Image and Volume Registration with AFNI

- Goal: bring images collected with different methods and at different times into alignment
- Facilitates comparison of data on a voxel-by-voxel basis
  - ✧ Functional time series data will be less contaminated by artifacts due to subject movement
  - ✧ Can compare results across scanning sessions once images are properly registered
- Most (all?) image registration methods now in use do pairwise aligment:
  - ✧ Given a base image **J(x)** and target image **I(x)**, find a geometrical transformation **T[x]** so that **I(T[x])≈J(x)**
  - ✧ **T[x]** will depend on some parameters
    - ➥ Goal is to find the parameters that make the transformed **I** a 'best fit' to **J**
  - ✧ To register an entire time series, each volume $\mathbf{I}_n(\mathbf{x})$ is aligned to **J(x)** with its own transformation $\mathbf{T}_n[\mathbf{x}]$, for n=0, 1, …
    - ➥ Result is time series $\mathbf{I}_n(\mathbf{T}_n[\mathbf{x}])$ for n=0, 1, …
    - ➥ User must choose base image **J(x)**

- Most image registration methods make 3 algorithmic choices:
  - How to measure mismatch **E** (for error) between **I(T[x])** and **J(x)**?
  - How to adjust parameters of **T[x]** to minimize **E**?
  - How to interpolate **I(T[x])** to the **J(x)** grid?
    - So can compare voxel intensities directly
- Existing AFNI programs match images by grayscale (intensity) values
  - **E** = (weighted) sum of squares differences = $\sum_x w(x) \cdot \{I(T[x]) - J(x)\}^2$
    - Only useful for registering 'like images':
      - SPGR↔SPGR, EPI↔EPI, but **not** SPGR↔EPI
  - Parameters in **T[x]** are adjusted by "gradient descent"
  - Several interpolation methods are available:
    - Default method is Fourier interpolation
    - Polynomials of order 1, 3, 5, 7 (linear, cubic, quintic, and heptic)
- Alternative method would be to match features computed from grayscale images:
  - Brain outline
  - Edges (places where image intensity changes abruptly in 1-2 pixels)
  - Such techniques can be used to match SPGR↔EPI volumes
    - Program **`3dAnatNudge`** can estimate SPGR↔EPI translations
    - But not rotations or warping

- AFNI program **3dvolreg** is for aligning 3D volumes
  - **T[x]** has 6 parameters:
    - Shifts along x-, y-, and z-axes; Rotations about x-, y-, and z-axes
  - Generically useful for intra- and inter-session alignment
  - Motions that occur within a single TR (2-3 s) cannot be corrected this way, since method assumes a rigid movement of the entire volume
- AFNI program **2dImReg** is for aligning 2D slices
  - **T[x]** has 3 parameters for each slice in volume:
    - Shift along x-, y-axes; Rotation about z-axis
  - Useful for sagittal EPI scans where dominant subject movement is 'nodding' motion that my be faster than TR
  - It is possible and sometimes even useful to run **2dImReg** to clean up nodding motion, followed by **3dvolreg** to deal with out-of-slice motion
- Hybrid 'slice-into-volume' registration:
  - Put each separate 2D image slice into the target volume with its own 6 movement parameters (3 out-of-plane as well as 3 in-plane)
  - Has been attempted by some AFNI users but the results are not wildly better than volume registration; method often fails on slices near edge of brain
    - More work is needed (i.e., send money)

- Intra-session registration:

```
3dvolreg -base 4 -heptic -clipit -zpad 4  \
         -prefix fred1_epi_vr             \
         -dfile fred1_vr_dfile            \
         fred1_epi+orig
```

  - ✧ **`-base 4`** ⇒ Selects sub-brick #4 of dataset `fred1_epi+orig` as base image **J(x)**
  - ✧ **`-heptic`** ⇒ Use 7th order polynomial interpolation (my personal favorite)
  - ✧ **`-clipit`** ⇒ Clip off negative voxels to zero
    - ➥ Negative voxels are artifacts of high-order interpolation methods
  - ✧ **`-zpad 4`** ⇒ Pad each target image, **I(x)**, with layers of zero voxels 4 deep on each face prior to shift/rotation, then strip them off afterwards
    - ➥ Zero padding is particularly desireable for **`-Fourier`** interpolation
    - ➥ Is also good for polynomial methods, since if there are large rotations, some data may get 'lost' when no zero padding if used (due to 4-way shift algorithm used for fast rotation)
  - ✧ **`-prefix fred1_epi_vr`** ⇒ Save output dataset into a new dataset with the given prefix name (e.g., `fred1_epi_vr+orig`)
  - ✧ **`-dfile fred1_vr_dfile`** ⇒ Save estimated movement parameters into a 1D (i.e., text) file with the given name
    - ➥ Can be plotted with command

```
1dplot -volreg -dx 5 -xlabel Time 'fred1_vr_dfile[1..6]'
```

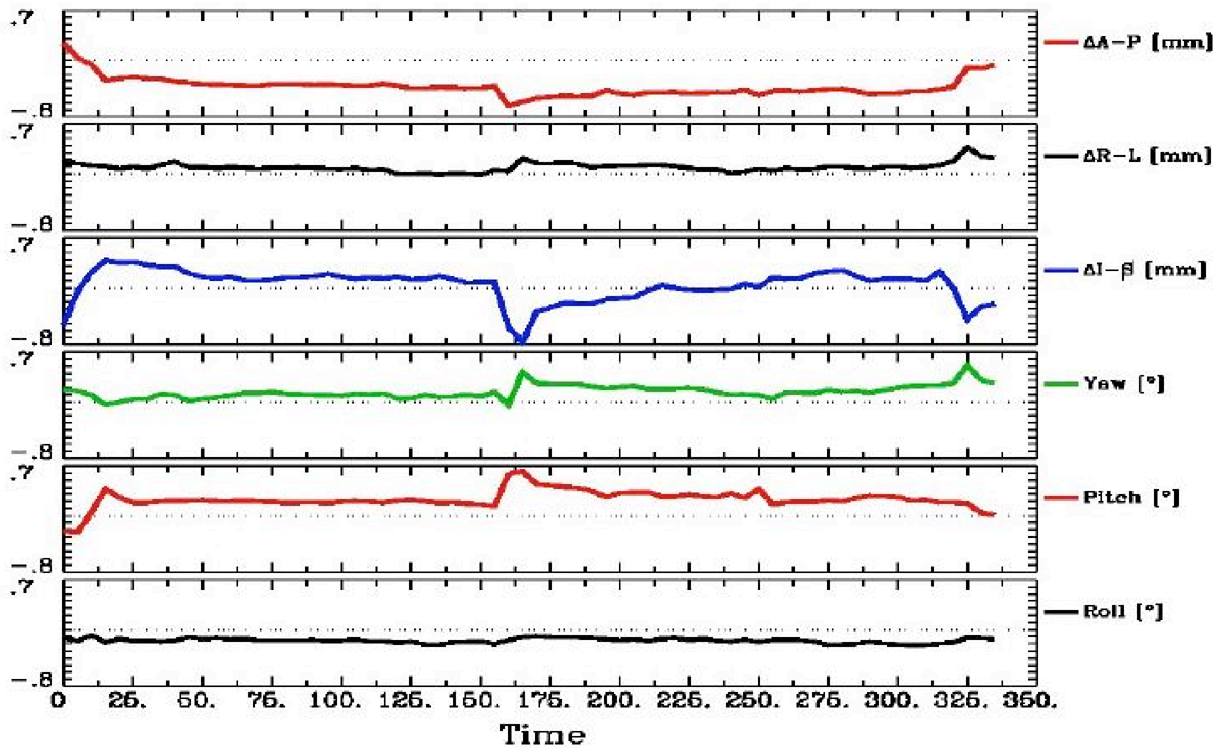
✧ Can now register second dataset from same session:

```
3dvolreg -base 'fred1_epi+orig[4]' -heptic -clipit -zpad 4 \
         -prefix fred2_epi_vr -dfile fred2_vr_dfile        \
         fred2_epi+orig
```

➥ Note base is from different dataset (**fred1_epi+orig**) than input (**fred2_epi+orig**)

➪ Aligning all EPI volumes from session to EPI closest in time to SPGR

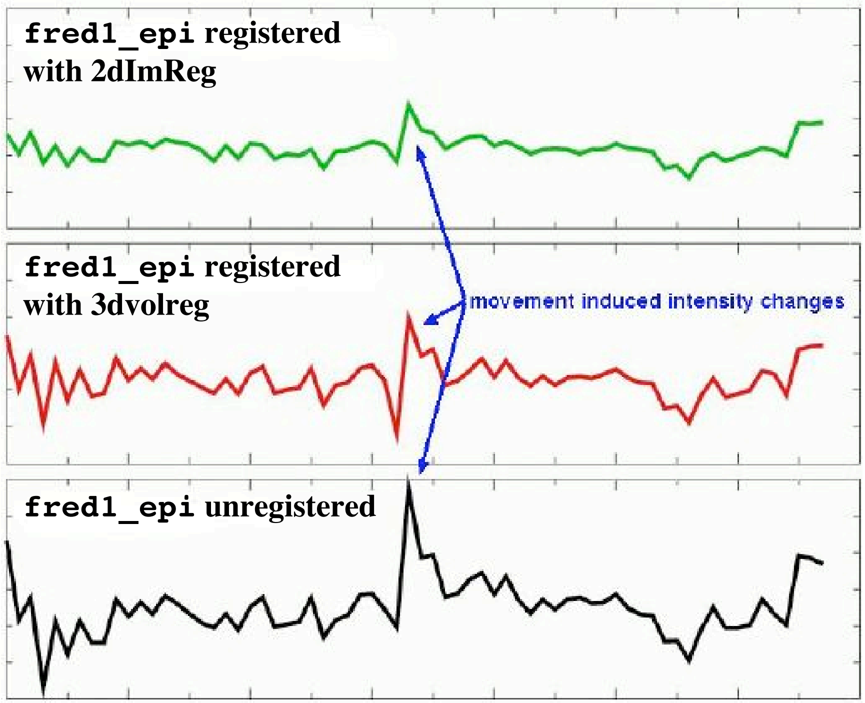
`1dplot -volreg -dx 5 -xlabel Time 'fred2_vr_dfile[1..6]'` gives:



➥ Note motion peaks at time ≈ 160s: subject jerked head up at that time

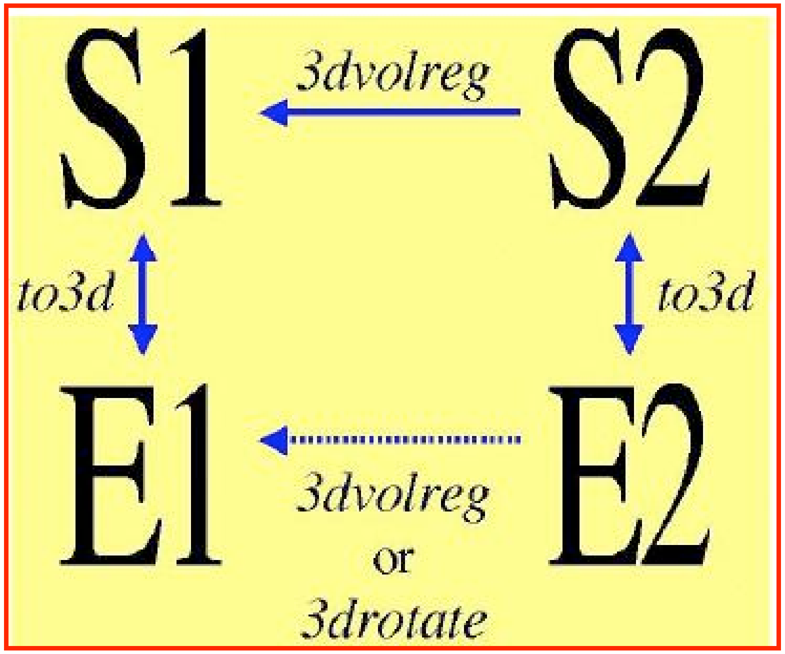
- Examination of time series **fred2_epi+orig** and **fred2_epi_vr_+orig** shows that head movement up and down happened within about 1 TR interval
  - Assumption of rigid motion of 3D volumes is not good for this case
  - Can do 2D slice-wise registration with command

```
2dImReg -input fred2_epi+orig        \
        -basefile fred1_epi+orig    \
        -base 4 -prefix fred2_epi_2Dreg
```



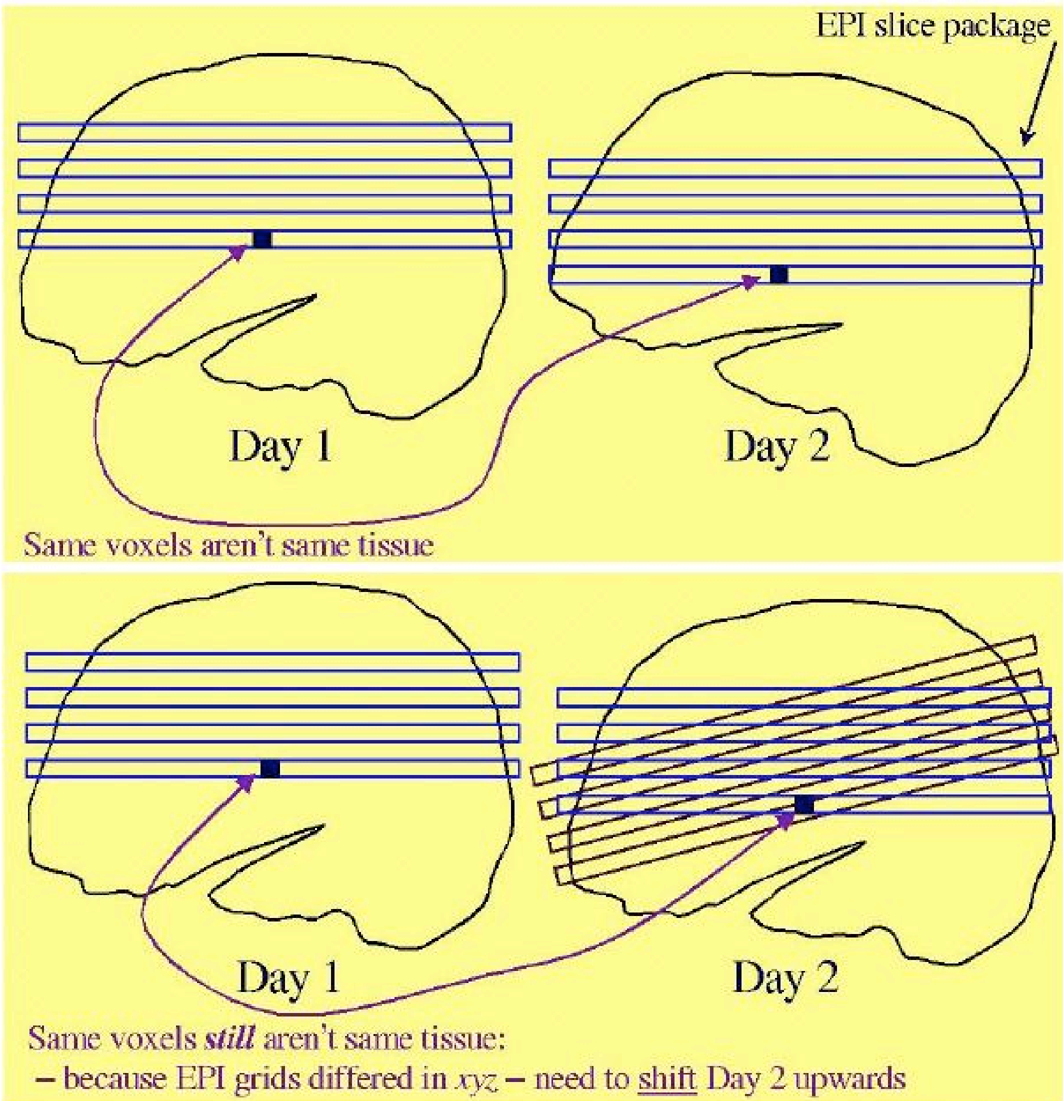
- Graphs of a single voxel time series near the edge of the brain:
  - Top = slice-wise alignment
  - Middle = volume-wise adjustment
  - Bottom = no alignment
- For this example, **2dImReg** appears to produce better results. This is because most of the motion is 'head nodding' and the acquisition is sagittal
- You should also use AFNI to scroll through the images (using the Index control) during the period of pronounced movement, on all 3 datasets

- Intra-subject, inter-session registration (for multi-day studies on same subject)
  - Longitudinal or learning studies; re-use of cortical surface models
  - Transformation between sessions is calculated by registering high-resolution anatomicals from each session
    - **`to3d`** defines defines relationship between EPI and SPGR in each session
    - **`3dvolreg`** computes relationship between sessions
    - So can transform EPI from session 2 to orientation of session 1



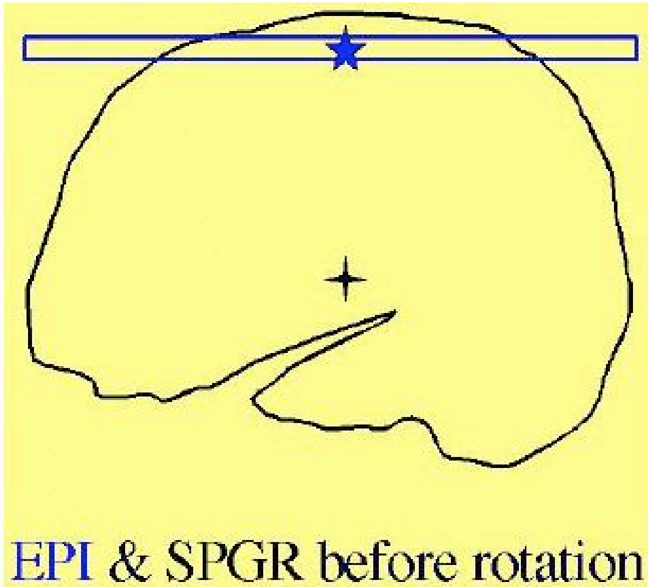
  - Issues in inter-session registration:
    - Subject's head will be positioned differently (in orientation and location)
      - xyz-coordinates and anatomy don't correspond
    - Anatomical coverage of EPI slices will differ between sessions
    - Geometrical relation between EPI and SPGR differs between session
    - Slice thickness may vary between sessions (try not to do this, OK?)

- Anatomical coverage differs

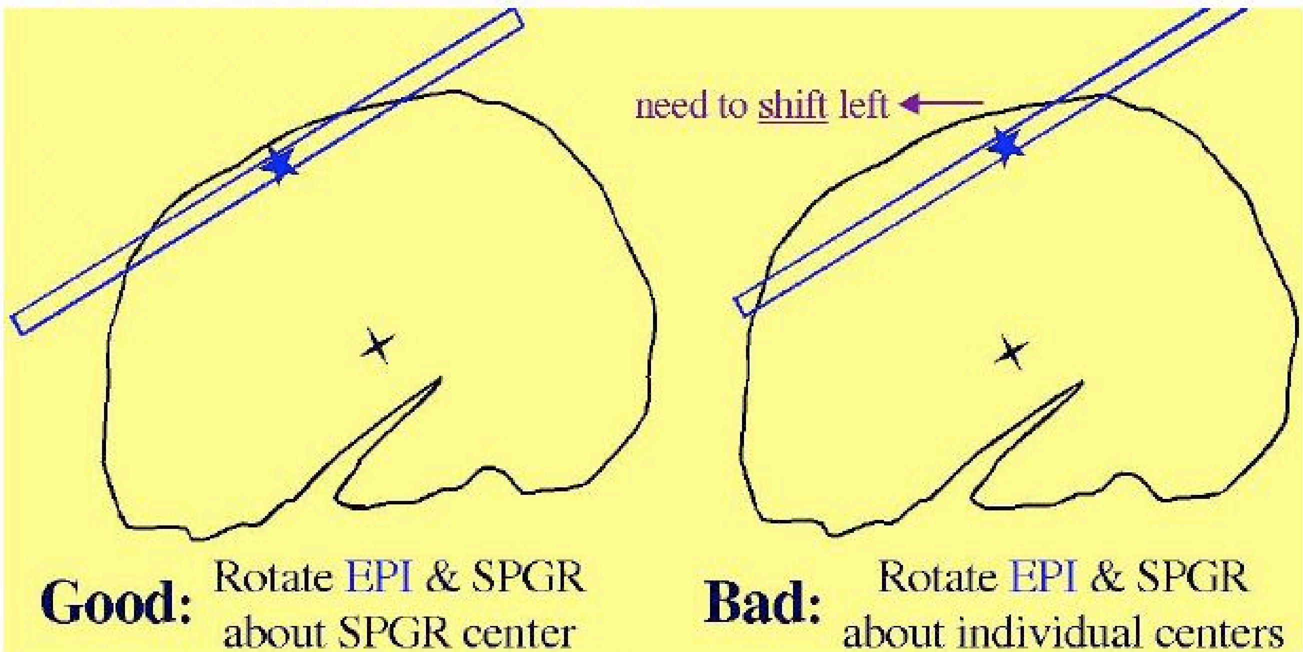


✧ At acquisition

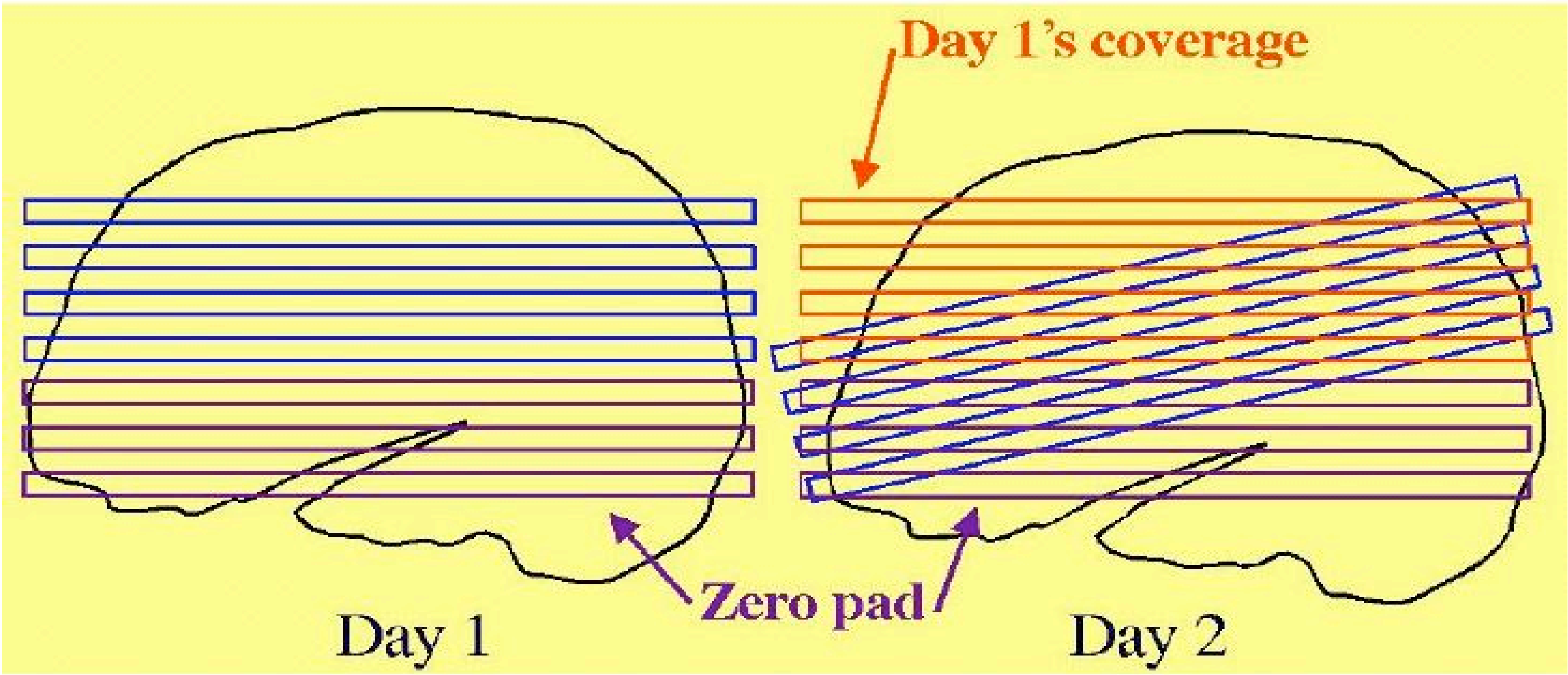
✧ After rotation to same orientation, then clipping to day 2 xyz-grid



- Another problem: rotation occurs around center of individual datasets



- Solutions to these problems:
  - Add appropriate shift to E2 on top of rotation
    - Allow for xyz shifts between days (E1-E2), and center shifts between EPI and SPGR (E1-S1 and E2-S2)
  - Pad EPI datasets with extra slices of zeros so that aligned datasets can fully contain all data from all sessions
  - Zero padding of a dataset can be done in **`to3d`** (at dataset creation time), or later using **`3dZeropad`**
  - **`3dvolreg`** and **`3drotate`** can zero pad to make the output match a "grid parent" dataset in size and location



- Recipe for intra-subject S2-to-S1 transformation:
  1. Compute S2-to-S1 transformation:
     ```
     3dvolreg -twopass -clipit -zpad 4 -base S1+orig \
              -prefix S2reg  S2+orig
     ```
  - Rotation/shift parameters are saved in **`S2reg+orig.HEAD`**
  2. If not done before (e.g., in **`to3d`**), zero pad E1 datasets:
     ```
     3dZeropad -z 4 -prefix E1pad  E1+orig
     ```
  3. Register E1 datasets within the session:
     ```
     3dvolreg -clipit -base 'E1pad+orig[4]' -prefix E1reg \
             E1pad+orig
     ```
  4. Register E2 datasets within the session, at the same time executing larger rotation/shift to session 1 coordinates that were saved in `S2reg+orig.HEAD`:
     ```
     3dvolreg -clipit -base 'E2+orig[4]'  \
             -rotparent S2reg+orig        \
             -gridparent E1reg+orig       \
             -prefix E2reg  E2reg+orig
     ```
  - **`-rotparent`** tells where the inter-session transformation comes from
  - **`-gridparent`** defines the output grid location/size of new dataset
    - Output dataset will be shifted and zero padded as needed to lie on top of **`E1reg+orig`**

- Recipe above does not address problem of having different slice thickness (EPI and/or SPGR) in different sessions
  - Best solution: pay attention when you are scanning, and always use the same slice thickness for the same type of image
  - OK solution: use **`3dZregrid`** to linearly interpolate datasets to a new slice thickness
- Recipe above does not address issues of slice-dependent time offsets stored in data header from **`to3d`** (e.g., 'alt+z')
  - After interpolation to a rotated grid, voxel values can no longer be said to come from a particular time offset, since data from different slices will have been combined
  - Before doing this spatial interpolation, it makes sense to time-shift dataset to a common temporal origin
  - Time shifting can be done with program **`3dTshift`**
    - Or by using the **-tshift** option in **`3dvolreg`**, which first does the time shift to a common temporal origin, then does the 3D spatial registration

- Further reading at the AFNI web site
  - File **README.registration** (plain text) has more detailed instructions and explanations about usage of **`3dvolreg`**
  - File **regnotes.pdf** has some background information on issues and methods used in FMRI registration packages