# *3dQwarp*
## and Its *Nwarp* Friends

Or, How I Learned to Stop Worrying and Love Getting My Datasets all Warped

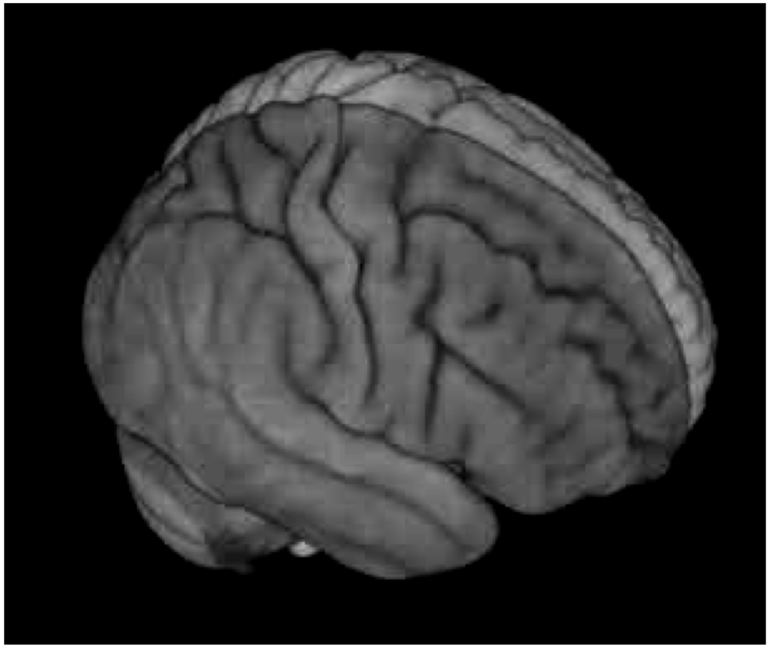
# Linear and Nonlinear Warping

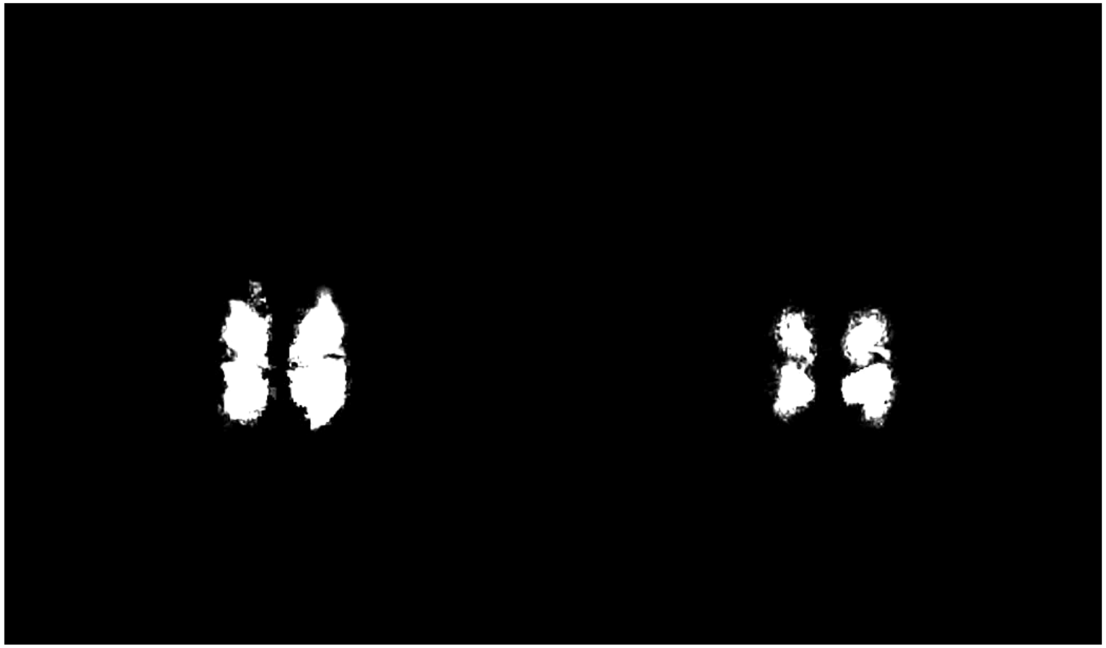
- The Central Equation:
  - $S(x)$ = source image $B(x)$ = base image
  - ➢ $\boxed{S(W(x)) \approx B(x)}$ where $W(x)$ = desired warp function = shows where each point x in B maps to in S
- *3dAllineate*: $W(x) = \mathbf{M}x$ where $\mathbf{M}$ = 4x3 matrix
  - M has 12 parameters to optimize
- *3dQwarp*: $W(x) = W_1( W_2( \ldots W_{n-1}( W_n(x) )) \ldots ))$
  - Each $W_k(x)$ is a 3D polynomial function over a "patch" that covers part of the 3D brain volume
  - Patches start big [$W_1(x)$] and shrink and shrink
  - Cubic patch = 24 parameters ; Quintic = 81 params
  - By the end, 1000s of parameters have been used

# The Good and The Ugly

- the Good:
  - Nonlinear warping can match anatomical structures between subjects more closely than linear transformation
  - Can also be used for intra-subject warping for high accuracy matching (e.g., pre- and post-surgery)
- the Ugly:
  - Nonlinear warping can seriously distort when it tries to match in regions that don't really "fit together" (e.g., 2 gyri in one person, 1 gyrus in another)
  - Extraneous small features can drive warping in strange ways (unlike linear transformation)
  - Partial brain coverage is a problem

# Start: Looking Good



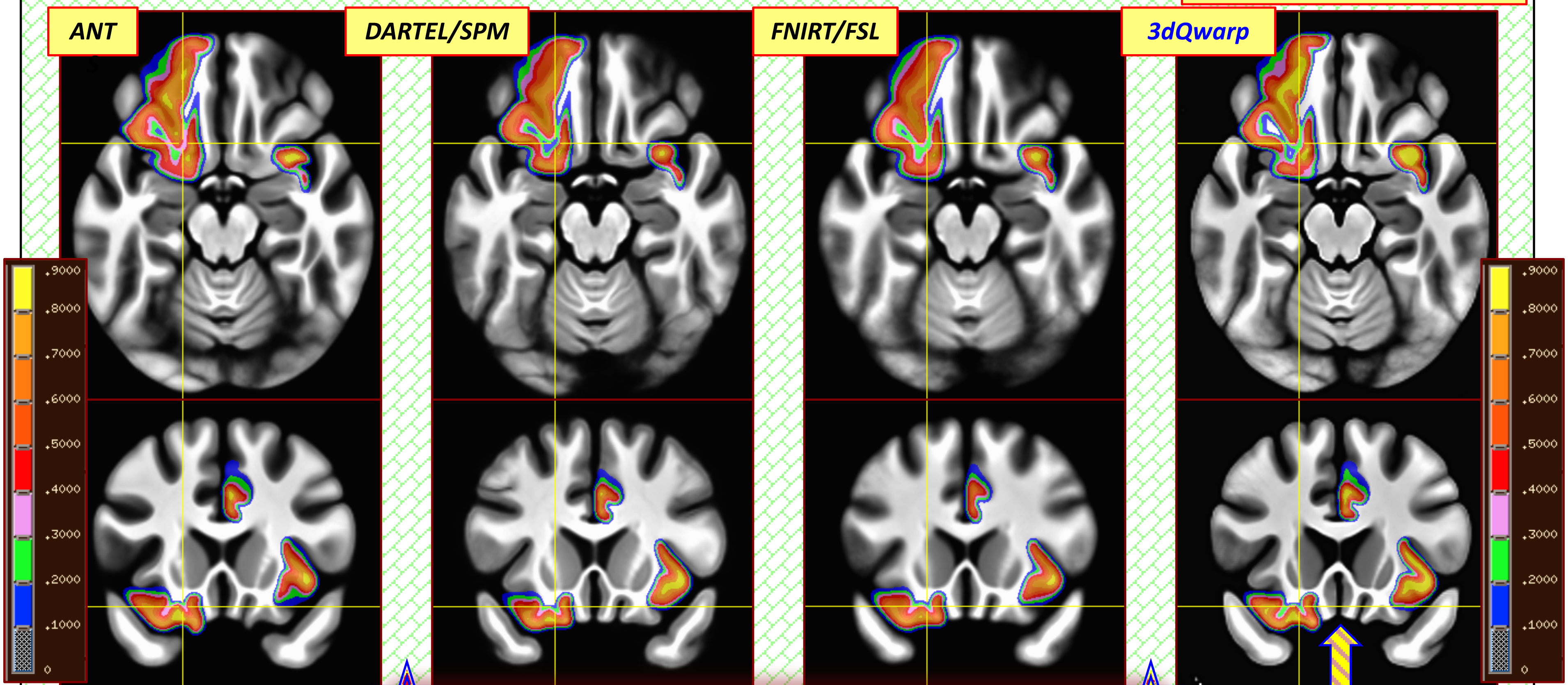


Compare FSL *FNIRT* vs AFNI *3dQwarp*
Average of 101 brain volumes warped to template

# Good Matches to Anatomical Labels

Align MindBoggle 101 $T_1$ Datasets to Separate Template:
Overlap Probability Maps for 3 of the Labeled Regions

**LH**: lateral orbital frontal
**RH**: caudal anterior cingulate
**RH**: insula

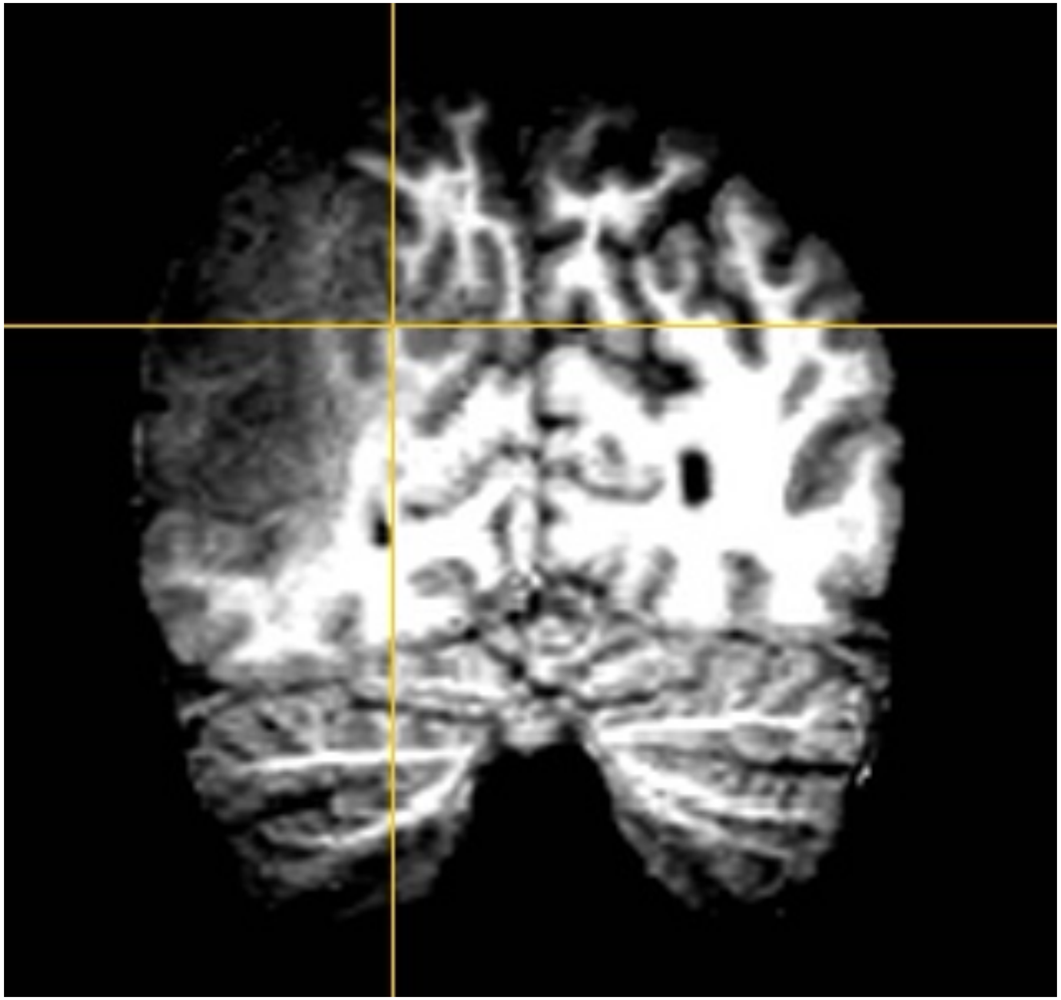



*ANTS* & *DARTEL* & *FNIRT* run with default settings

***3dQwarp*** distribution of overlap probabilities is 2nd order stochastically dominant in a majority of 62 labeled regions

More yellow in the overlay means more 90+% overlap in labels

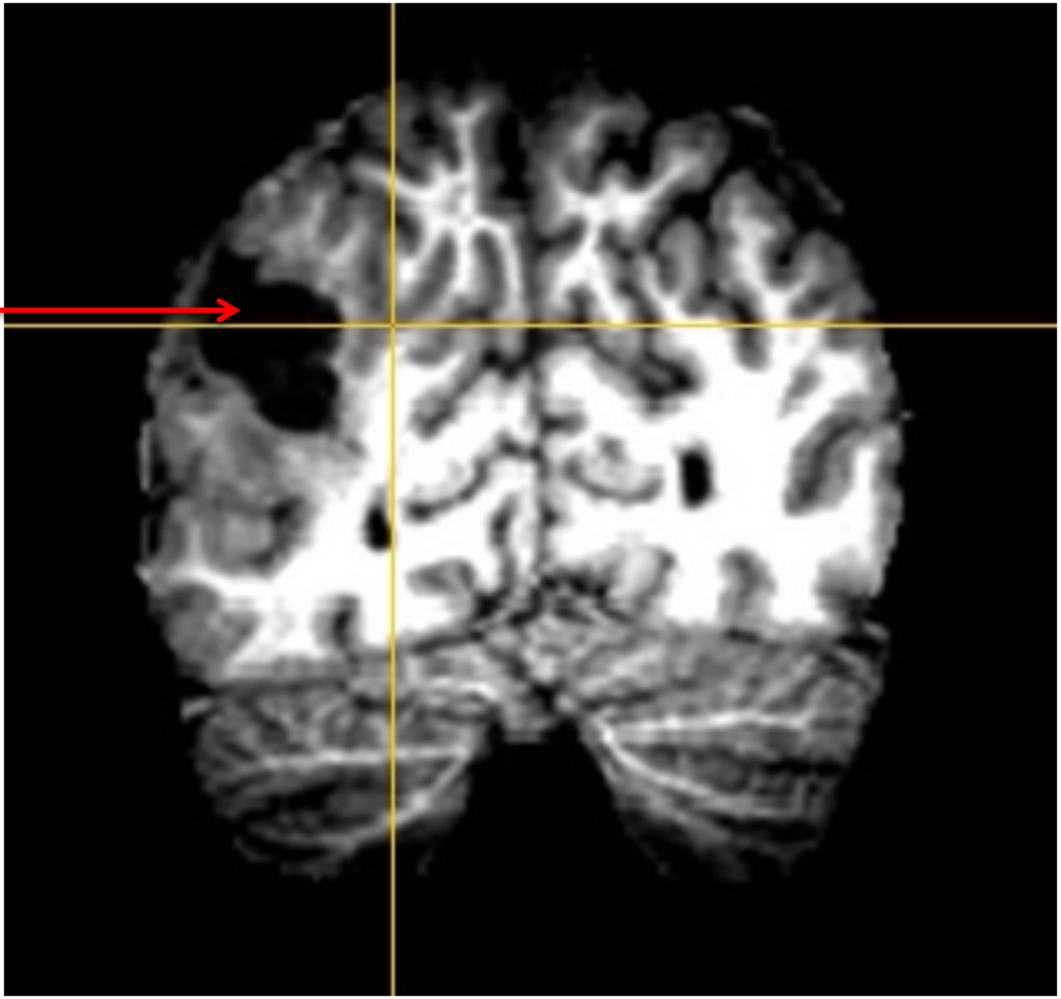
# Maybe Even Useful: Neurosurgery



Pre-surgical volume

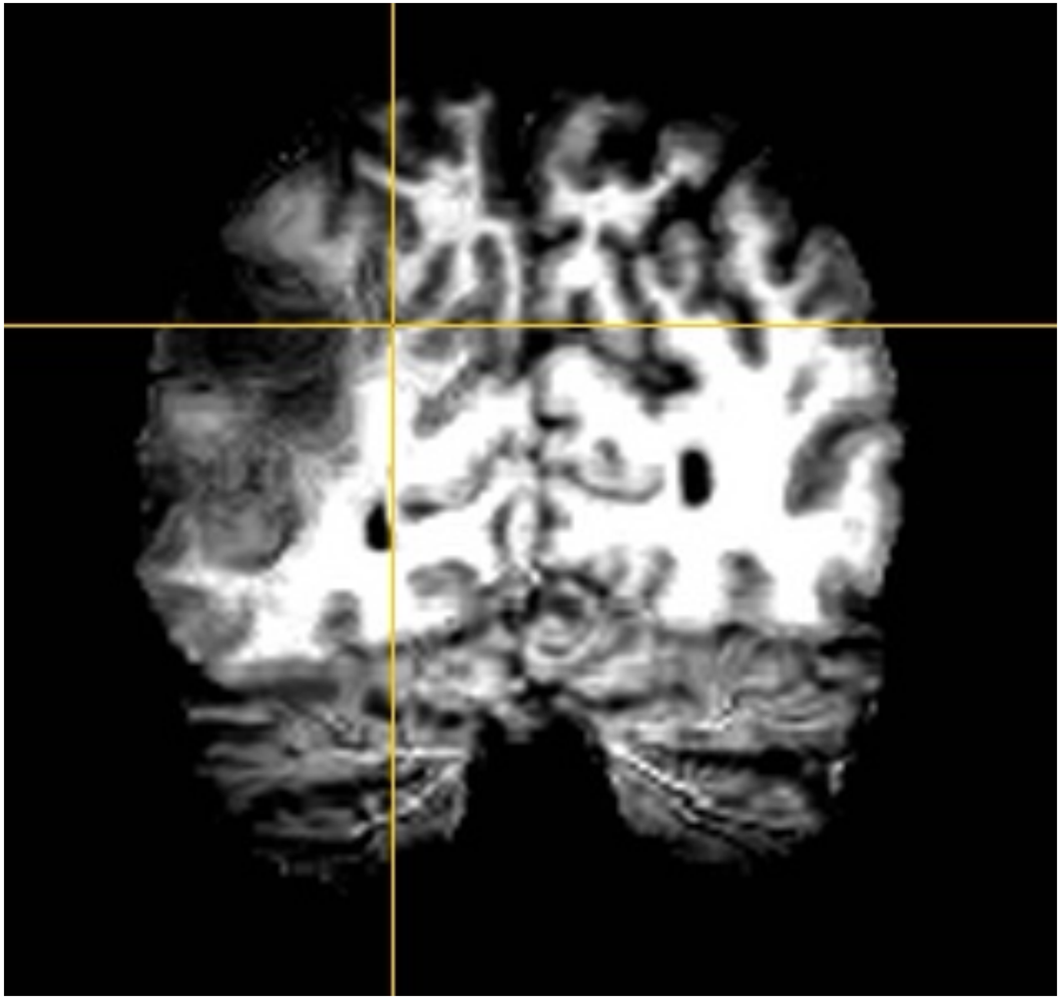
# Neurosurgery

Manually drawn "exclusion Mask"



Post-surgical volume

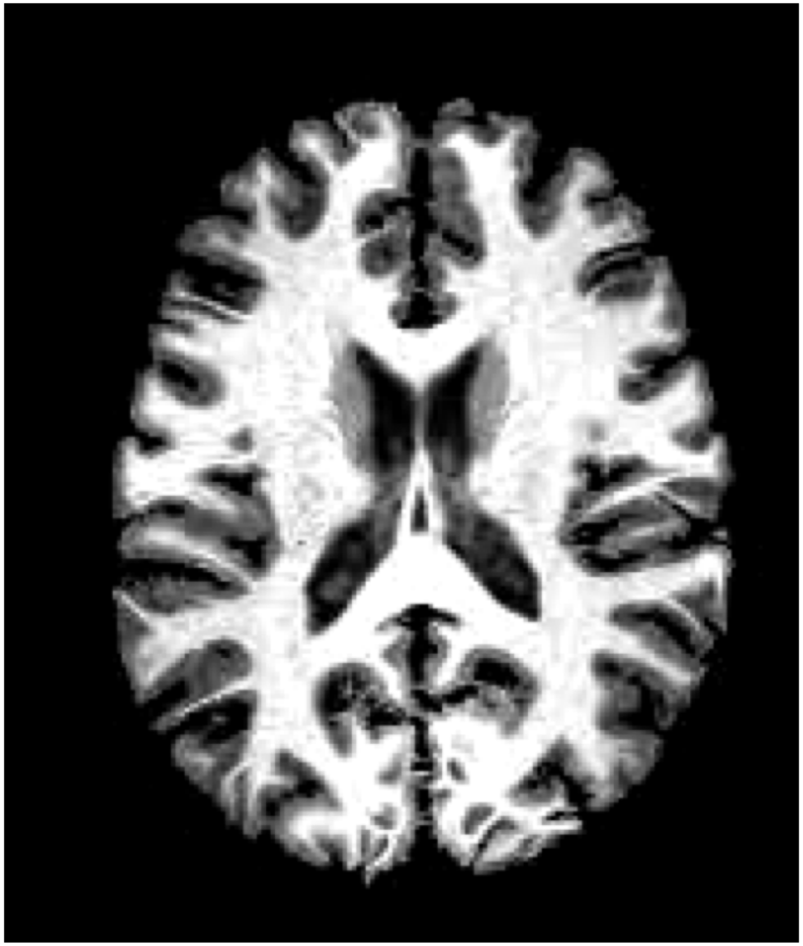
# Neurosurgery



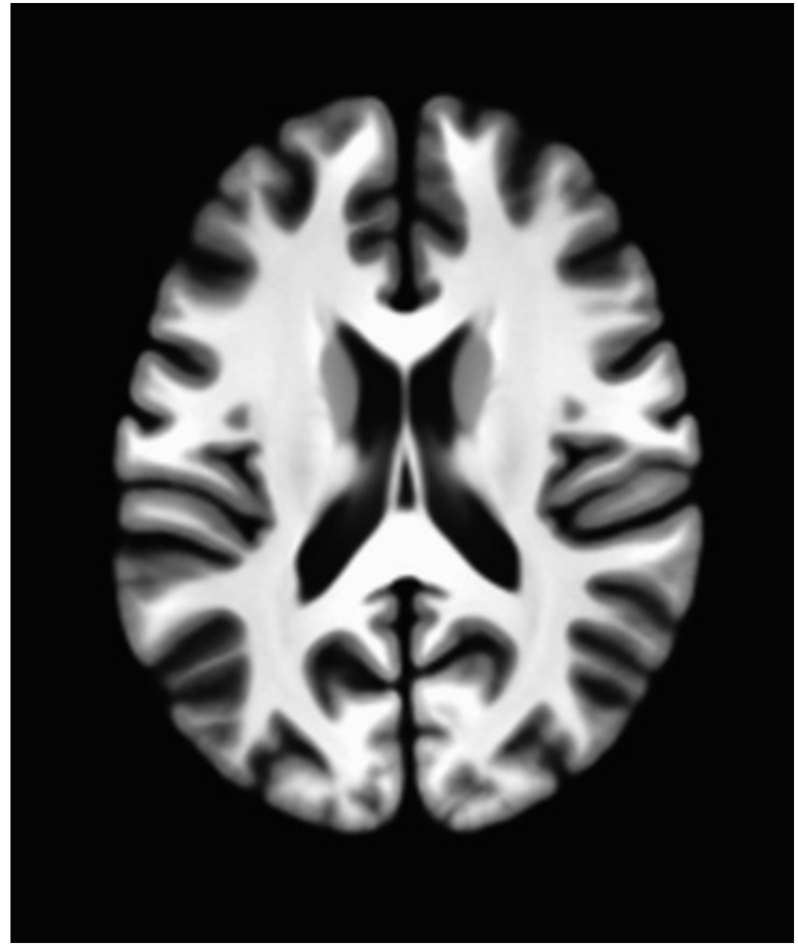
Pre-surgical volume

***3dQwarp***-aligned to Post-surgical volume

# But … Some Ugly



All 101 Volumes
After Warping



Mean of 101 Volumes
After Warping

# How to Make a Template

- Given a collection of skull-stripped structural ($T_1$-weighted) datasets
- Script *@toMNI_Awarp* pre-processes each dataset (*3dUnifize* and *@auto_tlrc*)
  - *3dUnifize* make the image intensity more uniform over the volume
- Script *@toMNI_Qwarpar* runs *3dQwarp* to collectively warp them together over finer and finer patch levels
- Has been used to create Haskins pediatric brain atlas (now distributed with AFNI)

# What Else to Do with a Warp?

- Warp another dataset the same way
  - *3dNwarpApply* (e.g., carry EPI to template)
- Warp some discrete points the same way
  - *3dNwarpXYZ* (e.g., eCog electrode locations)
- Compute voxel-wise functions of a warp
  - *3dNwarpFuncs* (e.g., volume distortion)
- Compose multiple warps together
  - *3dNwarpCat* and *3dNwarpCalc*
- Can compute inverse warp $W^{-1}(x)$, to map locations in $S(x)$ to matching locations in B
  - $S(W(x)) \approx B(x) \Rightarrow S(x) \approx B(W^{-1}(x))$

# How to Use *3dQwarp*

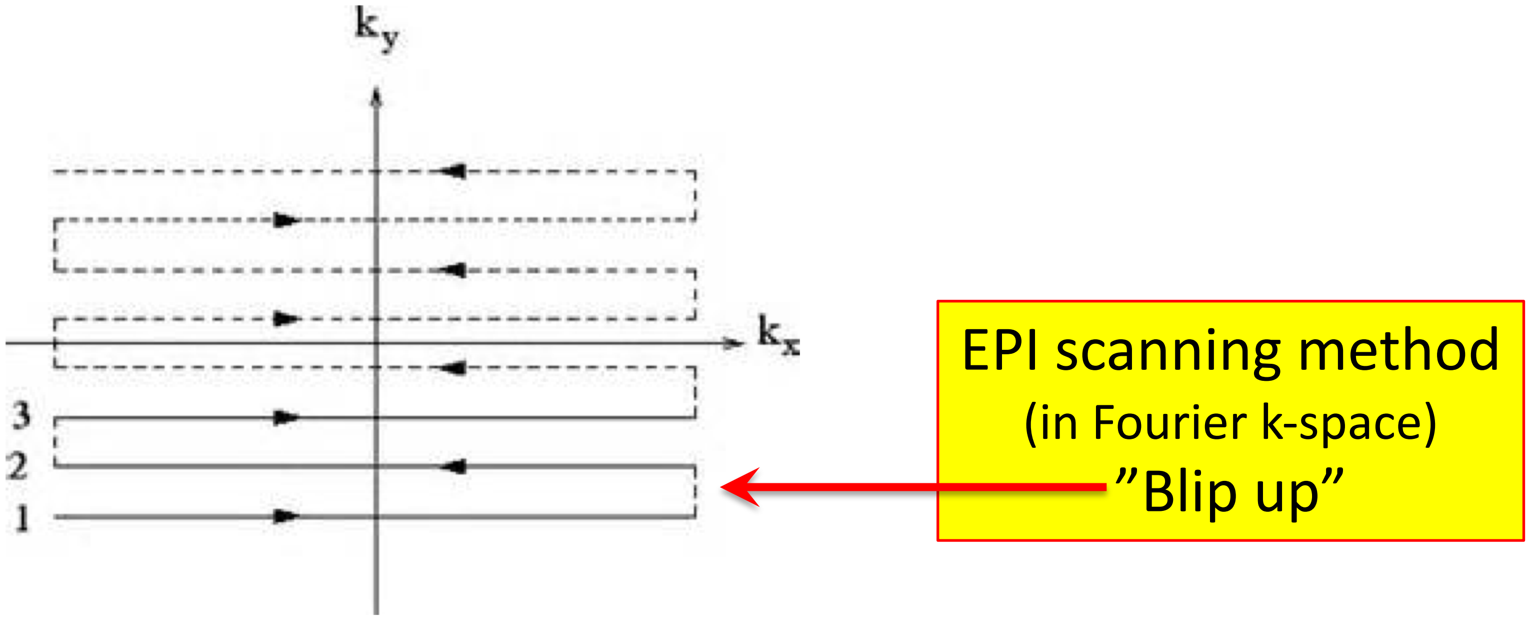
- Run it yourself (the “old school” or “real man” way)
- *auto_warp.py* (easier, less flexible)
- ☛ Use *@SSwarper* script to ***S***kull ***S***trip and ***W****arp* to MNI template
- Use ‘*-tlrc_NL_warp*’ option in *afni_proc.py* to have transformation to template space be done via *auto_warp.py*
- Use *@toMNI_Awarp* and *@toMNI_Qwarpar* to create a study (or population) specific template
- Use ‘*-plusminus*’ option in *3dQwarp* to warp blip-up and blip-down EPI datasets to “meet in the middle” (script *unWarpEPI.py*)

# Yet to Be Done

- ~~Incorporate more fully into *afni_proc.py* and *uber_subject.py*~~
  - ~~Warping to template; un-warping EPI distortions~~
- Explore how much nonlinear warping to a template can improve group analysis in functional and anatomical MRI
  - And improvements to *3dSeg* (segmentation)
- Extend matching algorithm to allow label-based matches, vs. existing intensity-based
- Speed the damn thing up*!*
- Write a paper about it*!*

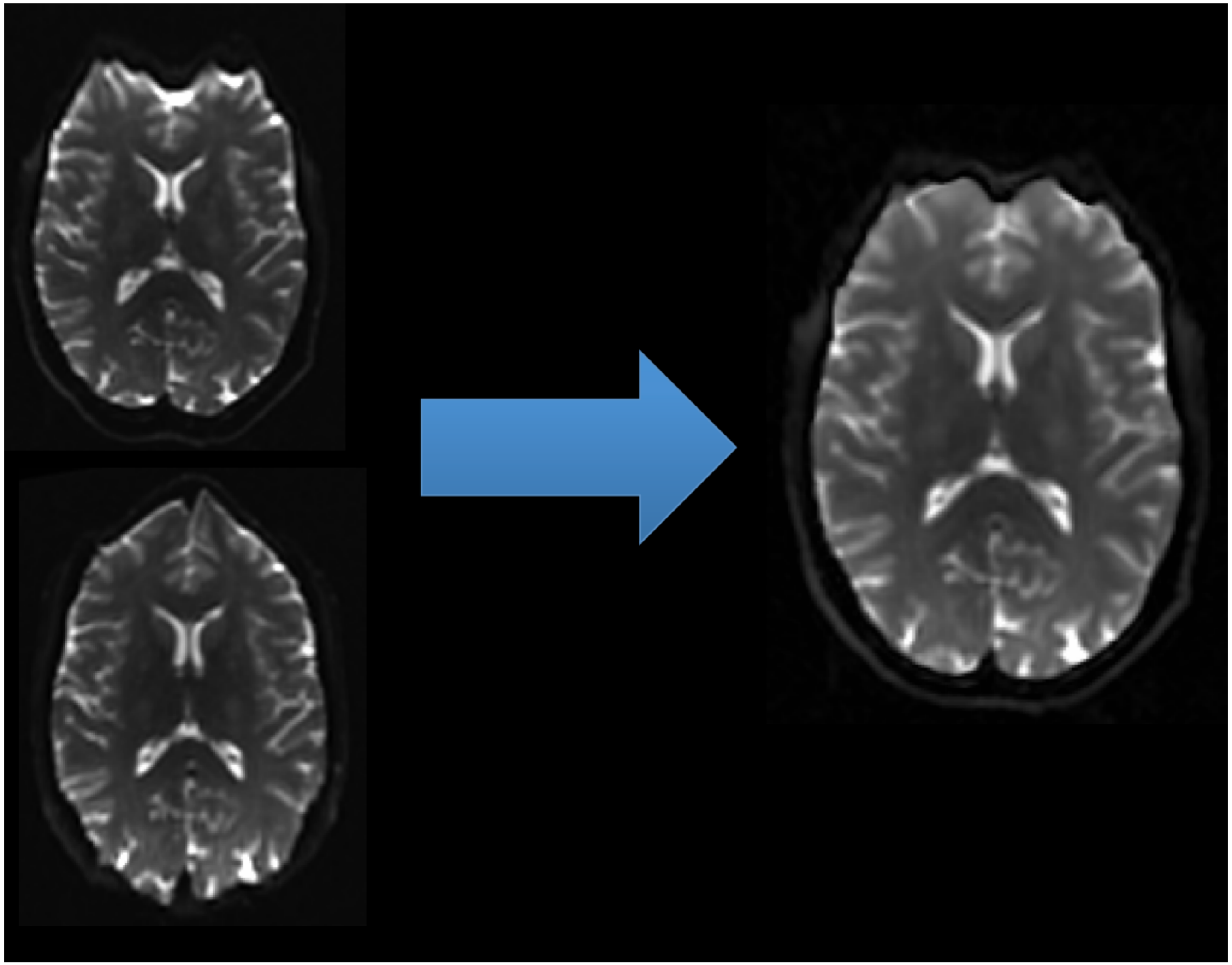
# Distortion Correction of EPI

- Acquire a few 3D images with opposite phase encoding method (“blip up” and “blip down”)
  - This will reverse the distortions of the EPI data
  - You don’t need many opposite blip images
  - Ideally, also reverse the slice-selection gradient to also reverse the small slice-selection distortions



# Distortion Correction of EPI

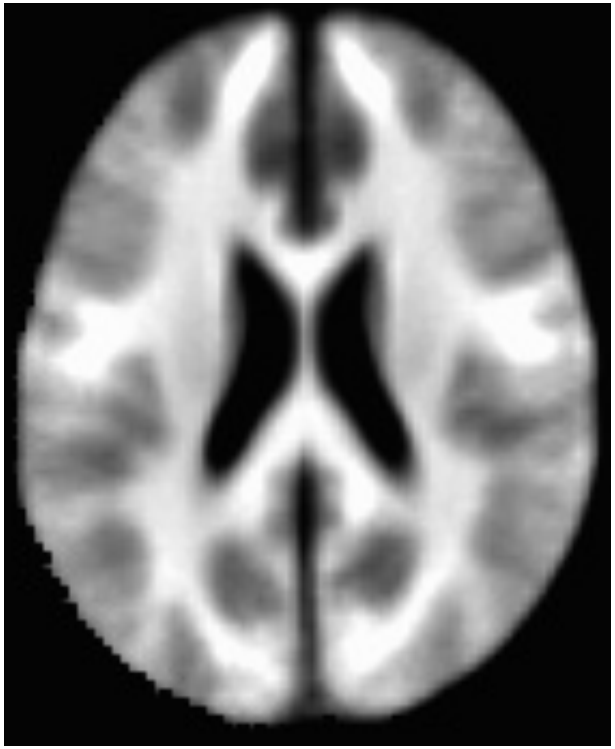
- Use *unWarpEPI.py* to fix the distortions

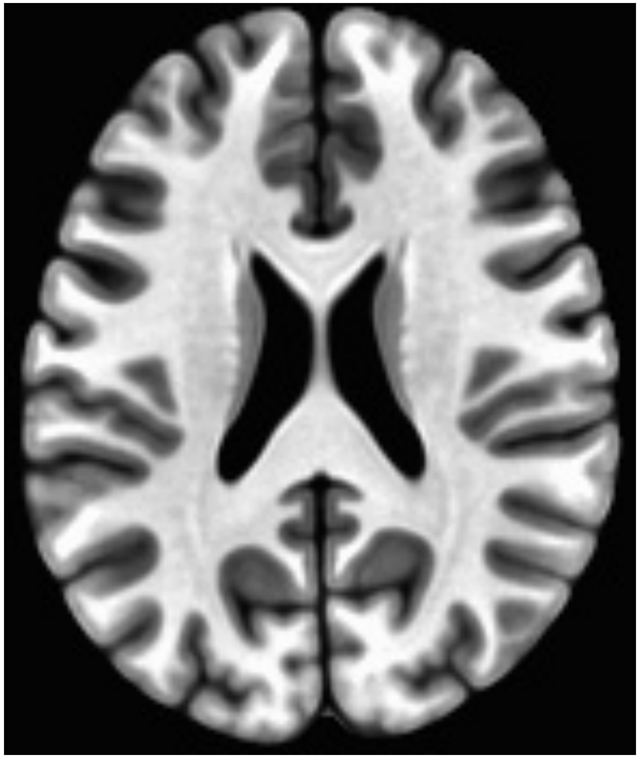


# Nonlinear Warping to MNI Template

- afni_proc.py *can* do the nonlinear warping for you
  - But, nonlinear warping is slow (in fact, slowly slow)
  - If you need to re-rerun subject analysis, nonlinear warping will slow the re-run script down *a lot*
- Solution: do the nonlinear warping *before* using afni_proc.py, then supply the warping results so that afni_proc.py will skip doing the warping itself
- Mechanism: the **@SSwarper** script (tcsh)
  - Does Skull Stripping (”SS”) and nonlinear warping
  - Base dataset is **MNI152_2009_template_SSW.nii.gz**
    - Nonlinearly warped, not too blurry

# Two MNI Templates





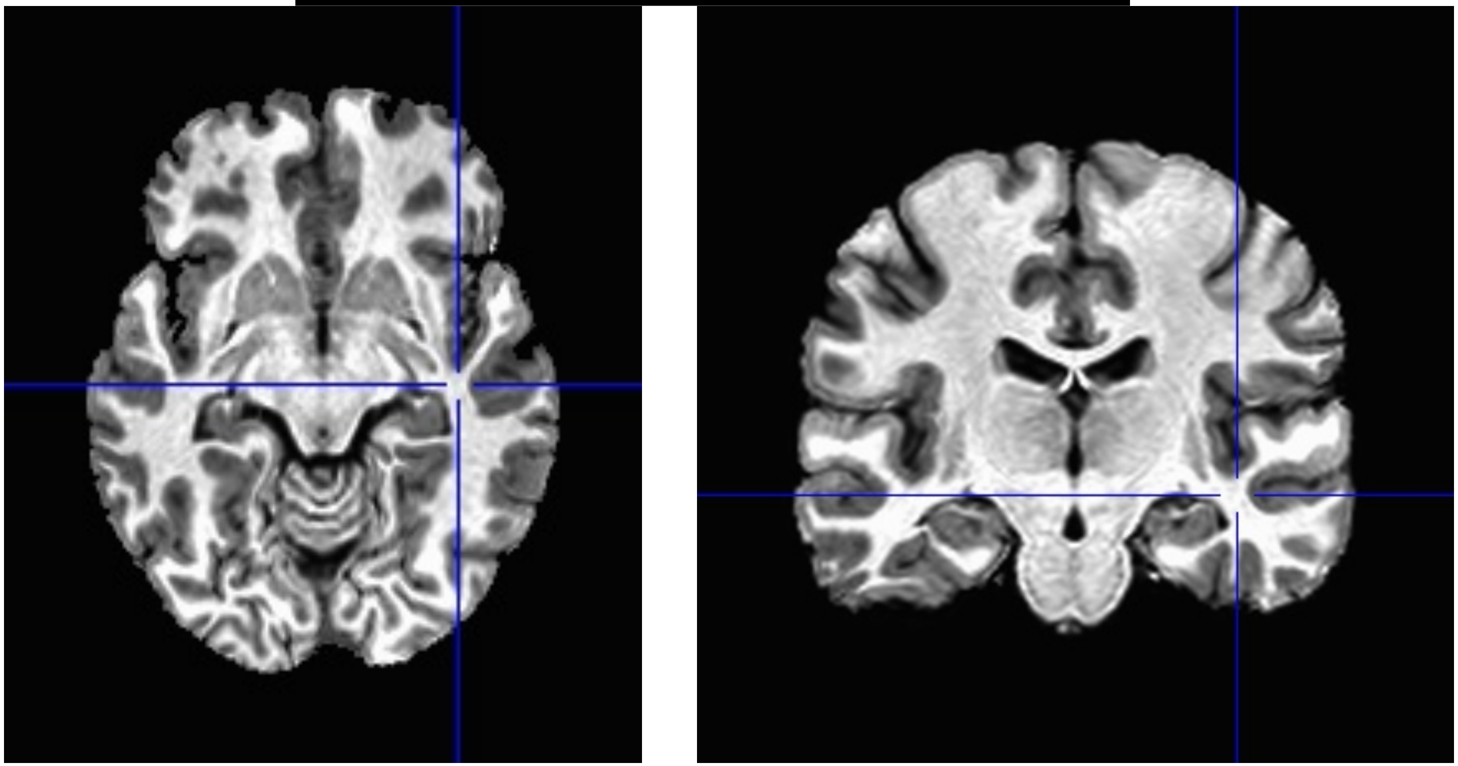
**MNI152_1mm_uni+tlrc**
Affine alignments

**MNI152_2009_template.nii.gz**
Nonlinear alignments

# What @SSwarper Reads and Writes

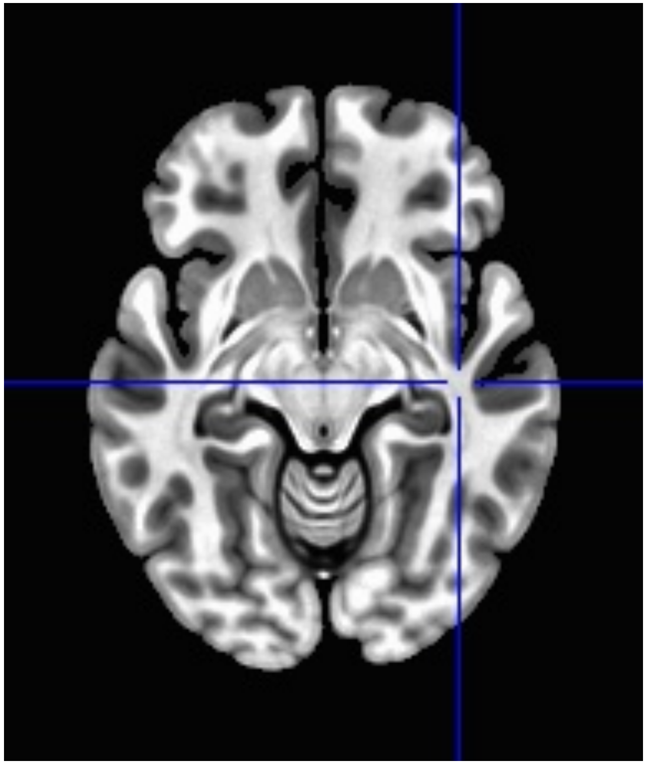
- Inputs:
  - T1-weighted anatomical image of subject (skull-on)
  - Subject ID code, for names of output files
- Outputs (subject ID = **sub007**):
  - **anatSS.sub007.nii**
    - skull-stripped dataset in original coordinates
  - **anatQQ.sub007.nii**
    - skull-stripped dataset, nonlinearly warped to MNI template
  - **anatQQ.sub007.aff12.1D**
    - affine matrix to transform original dataset to MNI template
  - **anatQQ.sub007_WARP.nii**
    - incremental warp from affine transformation to nonlinearly aligned dataset
- These files are needed for later use in afni_proc.py

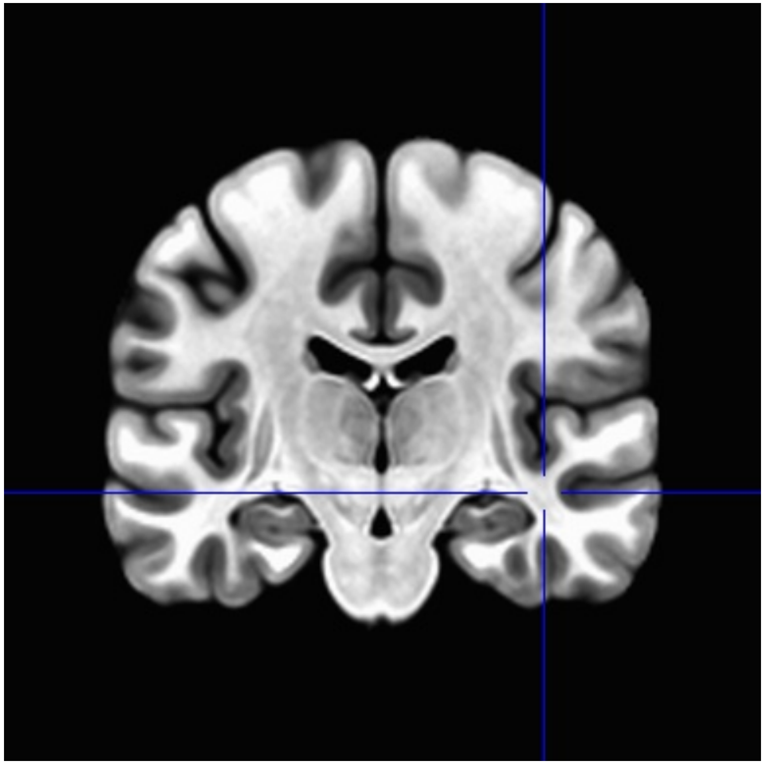
# @SSwarper Results



sub00440 from Beijing-Zang
in the FCON-1000 collection

# MNI Template Slices





For comparison