User's Guide
Neuroimage Processing ToolKit (NPTK) – Version.1.7 (beta)
fMRI Registration Software Pipeline for Functional Localization

Software Written by Ali Gholipour
SIP Lab, UTD, © 2005-2007
Revision 1.7 – May 2007

Table of Contents

1. Introduction
2. Data acquisition
3. Data formats
4. Pre-processing
    4.1. Pre-processing using **AFNI**
        4.1.1. Realignment using `3dVolreg`
        4.1.2. Co-registration using `3dAllineate`
        4.1.3. Skull-stripping using `3dSkullStrip`
    4.2. Pre-processing using **SPM2**
        4.2.1. Realignment using `spm_realign`
        4.2.2. Co-registration using `spm_coreg`
        4.2.3. Segmentation using `spm_segment`
    4.3. Pre-processing using **FSL**
        4.3.1. Realignment using `mcflirt`
        4.3.2. Co-registration using `flirt`
        4.3.3. Brain extraction using `bet`
    4.4. `rigidregistration`*
    4.5. `brainmask`*
5. `epicorrection`*
6. `brainwarp`*

Appendix
    A.1. AFNI user's example
    A.2. SPM user's example

* Software developed at SIP Lab, UTD. Important notes:
This software is developed on the basis of generic programming and the fundamental classes of NLM Insight Image Segmentation and Registration ToolKit (ITK) © 1999-2007 Insight Software Consortium.
THIS SOFTWARE IS DISTRIBUTED WITHOUT ANY WARRANTY; WITHOUT EVEN THE IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1. Introduction

Functional localization involves the application of a sequence of geometrical and statistical image processing operations in order to identify the location of brain activity or to produce functional/parametric maps with respect to the brain structure or anatomy. It should be noted that statistical analysis is performed in the native EPI space, and functional localization is carried out to obtain accurate overlays of extracted functional/parametric maps onto the high-resolution anatomy space. The main part of this procedure involves rigid/affine co-registration of EPI to MRI followed by `epicorrection` which corrects for the effect of local nonlinear geometric distortions in EPI.

2. Data acquisition

A typical fMRI analysis study involves two main image acquisitions:
(1) fMRI EPI time series, (~5-10 minutes)
(2) A reference high-resolution structural MRI (T1 or MPRAGE), (~5-10 minutes)

The recommended protocol for more accurate functional localization involves the acquisition of (3) a high-resolution EPI sequence (5 measurements, 1.8×1.8×2.5 mm) (32 seconds), and/or (4) a low-resolution dual echo field map image (3.5×3.5×3.5 mm) (47 seconds) (which includes phase and magnitude images)

These images are referred to by the colored text indicated below:

(1) fMRI EPI time series	`EPIts`
(2) Reference anatomical image	`REFimage`
(3) High-resolution EPI scans	`EPIimage`
(4) Low-resolution phase field map image	`FMphaseimage`
and the magnitude field map image	`FMmagimage`

3. Data formats

The pre-processing pipeline can be done with various software packages, which means that various image formats may be used by the user. The software developed at the UTD SIP Lab. (`brainmask`, `epicorrection`, and `brainwarp`) supports the ANALYZE 7.5 and NIfTI formats, but not the AFNI's HEAD and BRIK format. Therefore, for AFNI users conversion into ANALYZE or NIfTI is required for functional localization. This can be achieved within a pre-processing script file using `3dAFNItoANALYZE`. Since functional localization is carried out without any interaction with statistical analysis, AFNI users do not need to convert the EPI time series into the ANALYZE or NIfTI format. Only the `REFimage` and the pre-processed `EPIimage` and `FMphaseimage` need to be converted. The default ANALYZE 7.5 orientation (**LPI**) should be used when using the SIP Lab functional localization software:

```
3dAFNItoANALYZE -orient LPI -4D EPIimage EPIimage+orig
3dAFNItoANALYZE -orient LPI -4D REFimage REFimage+orig
3dAFNItoANALYZE -orient LPI -4D FMphaseimage FMphaseimage+orig
3dAFNItoANALYZE -orient LPI -4D FMmagimage FMmagimage+orig
```

4. Pre-processing

The EPI fMRI pre-processing pipeline as needed for functional localization is described in this section. It is assumed that appropriate pre-processing considerations (such as slice time shift correction, bias correction and transient removal) are applied beforehand to EPI time series (**EPIts**) for analysis purposes. The pre-processing pipeline for functional localization involves three main components: realignment, co-registration, and brain extraction:

- Realignment is applied to the last two scans (scans 4 and 5) of **EPIimage** for motion correction and also to obtain a less noisy EPI image for **epicorrection**. The images after realignment are shown by "mc" prefix, i.e. **mcEPIimage**, and **mcEPIts**.

- Co-registration to **REFimage** is a fundamental step for functional localization in the space of the high-resolution anatomy. The co-registered images are shown by an "r" prefix, i.e., **rEPIimage** and **rFMphaseimage**. A typical result of fMRI analysis is indicated here by **functionalimage**, which should also be mapped to the space of **REFimage**. This can be simply achieved by co-registering the native EPI space (**mcEPIts**) to **REFimage** and applying the computed transformation to **functionalimage**. The result would be **rfunctionalimage**. The command line structure for co-registering the images into the anatomical image space (**REFimage**) will be shown in the sections that follow.

- A smooth binary brain mask image is usually needed for functional localization, and is required by **epicorrection**. The brain can be extracted from **REFimage** by various tools in different software packages. Here, the brain extracted image is referred to as **REFimageBrain**.

4.1. Pre-processing using AFNI

4.1.1. Realignment using 3dVolreg

```
3dvolreg -prefix mcEPIimage -base 4 [options] EPIimage
% The following 3dTstat command calculates the average of the input scans
3dTstat -prefix mmcEPIimage [options] 'mcEPIimage[3-4]'

3dvolreg -prefix mcEPIts -base n [options] EPIts
3dTstat -prefix mmcEPIts [options] mcEPIts
```

4.1.2. Co-registration using 3dAllineate

```
% co-register high-resolution EPI to REFimage
3dAllineate -base REFimage -prefix rEPIimage mmcEPIimage

% co-register field map image to REFimage
3dAllineate -base REFimage -1Dfile FM2REF.1D FMmagimage
3dAllineate -1Dapply FM2REF.1D -prefix rFMphaseimage FMphaseimage

% co-register native EPI space and functional images to REFimage
3dAllineate -base REFimage -1Dfile EP2REF.1D mmcEPIts
3dAllineate -1Dapply EP2REF.1D -prefix rfunctionalimage functionalimage
```

4.1.3. Skull-stripping using 3dSkullStrip

```
3dSkullStrip -input REFimage -prefix REFimageBrain
```

4.2. Pre-processing using SPM2

4.2.1. Realignment using spm_realign

```
% GUI: realign (EPIimage1,... EPIimageN); output: meanEPIimage
% script:
spm_defaults;
P = ['EPIimage1';'EPIimage2';...;'EPIimageN'];
spm_realign(P,defaults.realign);
spm_reslice(P,struct('mean',1));
% results in meanEPIimage, rEPIimage1,..., rEPIimageN
% see help spm_realign, and help spm_reslice for more options
```

4.2.2. Co-registration using spm_coreg

```
% GUI: coregister (REFimage, meanEPIimage)-> rmeanEPIimage
%    coregister (REFimage, FMmagimage, FMphaseimage)-> rFMphaseimage
%    coregister (REFimage, mmcEPIts, functionalimage)-> rfunctionalimage
% script:
VG = spm_vol(fullfile(cd,'REFimage'));
VF = spm_vol(fullfile(cd,'meanEPIimage'));
x = spm_coreg(VG,VF);
spm_reslice([VG,VF],struct('which',1));
% results in rmeanEPIimage also referred to as rEPIimage
% see help spm_coreg, and help spm_reslice for more options
```

4.2.3. Segmentation using spm_segment

```
% GUI: segment (REFimage) -> REFimage_seg1 (seg1), seg2, seg3 ->
% imcalc (REFimage,seg1,seg2,seg3); i1.*i2+i1.*i3+i1.*i4 -> REFimageBrain
% script:
spm_defaults;
PF = fullfile(cd,'REFimage');
PG = fullfile(spm('Dir'),'templates','T1.mnc');
flags = defaults.segment;
spm_segment(PF,PG,flags);
V1 = spm_vol(fullfile(cd,'REFimage'));
V2 = spm_vol(fullfile(cd,'REFimage_seg1'));
V3 = spm_vol(fullfile(cd,'REFimage_seg2'));
V4 = spm_vol(fullfile(cd,'REFimage_seg3'));
Vo = V1;
Vo.fname = fullfile(cd,'REFimageBrain');
Vi = [V1;V2;V3;V4];
Vo = spm_imcalc(Vi,Vo,'i1.*i2+i1.*i3+i1.*i4',{0,0,0});
% results in REFimageBrain
% see help spm_segment for more options
```

4.3. Pre-processing using FSL

4.3.1. Realignment using mcflirt

```
mcflirt -in EPIimage -out mcEPIimage -meanvol [options]
```

```
% see mcflirt -help for more options to edit default parameters

mcflirt -in EPIts -out mcEPIts -meanvol [options]
```

4.3.2. Co-registration using `flirt`

```
% co-register high-resolution EPI to REFimage
flirt -in mcEPIimage -ref REFimage -out rEPIimage [options]
% see flirt -help for more options to edit default parameters

% co-register field map image to REFimage
flirt -in FMmagimage -ref REFimage -omat FM2REF.mat [options]
flirt -in FMphaseimage -init FM2REF.mat -ref REFimage -applyxfm -out
      rFMphaseimage

% co-register native EPI space and functional images to REFimage
flirt -in mmcEPIts -ref REFimage -omat EPI2REF.mat [options]
flirt -in functionalimage -init FM2REF.mat -ref REFimage -applyxfm
      -out rfunctionalimage
```

4.3.3. Brain extraction using `bet`

```
bet REFimage REFimageBrain [options]
% see bet -help for more options to edit default parameters
```

4.4. rigidregistration

- Input(s): `mcEPIimage, REFimage`
- Output(s): `rEPIimage`
- Usage:

```
rigidregistration REFimage mcEPIimage rEPIimage
```

- Description:
  `rigidregistration` is an alternative for co-registration tools discussed in previous sections (i.e. `spm_coreg`, `3dAllineate`, and `flirt`) which performs a conservative rigid registration (translation and weighted rotation) for multi-modality MRI scans obtained in one scanning session. The program starts with an initial identity transformation assuming minimum head motion inside the scanner. Optimum transformation parameters are obtained based on the maximization of mutual information.

4.5. brainmask

- Input(s): `REFimageBrain`
- Output(s): `BrainMaskImage`
- Usage:

```
brainmask REFimageBrain BrainMaskImage (thresholdValue – default .15)
% see brainmask -help for more options
```

- Description:

brainmask generates a smooth binary brain mask image for EPI correction. The input to the program (**REFimageBrain**) is a previously brain extracted ANALYZE 7.5 or NIfTI image with .img or .nii extension as obtained in sections 4.1.3, 4.2.3, or 4.3.3. **brainmask** is primarily applied to the high-resolution anatomical image but can also be used for different modalities including EPI. The argument **thresholdValue** may be set as input to expand or shrink the mask on the cortical surface. The suggested range is 0.12 to 0.20.

5. epicorrection

- Input(s): **rEPIimage, BrainMaskImage, rFMphaseimage, REFimage**
- Output(s): **deformationfieldImage, wEPIimage**
- Usage:
  epicorrection -e rEPIimage -b BrainMaskImage (-f rFMphaseimage) (-r REFimage) (-d deformationfieldImage) (-o wEPIimage) [options]

```
% see epicorrection -help
usage: epicorrection [options]

Available options are:
-e  EPIImageFileName (input)
-f  FMphaseimageFileName (input)
-r  ReferenceImageFileName (input)
-b  BrainMaskImageFileName (input)
-o  OutputImageFileName (output)
-d  DeformationFieldImageFileName (output) with the ".vtk" extension

Field map distortion correction input parameters are:
-te Echo time difference in ms (default 7.65-5.19 = 1.46 ms)
-bw Phase encoding bandwidth/resolution (Hz/Pixels) (default 2056/128)
-ir Field map phase image intensity range (default 4096)

-drd DistortionRegionDetection (1=ON or 0=OFF) (default = 1)
-nr  2ndStageNonrigidRegistration (1=ON, or 0=OFF) (default = 0)
```

- Description:
  epicorrection corrects the EPI spatial distortions by field map undistortion and/or non-rigid registration. All the input and output images, except **deformationfieldImage**, are in the ANALYZE 7.5 or NIfTI format (with .img or .nii extensions). The argument **deformationfieldImage** is a three dimensional vector image representing the computed deformation field with .vtk extension. **rEPIimage** is the output of co-registration as discussed in sections 4.1.2, 4.2.2 and 4.3.2; and **BrainMaskImage** is the output of **brainmask** as discussed in section 4.4. Either a reference anatomical image or a phase field map image should be provided. In case a field map is provided, appropriate values for echo time difference, phase encoding bandwidth/resolution and phase image intensity range should be provided. If these parameters are not defined, the default values listed in the command line help will be used.
  For full FOV **rEPIimage** and **REFimage**, the default DistortionRegionDetection option (-drd 1) is suggested. When applying to limited FOV images, –drd 0 should be used. If the –nr option is set to 1, a second stage non-rigid registration is performed after field map based distortion correction.

6. brainwarp

- Input(s): `rfunctionalimage, deformationfieldImage`
- Output(s): `wfunctionalimage`
- Usage:
  ```
  brainwarp rfunctionalimage deformationfieldImage wfunctionalimage
  ```
- Description:
  brainwarp warps an input image based on a pre-computed **deformationfieldImage** (e.g., output of **epicorrection** with .vtk extension). The input can be the outcome of the functional image analysis co-registered to the space of reference anatomical image (e.g., **rfunctionalimage** as discussed in section 4), and the output will be **wfunctionalimage** with the corrected spatial overlay on the brain anatomy.

Appendix A.1: AFNI user's example

```
% Important note: this is just an example and various parameters may need to be
defined for any specific image acquisition and processing protocol.
% try -help for most of the commands bellow to see a full list of options and to
change the default parameters
% AFNI files have been converted to ANALYZE format using 3dAFNItoANALYZE

3dvolreg -base 4 -prefix s01_epi_hr_reg.hdr s01_epi_hr.hdr
3dTstat -prefix ms01_epi_hr.hdr ‘s01_epi_hr_reg[3-4]’

3dvolreg -base 104 -prefix s01_epi_lr_ts_reg.hdr s01_epi_lr_ts.hdr
3dTstat -prefix ms01_epi_lr.hdr ‘s01_epi_lr_ts_reg[100-104]’

3dAllineate  -base  s01_mprage.hdr  -prefix  rs01_epi_hr.hdr  -mi  -clbin  30
ms01_epi_hr.hdr
3dAllineate -base s01_mprage.hdr -1Dfile FM2REF.1D -clbin 30 s01_B0map_mag.hdr
3dAllineate -base s01_mprage.hdr -1Dapply FM2REF.1D -prefix rs01_B0map_phase.hdr
s01_B0map_phase.hdr
3dAllineate -base s01_mprage.hdr -1Dfile -clbin 30 EP2REF.1D ms01_epi_lr.hdr
3dAllineate -base s01_mprage.hdr -1Dapply EP2REF.1D -prefix rfmaps.hdr fmaps.hdr

3dSkullStrip -input s01_mprage.hdr -prefix s01_mprage_brain.hdr
brainmask s01_mprage_brain.hdr s01_brainmask.hdr

epicorrection  -e  rs01_epi_hr.hdr  -b  s01_brainmask.hdr  -r  s01_mprage.hdr  -f
rs01_B0map_phase.hdr -o s01_epi_hr_nreg.hdr -d s01_df.vtk

brainwarp ms01_epi_lr.hdr s01_df.vtk ms01_epi_lr_nreg.hdr
brainwarp rfmaps.hdr s01_df.vtk fmaps_nreg.hdr
```

Appendix A.2: SPM user's example

```
% Important note: this is just an example and various parameters may need to be
defined for any specific image acquisition and processing protocol.
% This is a compact GUI example. Scripts may be used according to section 4.2.
% SPM 2.0 GUI commands: (defaults: co-register and reslice)

Realign (s01_epi_hr_0003, s01_epi_hr_0004) -> means01_epi_hr
Realign (s01_epi_lr_0005, …, s01_epi_lr_0104) -> means01_epi_lr
Coregister (s01_mprage, means01_epi_hr) -> rmeans01_epi_hr
Coregister (s01_mprage, s01_B0map_mag, s01_B0map_phase) -> rs01_B0map_phase
Coregister (s01_mprage, means01_epi_lr, s01_fmap) -> rmeans01_epi_lr, rs01_fmap

Segment (s01_mprage) -> s01_mprage_seg1, s01_mprage_seg2, s01_mprage_seg3
Imcalc (s01_mprage, _seg1,_seg2,_seg3, i1.*i2+i1.*i3+i1.*i4) -> s01_mprage_brain

brainmask s01_mprage_brain.img s01_brainmask.img

epicorrection -e rmeans01_epi_hr.img -b s01_brainmask.img -r s01_mprage.img -f
rs01_B0map_phase.img -o s01_epi_hr_nreg.img -d s01_df.vtk

brainwarp rmeans01_epi_lr s01_df.vtk rmeans01_epi_lr_nreg.img
brainwarp rfmaps.img s01_df.vtk fmaps_nreg.img
```