

3D Visualization Tools Documentation

Allison Pearce
Translational Neuroengineering Lab
University of Pennsylvania

Updated August 2012

Contents

1	Introduction	3
2	Package Contents	3
3	Initial Configuration	4
4	Converting Images from 2D DICOM or JPEG to 3D NIfTI	4
5	Explanation of coregister.m	6
5.1	User Input	6
5.2	Init	7
5.3	Main	7
5.4	Outputs	8
6	Running the Script	8
7	Viewing the Output in ITK-SNAP	9
7.1	Loading Images from the GUI	9
7.2	Loading Images from the Command Line	10
7.3	Labels File Options	10
7.4	Using the Toolboxes	11
8	Troubleshooting	11
8.1	Electrode Wires and Visual Noise	11
8.2	Electrodes Not Visible or Missing	12
8.3	Neck Still Attached to Brain	13
9	Acknowledgements	13

Appendix B: Using Screens	13
Appendix B: Marking the Resected Region	14
Appendix C: About digElectrodes.m	15

1 Introduction

The Coregister combines a T1¹ MRI image, a CT scan of electrode placement, and a standard brain template to produce a 3D reconstruction of the patient's brain with labeled regions and electrodes in the correct positions. This allows the user to determine the precise region of each electrode. The process has been consolidated into a single MATLAB script that requires minimal user input. The 3DVis package provides all brain imaging, registration, and normalization tools needed to run the script on a 64-bit Linux machine. This manual assumes no prior knowledge other than a basic familiarity with MATLAB and the terminal, and an ability to follow directions.

2 Package Contents

- `Coregister.m`: main MATLAB script
- `lib` Directory
 - `ants-bin`: directory containing ANTS binaries
 - `FSL`: directory containing FSL program files, including BET
 - `itksnap`: directory containing ITK-SNAP program files
 - `mricon`: directory containing files for MRICron's dcm2nii conversion tool
 - `etc`: directory with additional MATLAB functions and tools such as the `TileImageFilter`
- `util` Directory:
 - `init.txt`: for use with initial configuration
 - `convert_jpg.sh`: example shell script for converting .jpgs files to .dcm files
 - `templates`: directory with NIREP templates and labels
 - `labels`: directory with a variety labels that can be used when viewing the finished segmentation in ITK-SNAP

¹T1is preferred and more commonly used than T2 for structural information about adult brains. In theory, the process would also work for T2 if the user had a corresponding T2 labeled template

3 Initial Configuration

64-Bit Linux: All binaries, functions, and files necessary to run the Coregister script on are already included in the package. Nothing needs to be installed, but the user will have to add the appropriate directories to the path before running the script for the first time.

Step 1 Open a terminal window

Step 2 Navigate to the `util` directory inside the 3DVis folder

Step 3 Type the following command exactly as written². This ensures that the appropriate directories are accessible each time you log in.

```
sed '1 i VISPATH=\mnt\local\gdrive\public\3DVis' init.txt >> ~/.bash_profile
```

Step 3 Type the following command to set the paths for the current session without having to log out and back in.

```
source ~/.bash_profile
```

Other platforms: Mac users will have to download binaries for ANTs, FSL, ITK-SNAP, and MRICron's dcm2nii converter separately, place them in the “lib” folder, and then follow the initial configuration steps listed above. The folder hierarchy should be exactly as specified in the Package Contents section. Currently, there is no support for the 3DVis package on Windows.

4 Converting Images from 2D DICOM or JPEG to 3D NIfTI

The script requires that the template, T1, and CT images be in compressed 3D NIfTI format (.nii.gz). Sometimes clinicians provide the data in the form of a series of 2D DICOM (.dcm) or JPEG (.jpg) files, which must essentially be stacked into a 3D image. The tools needed to convert both formats are included in this package.

DICOM

If there are multiple pre-implant MRI studies, it is best to choose the one corresponding to the largest number of DICOMs to produce the highest resolution NIfTI output. You will need a DICOM viewer such as Osirix (<http://www.osirix-viewer.com/>) to

²If you are not using the 3DVis directory found in the public folder of Fourier, this path can be determined by navigating to the 3DVis directory within the terminal and typing `pwd`.

import images from a disk to your computer. You can then use MRICron's dcm2nii tool³ :

Step 1 Make sure all of the .dcm files are in the same folder.

Step 2 Navigate to the folder where you would like your .nii.gz file to be generated, and type “dcm2nii” followed by the path to one of the 2D Dicom images. For example,

```
dcm2nii ~/gDir/3DVis/images/Pt-001/MRI/IMG-0001a2002.dcm
```

The program should search the directory of the input image for all similar files and combine them into a 3D image. Depending on the settings, more than one .nii.gz file might be created. The file that begins with “co” (e.g. coMSEL397s003a1001.nii.gz) is the cropped, oriented image that should be used in Coregister. If there is no “co” file, use the file beginning with “o” (this means that the image did not need to be cropped, only oriented.)

JPEG

High quality reconstructions are difficult to achieve with JPEG data and it is not recommended for this process, but it can be converted using ITK's TileImageFilter:

Step 1 Make sure all of the .jpg images are in the same directory.

Step 2 Stack the first two images using the following command:

```
TileImageFilter 1 1 2 <input_1>.jpg <input_2>.jpg <desired_output_name>.nii.gz
```

Step 3 Use the example file `convert_jpg.sh` (found in the `util` directory) to write a shell script that will loop through the remaining images, adding each one to the stack of previous images.

Checking the Image Orientation

It is recommended that you check the result of the conversion before running it through the full script. Open a new ITK-SNAP window and load the NIfTI file as a grayscale image. Make sure that the right to left, anterior to posterior, and superior to inferior axis are oriented properly. If not, choose “Tools → Reorient Image” and type in a new RAI code. Each of the three letters in the code specifies the direction for a dimension of the patient coordinate system. For example, the code “RAI” indicates that each voxel's x-axis runs right to left, the y-axis runs anterior to posterior, and the z-axis is inferior to superior.

³You can change certain options for the dcm2nii tool, including those for anonymizing the output, by opening and editing the file “.dcm2nii/dcm2nii.ini”. See the dcm2nii online documentation: <http://www.mccauslandcenter.sc.edu/mricro/mricron/dcm2nii.html>.

5 Explanation of `coregister.m`

5.1 User Input

This section must be edited by the user before running the script for each patient. Filenames are to be typed without extensions and must be of the specified type.

- Directories and Data Filenames (Patient-specific)
 - **ptName**: Short patient identifier which will be included in the output filename
 - **vispath**: Path to the root 3DVis directory
 - **in_out**: Path to the directory containing the input files, and where the output files will be written
 - **T1**: Pre-resection T1 MRI (must be high-resolution and in `.nii.gz` format)
 - **CT**: Post-implant CT scan of electrode placement (`.nii.gz`)
 - **T2** (optional): Post-resection T2⁴ MRI (if post-resection data is available)
 - **resection** (optional): Binary mask of the resected region, created manually using ITK-SNAP (see Appendix C: Marking the Resected Region).
- Run Options
 - **segment**: Segmentation mode. Set to 1 to divide and label the brain into 35 regions before overlaying the electrodes or 0 for just brain and electrodes. Option 0 is significantly faster than option 1 (less than an hour vs. most of a day).
 - **resect**: Resection mode, set to 1 if you will be processing post-resection data
 - **unbury**: Set to 1 to include an intermediate processing step that ensures all electrodes are clearly visible on the surface of the brain. Set to 0 if depth electrodes were included in the implant.
 - **db**: Debug mode, set to 1 for command printouts (can be helpful for troubleshooting)
 - **cleanup**: Clean-up mode; set to 1 to delete intermediate files at end of script

⁴T2 is the preferred format, but T1 will usually suffice

- `electrode_thresh`: Intensity threshold for locating electrodes in CT scan. Assume electrodes are of higher intensity than “`electrode_thresh`” and everything else is lower (default is 2000).

5.2 Init

This section sets some of the paths for the MATLAB working environment, defines variable names, and reorganizes directories.

- `template` and `templateLabels`: brain segmentation template and labels provided in the `until` folder (both `.nii.gz`)
- `warpOutputPrefix`: Prefix for warp output filename (an intermediate file not used in the final result). Should not be the same as the template file main body.
- `MRF_smoothness`: smoothness parameter for prior-based segmentation. The larger the value, the smoother the segmentation. Normal range is 0.05 to 0.5 (default is 0.1).
- `log`: location of the log file where system output will be printed while the script is running

5.3 Main

Skull Stripping The first step calls `bet2` (a brain extraction tool, part of the FSL library) to strip the skull from the T1 image. It usually takes less than five minutes to run.

Non-Rigid Warping In segmentation mode, the second step uses `ANTs` (Advanced Normalization Tools) to perform a non-rigid image registration, warping the template to the skull-stripped T1 image. This process can take hours to complete. For more information, see the `ANTs` documentation: <http://www.picsl.upenn.edu/ANTs/>.

Prior-Based Segmentation on Warped Labels In segmentation mode, this step uses `Atropos` (a component of `ANTs`) on the warped template labels to segment the T1 data. This is the most computationally intensive part of the process and may take hours to complete.

CT Alignment and Electrode Extraction The script uses ANTS' rigid registration functionality to align the CT to the T1 MRI and extract the electrodes using the predefined `electrode_thresh`. This process is generally much faster than the non-rigid registration. The extracted electrodes are overlaid onto the T1 segmentation or the brain mask, depending on the segmentation options.

Post-Resection Alignment This step only occurs in resection mode. The pre- and post- resection MRI images are aligned, and then the resected region from the post-resection image is transformed onto the pre-resection MRI so that it can be incorporated into the segmentation with the electrodes.

Clean Up If the `cleanup` option is set to 1, the script finishes by deleting intermediate files created during the process, leaving only the segmentation files needed to view the 3D brain reconstruction. Note that these intermediate files can be useful for other purposes, including troubleshooting.

5.4 Outputs

The script outputs either one or two electrode segmentation files that can be viewed in ITK-SNAP. These files will appear in whichever directory is specified for inputs and outputs when the script is finished running. The first file, non-resection, is called `<ptName>.electrode_seg.nii.gz` and the optional post-resection segmentation is called `<ptName>_post_resection_seg.nii.gz`. Labels files to accompany these segmentations can be found in the `3DVis/util/labels` directory and are explained in section 7.3.

6 Running the Script

Before running `coregister.m` for the first time, follow the instructions in Section 3: Initial Configuration to set the paths.

Step 1 Choose or create a folder to be your inputs-output directory, and place the MRI and CT files in it. This directory can be located anywhere, but you need to know its path.

Step 2 Open the `coregister.m` file in the MATLAB editor and fill in the Required Inputs section.

Step 3 Make sure the location of `coregister.m` is on the MATLAB path and run

the script. Because of the extended processing time, it is best to start a new GNU screen and open MATLAB with the `-nodisplay` option (see Appendix B). You can also type `coregister` into the MATLAB command window, or choose “Cell → Evaluate Entire File” from the MATLAB editor window.

Step 4 After the script finishes, follow the instructions in Section 7: Viewing in ITK-SNAP to open the segmentation files and view the 3D reconstruction(s).

Checking the Log File

Many of the commands in the script, particularly `ANTS` and `Atropos`, provide output that is not displayed by MATLAB until the very end of the process but is written to a log file in real time. You can check the progress of the script at any time by viewing the log. This file will be located in the `temp` directory, which is created within the `inputs-outputs` directory. One option is to navigate to the appropriate directory from the Terminal and type `cat log_<ptName>.txt`. You can also open the file with your favorite text editor. Even if you don’t know what the output means, you can still measure the script’s progress by the amount of text in the log file.

7 Viewing the Output in ITK-SNAP

Three files are required for a 3D reconstruction in ITK-SNAP: a base image (called the ‘greyscale image’), a segmentation file, and labels file. For our purposes, the brain-only pre-resection MRI is the greyscale image, the segmentation file(s) generated by the script serve as the segmentation, and default labels files are included in the `util` directory.

7.1 Loading Images from the GUI

Step 1 Open a new window of the ITK-SNAP GUI (this can be done from the terminal by typing `itksnap`).

Step 2 Click “File → Open Greyscale Image” and select the file in the inputs and outputs directory called `<mri filename>_brain.nii.gz`.

Step 3 Click “Segmentation → Load From Image” and navigate to the electrode segmentation file (the default filenames are `<ptName>_electrode_seg.nii.gz` for pre-resection images and `<ptName>_post_resection_seg.nii.gz` for post-resection).

Step 4 Click “Segmentation → Load Label Descriptions” and navigate to the labels

file. Click “Update Mesh” to load the 3D segmentation.

Note: *You must enable X-forwarding to open the ITK-SNAP GUI when using a secure shell to remotely access a server where the files are being stored and processed. To do this, type `ssh -X username@serveraddress` when logging in with the secure shell.*

7.2 Loading Images from the Command Line

To load a greyscale image with segmentation and labels files, type `itksnap -g <filepath> -s <filepath> -l <filepath>`. For example, to load a pre-resection reconstruction with the default labels and segmentation output, type:

```
itksnap -g pt34_MRI_brain.nii.gz -s electrode_seg.nii.gz -l labels.txt
```

7.3 Labels File Options

The labels files specify the identifiers, colors and opacities of the different regions of the segmentation. The `util` folder has a sub-directory called `labels`, which contains a variety of labels files appropriate for the different run options. They are simple text files, so it is easy to create a new labels file tailored to a specific patient using an existing file as a template. You can also use the “Label Editor” in ITK-SNAP to modify and save the settings of an existing labels file.

- `labels_no_seg.txt`: Gray brain with red electrodes. Use this file if you set the `segment` option to 0.
- `labels_pre_resection.txt`: Color-coded 35 brain regions with red electrodes. Use this file if you set the `segment` option to 1 and `resect` to 0.
- `labels_pre_transBrain.txt`: Color-coded 35 brain regions with red electrodes and slightly translucent brain tissue. Use this file if you set the `segment` option to 1, `resect` to 0 and are having trouble seeing the electrodes, or if the implant included depth electrodes.
- `labels_resect`: Color-coded 35 brain regions with red electrodes and translucent green resection region. Use this file if you set the `segment` option to 1 and `resect` to 1.
- `labels_resect_transBrain`: Color-coded 35 slightly translucent brain regions with red electrodes and translucent green resected region. Use this file if you set the

`segment` option to 1, `resect` to 1 and are having trouble seeing the electrodes, or if the implant included depth electrodes.

7.4 Using the Toolboxes

The ITK-SNAP tools needed to view and manipulate images are fairly straightforward. The most useful in the Main Toolbox are:

- **Crosshairs:** Use this tool to navigate to different slices in the 2D windows. The Tool Options box displays the label for the selected region and its intensity value. If you have multiple SNAP windows open at the same time, you can use the Multisession Cursor to link the actions of the crosshairs so that all sessions respond to when you click on one.
- **Navigation (magnifying glass):** Use this tool to zoom and pan in the 2D panes. Click and drag with the left mouse button to pan; click and drag with the right mouse button to zoom. The Tool Options box displays the zoom magnitude and allows you link or separate the zoom options in multiple SNAP windows.
- **Paintbrush:** This tool is useful for editing noise and other minor segmentation problems. See the Troubleshooting section for more detail.

The relevant tools in the 3D Toolbox are:

- **3D Trackball:** This tool allows you to rotate, pan, and zoom in the 3D view. Click and drag with the left mouse button to rotate and the right mouse button to zoom. If your mouse has a middle button, you can use it to pan. The location of the crosshairs serves as the axis of rotation.
- **3D Navigation:** Use this tool to select the position of the crosshairs in the 3D window. Note that you can only select a point on the brain surface; to position the crosshairs in the brain interior, you must click inside one of the 2D slices.

8 Troubleshooting

8.1 Electrode Wires and Visual Noise

Occasionally, wires or other non-electrode bright spots from the CT appear in the resulting segmentation labeled as electrodes. You can wipe away extraneous floating bits by hand in ITK-SNAP. Click on the Paintbrush Tool in the Main Toolbox. Check the “3D Brush” box under Tool Options. You can also adjust the size and

shape of the brush. Under Segmentation Options, select “Clear Label” as the Active Drawing Label and select “Electrodes” in the Draw Over list. Use the 3D crosshairs to navigate to each fragment, paint over it in all of the 2D panes, and click “Update Mesh” to update the 3D view.

8.2 Electrodes Not Visible or Missing

The electrodes might not be visible because they are embedded beneath the tissue surface. It is also possible that they are not being displayed properly because they failed to transfer from the CT scan, or because they are being masked by ITK-SNAP’s smoothing process⁵.

Diagnosis

It is possible to hide the brain in the 3D view of ITK-SNAP so that only the electrodes are visible. On the right side of the screen, click on “Label Editor.” Click on each label under “Available Labels” and check the box marked “Hide label in the 3D Window.” If necessary, click “Update Mesh.” If all of the electrodes are now visible, see the “Embedded” section below. If not, try clicking “Tools → Display Options → 3D Rendering” and uncheck the “Gaussian Smoothing” box. If the electrodes still do not appear, see the “Failed to Transfer” section below.

Embedded

If you ran the script without the `unbury` option, which calls `digElectrodes.m`, try running it again with this option. Alternatively, you can use the Label Editor to make certain brain regions slightly transparent so that the electrodes are visible. If your primary interest is determining the brain region of each electrode, you can view the image with the brain hidden in the 3D view but visible in the 2D views. Use the 3D crosshairs to select an electrode in the 3D window and determine its region label from the 2D panes.

Failed to Transfer

The electrodes might not have been extracted from the CT scan because the electrode threshold was too high. To determine a more appropriate threshold, load the CT scan as a greyscale image in ITK-SNAP and use the crosshairs to click on the electrodes. Use the arrow keys to move the crosshairs small increments in all directions until you find the peak intensity value. Use this value for the variable `electrode_thresh` in the Optional Inputs section of the script. Alternatively, you

⁵This is a particular problem for images with severely anisotropic voxels. ITK-SNAP automatically employs Gaussian smoothing during the 3D rendering. If the voxels’ z-coordinates are much larger than the x and y coordinates, an electrode might only appear in one horizontal slice and will be lost during smoothing.

can try progressively decreasing the value for `electrode_thresh` by 200 until you achieve a better result.

8.3 Neck Still Attached to Brain

Particularly with low resolution images, problems can arise when isolating the brain during the skull stripping step. If the final segmentation includes portions of the neck, use the “Bias and Neck” filter option of BET. Open the `coregister.m` file and find the code that strips the skull in T1 (the first section under the `MAIN` heading). The command should read:

```
systemf_db(db, 'bet2 %s.nii.gz %s_brain -m >> %s', T1, T1, log)
```

Change `bet2` to `bet` and insert the `-B` option after the `-m`. The command should now read:

```
systemf_db(db, 'bet %s.nii.gz %s_brain -m -B >> %s', T1, T1, log)
```

Sometimes this option overcompensates and removes too much of the brain. An alternative is to open and edit the brain mask by hand, wiping away unwanted portions similarly to erasing electrode wires. Open the T1 file as a greyscale image in ITK-SNAP and load the brain mask as segmentation (the filename is the same as the T1 image with the suffix `_brain_mask.nii.gz`). Use the paintbrush in the Main Toolbox to erase any sections of the mask that are not over regions of brain (see Troubleshooting Electrode Wires and Visual Noise for more explicit Paintbrush instructions). Save the segmentation. Then comment out the skull-stripping step and run the `coregister.m` script again.

9 Acknowledgements

The main code in this package was adapted from an original script by John Wu at the Penn Image and Computing Science Lab (Jim Gee Lab) of the University of Pennsylvania.

Appendix A: Using Screens

With programs like Coregister than can require up to a day to finish, it is advantageous to use a GNU screen. A screen is an independent window in which one can

start a program on the server and then “detach” without interrupting or ending the process. This enables the user to log out and allow long programs to run overnight or over a weekend.

To start a screen: Log in to the server using a secure shell (ssh). Type `screen` to start a new session, or `screen -S` followed by a title to give your screen a descriptive name. For example, `screen -S coregister`.

To run Coregister from a screen: You must use the command-line only version of MATLAB with screens. Type `matlab -nodisplay` to open the program, then navigate to the appropriate directory, and type `coregister` to run the script.

To detach: Push `Ctrl+a`⁶ followed by `Ctrl+d`. Your session will continue to run and you may log out or use other programs.

To reattach: If you have only one screen running, type `screen -r` from any terminal window. If you have multiple screens, type `screen -r` followed by the name of the session you want to reopen.

To check the status, number, or names of current screens: The command `screen -ls` provides a list of all screens, their names, and whether they are detached or attached.

To terminate (kill) a screen: With the screen attached, type `Ctrl+a` followed by `Ctrl+k`. You will be asked if you really want to kill the screen. Type `y` for yes.

Appendix B: Marking the Resected Region

A binary mask for the resected region is required to delineate this area in the reconstruction. This must be done by hand in ITK-SNAP.

Step 1 Open the post-resection MRI as a greyscale image in ITK-SNAP.

Step 2 Find the resected area in each of the 2D views. Decide which plane provides the clearest view of the resection and maximize this view by clicking the plus sign next to it. Click “zoom to fit” so that the image fills the screen.

Step 3 Navigate through the slices to find the first one in which the resected region appears.

Step 4 Click on the Polygon Tool in the Main Toolbox. Select the piecewise linear option and make sure the segment length is set to a reasonable value, around 6.

Step 5 Trace around the resection, either by clicking and dragging or clicking a

⁶Almost all screen commands begin with `Ctrl+a`, and there are many more than those described here. For a full list of possibilities, see the online GNU screen documentation

series of points around the perimeter. Once you have made a closed loop, hit Enter twice to fill it in.

Step 6 Move to the next slice and repeat until you have marked each slice in which the resected area appears.

Step 7 Save the resected region in the `inputs-outputs` directory by clicking “Segmentation → Save as Image,” navigating to the appropriate directory, and inputting an appropriate file name.

Appendix C: About `digElectrodes.m`

The `digElectrodes` tool was created to solve the problem of electrodes occasionally being buried beneath brain tissue in the final segmentation. It is called automatically from the script when the `unbury` option is set to 1. It can also be used independently of the script. The MATLAB function takes as input the file paths to two 3D binary images: the first is a brain mask and the second is the set of electrodes extracted from the CT scan. Both are created as intermediate files during the coregistration process. Using a few simple calculations, the algorithm determines the magnitude and direction of a shift necessary to place the center of mass of each electrode on the outside surface of the brain. It then translates all voxels in that electrode by the calculated amount. This version cannot distinguish between depth and surface electrodes, so using the function on data with depth electrodes will result in strange and serious errors in the final segmentation.