

Aligning Anatomical Volumes with Oblique Axial Functional Volumes

Revision date: December 3, 2004

Overview

For certain experiments, such as imaging of the medial temporal lobe (MTL), it can be advantageous to acquire oblique axial functional imaging volumes. Given the tilt angle of the functional volumes, it is also possible to acquire the anatomical (typically FSPGR) volume at the same angle. Unfortunately, with the current GE graphical prescription program this is not always a straightforward process. As an interim measure, we have found it useful to acquire an axial FSGPR and then rotate the anatomical volume to the same angle as the functional volume. To do this, the user records the geometric coordinates of the functional and anatomical volumes and then runs a MATLAB-based processing routine (either directly from MATLAB or using a perl script).

Required Data

1. Prescribe and run your functional. Record the coordinates (e.g. R0.0, L0.0 P09.2 P19.4 S10.5 S22.9). Double-check the numbers you have recorded.
2. Prescribe and run your anatomical. Record the coordinates (e.g. I70.6 S89.3 R1.2 R1.2 A14.8 A14.8). Double-check the numbers you have recorded.

Processing the data

1. You can process the data either in the MATLAB environment or using a perl script that will call MATLAB. In order to do so, you must have MATLAB installed on the computer that you are using.
2. You will need to convert coordinates into the following form [R/L Coordinates A/P Coordinates S/I Coordinates], where RAI values are negative while LPS values are positive. Because the program is using the coordinates only to find the center of the volume, the relative order of R/L (or A/P or S/I) coordinates does not matter. For example, R0.0 R0.0 A24.1 A32.9 I54.1 I31.8 could be converted to [0.0 0.0 -24.1 -32.9 -54.1 -31.8] or [0.0 0.0 -32.9 -24.1 -31.8 -54.1], etc. Similarly R0.0, L0.0 P09.2 P19.4 S10.5 S22.9 would be converted to [0.0 0.0 9.2 19.4 10.5 22.9] or [0.0 0.0 19.4 9.2 22.9 10.5], etc.
3. You should also note whether the oblique functionals are rotated clockwise (posterior more inferior than anterior) or counterclockwise (anterior more inferior than posterior). For MTL imaging, you will most likely have a counterclockwise rotation while for visual cortex (e.g. aligned along calcarine sulcus) you will most likely have a clockwise rotation.
4. The MATLAB function `Oblique_Reg` uses the following syntax:
`Oblique_Reg(brikdir,func_brik,anat_brik,out_brik,func_Rx,anat_Rx,func_angle_dir);`

brikdir: Directory with briks. Examples: `'/mnt/raid3/sdc/data'` or `'.'` to denote current directory.

target_brik: Name of functional brik (without +orig) , e.g. `'func1brikreg_e01'`

anat_brik: Name of anatomical brik (without +orig) , e.g. `'anat1'`

out_brik: Name of output (rotated) brik,e.g. `'anat4'`

func_Rx: Coordinates of functional brik, e.g. `[0.0 0.0 -24.1 -32.9 -54.1 -31.8]`

anat_Rx: Coordinates of anatomical brik, e.g. `[0.0 0.0 9.2 19.4 10.5 22.9]`

func_angle_dir: Direction of functional rotation 1 = counterclockwise, -1 = clockwise.

EXAMPLES

Example of Usage from MATLAB:

To use the Oblique_Reg program within MATLAB, it is recommended that the user create or modify a MATLAB script with the commands and then execute the script from within MATLAB. Here is an example script we call obexample.m.

```
%obexample.m
% Example of using Oblique_Reg program

brikdir='.';
func_brik='func1brikreg_e01';
anat_brik='anat';
out_brik = 'anat4';

func_Rx=[0 0 -24.1 -32.9 -54.1 -31.8];
anat_Rx=[2.9 2.9 -24.8 -24.8 -87.0 72.9];
Oblique_Reg(brikdir,func_brik,anat_brik,out_brik,func_Rx,anat_Rx,1);
```

To run this script from within matlab, the user simply types obexample at the MATLAB prompt

Example of Usage with a PERL script

To use the Oblique_Reg program directly from UNIX, it is recommended that the user create or modify a script with the commands and then execute the script from UNIX. The script will call MATLAB to execute the commands. Here is an example perl script called obreg. To execute this script the user simply types obreg at the UNIX prompt.

```
#!/usr/bin/perl
{
# Script Name: obreg
# THIS IS AN EXAMPLE PERL SCRIPT FOR ROTATING ANATOMICALS
# TO ALIGN WITH OBLIQUE AXIAL FUNCTIONAL SCANS

# PUT NAME OF FUNCTIONAL BRIK HERE
$func = "func1brikreg_e01";
# PUT NAME OF ANATOMICAL BRIK HERE
$anat = "anat";
# PUT NAME OF ANATOMICAL BRIK HERE
$anatout = "anat_rot2";
# PUT FUNCTIONAL RX HERE
$func_rx = "[0 0 -24.1 -32.9 -54.1 -31.8]";
# PUT ANATOMICAL RX HERE
$anat_rx = "[2.9 2.9 -24.8 -24.8 -87.0 72.9]";
# INDICATE TARGET ANGLE DIRECTION HERE
# 1 = counterclockwise
# -1 = clockwise
$angle = "1";
# INDICATE DIRECTORY WITH BRIKS HERE
$sdir = ".";
$command1 = "matlab -nojvm -nosplash -r ";
$command2= "\"Oblique_Reg('\$sdir', '\$func', '\$anat', '\$anatout', \$func_rx, \$anat_rx, \$angle);quit\"";
$command = "$command1 $command2";
print "$command\n";
system($command);
print "DONE WITH ROTATION\n"
}
```

INSTALLATION

1. Copies of Oblique_Reg.m along with example MATLAB and perl scripts and datasets can be found on cfMRI under /mnt/raid3/sdc/data/obreg_example
2. Copy the necessary files onto your computer.
3. Add the directory containing Oblique_Reg.m to your MATLAB search path – you may need to consult your local computing guru to help out with this. On UNIX machines, this can be done by adding the path within the startup.m file.

MISCELLANEOUS ISSUES

1. So far, the software has only been tested with axial anatomical volumes. If you have acquired your anatomical volume in an orientation that is not axial (e.g. coronal or sagittal) you will need to resample the volume in a RAI format with the AFNI command 3dresample prior to calling the program. Example: 3dresample -prefix out -orient RAI anat+orig