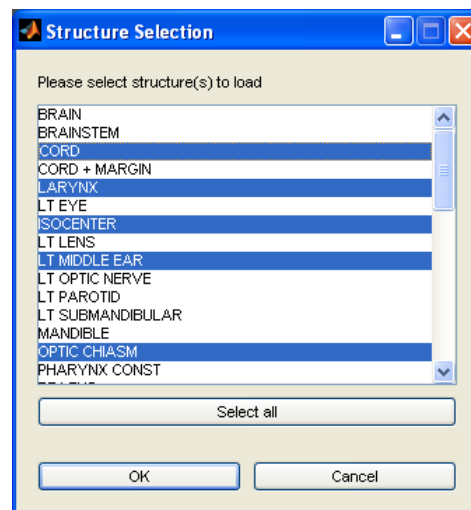
15.3.1 Loading structures from CERR plans

DIRART loads structures from CERR plans. To do so, users can use the menu “ART Tools → Structure Contours → Load Structure from CERR Plan”, or start inside the structure manager.

After a CERR plan file is picked, DIRART shows the list of structures available in the CERR plan (Figure 47b). User can select all or some of the available structures to load into DIRART.



(a) Structure related menus



(b) Selection of structures to load from a CERR plan

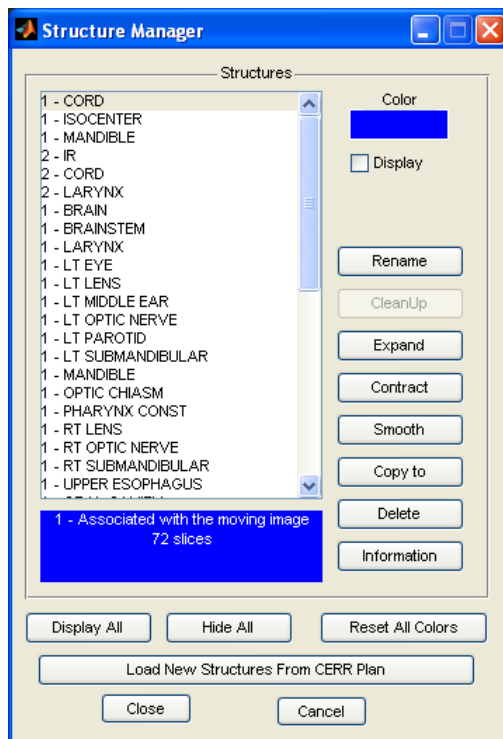
Figure 47: (a) Structure related GUI menus. (b) Loading structures from a CERR plan

15.3.2 Mesh-based structure presentation in DIRART

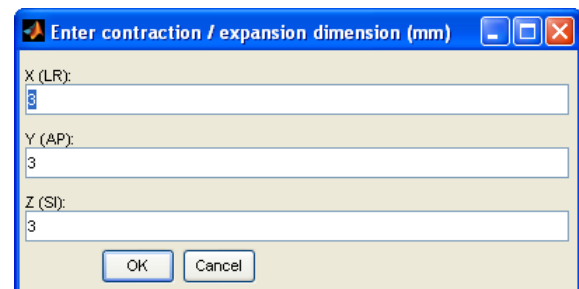
All structures are stored as meshes in DIRART. When DIRART loads a structure from a CERR plan, DIRART will compute meshes for the structure.

15.3.3 Structure manager

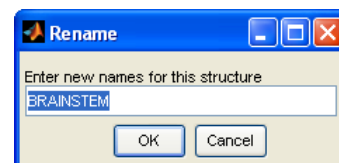
The structure manager is the central control of all structures loaded in DIRART. It can be invoked at “ART Tools → Structure Contours → Structure Manager ...”. Functions of the structure manager are accessible from the buttons on the GUI (Figure 48a).



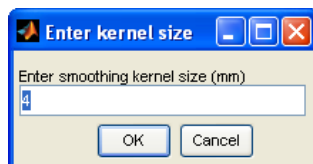
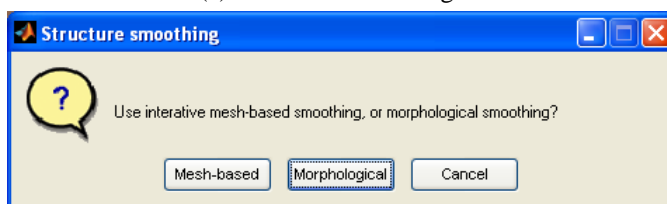
(a) The structure manager



(c) Expand / Contract a structure



(d) Rename a structure



(b) Smoothing a structure



(e) Change the color for a structure

Figure 48: Screen shots of the structure manager and its functions

15.3.4 Smoothing a structure

To smooth a structure, use the “smooth” button inside the structure manager. DIRART offers two ways to smooth a structure: mesh-based smoothing and morphological filter based

smoothing (Figure 48b). Each way has its own workflow (Figure 49 and Figure 50). Example results are shown in Figure 51.



Figure 49: Workflow of morphologic filter based structure smoothing

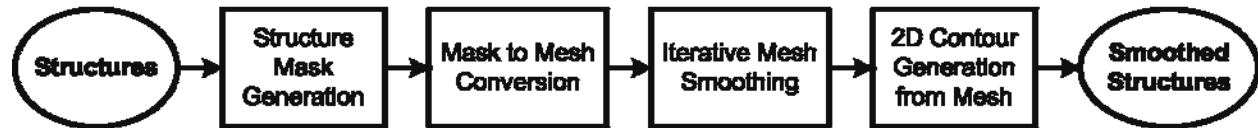


Figure 50: Workflow of mesh-based structure smoothing

15.3.5 Contract and expand a structure

To contract or expand a structure, use the “Contract” or “Expand” button inside the structure manager. Contraction or expansion is performed using morphological filters. User needs to enter the parameters shown in Figure 48c. Example results are shown in Figure 51.

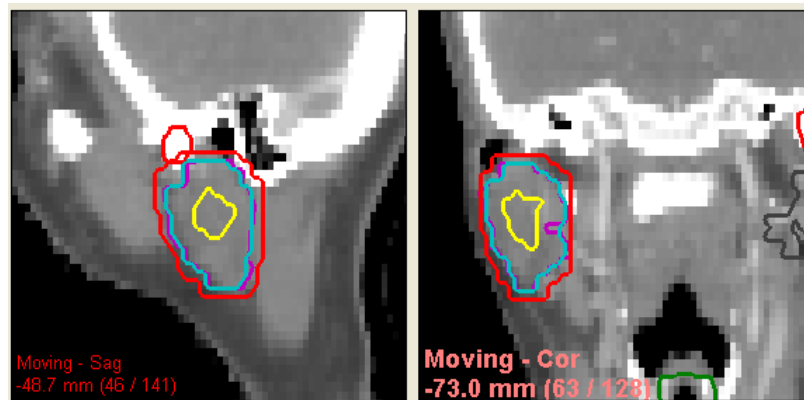


Figure 51: Demonstration of structure smoothing, expansion and contraction. Purple: the original structure. Light blue: the smoothed structure. Yellow: the contracted one. Red: the expanded one. The left figure is the sagittal view and the right figure is the coronal view.

15.3.6 Copy a structure to the other image

To copy a structure associated with one image to the other image, use the “Copy to” button inside the structure manager. A new structure will be created with the name of the original structure, appended by “_copied”. Please pay attention to the structure names listed in the structure manager (Figure 52). All structure names are preceded with a number. The number 1 means that the structure is associated with the moving image, and number 2 means the fixed image.

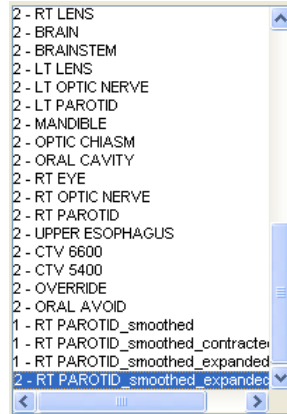
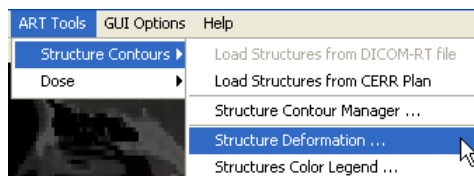


Figure 52: Structure list in the structure manager.

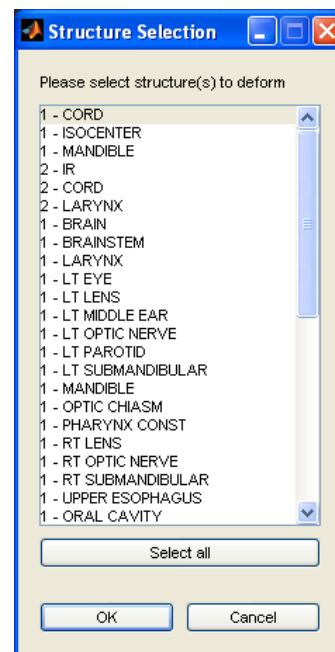
The smoothing, expansion, contraction and copying operations all create new structures from the original structures the operation is performed on. Names of the new structures are copied from the original structure and appended.

15.3.7 Deform structures

After DVF is computed, it could be used to deform the loaded structures in DIRART. To do so, use the menu “ART tools → Structures → Structure Deformation” (Figure 53a), then select one or more structures (Figure 53b) to deform.



(a) Structure deformation menu screen shot



(b) Screen shot of the structure selection window

Figure 53: Structure deformation

However, it is not straightforward to deform a structure from one image to the other image given a DVF from deformable image registration. DIRART has implemented two relative complicated workflows. One is based on structure masks (Figure 54), and the other one is based on the structure mesh presentation (Figure 55).

After a structure is deformed, a new structure is created as the deformation result. The new structure is associated with the same image as the structure being deformed. (This is slightly counterintuitive. In fact, the new structure could be associated to either image). User could also apply smoothing on the new structure. Then the new structure could be copied to the other image.

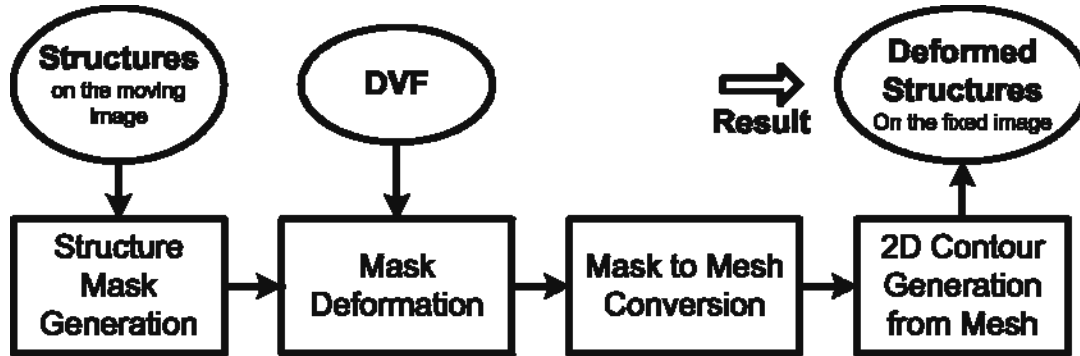


Figure 54: Work flow of mask based structure deformation

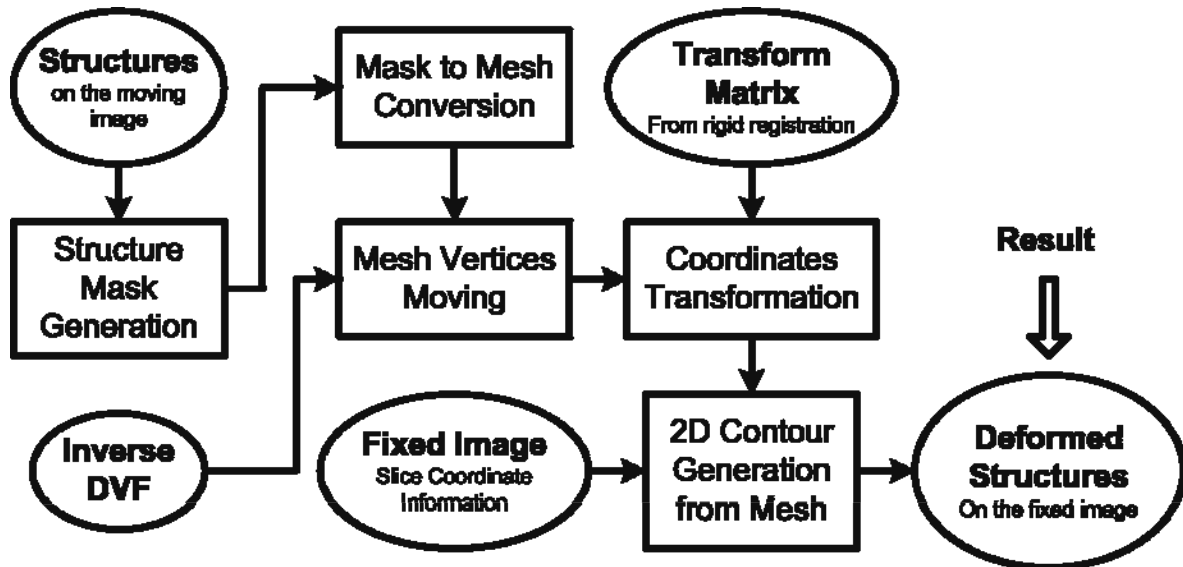


Figure 55: Work flow of mesh-based structure deformation

15.3.8 Export structures to CERR plan

Structures in DIRART could be put back to CERR plan. To do so, use the menu “ART Tools → Structures → Add Structures to CERR Plan”. Structures in DIRART are associated with either the moving image or the fixed image. Each image in DIRART has its own UID. DIRART will match the image UID with the scan UID in the CERR plan to determine if a structure can be associated with the scan in the CERR plan. If yes, then the structure can be exported to the CERR plan. DIRART is able to modify the CERR plan to insert the structure into it, and prompt user to save the modified CERR plan to a new file to the overwrite the existing CERR plan file. User can then load the modified CERR plan in CERR to view the new structure(s), and perform dose metric computation using the new structure(s).

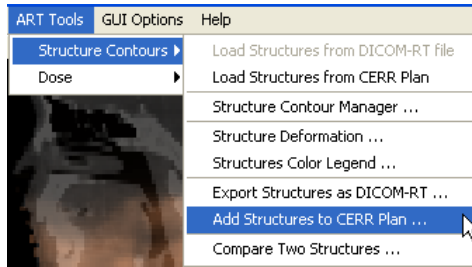


Figure 56: Screen shot of the GUI menu to export structures to CERR plans.

15.3.9 Compare two structures

DIRART can compare two structures by computing the Dice similarity metric, and compute the volume difference. See Table 16 for examples.

Table 16: Example outputs of structure comparison

Dice similarity ratio for "RT PAROTID_smoothed" and "RT PAROTID" = 0.97 Volume of "RT PAROTID_smoothed" = 20.35 cc Volume of "RT PAROTID" = 21.12 cc Volume of difference = 1.06 cc
--

15.4 Doses

DIRART supports multiple doses. Each dose is associated with either the moving image or the fixed image.

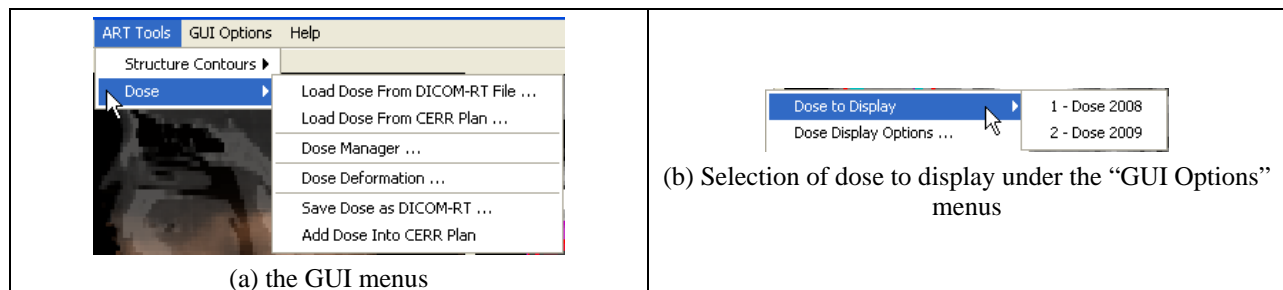


Figure 57: Screen shot of dose related GUI menus

15.4.1 Load doses from CERR plans

Doses are loaded in DIRART from CERR plans. Please note that DIRART can load a DICOM dose file without using CERR. However, CERR translates the coordinates for the images. If the images are loaded using CERR, then DIRART requires the dose also loaded from the CERR plans.

15.4.2 The dose manager

The dose manager is able to load new doses, rename doses, rescale doses, delete doses, and combine two doses by taking the difference or summation. It can also assign the dose to be associated with the other image (there are situations to do this). See Figure 58 for the screen shot of the dose manager.

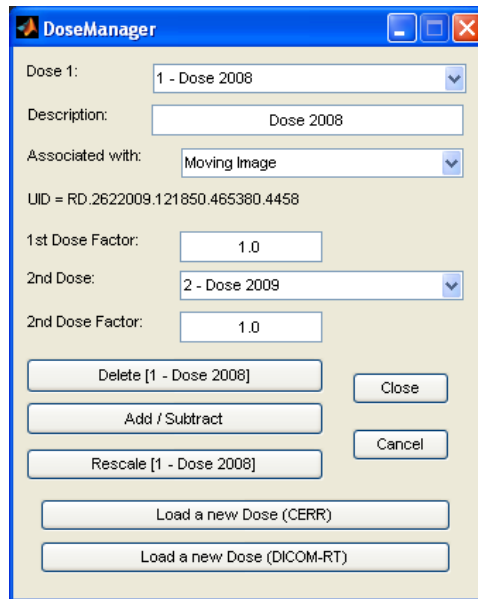


Figure 58: Screen shot of the dose manager

At the time of writing this instruction manual, the dose add/subtract feature is not finished.

15.4.3 Dose deformation

Dose deformation is easier than structure deformation. DIRART allows the deformed dose to be associated to either image, by fault, to the other image.

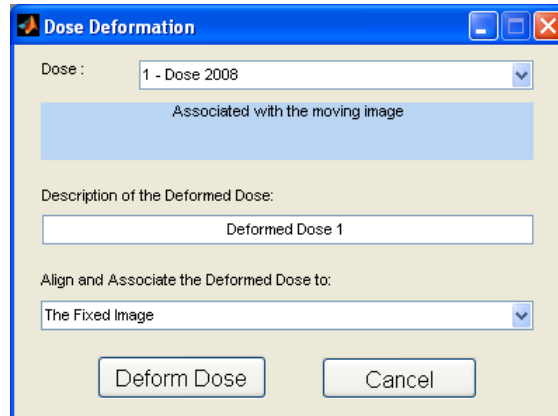


Figure 59: Screen shot of dose deformation GUI

15.4.4 Export processed dose to CERR plan

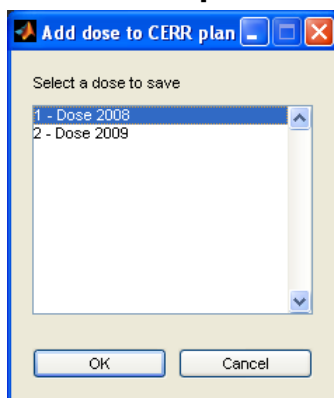


Figure 60: Screen shot of the GUI to select a dose to export

15.5 Other ART tasks

We are planning to implement more ART tasks into DIRART GUI in the near future.

Tasks	Explanation
MVCT FOV completion	After registration, using the kVCT to replace the MVCT outside the MVCT FOV circle, and use the completed image for daily dose computation.
Converting dose to structures	
Dose metrics computation inside DIRART	
Automatic segmentation	

16 Data inside DIRART

16.1 Access to DIRART internal data

In MATLAB command line, use command:

```
dirart_data = guidata(h);
```

`h`, which is the return value when the DIRART program starts, is the window handle of the DIRART main figure window. If the DIRART window is the current active figure window of MATLAB, this command will also work:

```
dirart_data = guidata(gcf);
```

If the data is modified outside DIRART, users can insert it back to DIRART by command:

```
guidata(h,dirart_data);
```

16.2 The DIRART data structure

DIRART's data is organized in a structure. The top level fields of this structure are:

```
gui_handles: [1x1 struct]
info: [1x1 struct]
gui_options: [1x1 struct]
reg: [1x1 struct]
ART: [1x1 struct]
images: [1x2 struct]
```

The fields inside the structure are named in a self-explanatory way. Here are the important fields that require further explanations:

16.2.1 Images

Images and related meta-information are stored in `dirart_data.images(k)`, where $k = 1, 2$. There are two sets of image data. $K = 1$ is for the moving image and $k = 2$ is for the fixed image. Here is the example for one image dataset.

```
image: [134x160x86 single]
voxelsize: [2 2 3]
origin: [67.8281 -160.5469 -317.9999]
voxel_spacing_dir: [-1 1 1]
original_voxelsize: [1.1719 1.1719 3.0000]
image_deformed: [134x160x86 single]
filename: [1x63 char]
type: 'unknown'
class: 'int16'
UID: 'CT.1412009.165440.405190.9237'
DICOM_Info: [1x1 struct]
original_CERR_Scan_Struct: [1x1 struct]
LoadFrom: 'CERR'
```

16.2.2 DVFs

Both the regular DVF and the inverse DVF are stored in `dirart_data.reg`. Here are the fields for the `dirart.reg.dvf`:

```
y: [73x74x51 single]
x: [73x74x51 single]
z: [73x74x51 single]
info: [1x1 struct] →
    voxelsize: [2 2 3]
    origin: [72.9492 -74.1211 -25]
    voxel_spacing_dir: [-1 1 1]
    type: 'DVF'
    class: 'single'
    UID: 'DVF.2322009.101924.439379.481'
    Fixed_Image_UID: 'CT.1412009.165134.797173.3886'
    Moving_Image_UID: 'CT.1412009.165440.405190.9237'
    GenerateBy: 'Reg3dGUI'
```

16.2.3 Structures

Structures related data are stored in `dirart.ART`. Here are the related fields:

```
structures: {1x16 cell}
structure_colors: [16x3 double]
structure_display: [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
structure_names: {1x16 cell}
structure_assocScanIDs: [1 2 4 4 4 4 4 4 4 4 5 5 4 5 5]
structure_scanInfos: {1x5 cell}
structure_structInfos: {1x16 cell}
structure_assocImgIdxes: [2 1 1 1 1 1 1 1 1 1 1 1 2 1 2]
```

16.2.4 Doses

Structures related data are stored in `dirart.ART.dose`, which supports multiple doses. Here are the related fields for one dose:

```
image: [51x51x52 single]
voxelsize: [3 3 3.0000]
origin: [74.8384 -73.7811 128]
voxel_spacing_dir: [-1 1 -1]
original_voxelsize: [3 3 3.0000]
filename: [1x63 char]
type: 'Dose'
class: 'single'
UID: 'RD.1412009.165134.797803.3644'
DICOM_Info: [1x1 struct]
LoadFrom: 'CERR'
ys: [1x51 double]
xs: [1x51 double]
zs: [1x52 double]
```

```

Description: 'CBCT Dose'
assocScanUID: 'CT.1412009.165134.797173.3886'
original_CERR_Dose_Struct: [1x1 struct]
Units: 'cGys'
association: 2

```

17 Advanced topics

17.1 Using the registration software without GUI

Both registration frameworks can be used on the MATLAB command line without the GUI. The main function of the asymmetric registration framework is: *multigrid_nogui6.m*. The inverse consistency framework main MATLAB function is *multigrid_rc_nogui.m*.

17.2 Important utility MATLAB functions

Images_info	Compute MI, CC, MSRE, entropy for two images
GPReduce, MaxReduce	Image half-sampling functions
Load_3d_image_dicom	Load dicom images into MATLAB
Lowpass3d	3D lowpass filtering
resample_3D_image	3D image resampling
View3dgui	3D matrix data viewer
invert_motion_field	Deformation vector field inversion
Move3dimage	3D volume image deformation using computed motion fields
compute_jacobian	Compute Jacobian for the deformation vector fields

18 Contacts

Please contact Deshan Yang, PhD for any questions by emails at:

dyang @ radonc . wustl . edu (without space characters)

or dyangchicago @ yahoo . com (without space characters)

19 References

¹J. O. Deasy, A. I. Blanco, and V. H. Clark “CERR: a computational environment for radiotherapy research,” Med Phys **30**, 979-85 (2003)

²D. Yang, S. R. Chaudhari, S. M. Goddu, D. Pratt, D. Khullar, J. O. Deasy, and I. El Naqa “Deformable registration of abdominal kilovoltage treatment planning CT and tomotherapy daily megavoltage CT for treatment adaptation,” Med Phys **36**, 329-338 (2009)

³D. Yang, H. Li, D. A. Low, J. O. Deasy, and I. El Naqa “A fast inverse consistent deformable image registration method based on symmetric optical flow computation,” Phys Med Biol **53**, 6143-65 (2008)

⁴J. L. Barron, D. J. Fleet, S. S. Beauchemin, and T. A. Burkitt “Performance of optical flow techniques,” Int J Comput Vision **12**, 43-77 (1994)

⁵J. P. Thirion “Image matching as a diffusion process: an analogy with Maxwell's demons,” Med Image Anal **2**, 243-60 (1998)

⁶S. L. Iu and Y. T. Lin, “Re-examining the optical flow constraint. A new optical flow algorithm with outlier rejection,” presented at the Int Conf Image Proc, 1999

⁷B. K. P. Horn and B. G. Schunck “Determining Optical Flow,” Artif Intell **17**, 185-203 (1981)

⁸B. D. Lucas and T. Kanade “An Iterative Image Registration Technique with an Application to Stereo Vision,” Proceedings of the 7th International Joint Conference on Artificial Intelligence 674-679 (1981)

⁹D. Yang, J. O. Deasy, D. A. Low, and I. M. El Naqa, “Level set motion assisted non-rigid 3D image registration,” presented at the Medical Imaging, Proceedings of SPIE, San Diego, CA, 2007 San Diego, CA:

¹⁰H. Wang, L. Dong, J. O'Daniel, R. Mohan, A. S. Garden, K. K. Ang, D. A. Kuban, M. Bonnen, J. Y. Chang, and R. Cheung “Validation of an accelerated 'demons' algorithm for deformable image registration in radiation therapy,” Phys Med Biol **50**, 2887-2905 (2005)

¹¹W. Lu, M. L. Chen, G. H. Olivera, K. J. Ruchala, and T. R. Mackie “Fast free-form deformable registration via calculus of variations,” Phys Med Biol **49**, 3067-87 (2004)